

Transferring Trust to the Client Platform

Security Hardware in Theory and Practice

Dagstuhl

20th of June 2008

Endre Bangerter¹, Maksim Djackov¹ and Ahmad Sadeghi²

¹Bern Univ. of Applied Sciences and ²Univ. of Bochum

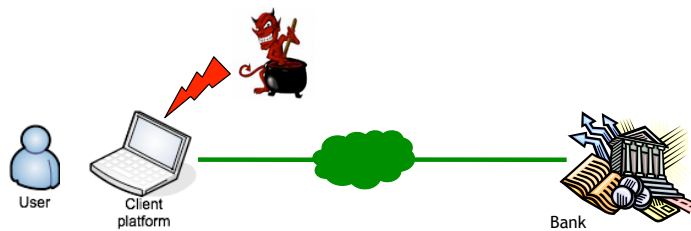
Overview

- Client platforms (desktops, laptops, etc.) are insecure
 - End users don't manage machines and behave securely
 - Operating systems, browsers etc. are not secure enough
 - So that also sophisticated and careful users are affected
- Probably #1 security problem for consumer type of end users, attackers are more and more criminals going after money
- Transaction security
 - Collaboration with AXSionics
- Ad-hoc attestation - new results on using TPM to attest client platforms
 - Joint work with Ahmad Sadeghi and Maksim Djackov
- What is happening in the real world
- No meant to be an exhaustive overview

Agenda

- Transaction security
- Ad-hoc attestation
- What is happening in the real world
- Conclusions

Transaction insecurity



- Transactions modified by malware on client platform after user is authenticated (often called transaction generators, browser in the middle attacks)
- These attacks are happening, mostly targeted at e-banking transactions
- What do banks do: push liability on end-users (not yet enforced), evaluating new technologies
- The hardest part is not modifying transactions, but to launder the money. Risk is born by money mules, not the attacker in the background
- Once money laundering problem is solved (e.g., using online games, online gambling) the “dam might burst”, and banks will enforce non-liability clause
- There is more: phishing, identity theft etc.

Trusted tokens to secure transactions

- An emerging market for transaction security?
- Different companies (e.g., AXSionics, IBM research both in Switzerland) are working on next generation security tokens to authenticate transactions and not only users
- RSA tokens of the future?



Trusted tokens to secure transactions

- Different products, common idea:

trusted token with display

plus

end to end secure channel from token to transaction server

allow

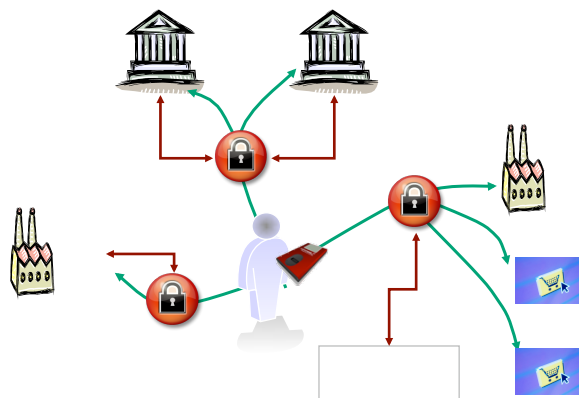
User to view and confirm / reject transactions on token

- **Trust model:** full trust in token, no trust in client platform
- Secure, but paranoid model

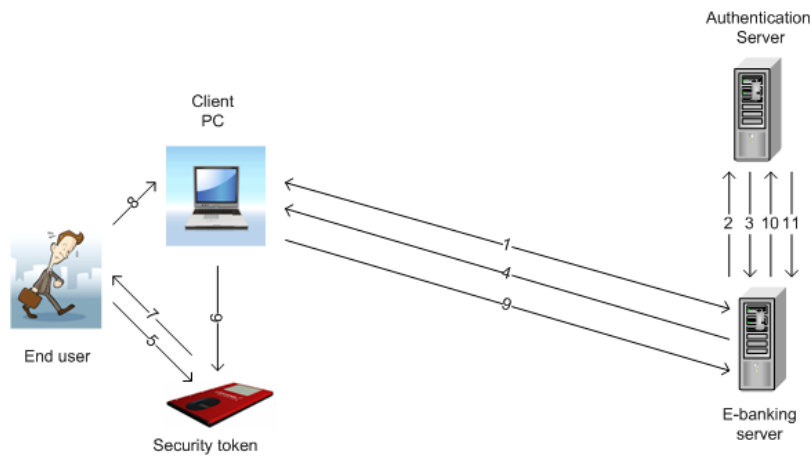
AXSionics security token - Demo

- Demo e-banking security: <http://axsportal.com>

Identity federation



Architecture overview



AXSionics security token - Features

- Can be combined with white-listing approach to reduce number of transactions to be confirmed
- Key features: security and ease of use
 - E.g., fingerprint reader, wireless optical transmission
- Engineering type of novelty, not research type of novelty → token actually works
 - Many aspects of the token were published earlier or developed independently

