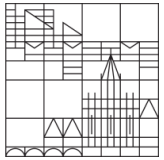


hPIN/hTAN: A Lightweight and Low-Cost e-Banking Solution against Untrusted Computers

Shujun Li[1], Ahmad-Reza Sadeghi[2,3], Sören Heisrath[3], Roland Schmitz[4], Junaid Jameel Ahmad[1]

[1]University of Konstanz, Germany

[2] Darmstadt University of Technology and Fraunhofer SIT, Darmstadt, Germany

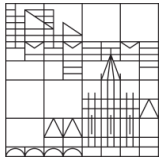
[3]Ruhr-University of Bochum, Germany

[4]Stuttgart Media University (HdM), Germany

March 2, 2011

The Big Picture

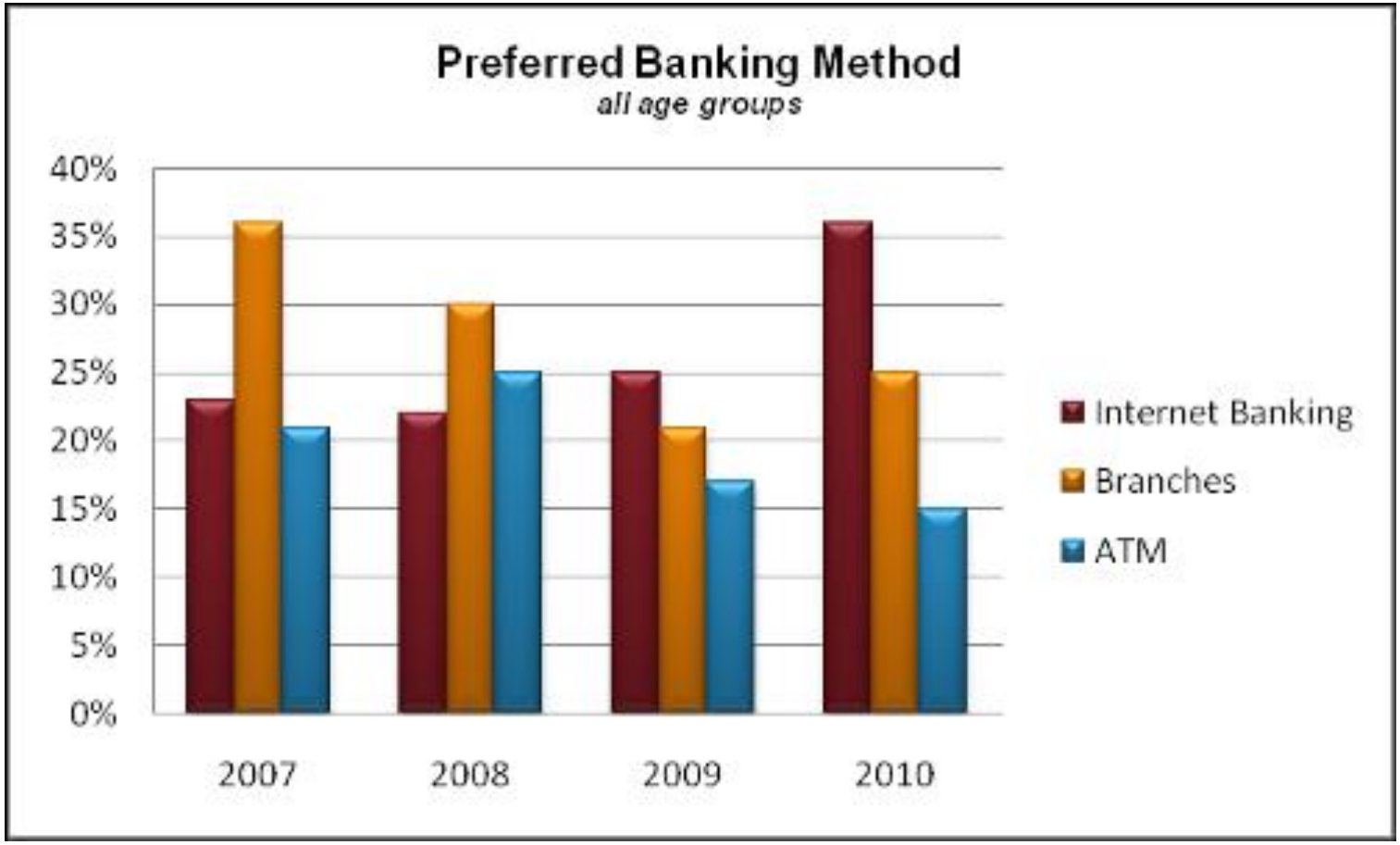
- Our motivation
  - Untrusted computers are a big problem for e-banking
  - Existing solutions suffer from a security-usability dilemma
- Our solution: hPIN/hTAN
  - Simplistic design + Open framework
  - Two parts: hPIN for login + hTAN for transaction
  - Three h-s: hardware (USB token) + hashing + human
  - Three no-s: no keypad + no OOB channel + no encryption
  - Proof-of-concept system + User study
  - A better security-usability balance
  - Live demo available



