Technology Transfer from Security Research Projects

A Personal Perspective

N. Asokan

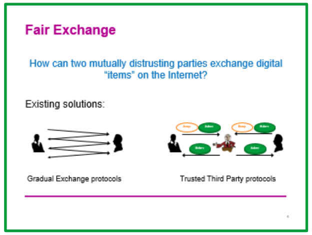
https://asokan.org/asokan/research

Aalto University

UNIVERSITY OF WATERLOO

Five examples

- Optimistic Fair Exchange
- Generic Authentication Architecture
- Channel Binding in Protocol Composition
- Secure Device Pairing
- On-board Credentials



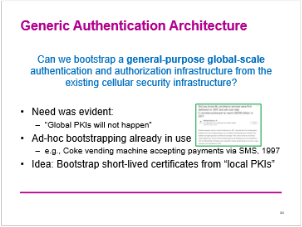
Fair Exchange

How can two mutually distrusting parties exchange digital "items" on the Internet?

Existing solutions:

Gradual Exchange protocols

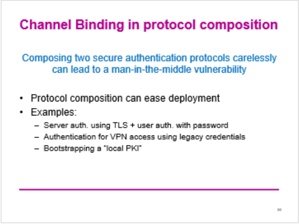
Trusted Third Party protocols



Generic Authentication Architecture

Can we bootstrap a **general-purpose global-scale** authentication and authorization infrastructure from the existing cellular security infrastructure?

- Need was evident:
  - "Global PKIs will not happen"
- Ad-hoc bootstrapping already in use
  - e.g., Coke vending machine accepting payments via SMS, 1997
- Idea: Bootstrap short-lived certificates from "local PKIs"



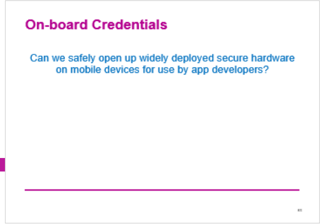
Channel Binding in protocol composition

Composing two secure authentication protocols carelessly can lead to a man-in-the-middle vulnerability

- Protocol composition can ease deployment
- Examples:
  - Server auth. using TLS + user auth. with password
  - Authentication for VPN access using legacy credentials
  - Bootstrapping a "local PKI"

Secure Device Pairing

How can the process of pairing two devices be made easy to use without compromising security or adding to cost?



On-board Credentials

Can we safely open up widely deployed secure hardware on mobile devices for use by app developers?

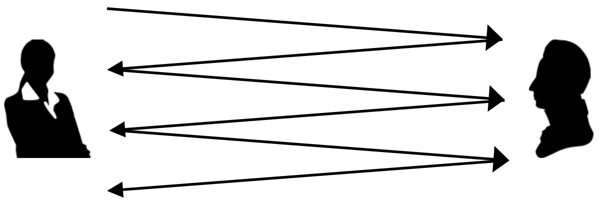
Five examples

- Optimistic Fair Exchange
- Generic Authentication Architecture
- Channel Binding in Protocol Composition
- Secure Device Pairing
- On-board Credentials

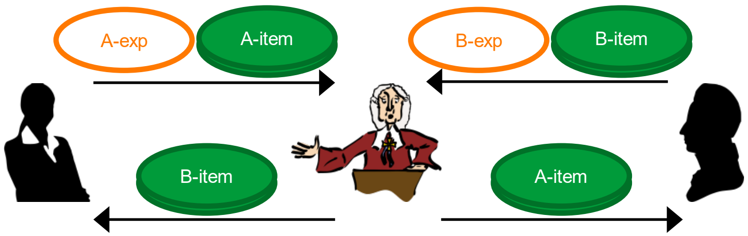
Fair Exchange

How can two mutually distrusting parties exchange digital "items" on the Internet?

Existing solutions:



Gradual Exchange protocols



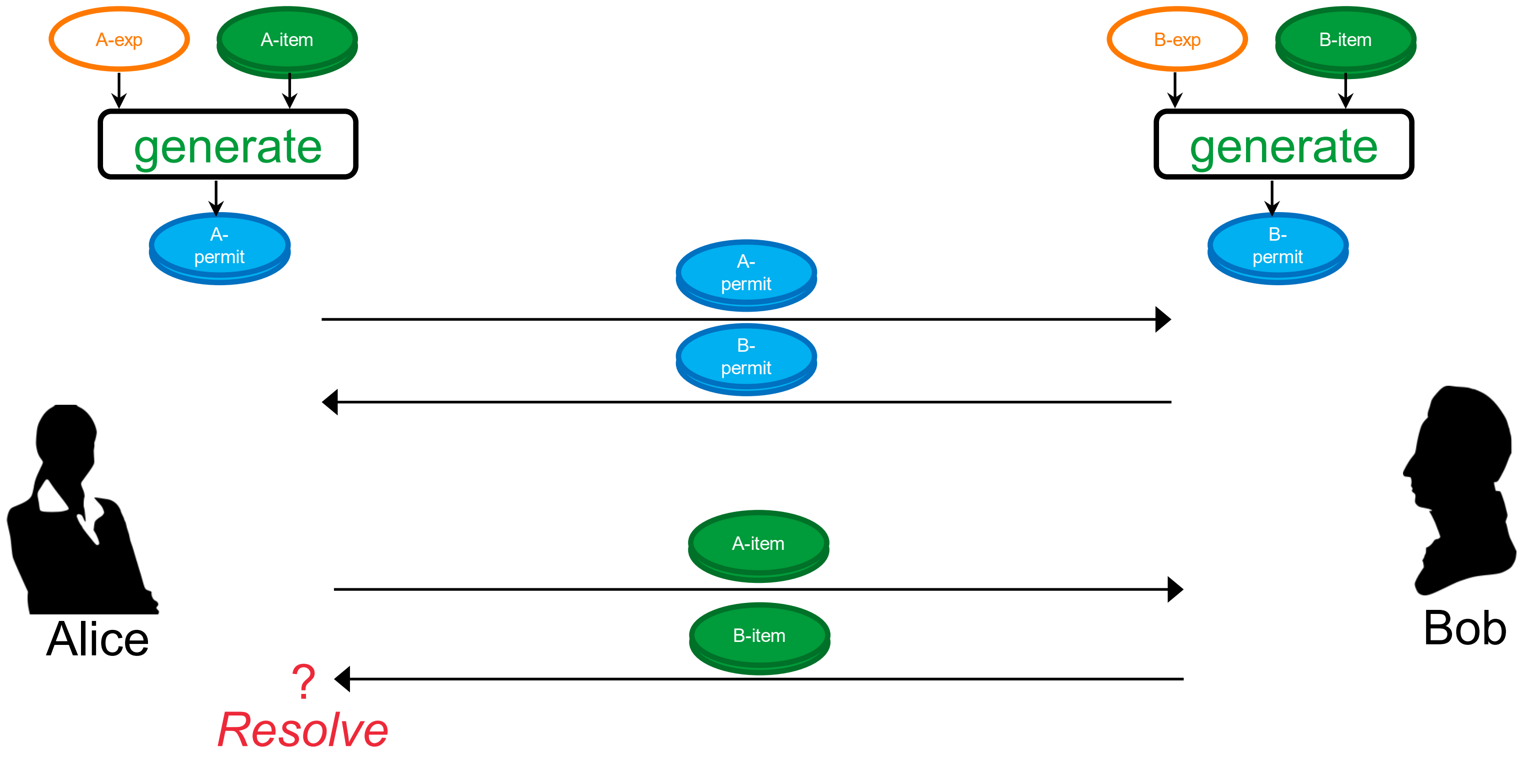
Trusted Third Party protocols

Fair Exchange: design choices

- Common case: both *want to* complete the exchange
  - design protocol that is efficient for the common case
  - but allows recovery in case of exceptions

- Requirements
  - Effectiveness
  - Fairness
  - Timeliness
  - (Non-invasive)

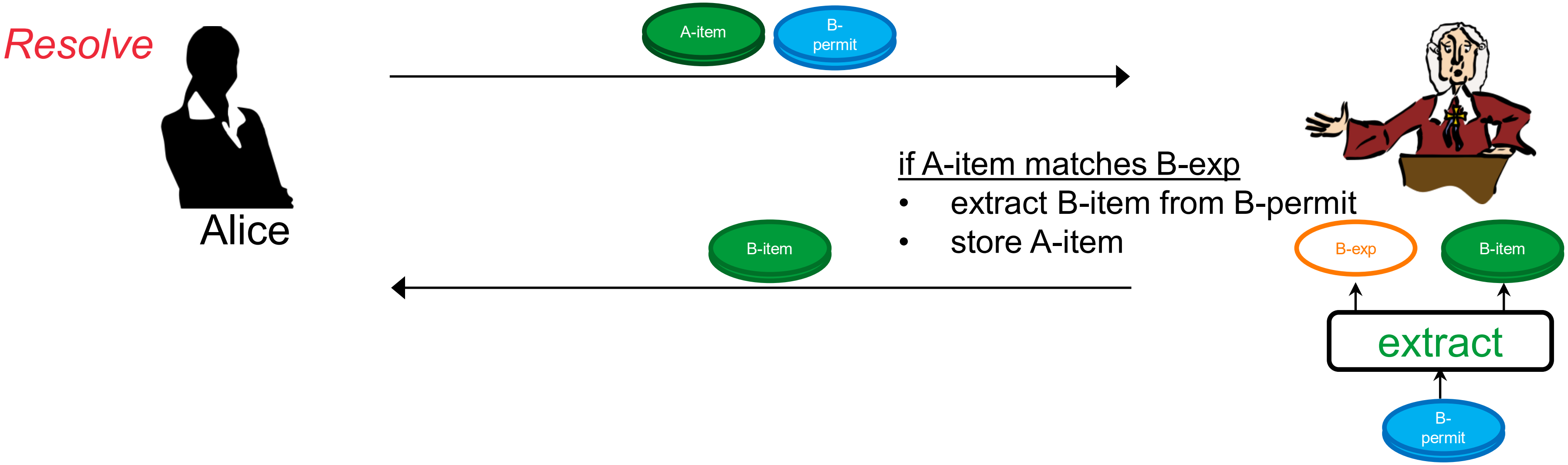
Optimistic Fair Exchange



SEMPER

http://www.semper.org/
https://web.archive.org/web/20200627000350/http://www.semper.org/
https://semper.schunter.org/

