Trusted Location Based Services

Masterprojects @IAIK

Outline

Overview: Location Based Services (LBS)

Security ? - Trusted Location Based Services (T-LBS)

- Prototype 1
- Prototype 2
- Security Analysis

Location Based Services

Numerous applications with users' current location

- Augmented reality
- Navigation
- Context - Awareness

Simple goal: improve user experience

no applications that require trustworthy location

- how to acquire trustworthy information?

Trusted - Location Based Services

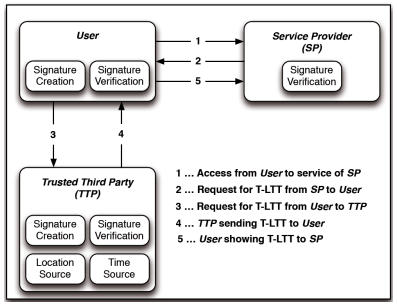
Location-Time-Ticket (LTT)

- position
- time

Attested by TTP

- trustworthy location
- for services

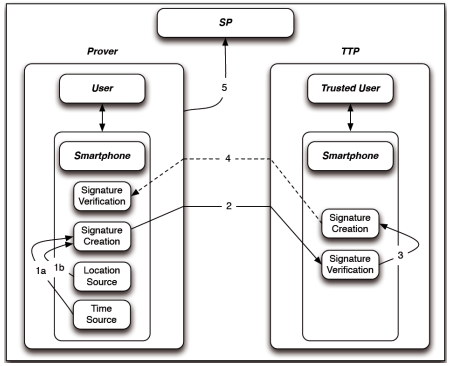
→ T-LTT



Prototype 1: Peer-to-Peer

Two Smartphones
- one user acts as TTP

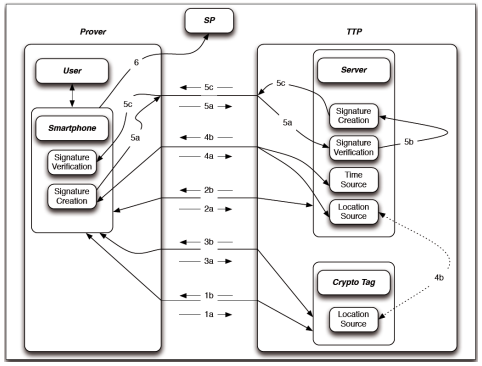
A-Trust Mobile Phone Signature
- attests users' identity
- integrity
- non repudiation



Prototype 2: NFC tag + server as TTP

NFC Crypto tag
- unique id
- electronic signature

Server
- knows location of tag
- timestamp



Use Cases

Prototype 1
- Accident report
- Commissioning (photo, confirmation)
- ...

suitable if at least one person has interest in correctness

Prototype 2
- Check-in (Foursquare, Facebook)
- Geocaching
- ...