The Problem

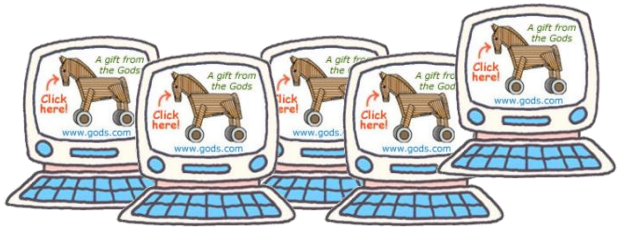
e-banking: Bank customer's first choice now!

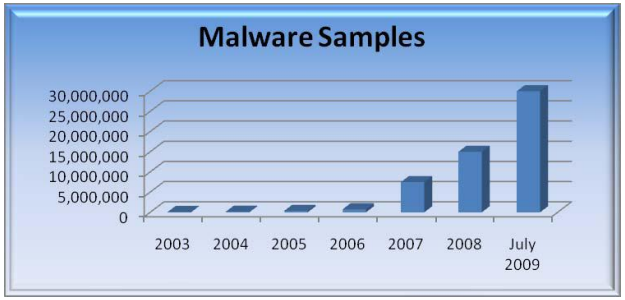
- American Bankers Association survey (September 2010)



Untrusted computers everywhere!

- We are living in a digital world full of insecurities…



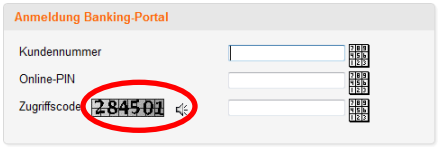


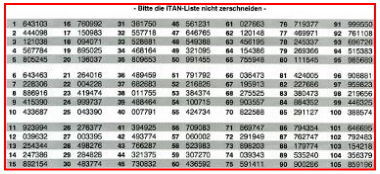
- Real cases of banking malware have been reported!
  - German police (Oct. 2010): ≥1.65 million Euro transactions manipulated by real-time (MitM) banking Trojans…

And the Solution???

E-banking security measures

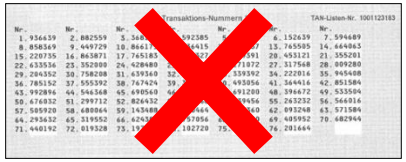
- An incomplete list…
  - login CAPTCHAs
  - indexed TAN
  - transaction CAPTCHAs
  - mobile TAN
  - hardware TAN generators
  - photoTAN
  - HBCI/FinTS
  - IBM ZTIC
  - …





Security-usability dilemma

- indexed TAN
  - Insecure against MitM attack



- mobile TAN
  - Insecure against mobile malware
  - No out-of-band (OOB) channel for mobile banking
  - Unavoidable additional costs (SMS)
  - Untrusted telecommunication service provider (real case reported)

- photoTAN
  - Insecure against mobile malware

- e-banking CAPTCHAs
  - Insecure against automated attacks [Li et al., ACSAC2010]

Security-usability dilemma

- Dedicated hardware-based solutions
  - Some are insecure (e.g. RSA SecurID)