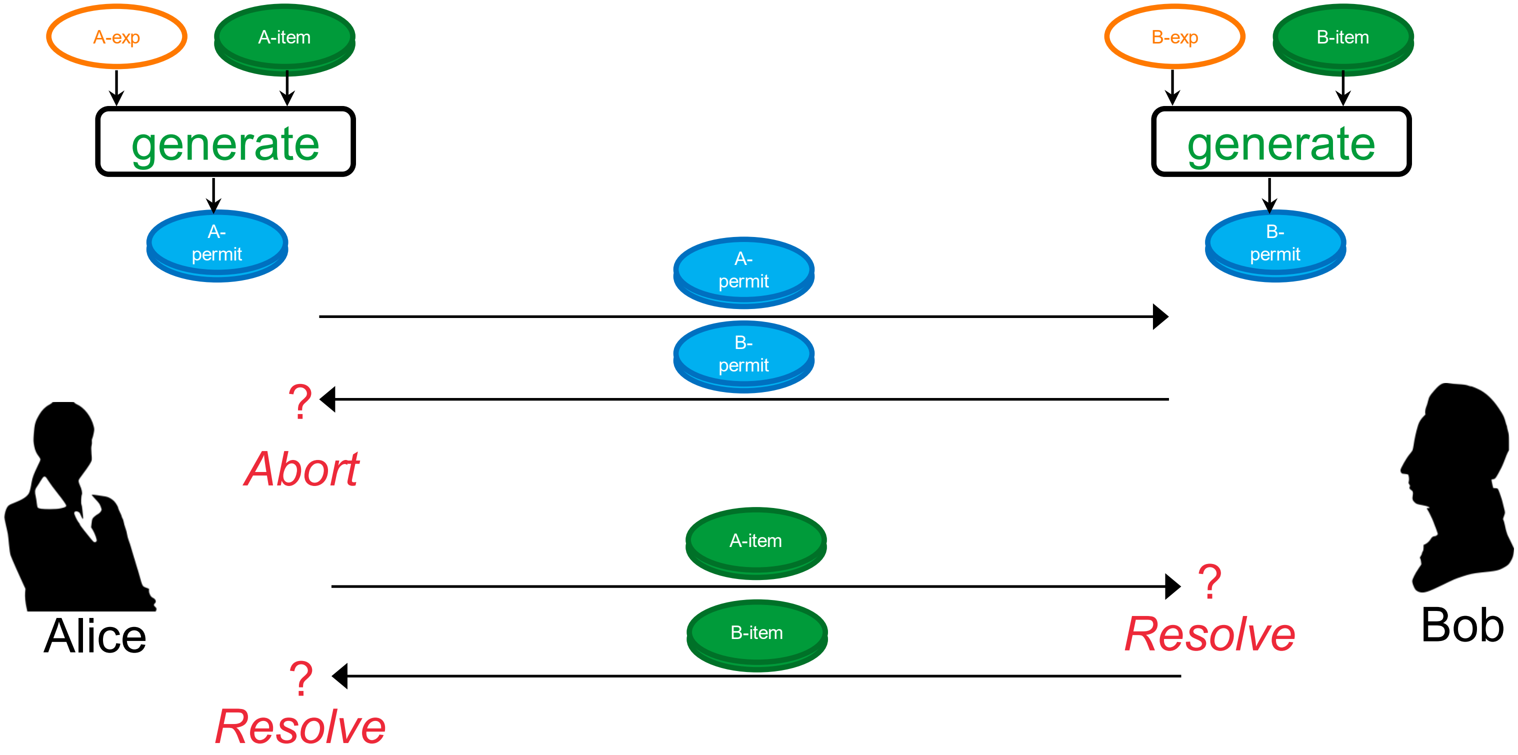
Optimistic Fair Exchange: Recovery



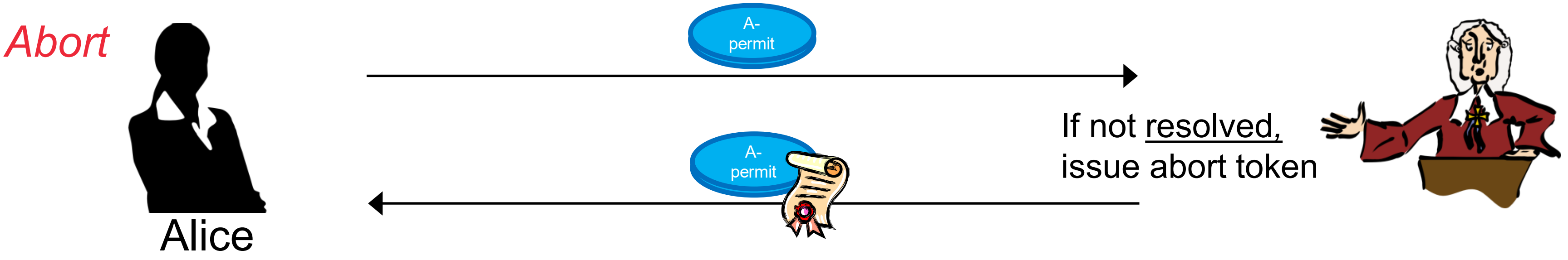
<u>if A-item matches B-exp</u>

- extract B-item from B-permit
- store A-item

Optimistic Fair Exchange

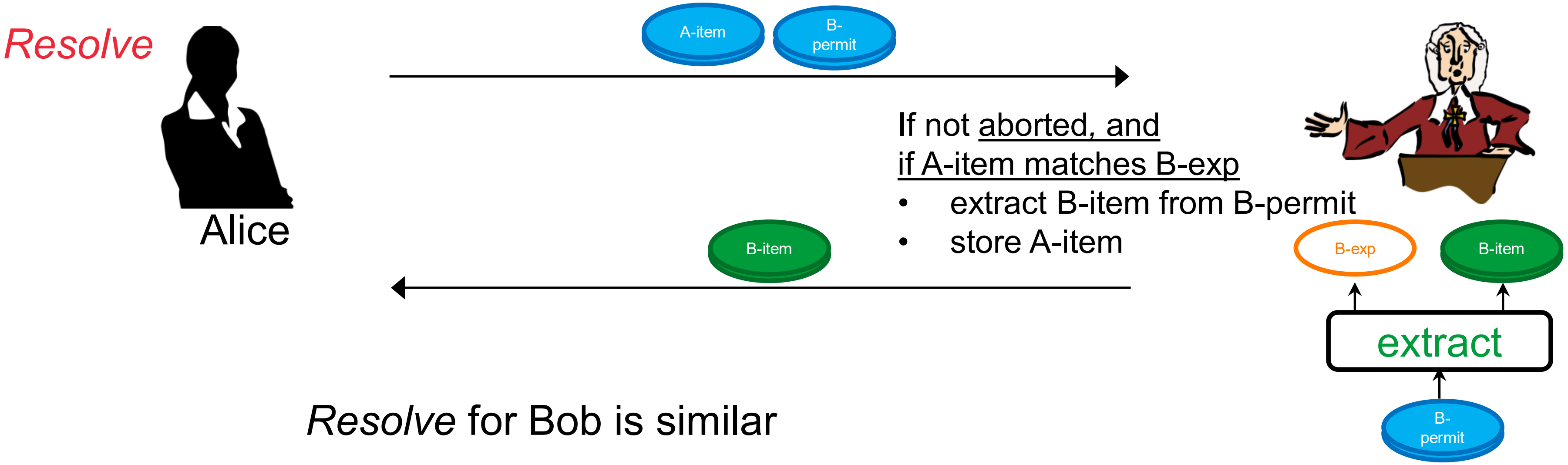


Optimistic Fair Exchange: Recovery



Abort

If not resolved, issue abort token



Resolve

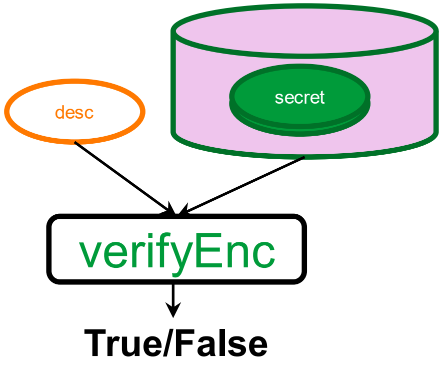
If not aborted, and if A-item matches B-exp

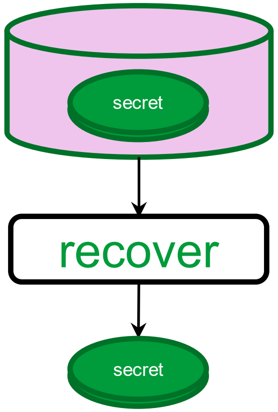
- extract B-item from B-permit
- store A-item

Resolve for Bob is similar

Verifiable Encryption

Analogy - jewelry in a glass box: can see but can’t touch

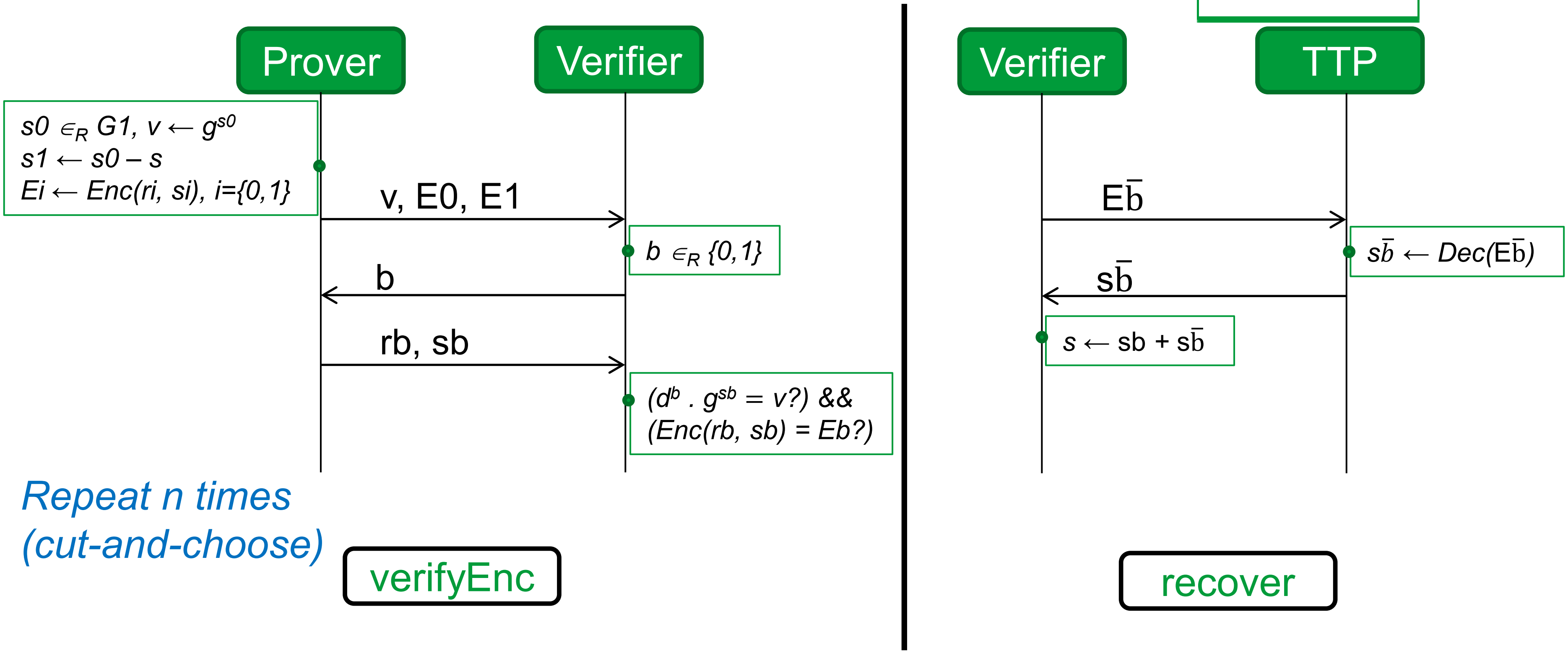




Verifiable Encryption of discrete logs

Setting: secret = s $\in$ G1, desc d = g^s (in G2)

$s\bar{b} \leftarrow Dec(E\bar{b})$



Prover

Verifier

$s0 \in_R G1$, $v \leftarrow g^{s0}$
$s1 \leftarrow s0 - s$
$Ei \leftarrow Enc(ri, si)$, $i=\{0,1\}$

v, E0, E1

$b \in_R \{0,1\}$

b

rb, sb

$(d^b \cdot g^{sb} = v?)$ &&
$(Enc(rb, sb) = Eb?)$

Repeat n times
(cut-and-choose)

verifyEnc

Verifier

TTP

$E\bar{b}$

$s\bar{b} \leftarrow Dec(E\bar{b})$

$s\bar{b}$

$s \leftarrow sb + s\bar{b}$

recover

From Verifiable Encryptions to Permits

A-exp = desc. of B-item

A-permit = Verifiable Encryption of A-exp + A-item

[ASW97] “Optimistic Protocols for Fair Exchange”, ACM CCS ‘97
[ASW98] “Asynchronous Protocols for Optimistic Fair Exchange”, IEEE S&P ‘98
[ASW00] “Optimistic Fair Exchange of Digital Signatures”, JSAC 18(4): 593-610 (2000)

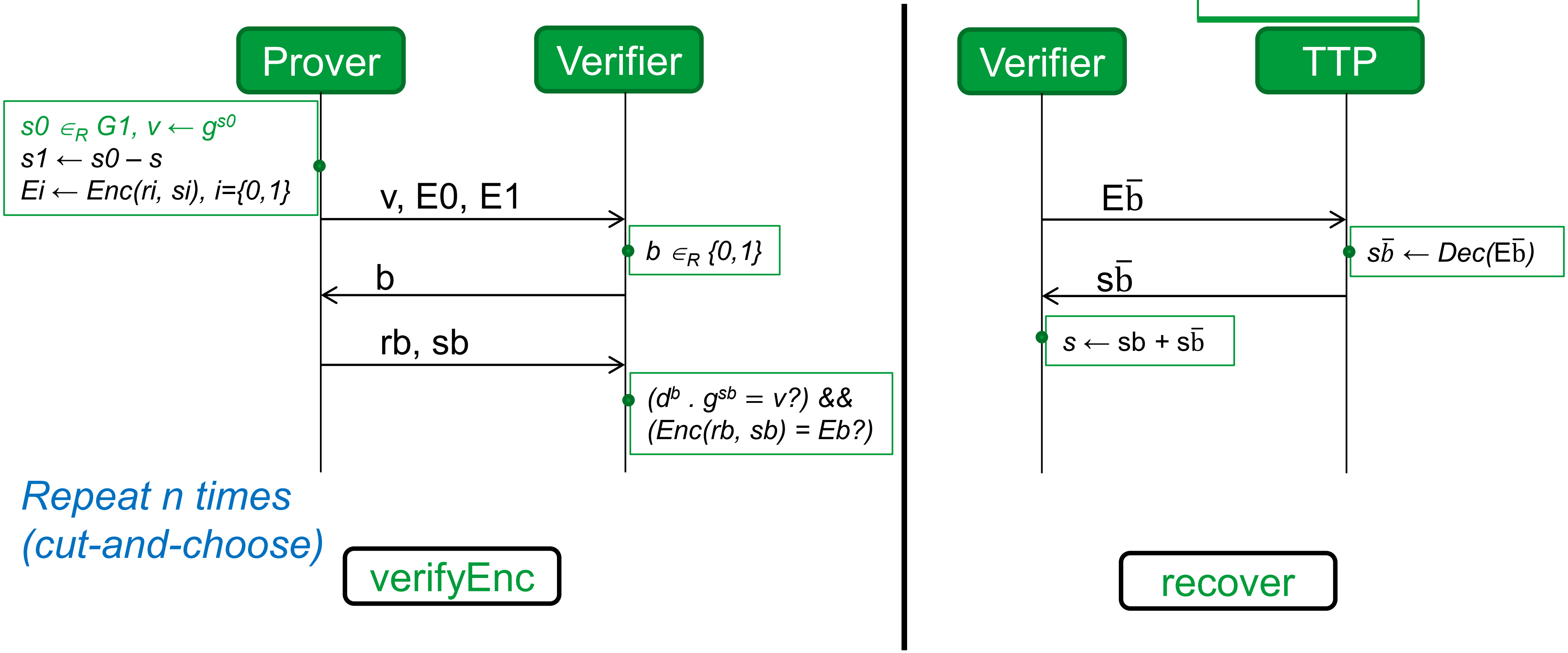
Optimistic Fair Exchange: the aftermath

- Someone has to run the Third Party
  - Wants to monetize *every* transaction!