no limitations in trustworthiness

Prototype 1: Project goals

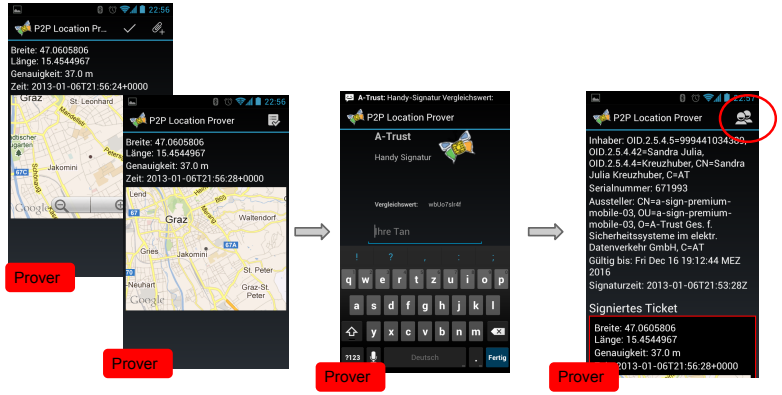
Extend exisiting prototype

Refine used protocol

Provide reusable code components

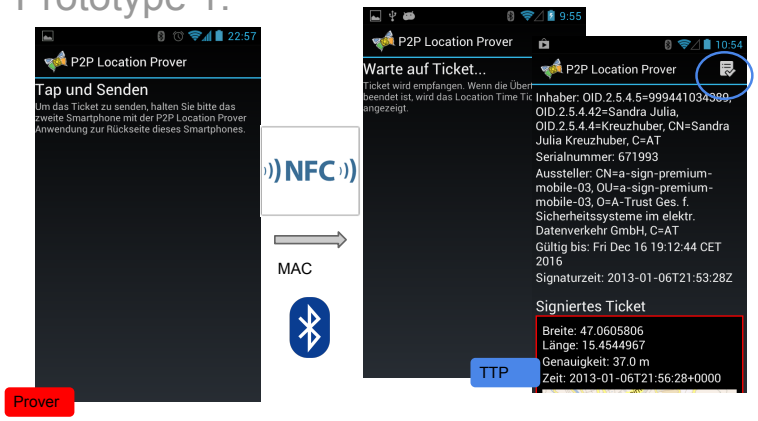
→ Android library + simple demonstrator application

Prototype 1:



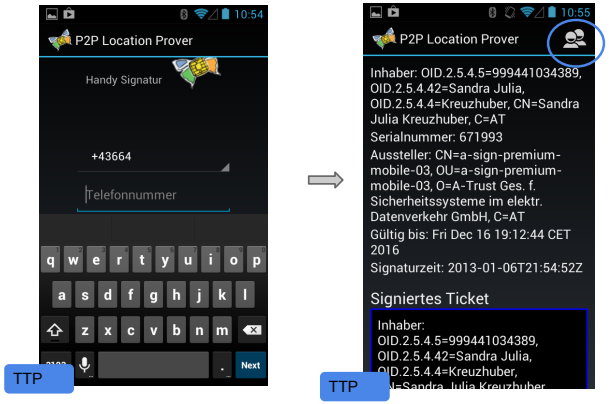
Prover creates ticket, checks it, signs it, checks signature

Prototype 1:



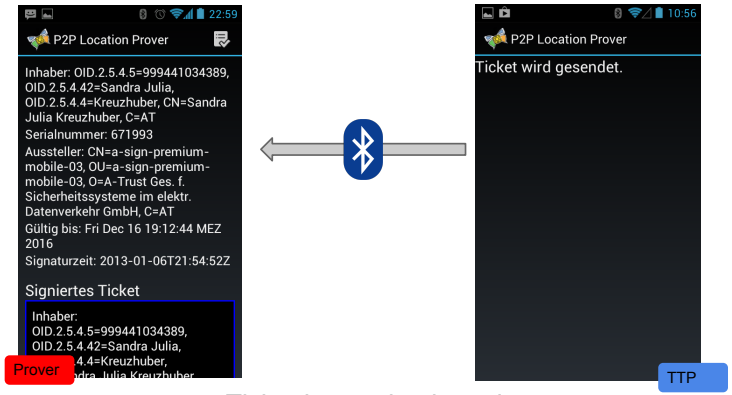
Share ticket, TTP receives ticket, checks ticket

Prototype 1:



Sign ticket, check signature, share ticket again

Prototype 1:



Ticket is sent back again

Prototype 1: Library components

Prototype 1: Lessons learned

A-Trust Handy Signature:

- Bug: sign ticket twice, same id in signature tag, signatures not verifiable, now fixed

Bluetooth:

- random bytes lost, large files, implemented SHA256 checksum

Prototype 1: Possible extensions

Improvements in User Experience

- Storage dialogs (now "hardcoded" in special dir on SD)
- more defined error messages

Signature Verification on the phone (other bachelor project...)

- display notification to user

Several "CommunicationDevices" - now Bluetooth, Wifi Direct?

- Android Bluetooth quite a mess...