AXSionics security token - Features

Some technical features:

- All computation and storage inside ARM secure core CPU
- Custom firmware
- Display resolution = 128 * 96 pixels
- Can use fingerprint reader for on-screen navigation
- Flickering based on animated GIF, Flash, Java, etc. works universally (browsers & monitors)
- Flickering frequency - 50 - 60 Hz
- Flickering bandwidth - 150 bits/sec
- Straightforward symmetric crypto, based on pre shared keys per token (i.e., 128 per token)
- USB port, currently only used for firmware upgrades and as power supply

Beyond transaction security

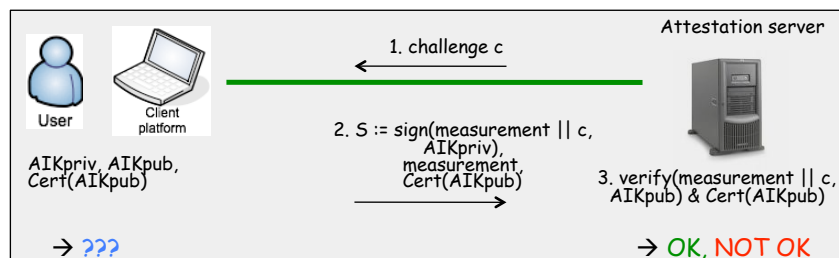
- Trust model: “full trust in token, no trust in client platform” OK for (sensible) transactions
- But not trusting the client platform means throwing away too (?) much functionality and convenience
 - → cannot replace PCs by tokens
- A more attractive opportunity is transferring trust from token to the client platform

Agenda

- Transaction security
- Ad-hoc attestation
- Trusted storage in the real world
- Outlook

Trusted computing: Remote attestation

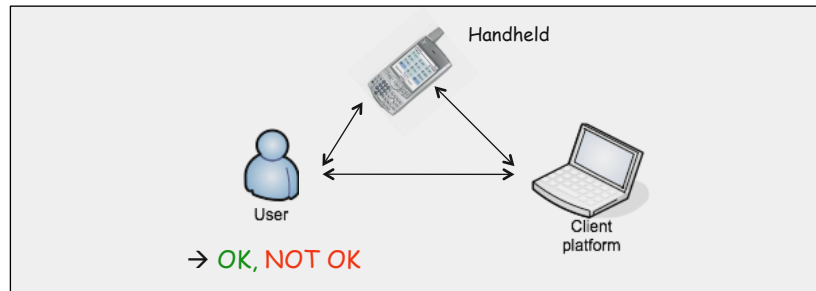
- Improving trust in platforms is the mission of trusted computing
- Trusted Platform Module (TPM)*: Trusted chip built into platforms, performs and reports integrity measurements (i.e., hashes of files)
- Remote attestation*:



- Remote attestation is adequate in some scenarios (e.g., verify client before connecting to company intranet), but not for improving the user's trust in the client platform

Trusted computing: Ad-hoc attestation

- *Ad-hoc attestation*: User walks up to a computing platform and wants to find out whether **that platform in front of her**, which she can identify physically (e.g., by seeing or touching it), **is trustworthy**.
- Protocol where user assisted by trusted handheld device (→ transfer trust to client)

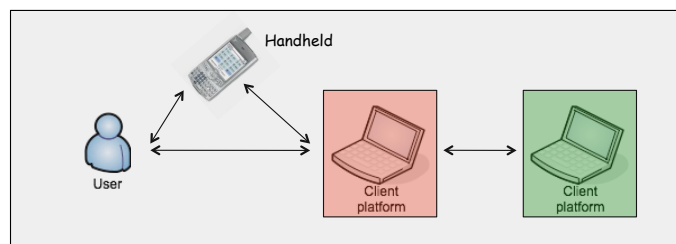


- *No pre-shared state specific to client platform* between user, handheld, client platform.

Open ad-hoc attestation challenges

[Perrig et al. Turtles all the way down: Research challenges in user-based attestation. Usenix Hot Topics in Security 2007]

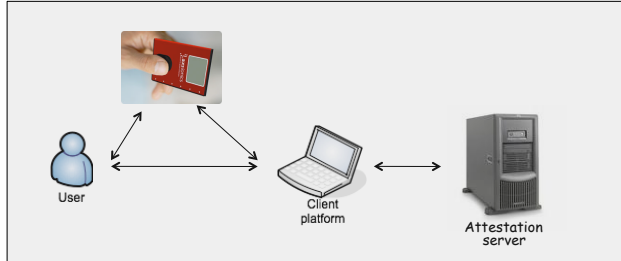
- *Platform in the middle attack*:



- *Usability & viability*:
 - User device: affordable, commodity hardware, small form factor
 - Universal connectivity between the user device and the target platform → any platform can be ad-hoc attested
 - User device itself has to be trustworthy and resilient against attacks
 - Device and the ad-hoc attestation protocol shall be intuitive and easy to use
- *Management and evaluation of integrity measurements*:
 - Store DB of known good integrity measurements on handheld? How to keep up to date?