Verifiable Encryption of discrete logs

$s\bar{b} \leftarrow Dec(E\bar{b})$

Setting: secret = s $\in$ G1, desc d = g^s (in G2)

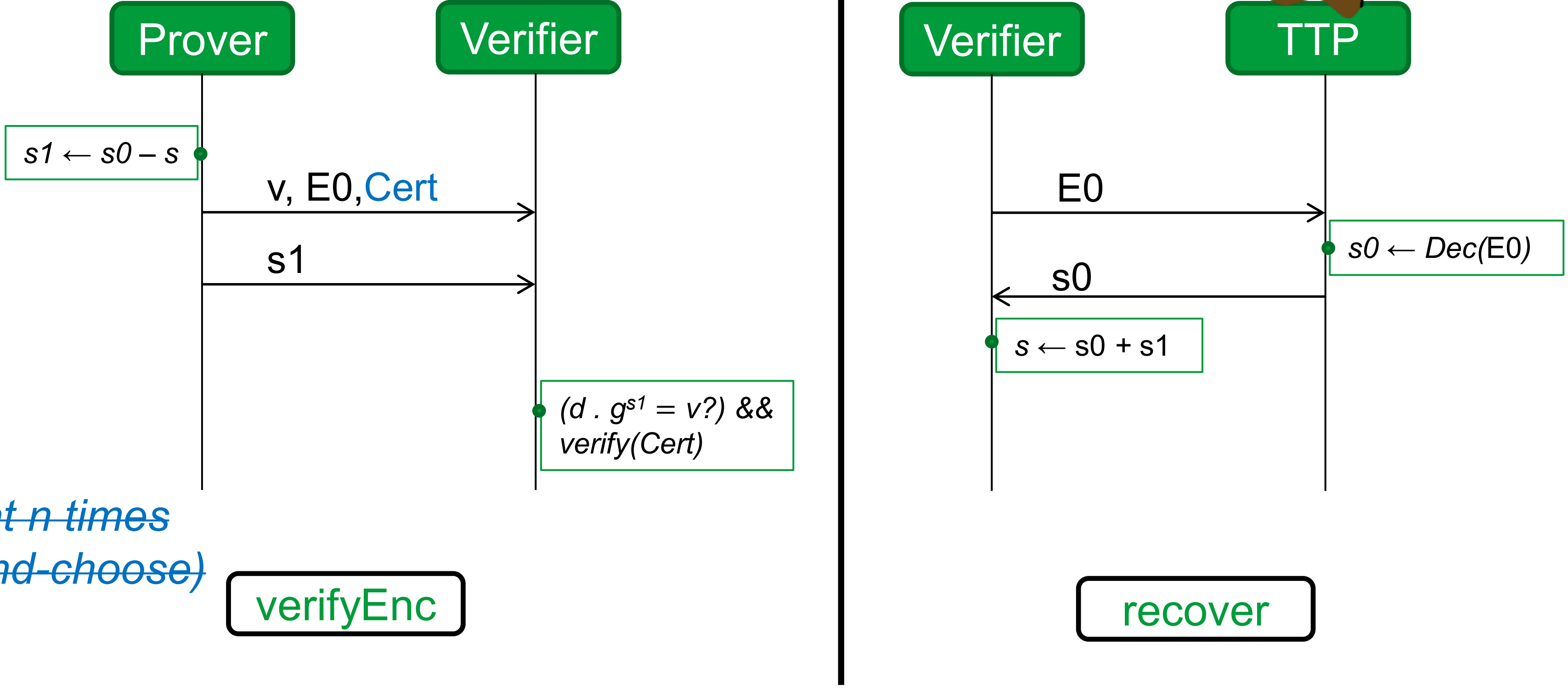

Prover
Verifier
$s0 \in_R G1,\ v \leftarrow g^{s0}$
$s1 \leftarrow s0 - s$
$Ei \leftarrow Enc(ri, si),\ i=\{0,1\}$
v, E0, E1
$b \in_R \{0,1\}$
b
rb, sb
$(d^b \,.\, g^{sb} = v?)$ &&
$(Enc(rb, sb) = Eb?)$
verifyEnc
Verifier
TTP
$E\bar{b}$
$s\bar{b} \leftarrow Dec(E\bar{b})$
$s\bar{b}$
$s \leftarrow sb + s\bar{b}$
recover

Repeat n times (cut-and-choose)

Verifiable Encryption of discrete logs

$s0 \in_R G1$, $v \leftarrow g^{s0}$
$E0 \leftarrow Enc(r0, s0)$
$Cert \leftarrow Sig_{TTP}(v, E0)$

Setting: secret = $s \in G1$, desc $d = g^s$ (in G2)



Prover

Verifier

$s1 \leftarrow s0 - s$

v, E0, Cert

s1

$(d \cdot g^{s1} = v?)$ && verify(Cert)

verifyEnc

Verifier

TTP

E0

$s0 \leftarrow Dec(E0)$

s0

$s \leftarrow s0 + s1$

recover

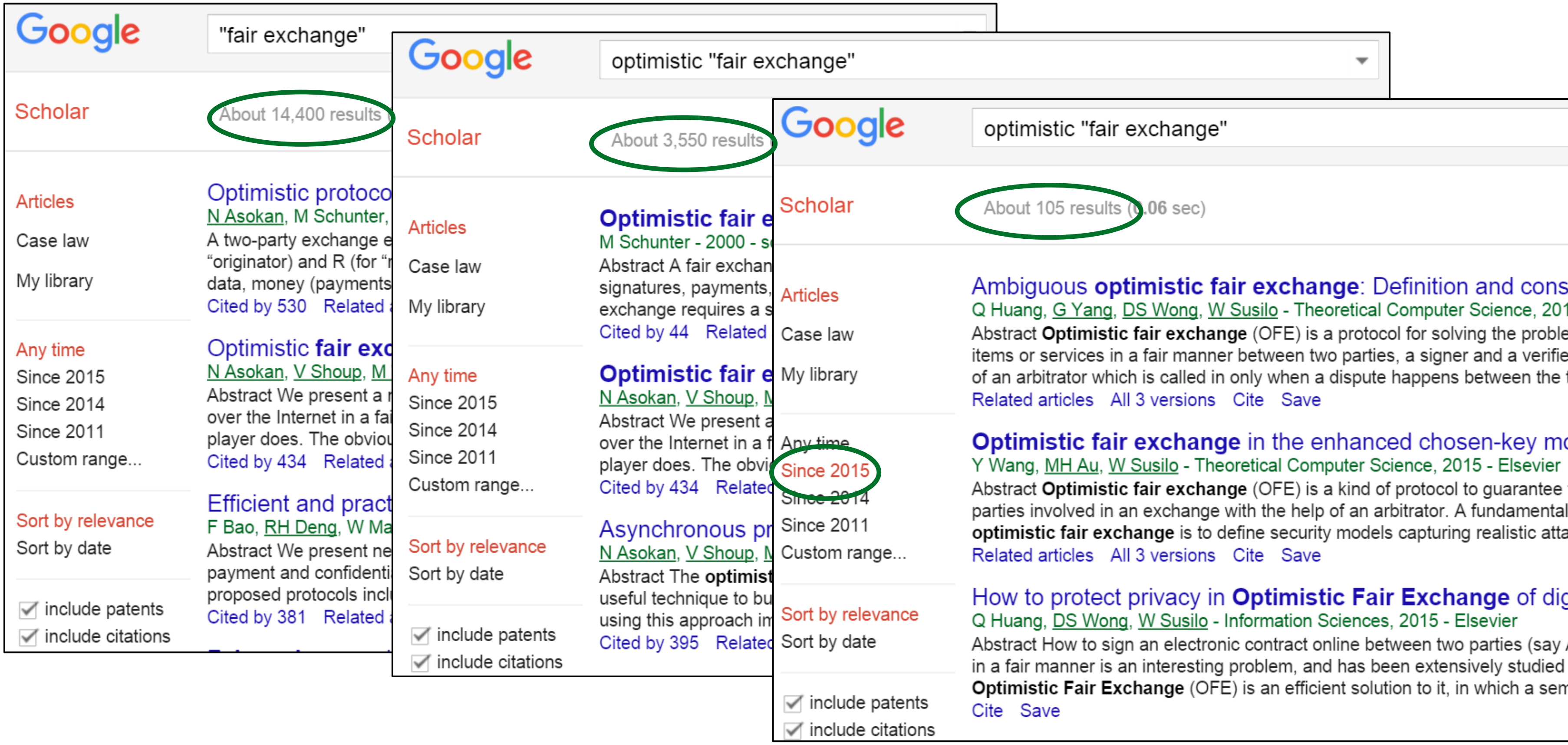
~~Repeat n times (cut-and-choose)~~

Pre-paid coupons bought from the TTP to be used for every optimistic transaction!

Optimistic Fair Exchange: the aftermath

- Someone has to run the Third Party ✓
  - Wants to monetize *every* transaction!
- Two decades on, current status:
  - Reputation systems
  - In-line TTP (e.g., E-bay escrow service)

Continuing "impact" in research circles!



Google "fair exchange"

Scholar About 14,400 results

Articles
Case law
My library

Any time
Since 2015
Since 2014
Since 2011
Custom range...

Sort by relevance
Sort by date

☑ include patents
☑ include citations

Optimistic protoco
N Asokan, M Schunter,
A two-party exchange
"originator) and R (for "
data, money (payments
Cited by 530 Related

Optimistic fair exc
N Asokan, V Shoup, M
Abstract We present a
over the Internet in a fai
player does. The obviou
Cited by 434 Related

Efficient and pract
F Bao, RH Deng, W Ma
Abstract We present ne
payment and confidenti
proposed protocols incl
Cited by 381 Related

Google optimistic "fair exchange"

Scholar About 3,550 results

Articles
Case law
My library

Any time
Since 2015
Since 2014
Since 2011
Custom range...

Sort by relevance
Sort by date

☑ include patents
☑ include citations

Optimistic fair e
M Schunter - 2000 - s
Abstract A fair exchan
signatures, payments,
exchange requires a s
Cited by 44 Related

Optimistic fair e
N Asokan, V Shoup, M
Abstract We present a
over the Internet in a f
player does. The obvi
Cited by 434 Relate

Asynchronous pr
N Asokan, V Shoup, M
Abstract The optimist
useful technique to bu
using this approach in
Cited by 395 Relate

Google optimistic "fair exchange"

Scholar About 105 results (0.06 sec)

Articles
Case law
My library

Any time
Since 2015
Since 2014
Since 2011
Custom range...

Sort by relevance
Sort by date

☑ include patents
☑ include citations

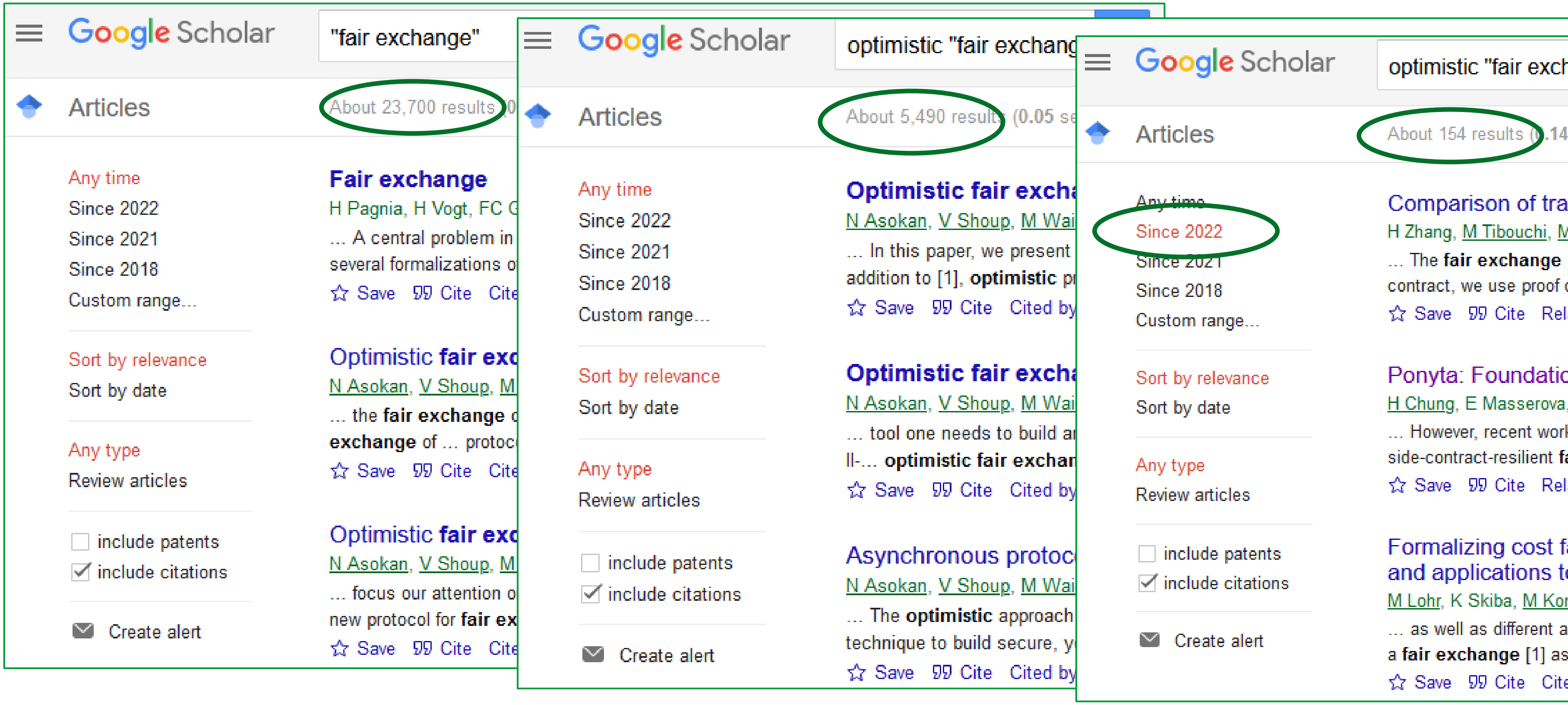
Ambiguous **optimistic fair exchange**: Definition and cons
Q Huang, G Yang, DS Wong, W Susilo - Theoretical Computer Science, 201
Abstract **Optimistic fair exchange** (OFE) is a protocol for solving the proble
items or services in a fair manner between two parties, a signer and a verifie
of an arbitrator which is called in only when a dispute happens between the
Related articles All 3 versions Cite Save

Optimistic fair exchange in the enhanced chosen-key mo
Y Wang, MH Au, W Susilo - Theoretical Computer Science, 2015 - Elsevier
Abstract **Optimistic fair exchange** (OFE) is a kind of protocol to guarantee
parties involved in an exchange with the help of an arbitrator. A fundamental
optimistic fair exchange is to define security models capturing realistic atta
Related articles All 3 versions Cite Save

How to protect privacy in **Optimistic Fair Exchange** of dig
Q Huang, DS Wong, W Susilo - Information Sciences, 2015 - Elsevier
Abstract How to sign an electronic contract online between two parties (say
in a fair manner is an interesting problem, and has been extensively studied
Optimistic Fair Exchange (OFE) is an efficient solution to it, in which a sem
Cite Save

Autumn 2015

Continuing “impact” in research circles!