Prototype 2: Overview

SP

Prover = Android App (on phone) + User

TTP = Server (on Internet) + Crypto Tag (fixed location)

Specific P2 Goals

Extend ACN project rapid prototype (without server...)

Feasibility (performance and usability: tag, internet)

Design protocol

GAE/GWT Know-how build-up

Prototype 2: Crypta

CRYPTA cryptographic protected tags

CRYptographic Protected TAg by IAIK, ams & RF-iT solutions

NFC-enabled (passive, ISO14443A)
- ISO7816 APDU commands

NFC Forum Type 4 Tag compliant
- read URL for application or Certificate (proof of originality)

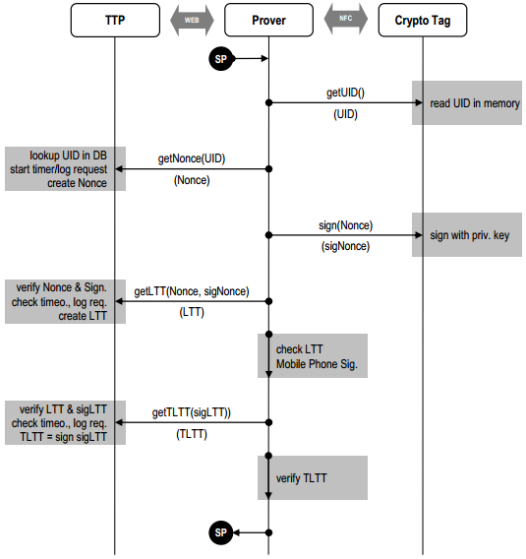
7 byte **UID**

Crypto: ECDSA, ASE
- **ECDSA**: sign 16 bit value (192 bit key size)

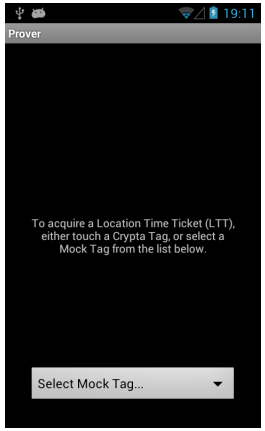
=> Mounted at fixed location; registered with TTP server