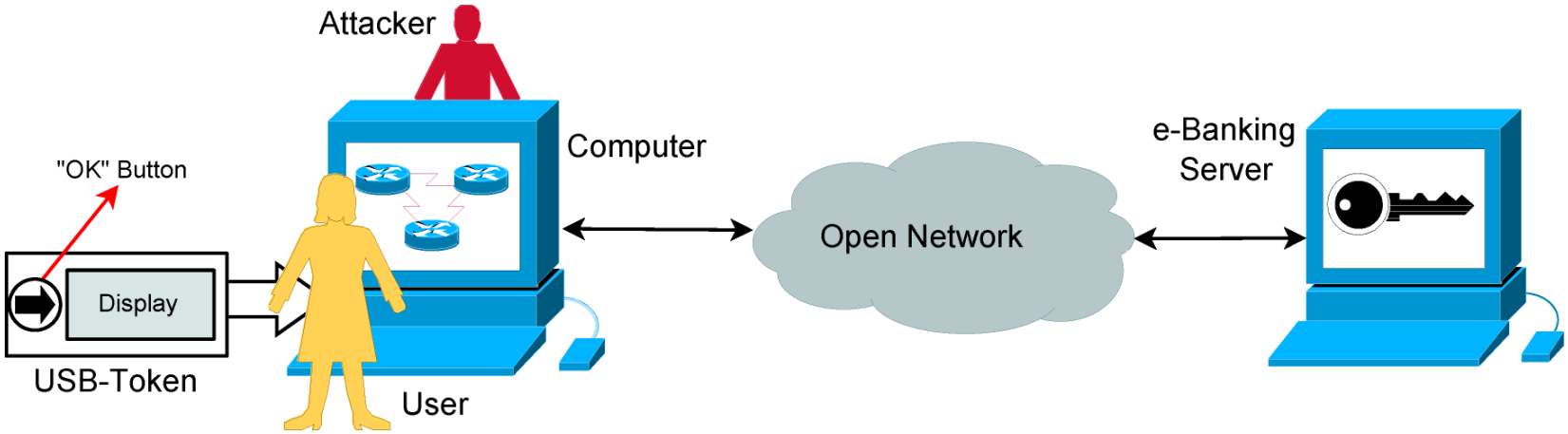
  - High costs (no free lunch, > 10 €)
  - Not very portable (TAN generator, HBCI/FinTS)
  - No PIN protection (IBM ZTIC)

  - High complexity: keypad or optical sensor, encryption, digital signature, SSL/TLS engine, HTTPS parser/embedded web browser, dependency on external website, etc.
  - ⇒ Resources of the untrusted computer are not well exploited!

Our Solution: hPIN/hTAN

The threat model and security requirements

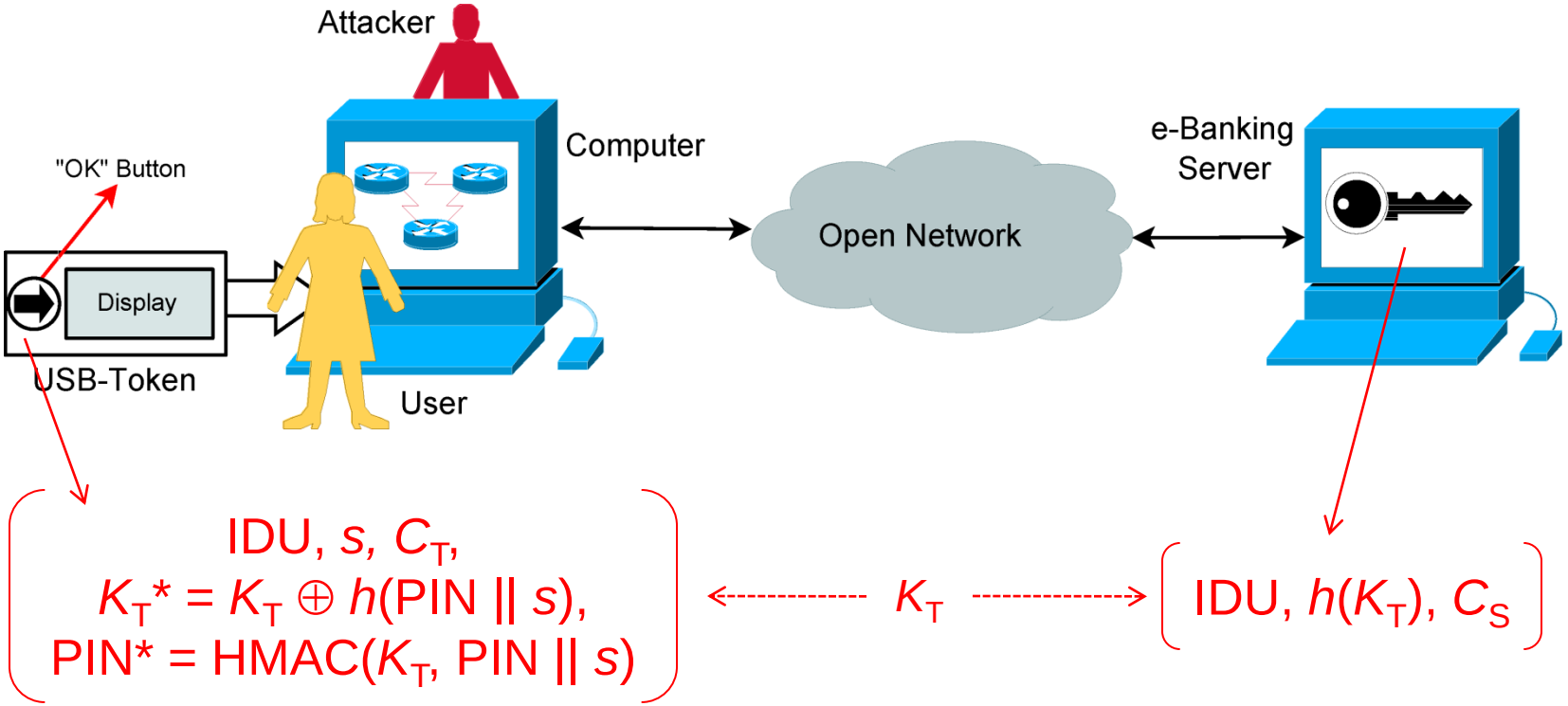
- Assumption
  - The attacker has ***full*** control of the user's computer.