Google Scholar "fair exchange"

Articles About 23,700 results (0

Any time
Since 2022
Since 2021
Since 2018
Custom range...

Sort by relevance
Sort by date

Any type
Review articles

□ include patents
☑ include citations

✉ Create alert

Fair exchange
H Pagnia, H Vogt, FC G
... A central problem in
several formalizations o
☆ Save ❞ Cite Cite

Optimistic **fair exc**
N Asokan, V Shoup, M
... the **fair exchange** c
exchange of ... protoc
☆ Save ❞ Cite Cite

Optimistic **fair exc**
N Asokan, V Shoup, M
... focus our attention o
new protocol for **fair ex**
☆ Save ❞ Cite Cite

Google Scholar optimistic "fair exchang

Articles About 5,490 results (0.05 se

Any time
Since 2022
Since 2021
Since 2018
Custom range...

Sort by relevance
Sort by date

Any type
Review articles

□ include patents
☑ include citations

✉ Create alert

Optimistic fair excha
N Asokan, V Shoup, M Wai
... In this paper, we present
addition to [1], **optimistic** pr
☆ Save ❞ Cite Cited by

Optimistic fair excha
N Asokan, V Shoup, M Wai
... tool one needs to build ar
II-... **optimistic fair exchan**
☆ Save ❞ Cite Cited by

Asynchronous protoc
N Asokan, V Shoup, M Wai
... The **optimistic** approach
technique to build secure, y
☆ Save ❞ Cite Cited by

Google Scholar optimistic "fair exch

Articles About 154 results (0.14

Any time
Since 2022
Since 2021
Since 2018
Custom range...

Sort by relevance
Sort by date

Any type
Review articles

□ include patents
☑ include citations

✉ Create alert

Comparison of tra
H Zhang, M Tibouchi, M
... The **fair exchange**
contract, we use proof o
☆ Save ❞ Cite Rel

Ponyta: Foundatio
H Chung, E Masserova,
... However, recent wor
side-contract-resilient **fa**
☆ Save ❞ Cite Rel

Formalizing cost f
and applications t
M Lohr, K Skiba, M Kor
... as well as different a
a **fair exchange** [1] as
☆ Save ❞ Cite Cite

Nov 2022

Optimistic Fair Exchange: the aftermath

- Someone has to run the Third Party
  - Wants to monetize *every* transaction!

- Two decades on, current status:
  - Reputation systems
  - In-line TTP (e.g., E-bay escrow service)

- Impact in academia vs. real world impact

- Biggest impact of SEMPER?

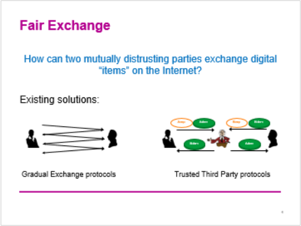
http://logging.apache.org/log4j/2.x/

Optimistic Fair Exchange: lessons learned

- Don’t just guess security requirements; Ask stakeholders
- Desiderata for deployment and research can be different
  - “the more (independent) parties you require for your scheme, the less likely it will be deployed”
- Capturing researcher interest $\nrightarrow$ (Tech transfer) Impact
  - MANETs anyone?
- “90-10 rule” applies to deploying security
  - “Good enough beats perfect”

Five examples

- Optimistic Fair Exchange
- Generic Authentication Architecture
- Channel Binding in Protocol Composition
- Secure Device Pairing
- On-board Credentials



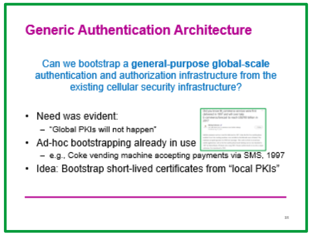
Fair Exchange

How can two mutually distrusting parties exchange digital "items" on the Internet?

Existing solutions:

Gradual Exchange protocols

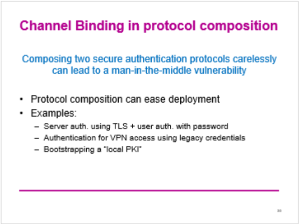
Trusted Third Party protocols



Generic Authentication Architecture

Can we bootstrap a **general-purpose global-scale** authentication and authorization infrastructure from the existing cellular security infrastructure?

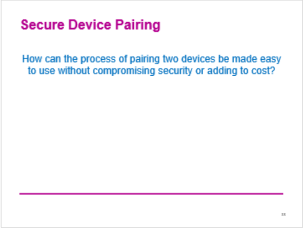
- Need was evident:
  - "Global PKIs will not happen"
- Ad-hoc bootstrapping already in use
  - e.g., Coke vending machine accepting payments via SMS, 1997
- Idea: Bootstrap short-lived certificates from "local PKIs"



Channel Binding in protocol composition

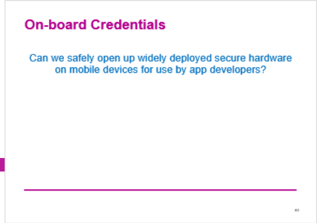
Composing two secure authentication protocols carelessly can lead to a man-in-the-middle vulnerability

- Protocol composition can ease deployment
- Examples:
  - Server auth. using TLS + user auth. with password
  - Authentication for VPN access using legacy credentials
  - Bootstrapping a "local PKI"



Secure Device Pairing

How can the process of pairing two devices be made easy to use without compromising security or adding to cost?



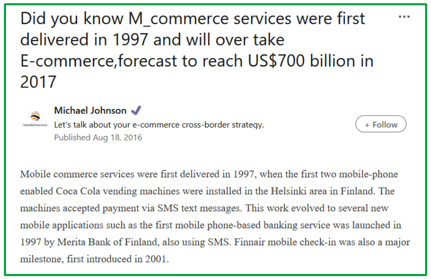
On-board Credentials

Can we safely open up widely deployed secure hardware on mobile devices for use by app developers?

Generic Authentication Architecture

Can we bootstrap a **general-purpose global-scale** authentication and authorization infrastructure from the existing cellular security infrastructure?

- Need was evident:
  - "Global PKIs will not happen"
- Ad-hoc bootstrapping already in use
  - e.g., Coke vending machine accepting payments via SMS, 1997
- Idea: Bootstrap short-lived certificates from "local PKIs"



Did you know M_commerce services were first delivered in 1997 and will over take E-commerce,forecast to reach US$700 billion in 2017

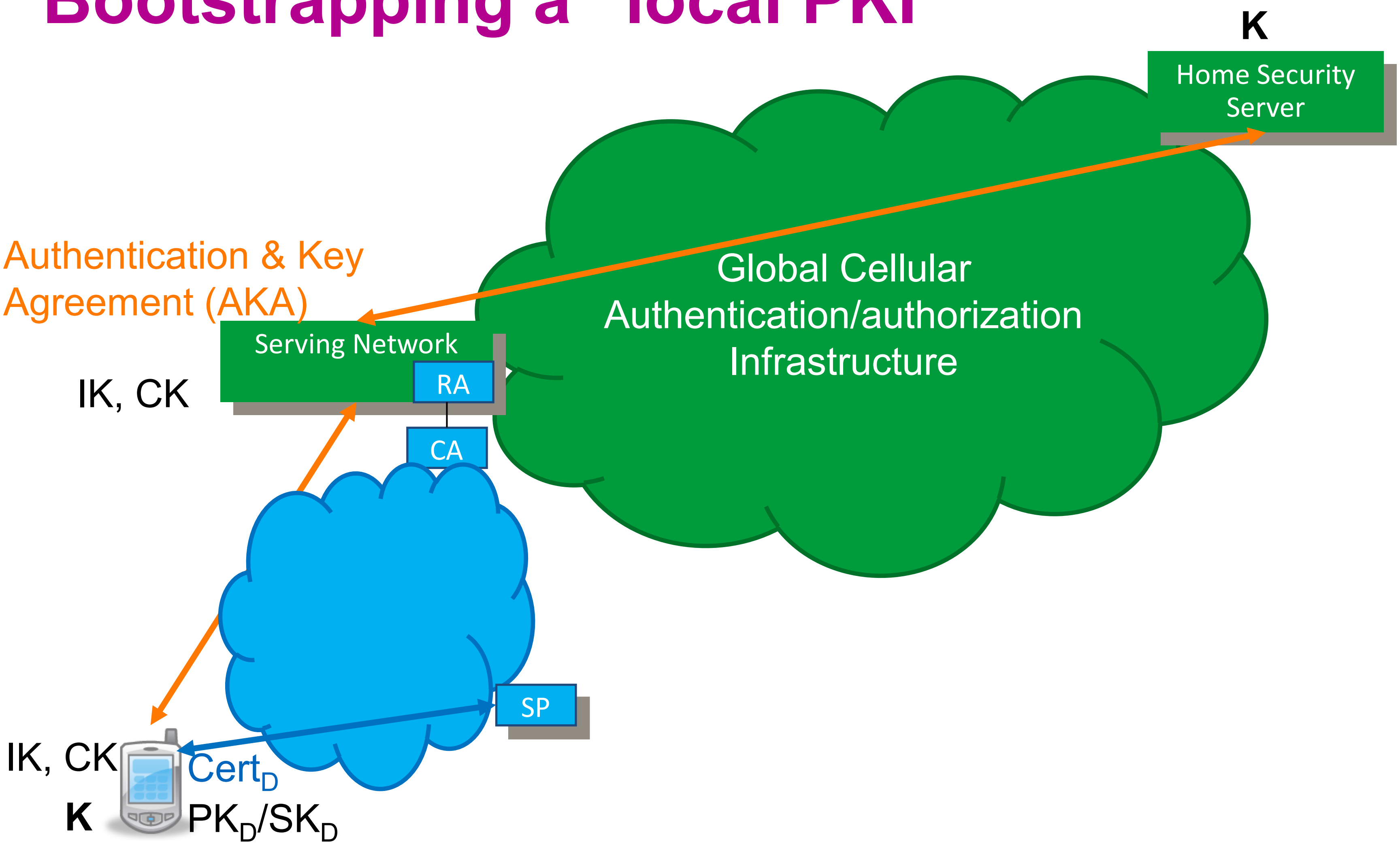
Michael Johnson

Let's talk about your e-commerce cross-border strategy.

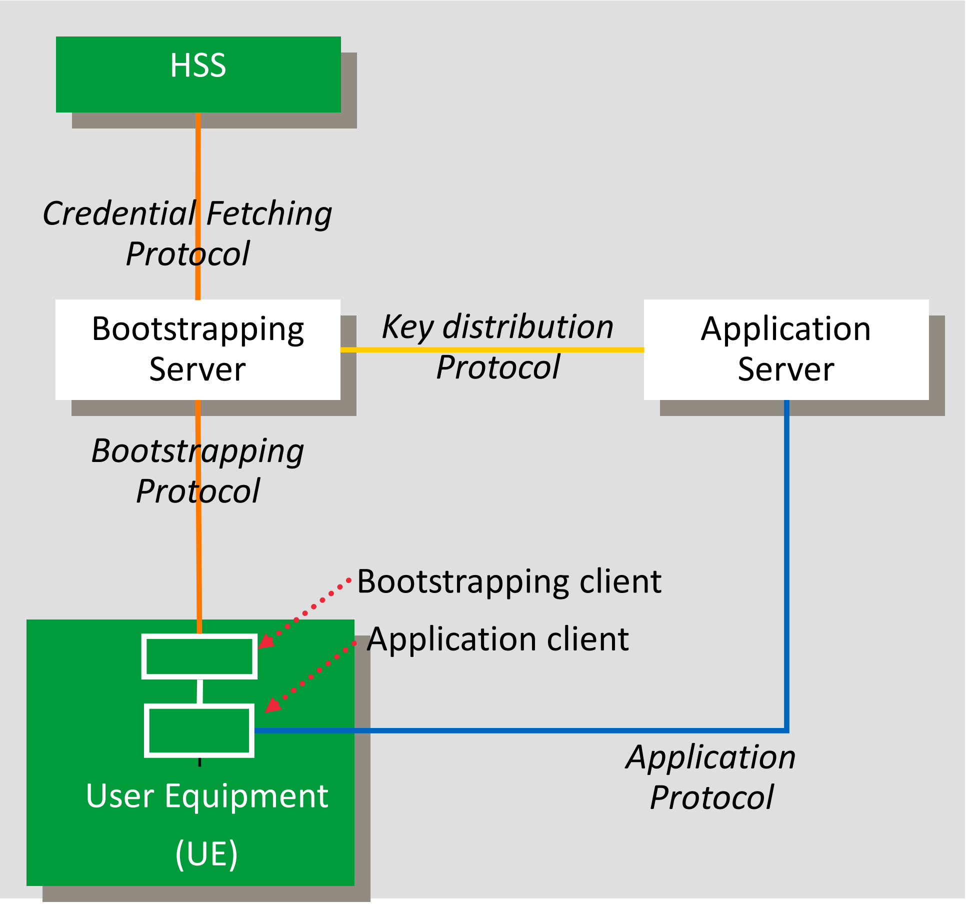
Published Aug 18, 2016

Mobile commerce services were first delivered in 1997, when the first two mobile-phone enabled Coca Cola vending machines were installed in the Helsinki area in Finland. The machines accepted payment via SMS text messages. This work evolved to several new mobile applications such as the first mobile phone-based banking service was launched in 1997 by Merita Bank of Finland, also using SMS. Finnair mobile check-in was also a major milestone, first introduced in 2001.