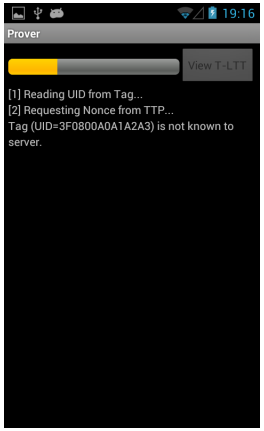
Prototype 2

Protocol

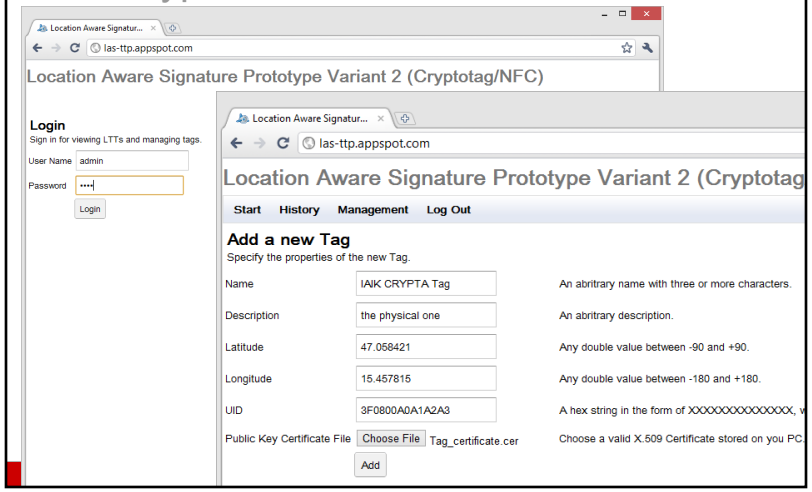


Prototype 2: Live Demo /1

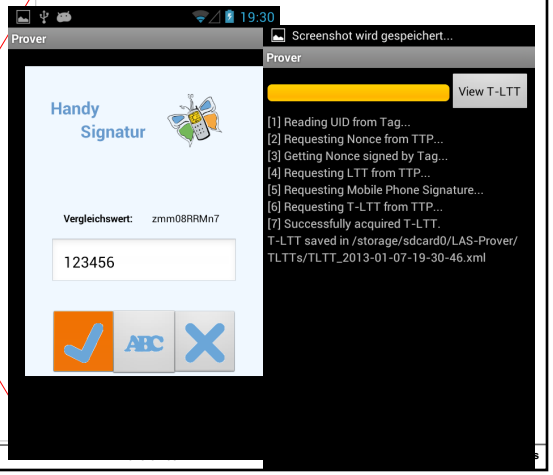




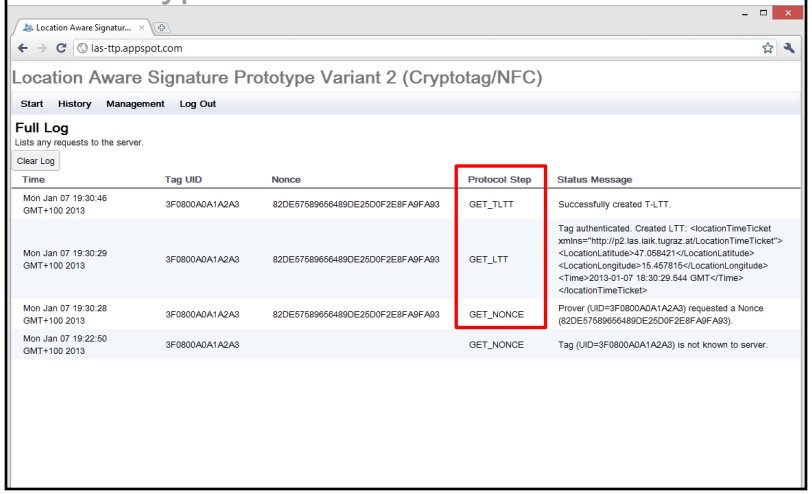
Prototype 2: Live Demo /2



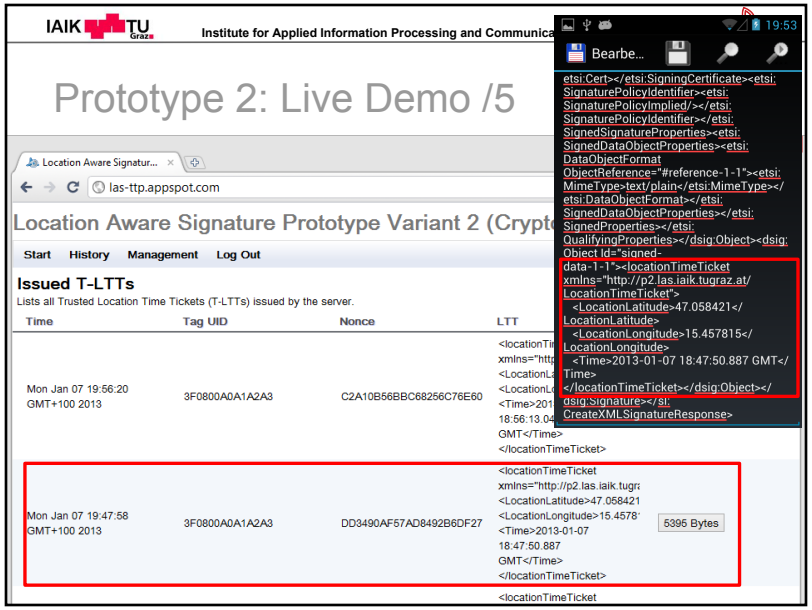
Prototype 2: Live Demo /3



Prototype 2: Live Demo /4



Prototype 2: Live Demo /5



Prototype 2: Implementation

TTP - Server

- hosted on GAE/J & datastore (DB of tags in the wild)
- UI for Administration and History (GWT)
- API implementation for protocol (Hessian)