- Security requirements
  - PIN confidentiality + User authenticity + Server authenticity + Transaction integrity/authenticity

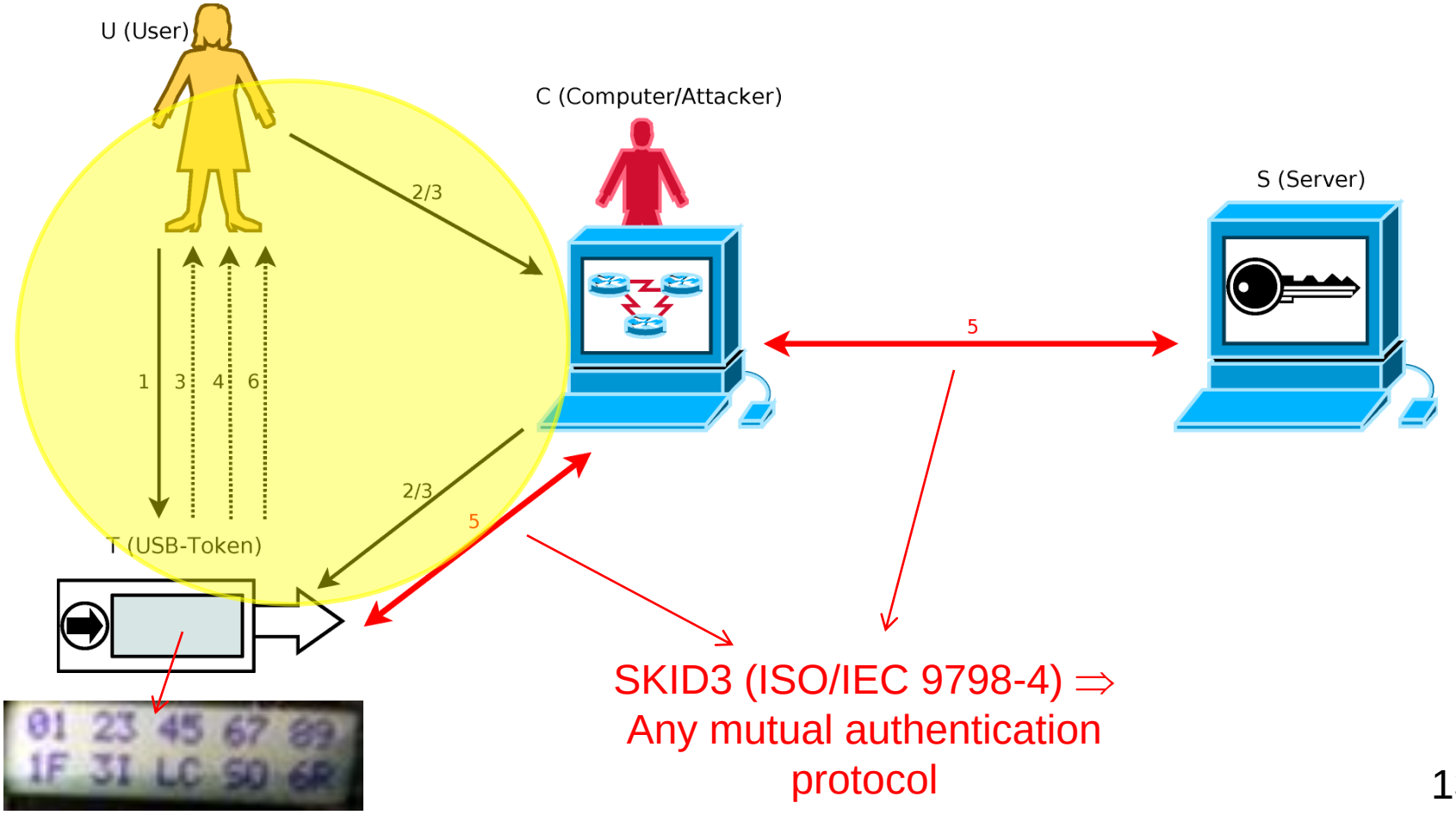
System requirements