Bootstrapping a “local PKI”


K
Home Security Server
Authentication & Key Agreement (AKA)
Global Cellular Authentication/authorization Infrastructure
Serving Network
RA
IK, CK
CA
SP
IK, CK
K
Cert_D
PK_D/SK_D

3GPP “Generic Authentication Architecture”



Two-layer architecture

- Generic Bootstrapping Architecture (GBA)
- Specialized Application Servers
  - E.g., for “subscriber certificates”

[HLGNA08] “Cellular Authentication for Mobile and Internet Services”, Wiley, 2008
Relevant 3GPP documents: E.g., [33.919], [33.220]

GAA: the aftermath

- Standardized in 3GPP
  - Variants: GBA and GBA_U (implemented in the smartcard, UICC)
  - GBA implemented for some services
  - none of which has taken off (e.g., Mobile TV)
    - At least not yet!
- Today's solutions:
  - Bootstrapping: Facebook, Google, …
    - Some mobile carriers even deployed PKI-enabled SIM cards
  - Roaming: iPass, Shibboleth, …
- Variants of the idea had more success
  - E.g., EAP SIM

GAA: lessons learned

- (Standardization) Politics can suffocate a good idea
- (Tech transfer) Impact ↛ Capturing researcher interest
- “90-10 rule” applies to deploying security

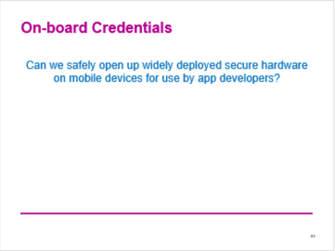
Lessons Learned

https://asokan.org/asokan/research

- How to choose the "right" problems?
  - Don't just guess security requirements; Ask stakeholders
  - Desiderata for deployment and research can be different
  - "90-10 rule" applies to deploying security
- How to identify "good" results?
  - Negative results are useful for security practitioners
  - Capturing researcher interest ↛ (Tech transfer) Impact
  - (Tech transfer) Impact ↛ Capturing researcher interest
- How to find paths to deployment?
  - Address pain points - builds credibility with stakeholders
  - (Standardization) Politics can suffocate a good idea
  - Standardization can make a good idea see light of day

The remaining examples

- Channel Binding in Protocol Composition
  - Do we tend to compose two secure authentication protocols carelessly? (Greater awareness, but continue to recur)
- Secure Device Pairing
  - How to make pairing secure but easy-to-use? (Bluetooth Secure Simple Pairing)
- On-board Credentials
  - How to make hardware TEEs safely accessible to developers? (Deployments in Nokia devices, but quietly!)
- (New) lessons learned
  - (Tech transfer) Impact   Capturing researcher interest
  - Negative results are useful for security practitioners
  - Address pain points - builds credibility with stakeholders
  - Standardization can make a good idea see light of day

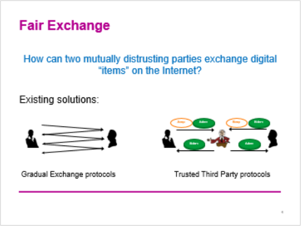


On-board Credentials

Can we safely open up widely deployed secure hardware on mobile devices for use by app developers?

Five examples

- Optimistic Fair Exchange
- Generic Authentication Architecture
- Channel Binding in Protocol Composition
- Secure Device Pairing
- On-board Credentials



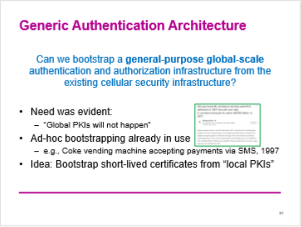
Fair Exchange

How can two mutually distrusting parties exchange digital "items" on the Internet?

Existing solutions:

Gradual Exchange protocols

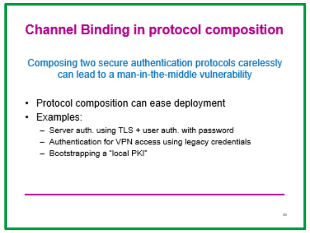
Trusted Third Party protocols



Generic Authentication Architecture

Can we bootstrap a **general-purpose global-scale** authentication and authorization infrastructure from the existing cellular security infrastructure?

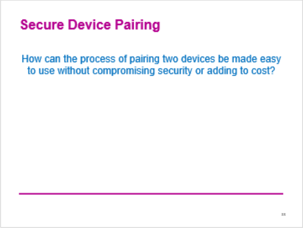
- Need was evident:
  - "Global PKIs will not happen"
- Ad-hoc bootstrapping already in use
  - e.g., Coke vending machine accepting payments via SMS, 1997
- Idea: Bootstrap short-lived certificates from "local PKIs"



Channel Binding in protocol composition

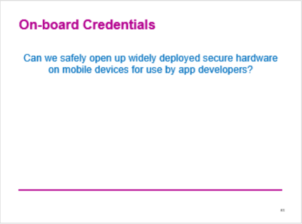
Composing two secure authentication protocols carelessly can lead to a man-in-the-middle vulnerability

- Protocol composition can ease deployment
- Examples:
  - Server auth. using TLS + user auth. with password
  - Authentication for VPN access using legacy credentials
  - Bootstrapping a "local PKI"



Secure Device Pairing

How can the process of pairing two devices be made easy to use without compromising security or adding to cost?



On-board Credentials

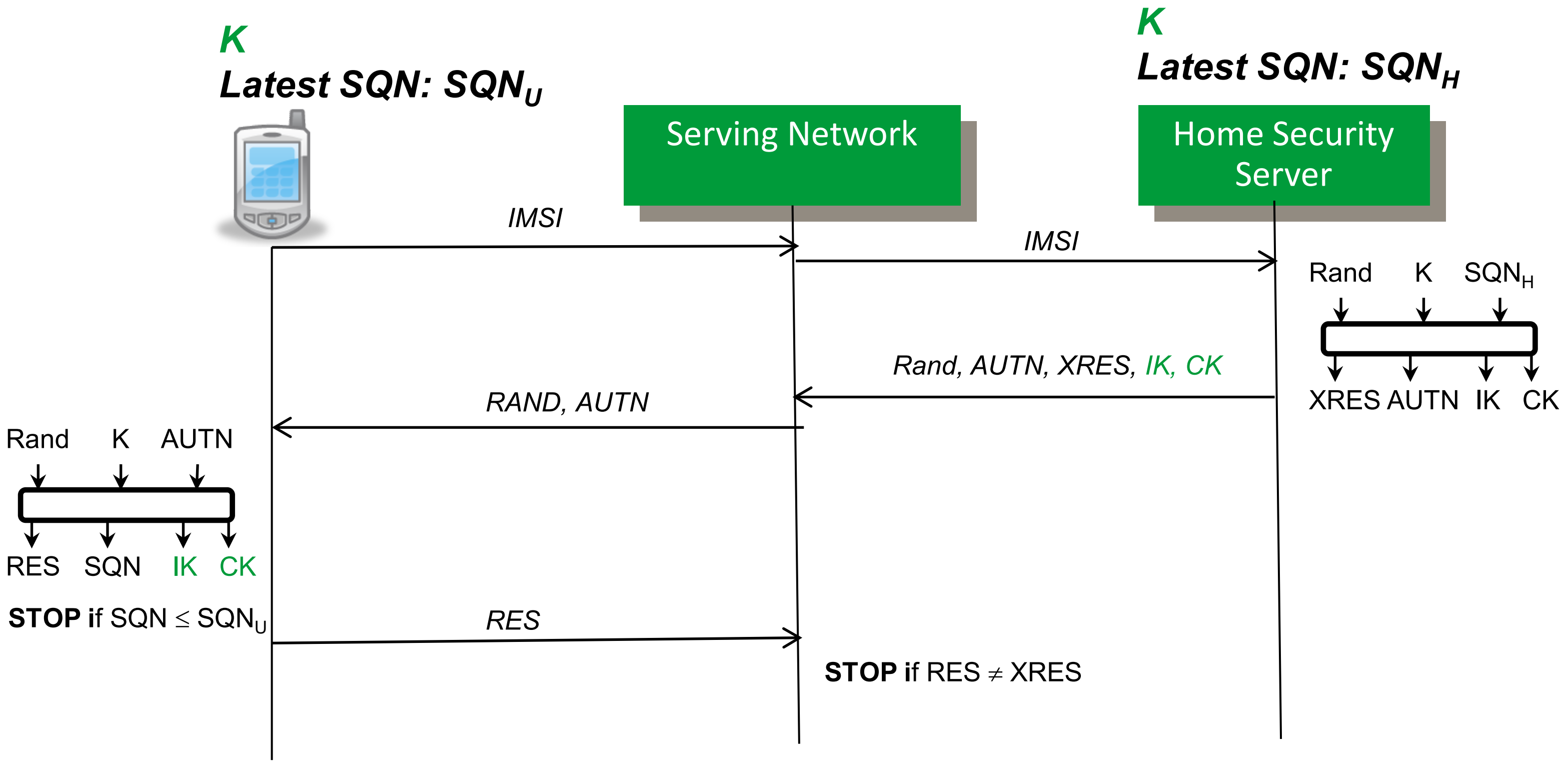
Can we safely open up widely deployed secure hardware on mobile devices for use by app developers?

Channel Binding in protocol composition

Composing two secure authentication protocols carelessly can lead to a man-in-the-middle vulnerability

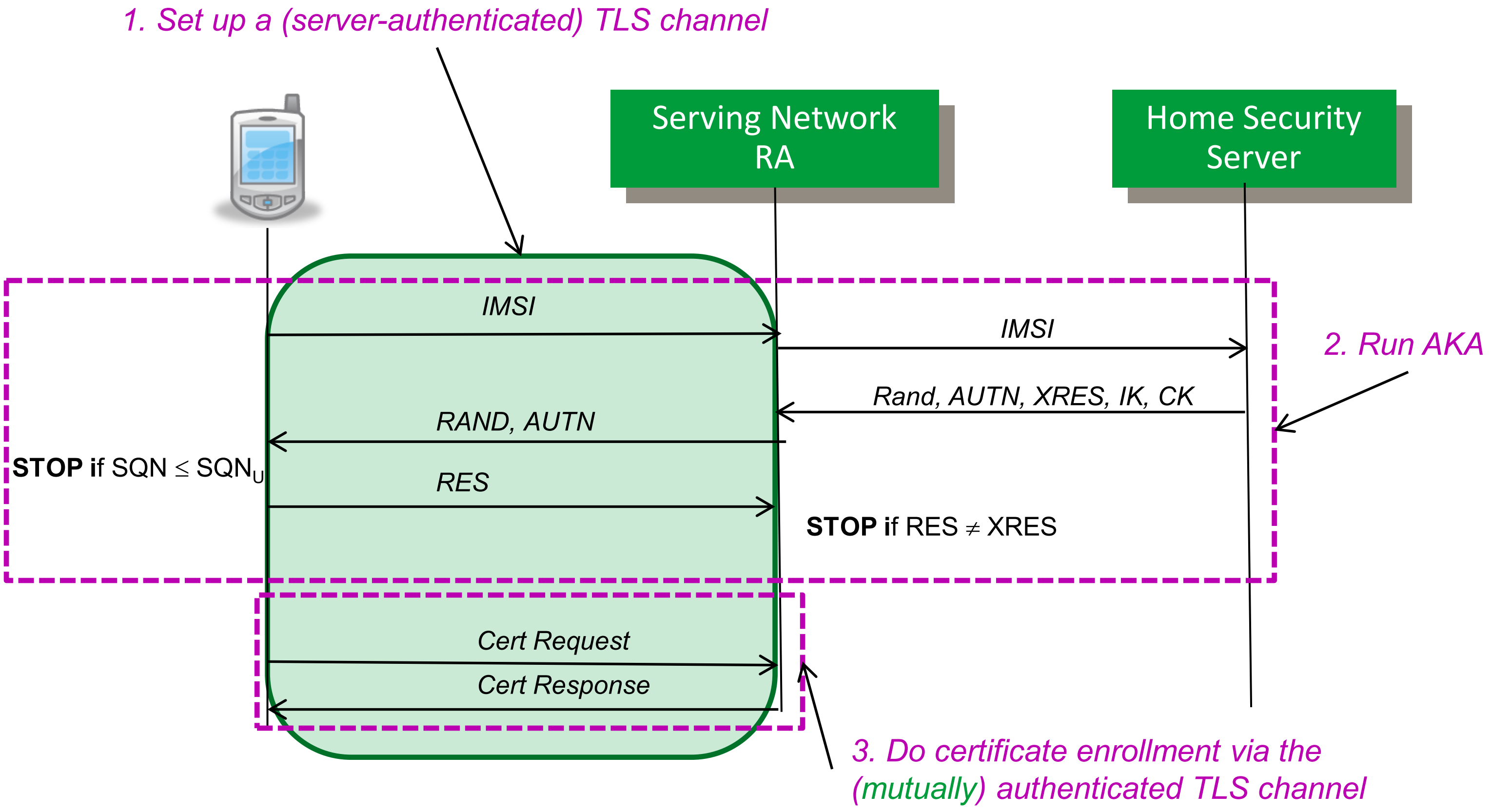
- Protocol composition can ease deployment
- Examples:
  - Server auth. using TLS + user auth. with password
  - Authentication for VPN access using legacy credentials
  - Bootstrapping a “local PKI”