TTP - Crypto Tag (black box - public & private key provided)

- Mock Tags (for unit testing or testing without physical crypto tag)

Prover (Android App)

- RPC to API (Hessian)
- NFC communication with Crypto Tag
- Mobile Phone Signature (using Library from P1)

Prototype 2: Lessons Learned/Future

Troublesome Mobile Phone Signature on GAE
GAE SDK >= 1.7 (mid 2012): register **own** JCE provider
RPC: Android/GAE => Hessian (after setup troubles)

Fix issue in protocol by either:
- SSL encryption for communication Prover<->API
- simplify protocol by combining getNonce & getLTT

Store application download URL on Crypto Tag (Personaliz.)
Public Key Infrastructure for Crypto Tags (+ Cert. onto Tag)
Modelling of Accuracy?

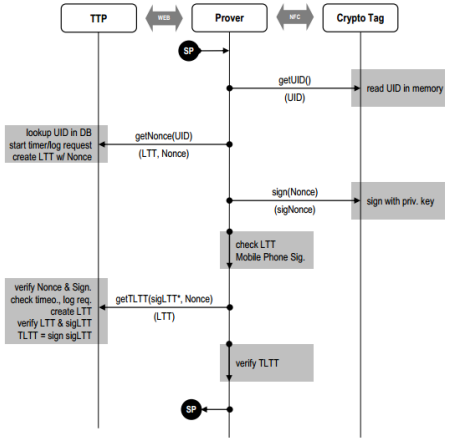
MOA on Google App Engine (GAE)

TTP (Server): requires MOA-SS and MOA-SP
GAE: limited, sandboxed JRE (JRE Class White List)
- read-only access to the filesystem (gae-filestore)
- ~~IAIK security provider~~ (resolved since GAE 1.7)
- No sockets can be opened

[MOAs in der Cloud, Bernd Zwattendorfer]

Prototype 2: Protocol Extension

Before Prover requests T-LTT by sending **getTLtt(signedLTT)** to the TTP Server, an attacker might send an unmodified LTT, but signed by himself to the TTP.



Security Analysis: Common Aspects

Protocol design & implementation
Protection of cryptographic material

Possible malicious entities
- LP ... Location Prover
- EXT ... Some 3rd party
- TTP ... Trusted Third Party

Security Analysis: Prototype 1

TTP = 2nd Smartphone and its User
- User = real person: pros and cons
- TTP has to have strong interest in use case

LP: Malware onto TTP; Credentials; Distraction of TTP
TTP: Malware onto LP; Credentials; access to T-LTT
EXT: Malware onto LP, TTP, or TTP and LP
LP+TTP
TTP+EXT

Security Analysis: Prototype 2

TTP = Tag (fixed location) + Server
- trusted as long as not compromised by external attack
- LTT generation on server
- thoroughly evaluated before deployment (in theory)
- TTP is not a real person

LP: key extraction of tag, side-channel attacks
LP: proxy to malicious person (tag is not human)
EXT: malware on LP smartphone

=> more possible scenarios, but also more complex

Security Analysis: Risks/Conclusion

1. Cooperation of LP and TTP in P1
   => applications where TTP has strong interest

2. Cooperation of LP and EXT in P2
   => TTP-Server cannot verify identity of LP

3. Stealing of credentials for Mobile Phone Signature (both)
   => quite difficult (two factor authentication), but possible

4. Malware injection on any involved smartphone

Conclusion

Both prototypes demonstrate technical possibility
(but Limitations with P2/GAE)

P1 serves components for P2 (Mobile Phone Signature)

Security:
- Aspect 1: Protocol & Implementation
- Aspect 2: General Principle of our TLBS
  => use case limitations

Trusted Location Based Services

Thank you for your Attention!

Q&A