Privacy-ABC Technologies – on Mobile Phones

Dr. Gert Læssøe Mikkelsen,
Alexandra Institute A/S

ABC4Trust Summit Event
Brussels, January 20, 2015
Challenges and possibilities.

• Pilots and Reference implementation in ABC4Trust
  ▪ Focus on Client(PC)-Server and smartcards
• Users are using mobile devices
• Users bring their smart phones everywhere
• New Use cases – e.g., in the physical world.
  ▪ Now even iPhones come with NFC – currently very restricted!

NEM ID  Apple Pay  Google wallet
Challenges on mobile devices

• **Platform?:**
  - Native – very diverse
    - Android, iOS, Windows Phone etc.
  - Common language: JavaScript?
  - Cloud IdMaaS?
  - HW support?

• Computational power?
• Storage of keys and credentials.
• Usability
Smart Phone Feasibility Study

- Focus on what can be done with current technology
- Focus on functionality

- 3 Proof of Concepts
  - Smart Card emulation
  - Native App
  - Java Script

- Relevant roles
  - User
  - Part of User’s SW (Smart Card emulation)
  - Verifier
  - Inspection

- Not so relevant roles
  - Issuer
  - Revocation authority
Smart Card emulation

- Still Client(PC)-server setup
  + Development time
  + Performance
  + User Convenience
  + User interface
  + No additional HW
- Security
- Devices
Native App

user service of ABC4Trust reference implementation as mobile service-app

• Android!
  ▪ ABC4Trust Reference implementation in Java

• Security
  ▪ Keys/Credential stored in ABC4Trust App’s internal memory

• Usability?
Native App

1: Request
2: Policy/Verifier URL
4: Presentation Token
5: Session Key and revealed Attributes
6: Session Key
7: Session Key

Smartphone
Application
p-ABC App

Service Provider
Verifier

Gert Løssøe Mikkelsen, Alexandra Institute A/S
Native App

Danish Literature
The-very-cool-course
Privacy and Security 101

The following request was sent from Course evaluation of The-very-cool-course:

Show that you own a Course Credential and reveal the value of course id (which is The-very-cool-course).

Show that you own a University Credential

Present a scope-exclusive pseudonym for the scope urn:patras:evaluation.
MS U-Prove Native App.

- MS U-Prove C# version can run on Windows Phones
JavaScript?

- JavaScript is highly cross-platform
  - Every device with a modern browser
  - Not built for security/Cryptography
  - How to verify the code?
    - Has someone changed the code server side?
    - Do I get the same code as everyone else?
  - Where to store keys/credentials securely?
    - Server side?
    - Cookies?
    - Local storage?
    - Issue when needed?
JavaScript Prototype

- Prototype implementation of MS U-Prove
  - U-Prove is simpler than the ABC4Trust reference implementation and Identity Mixer.
- Compatible with MS U-Prove C# library
Smartphone

Browser

ABC4Trust.eu only!

Service Provider

1: Request

3: Presentation token

2: Crypto.js

ABC4Trust.eu

ABC4Trust.eu only!
JavaScript Performance

• Very dependent on platform, and use of libraries!
• Our implementation:
  ▪ 2.1 sec (Galaxy Nexus, default browser)
  ▪ 30 sec (iPhone 5, Safari)

• Microsoft implementation: iPhones nearly as fast as Androids.
JavaScript the new language for Crypto?

A lot is happening - Since this task of the project was finished:

- Microsoft U-Prove JavaScript (July 2014)
- Microsoft Research JavaScript Cryptography Library (August 2014)
- Google End-to-end Chrome Extension (June 2014)
- W3C Cryptography API
Security Mobile Devices.

- Subject to malware attacks
- Subject to physical theft

- Define a threat model
- Security improvements
  - Secure elements
  - Direct Anonymous Attestation TPM
  - SIM cards
  - Smart card read by the smartphone.
Conclusion

• Using p-ABC’s on mobile devices is feasible
  ▪ both as native applications and JavaScript.

• New use cases/improved user experience.

• New security issues
  ▪ Mobile devices vulnerable to a number of attacks - should be addressed according to the threat model.

• A lot is happening on JavaScript right now.

• D4.4 Smartphone feasibility analysis
  www.abc4trust.eu