3G AKA

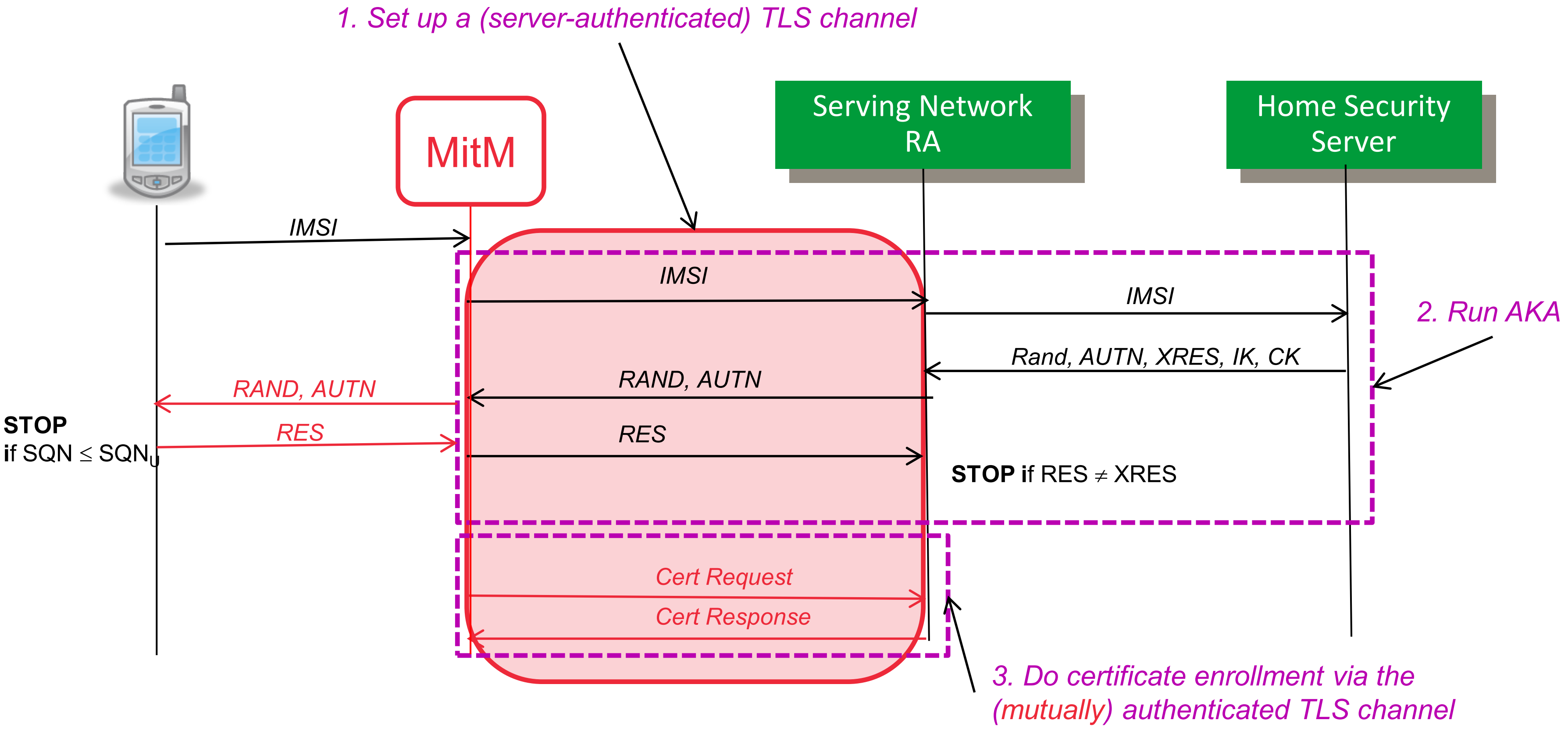


Provides mutual authentication

Bootstrapping certificate enrollment



Bootstrapping certificate enrollment

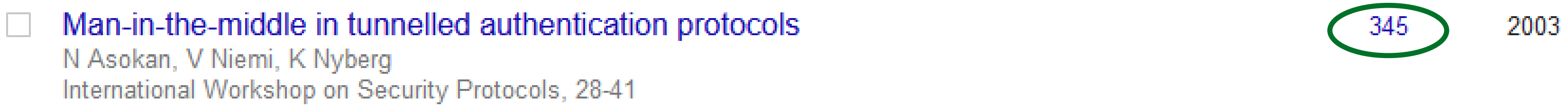


Channel binding: Use of **cryptographic binding** to compose two authenticated channels

[ANN03] “Man-in-the-middle in Tunnelled Authentication Protocols”, Security Protocols, 2003

Channel binding: the aftermath

- Fiery reception at Security Protocols workshop!
  - “But you are using the worst rackets in industry as a justification for what you’re doing. There are all sorts of people just generating garbage protocols, a couple of which you have already mentioned here. We’re trying to reverse their work, whereas you’re trying to advocate we use all these garbage protocols.”
  - For an entertaining read, see transcript of discussion during my talk at SPW ’03!
- Impact in IETF
  - Closing down of *ipsra* working group; channel binding in IKEv2
  - Continued attention: e.g., RFC 6813



□ Man-in-the-middle in tunnelled authentication protocols 345 2003
N Asokan, V Niemi, K Nyberg
International Workshop on Security Protocols, 28-41

Channel Binding: lessons learned

- Negative results are useful for security practitioners
- Standardization can make a good idea see light of day
- (Tech transfer) Impact ↛ Capturing researcher interest

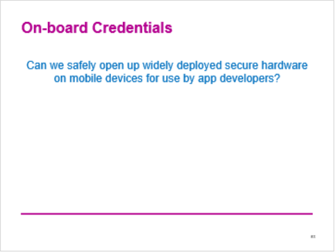
The remaining examples

Lessons Learned

https://asokan.org/asokan/research

- How to choose the “right” problems?
  - Don’t just guess security requirements; Ask stakeholders
  - Desiderata for deployment and research can be different
  - “90-10 rule” applies to deploying security
- How to identify “good” results?
  - Negative results are useful for security practitioners
  - Capturing researcher interest $\nrightarrow$ (Tech transfer) Impact
  - (Tech transfer) Impact $\nrightarrow$ Capturing researcher interest
- How to find paths to deployment?
  - Address pain points - builds credibility with stakeholders
  - (Standardization) Politics can suffocate a good idea
  - Standardization can make a good idea see light of day

- Secure Device Pairing
  - How to make pairing secure but easy-to-use? (Bluetooth Secure Simple Pairing)
- On-board Credentials
  - How to make hardware TEEs safely accessible to developers? (Deployments in Nokia devices, but quietly!)
- New lessons learned
  - Address pain points - builds credibility with stakeholders

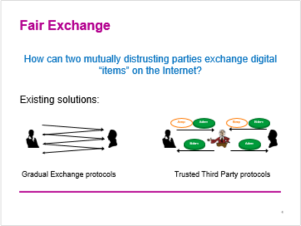


On-board Credentials

Can we safely open up widely deployed secure hardware on mobile devices for use by app developers?

Five examples

- Optimistic Fair Exchange
- Generic Authentication Architecture
- Channel Binding in Protocol Composition
- Secure Device Pairing
- On-board Credentials



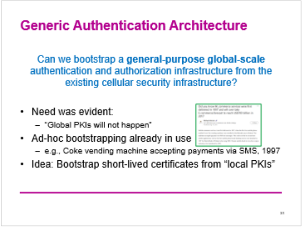
Fair Exchange

How can two mutually distrusting parties exchange digital "items" on the Internet?

Existing solutions:

Gradual Exchange protocols

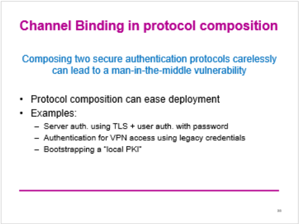
Trusted Third Party protocols



Generic Authentication Architecture

Can we bootstrap a **general-purpose global-scale** authentication and authorization infrastructure from the existing cellular security infrastructure?

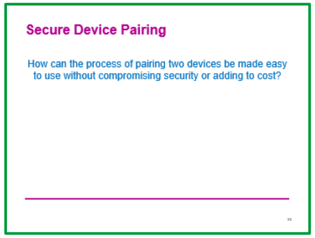
- Need was evident:
  - "Global PKIs will not happen"
- Ad-hoc bootstrapping already in use
  - e.g., Coke vending machine accepting payments via SMS, 1997
- Idea: Bootstrap short-lived certificates from "local PKIs"



Channel Binding in protocol composition

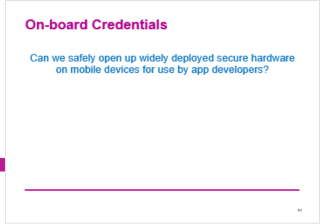
Composing two secure authentication protocols carelessly can lead to a man-in-the-middle vulnerability

- Protocol composition can ease deployment
- Examples:
  - Server auth. using TLS + user auth. with password
  - Authentication for VPN access using legacy credentials
  - Bootstrapping a "local PKI"



Secure Device Pairing

How can the process of pairing two devices be made easy to use without compromising security or adding to cost?



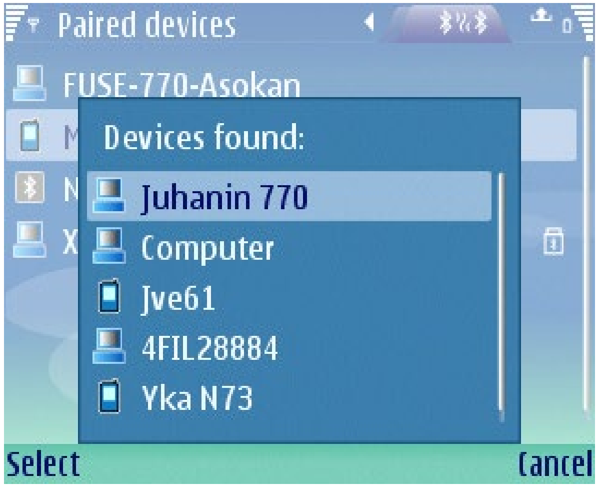
On-board Credentials

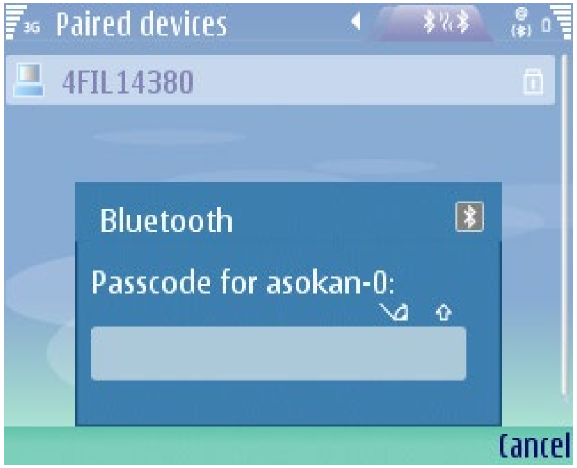
Can we safely open up widely deployed secure hardware on mobile devices for use by app developers?

Secure Device Pairing

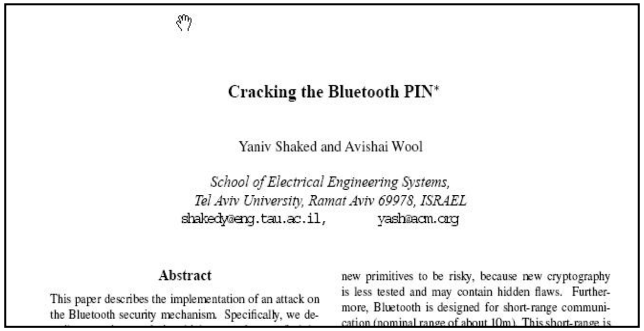
How can the process of pairing two devices be made easy to use without compromising security or adding to cost?