- USB token = a processing unit + memory units (for program and data) + a communication (USB) module + an “OK” button + a trusted display



An open framework

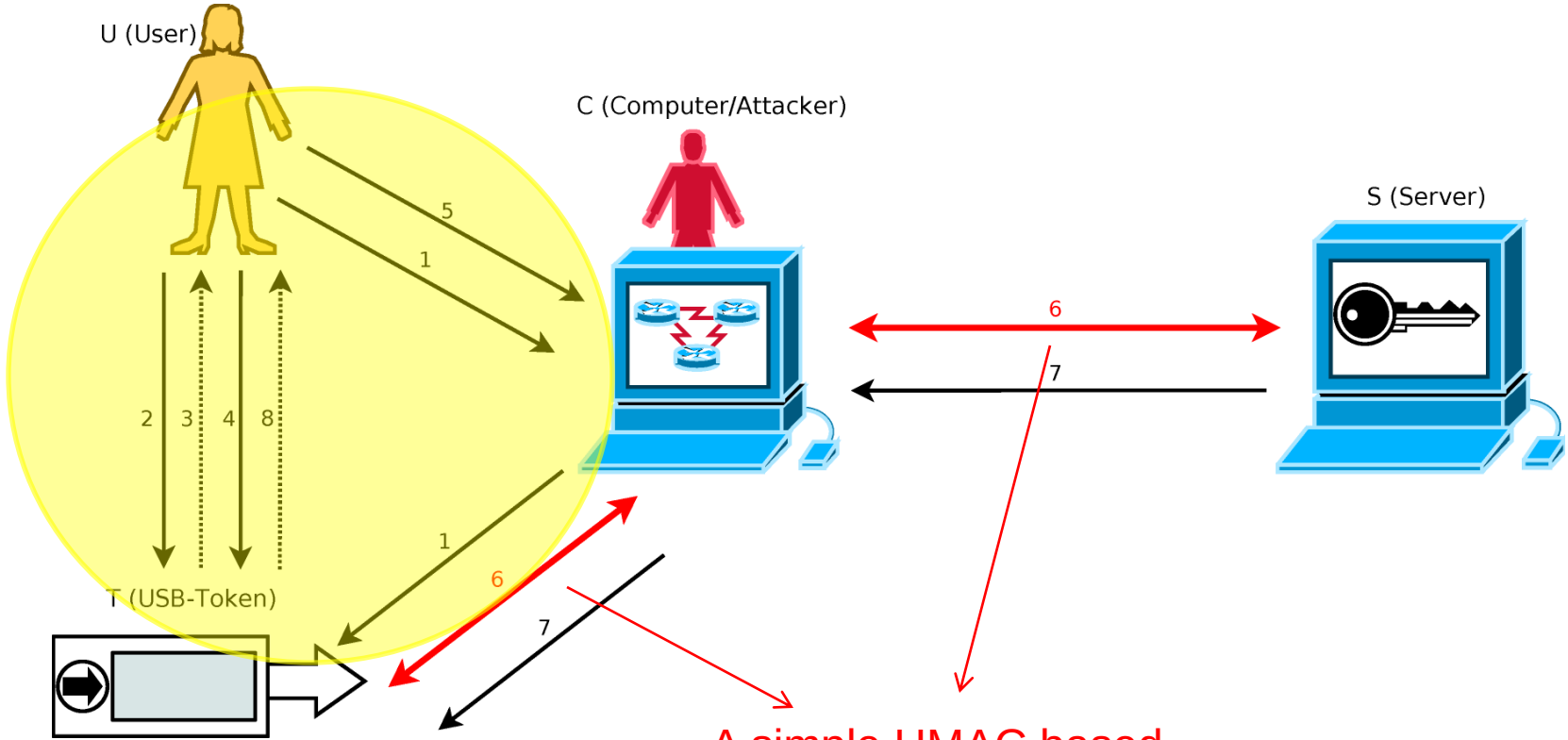
- hPIN (for login)