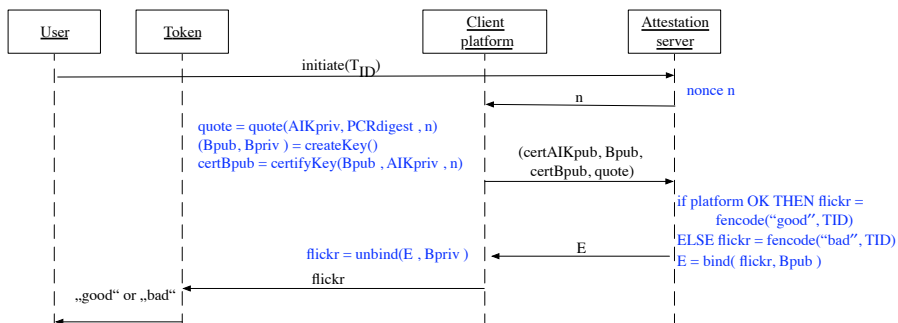
Our solution - Overview

- Novel solution(s) living up to the challenges, based on same architecture & technology:



- Main ideas and features:
 - Inherits ease of use of AXSionics system → demonstrative ad-hoc attestation
 - Attestation server centrally manages all known good integrity measurements and performs integrity evaluation → could be run as a service like anti-virus signatures
 - Client to attestation server run standard remote attestation protocol
 - We provide protocols that allow attestation server to securely report client state to AXS token asserting that user can identify the client in question
- Based on the assumption that there are adequate integrity evaluation technique, we did not tackle how to improve attestation techniques

Ad-hoc attestation protocol details



Comments

- **Security:** Based on the assumption that good platform can keep flickering private
 - No malicious process running on good platform that forwards flickering signal to an impersonating platform
 - On typical OS this is a strong assumption, since all processes can read screen buffer and thus get flickering, no ACL to screen contents
 - Non SW attack: filming pattern
 - Assumption OK on secure OS (e.g., Perseus / TURAYA)

- Other variant of the protocol, based on assumption that client platform can assert integrity of fingerprint measurement
 - More realistic for current OS, can use ACL mechanism to protect fingerprint reader and attestation component
- Protocols independent of static or runtime attestation
- We see, once trusted computing technologies built into OS, there are viable solutions for ad-hoc attestation

- Problem: trusted computing is far(?) away

Future work

- Have implemented research prototype of ad-hoc attestation protocols

- Turn it into a really usable system
 - Integrate with Perseus / Turraya secure OS
 - Enforce and implement OS security features on which attestation protocol relies
 - Minimize pre-installed components on client platform, e.g., load client attestation SW from attestation server

Agenda

- Transaction security
- Ad-hoc attestation
- What is happening in the real world
- Conclusions

Trusted launch of trusted browser from token

- Kobil mIdentity being deployed by banks
- USB token that serve as classic authentication token (i.e., user and not transaction authentication), which contain hardened version of Firefox browser
- Hardened browser:
 - Locked down to bank URL
 - All extension mechanisms removed, locked down
 - Anti-reverse engineering techniques applied
 - Some integrity controls built into code
- USB token is read-only, except for authenticated browser updates
- What is interesting about this solution is: Similar idea as with ad-hoc attestation: transfer trust from token to platform
- But more realistic (?) version thereof: can be deployed today, less secure (protects against current malware, won't stand dedicated malware), and cheap

Agenda

- Transaction security
- Ad-hoc attestation
- What is happening in the real world

- Conclusions

Conclusions

- HW token technologies *driven by client platform insecurity*
- Transaction security based on exclusively trusted security tokens, neglecting the functionality of client platform
- Practically viable ad-hoc attestation protocol, based on trusted computing technologies, which are *not yet (?) widely used*
- Solutions with limited trust guarantees, but lightweight make it to the market
- Room for novel research & technologies where handheld devices test client integrity in a more clever and yet practical way?

Further info and reading

- Bangerter, Djackov, and Sadeghi. A Demonstrative Ad-hoc Attestation System. ISC 2008. Springer LNCS. To be published.
- AXSionics <http://www.axsionics.com/>
- IBM token <http://www.zurich.ibm.com/ztic/>
 - Weigold et al. The Zurich Trusted Information Channel - An Efficient Defence against Man-in-the-Middle and Malicious Software Attacks. TRUST 2008, LNCS 4968, pp. 75-91, 2008.
 - Baentsch et al. A Banking Server's Display on your Key Chain. ERCIM News 73, online edition, April 2008
- KOBIL mIDentity
http://www.kobil.de/fileadmin/download/support/download/informationmaterial/en/ONLINEFLYER-MIDENTITY-FUTUREBANKING_1V02_20051102_UK.pdf