Secure Device Pairing: ca. 2005





Passcode for asokan-0:



Cracking the Bluetooth PIN*

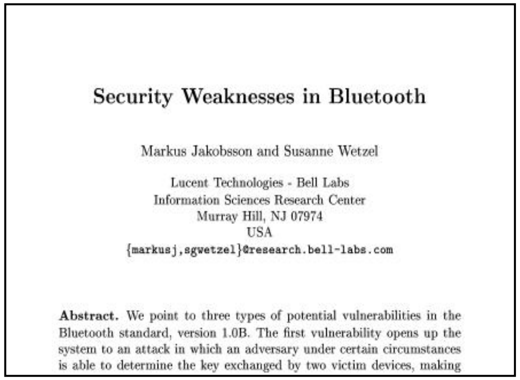
Yaniv Shaked and Avishai Wool

School of Electrical Engineering Systems,
Tel Aviv University, Ramat Aviv 69978, ISRAEL
shakedy@eng.tau.ac.il, yash@acm.org

Abstract

This paper describes the implementation of an attack on the Bluetooth security mechanism. Specifically, we de-

new primitives to be risky, because new cryptography is less tested and may contain hidden flaws. Furthermore, Bluetooth is designed for short-range communication (nominal range of about 10m). This short-range is



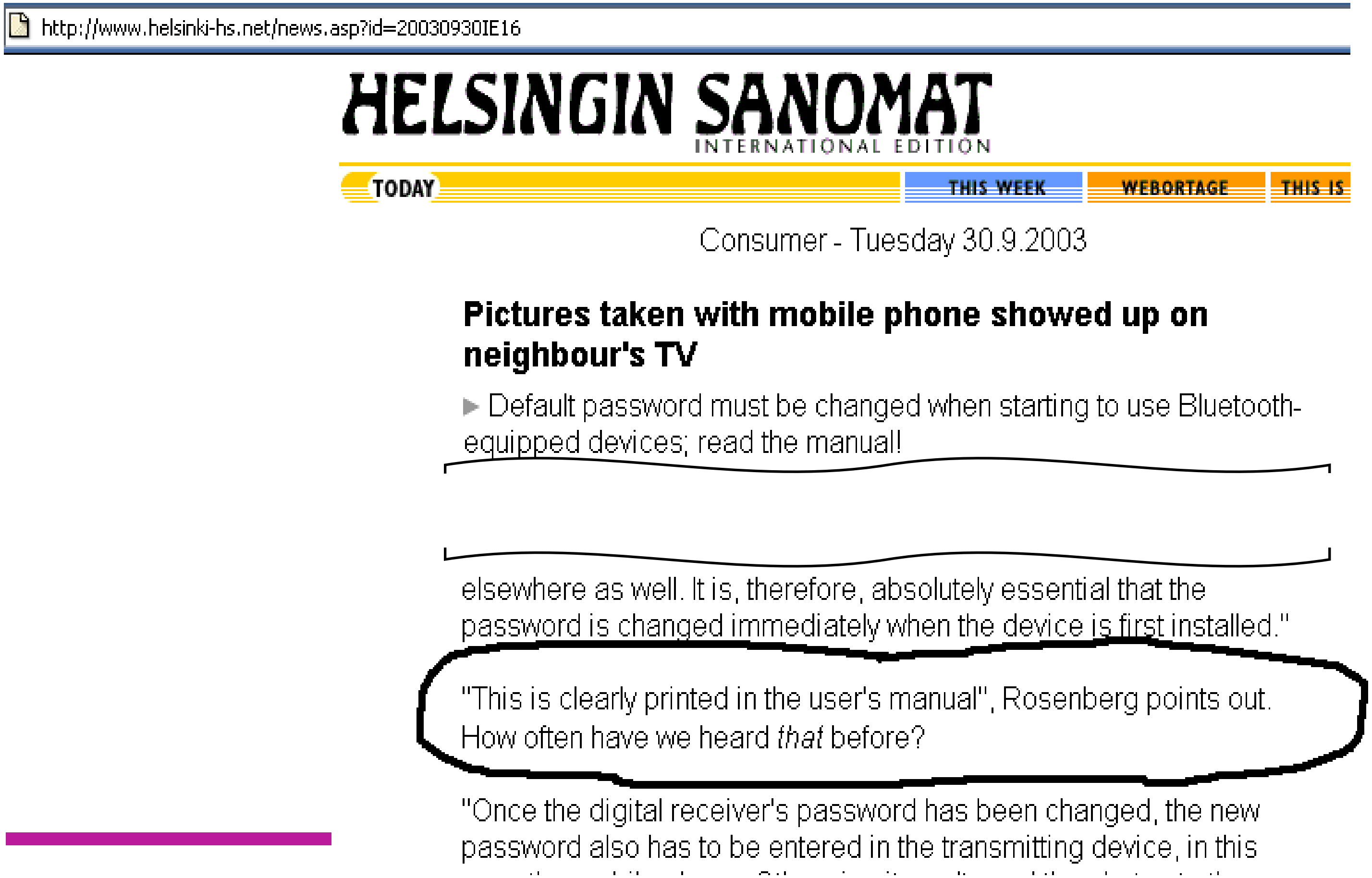
Security Weaknesses in Bluetooth

Markus Jakobsson and Susanne Wetzel

Lucent Technologies - Bell Labs
Information Sciences Research Center
Murray Hill, NJ 07974
USA
{markusj,sgwetzel}@research.bell-labs.com

Abstract. We point to three types of potential vulnerabilities in the Bluetooth standard, version 1.0B. The first vulnerability opens up the system to an attack in which an adversary under certain circumstances is able to determine the key exchanged by two victim devices, making

Naïve usability measures damage security



http://www.helsinki-hs.net/news.asp?id=20030930IE16

HELSINGIN SANOMAT
INTERNATIONAL EDITION

Consumer - Tuesday 30.9.2003

Pictures taken with mobile phone showed up on neighbour's TV

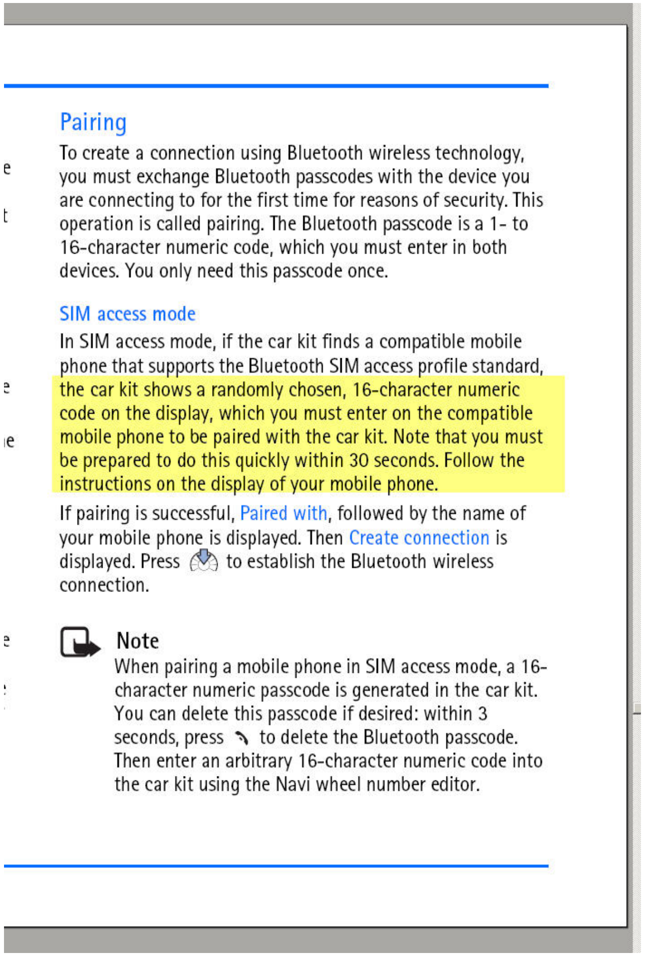
▶ Default password must be changed when starting to use Bluetooth-equipped devices; read the manual!

elsewhere as well. It is, therefore, absolutely essential that the password is changed immediately when the device is first installed."

"This is clearly printed in the user's manual", Rosenberg points out. How often have we heard *that* before?

"Once the digital receiver's password has been changed, the new password also has to be entered in the transmitting device, in this

Naïve security erodes usability



Pairing

To create a connection using Bluetooth wireless technology, you must exchange Bluetooth passcodes with the device you are connecting to for the first time for reasons of security. This operation is called pairing. The Bluetooth passcode is a 1- to 16-character numeric code, which you must enter in both devices. You only need this passcode once.

SIM access mode

In SIM access mode, if the car kit finds a compatible mobile phone that supports the Bluetooth SIM access profile standard, the car kit shows a randomly chosen, 16-character numeric code on the display, which you must enter on the compatible mobile phone to be paired with the car kit. Note that you must be prepared to do this quickly within 30 seconds. Follow the instructions on the display of your mobile phone.

If pairing is successful, Paired with, followed by the name of your mobile phone is displayed. Then Create connection is displayed. Press to establish the Bluetooth wireless connection.

Note

When pairing a mobile phone in SIM access mode, a 16-character numeric passcode is generated in the car kit. You can delete this passcode if desired: within 3 seconds, press to delete the Bluetooth passcode. Then enter an arbitrary 16-character numeric code into the car kit using the Navi wheel number editor.

Car kits

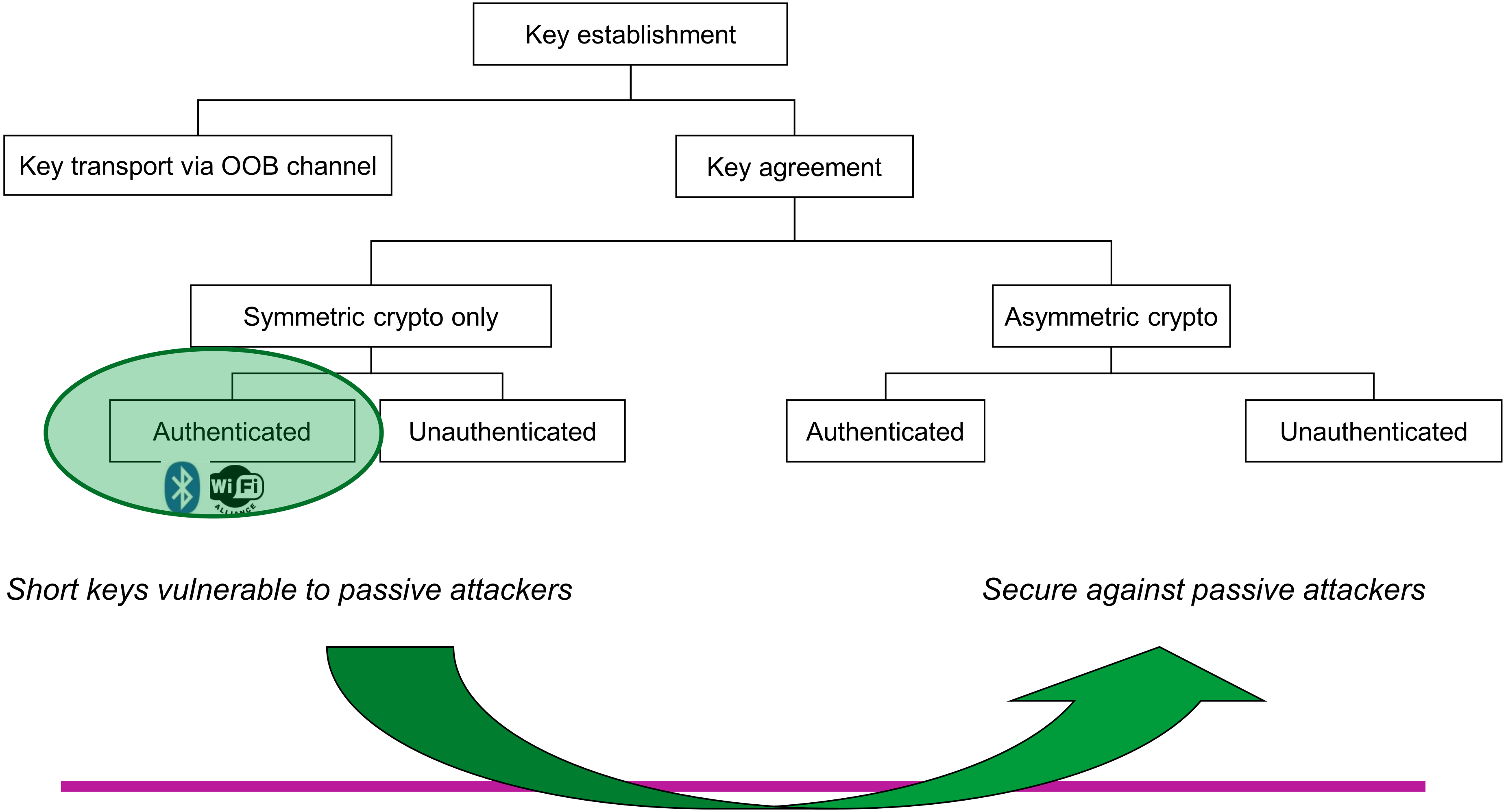
- Allow hands-free phone usage in cars
- Retrieve/use session keys from phone SIM
- require higher level of security

➢ users must enter 16-character passcodes

More secure = Harder to use?