SKID3 (ISO/IEC 9798-4) ⇒
Any mutual authentication protocol

An open framework

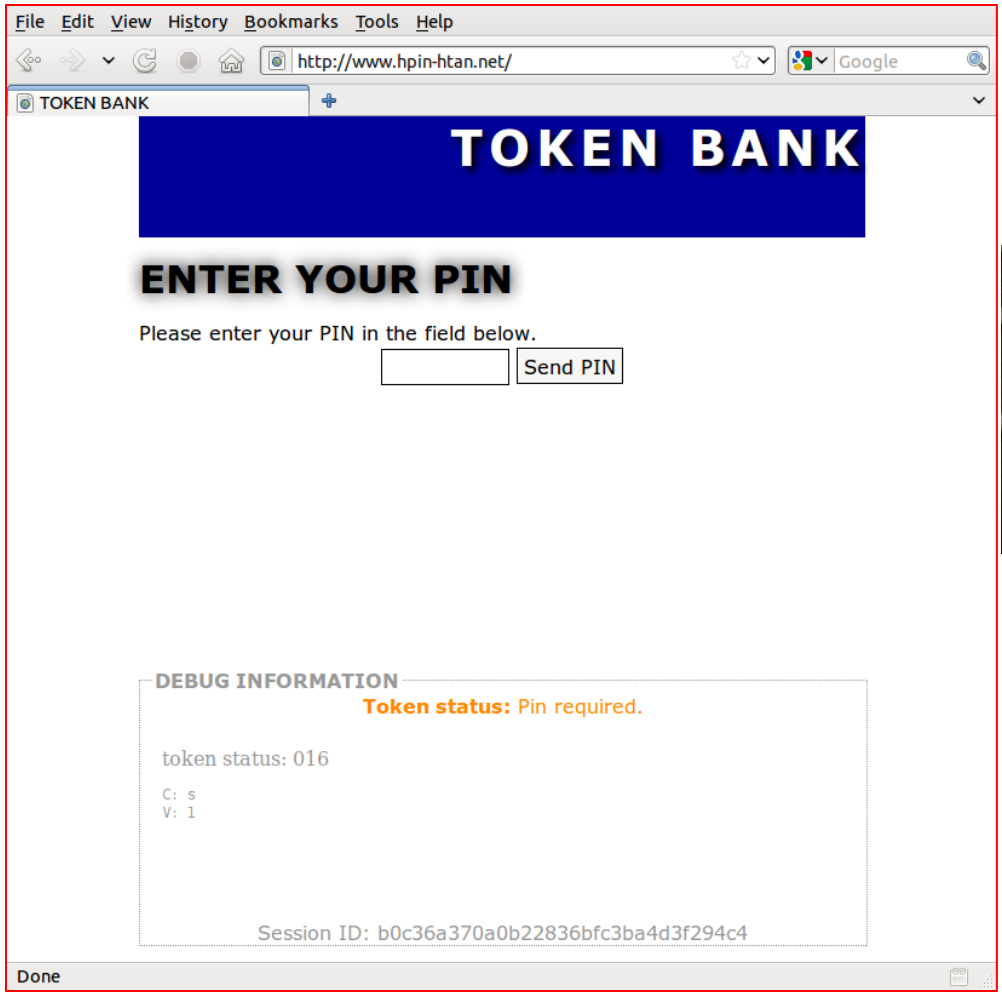
- hTAN (for transaction)



A simple HMAC based protocol ⇒ Any message authentication protocol

Prototype and live demo

- http://www.hPIN-hTAN.net



Security aspects

- PIN confidentiality
  - The one-time random code prevents exposing PIN to malware.
- User/Server authenticity
  - Guaranteed by the mutual authentication protocol in hPIN.
- Transaction integrity/authenticity
  - HCT (human-computer-token) protocol ensures transaction data integrity (H⇒T).
  - Message authentication protocol ensures STD integrity (T⇒S).
- Simplistic design ⇒ Less bugs and security holes.

Usability aspects

- A small-scale user study at our universities
  - 20 users (students & staff members, 25-49 years old)
  - Overall success login rate: 60/66 ≈ 91%
  - Median login time: 27.5 seconds
  - Median time for completing a transaction with 55 characters: 70 seconds (1.27 seconds per character)
  - Users' opinions on overall usability
    - Mean opinion score: 3.65 (moderately usable to very usable)
    - Median opinion score: 4 (very usable)

How lightweight is the token?

- Hardware
  - Microcontroller: ATmega32 @ 16 MHz
  - Program memory (Flash): 32 KB
  - Program memory (EEPROM): 1 KB
  - Data memory (RAM): 2 KB

- Software
  - Size of firmware ≈ 10 KB (can be downsized to 5-6 KB)
  - Number of lines of C code ≈ 1500 (own code) + 1100 (other's code for LCD and the SHA-1 hash function)

How costly is the token?

- Our costs: 3-5 € per token
  - Microcontroller: 1 €
  - Display: 1-3 €
  - Case: < 1 €
  - Other hardware stuff: ≤ 1 €
  - Programmer (Sören Heisrath): 0 € ☺

- Actual costs of mass production: ≤ 5 € per token?
  - Batch purchase is always much cheaper!
  - Programming costs per token is negligible: 3 man months / $O(100{,}000) \ll 1$ €.
  - The gap between the token vendor and bank customers…

hPIN/hTAN vs. Existing solutions

	Mobile /PDA	Trusted keypad	Encry ption	Optical sensor	External dependency	Smart card*
hPIN/hTAN	**No**	**No**	**No**	**No**	**No**	**No**
mTAN	Yes	No	No	No	Yes	Yes
sm@rtTAN plus	No	Yes	No	No	No	Yes
sm@rtTAN optic	No	Yes	No	Yes	No	Yes
FINREAD/FinTS	No	Yes	Yes	No	No	Yes
photoTAN	Yes	Yes	Yes	Yes	No	No
“Open Sesame”	Yes	Yes	Yes	Yes	Yes	Yes
QR-TAN	Yes	Yes	Yes	Yes	No	No
IBM ZTIC	No	No	Yes	No	No	No
AXSionics	No	No	Yes	Yes	Yes	No
MP-Auth	Yes	Yes	Yes	No	No	No

* As a compulsory component: a SIM card, a banking card, etc.

hPIN/hTAN: A summary

- Pros
  - Security guaranteed + Usability not compromised + User experience enhanced + Low cost + Scalability
- Cons
  - Changes to the server: required (same for any new e-banking solution)
  - Changes to the client (untrusted) computer: required – for communication between the web page and the USB token
  - A USB extension cable is needed?

Thanks for your attention!

Questions?

Find more at http://www.hooklee.com/default.asp?t=hPIN/hTAN

Security against other attacks

- Timing attack
  - Q: Does the user input different PIN letters with different response time?
  - A: Not likely, because she does not need to scan the whole look-up-table from left to right, but simply gaze at the position just below the next PIN letter she is going to enter.
- Physical attack
  - Getting PIN* by physically breaking the token or via a side-channel attack like power analysis: a brute force search may work since PIN is too short.
  - Possible solutions: 1) increase the PIN length; 2) increase the alphabet size; 3) slowing down the hashing process deliberately.

Security against other attacks

- Social engineering
  - PIN can be socially engineered, but K_T cannot as it is invisible to the user (so she doesn't know it, neither its existence if not told).
- Malicious code injection
  - The token is designed to be read-only at the user's end.
  - The firmware should only be updated at the bank counter.
- Insider attack
  - hPIN/hTAN can be enhanced to make it secure as long as the attacker has no simultaneous access to the communications between the user and the server.

Security against other attacks

- Collusion attack
  - Insider attack + Physical attack
  - Insider attack + MitM attack

  - = Untrusted server + Untrusted client

  - Is it possible to have a solution secure under this situation?
  - We don’t think the answer is yes.