Cost:
Calls to Customer

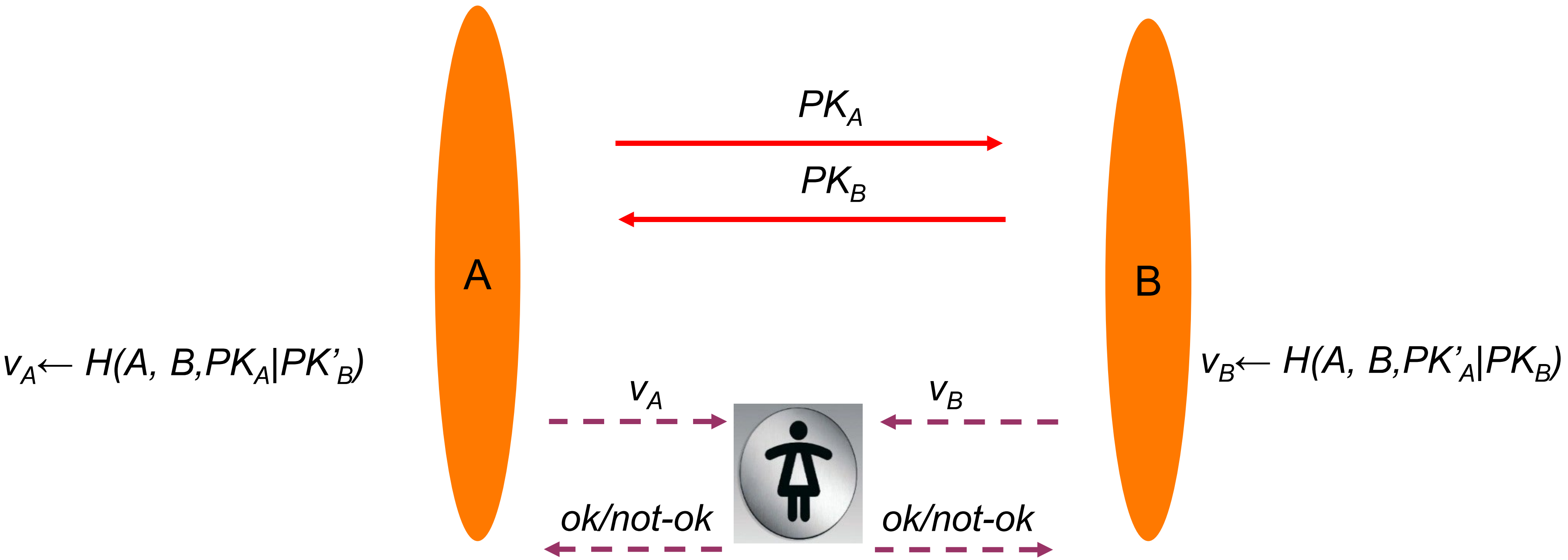
Key establishment for secure pairing ~2005



Short keys vulnerable to passive attackers

Secure against passive attackers

Authentication by comparing short strings



PK_A

PK_B

A

B

$v_A \leftarrow H(A, B, PK_A | PK'_B)$

$v_B \leftarrow H(A, B, PK'_A | PK_B)$

v_A

v_B

ok/not-ok

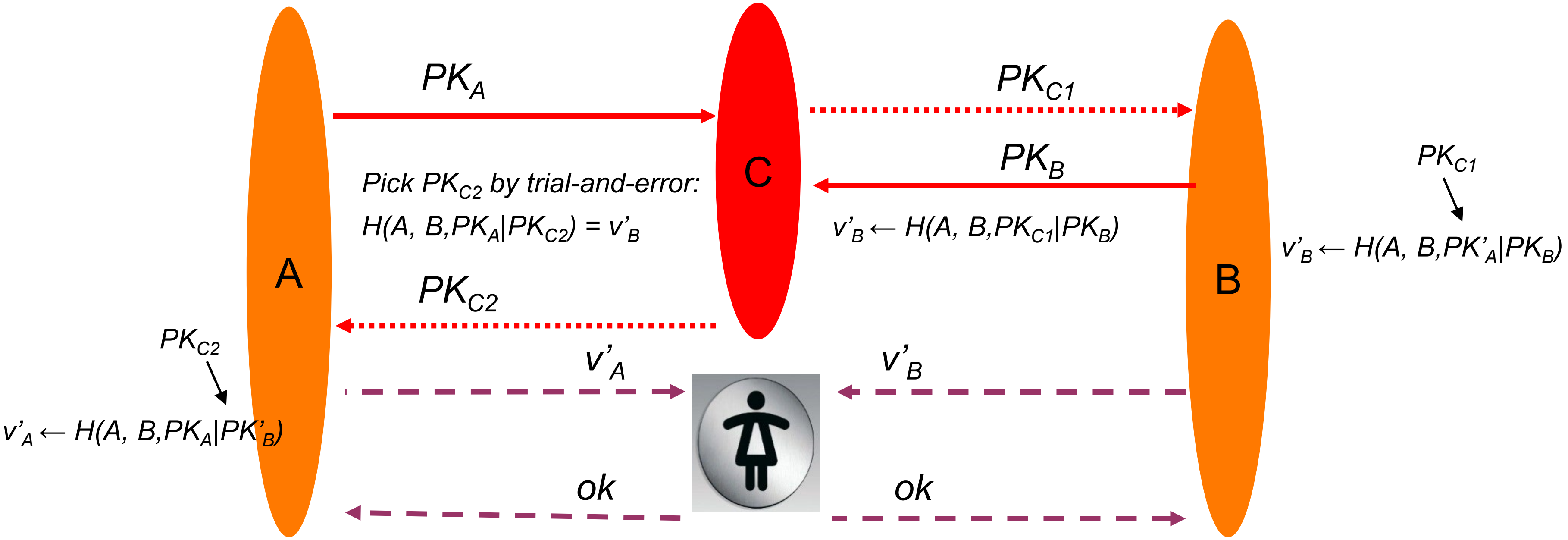
ok/not-ok

v_A and v_B are short strings (e.g., 4 digits),

User approves acceptance if v_A and v_B match

A man-in-the-middle can easily defeat this protocol

MitM in comparing short strings

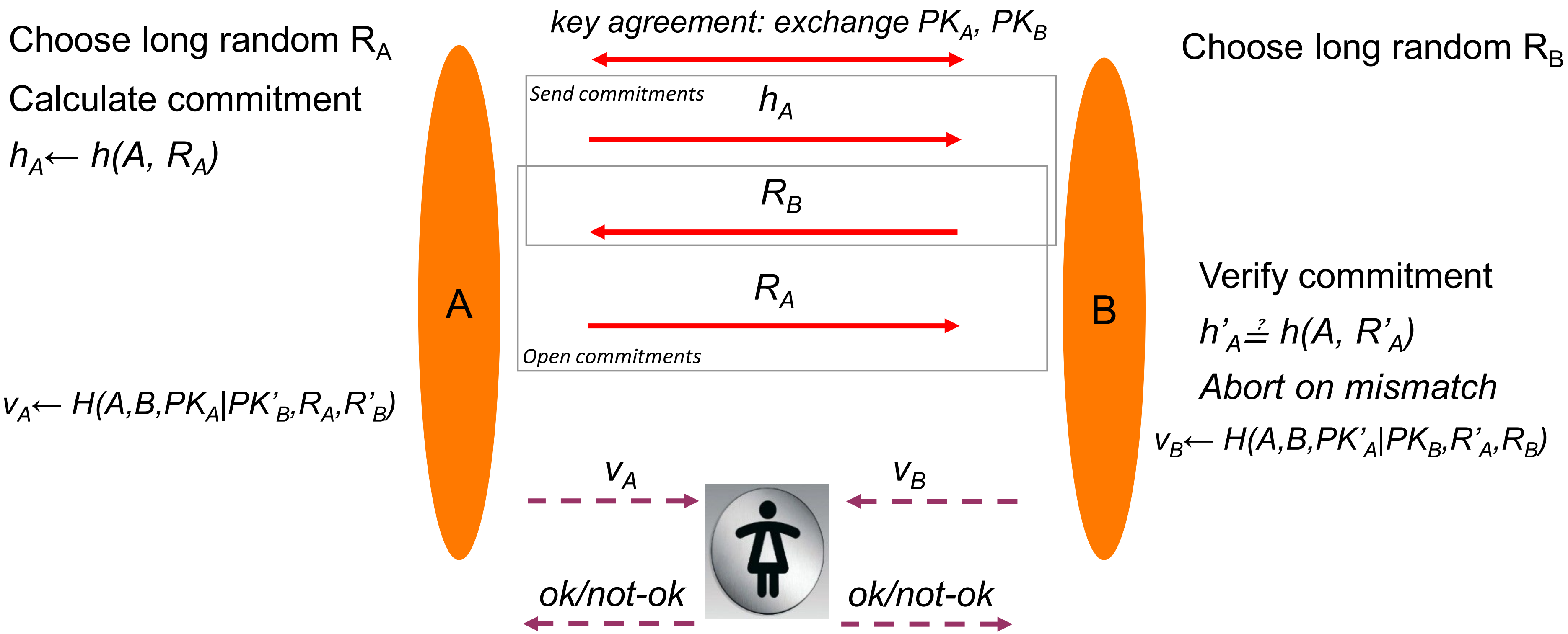

A
C
B
PK_A
PK_{C1}
PK_B
PK_{C1}
Pick PK_{C2} by trial-and-error:
$H(A, B, PK_A|PK_{C2}) = v'_B$
$v'_B \leftarrow H(A, B, PK_{C1}|PK_B)$
$v'_B \leftarrow H(A, B, PK'_A|PK_B)$
PK_{C2}
PK_{C2}
v'_A
v'_B
$v'_A \leftarrow H(A, B, PK_A|PK'_B)$
ok
ok

Guess a value SK_{C2}/PK_{C2} until $H(A, B, PK_A|PK_{C2}) = v'_B$

If v'_B is n digits, attacker needs at most 10^n guesses; Each guess costs one hash calculation

A typical modern PC can calculate 100000 MACs in 1 second

Authentication by comparing short strings



Choose long random R_A

Calculate commitment

$h_A \leftarrow h(A, R_A)$

key agreement: exchange PK_A, PK_B

Choose long random R_B

Send commitments h_A

R_B

R_A

Open commitments

A

B

Verify commitment

$h'_A \stackrel{?}{=} h(A, R'_A)$

Abort on mismatch

$v_A \leftarrow H(A,B,PK_A|PK'_B,R_A,R'_B)$

$v_B \leftarrow H(A,B,PK'_A|PK_B,R'_A,R_B)$

v_A

v_B

ok/not-ok

ok/not-ok

User approves acceptance if v_A and v_B match

2^{-l} (“unconditional”) security against man-in-the-middle (l is the length of v_A and v_B)

h() is a hiding commitment; in practice SHA-256

[LAN05] MANA IV, IACR report; [LN06] CANS ‘06

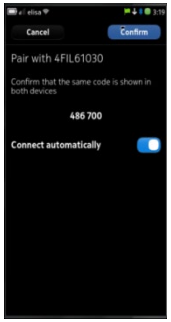
Key establishment for secure pairing ~2008

	Unauthenticated Diffie-Hellman	Authenticated Diffie-Hellman		
		short-string comparison	short PIN	Out-of-band channel
WiFi Protected Setup	"Push-button"		√	NFC
Bluetooth 2.1	"Just-works"	√	√	NFC
Wireless USB		√		USB Cable

[AN10] "Security associations for wireless devices" (Overview, book chapter)
[SVA09] "Standards for security associations in personal networks: a comparative analysis" IJSN 4(1/2):87-100 (survey of standards)

Secure Pairing: the aftermath

- Widely deployed (Bluetooth SSP, WiFi Protected Setup)
- **Improving usability/security → fundamental protocol changes**

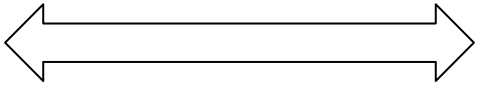


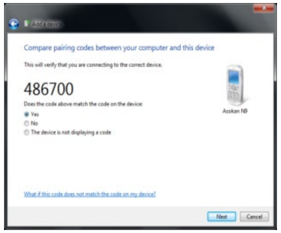
Pair with 4FIL61030

Confirm that the same code is shown in both devices

486 700

Connect automatically ●





Compare pairing codes between your computer and this device

This will verify that you are connecting to the correct device.

486700

Does the code above match the code on the device:

● Yes
○ No
○ The device is not displaying a code

What if this code does not match the code on my device?

[UKA07] “Usability Analysis of Secure Pairing Methods”, USEC ‘07

Secure Device Pairing: lessons learned

- Address pain points - builds credibility with stakeholders
- Don’t just guess security requirements; Ask stakeholders
- Desiderata for deployment and research can be different
- Standardization can make a good idea see light of day

The remaining examples

- On-board Credentials
  - How to make hardware TEEs safely accessible to developers? (Deployments in Nokia devices, but quietly!)

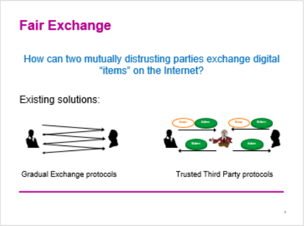
Lessons Learned

https://asokan.org/asokan/research

- How to choose the "right" problems?
  - Don't just guess security requirements; Ask stakeholders
  - Desiderata for deployment and research can be different
  - "90-10 rule" applies to deploying security
- How to identify "good" results?
  - Negative results are useful for security practitioners
  - Capturing researcher interest $\not\rightarrow$ (Tech transfer) Impact
  - (Tech transfer) Impact $\not\rightarrow$ Capturing researcher interest
- How to find paths to deployment?
  - Address pain points - builds credibility with stakeholders
  - (Standardization) Politics can suffocate a good idea
  - Standardization can make a good idea see light of day

Five examples

- Optimistic Fair Exchange
- Generic Authentication Architecture
- Channel Binding in Protocol Composition
- Secure Device Pairing
- On-board Credentials



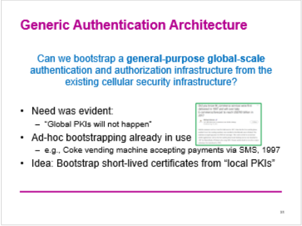
Fair Exchange

How can two mutually distrusting parties exchange digital "items" on the Internet?

Existing solutions:

Gradual Exchange protocols

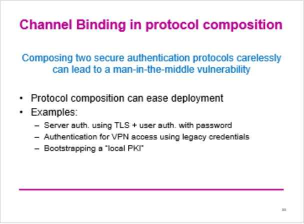
Trusted Third Party protocols



Generic Authentication Architecture

Can we bootstrap a **general-purpose global-scale** authentication and authorization infrastructure from the existing cellular security infrastructure?

- Need was evident:
  - "Global PKIs will not happen"
- Ad-hoc bootstrapping already in use
  - e.g., Coke vending machine accepting payments via SMS, 1997
- Idea: Bootstrap short-lived certificates from "local PKIs"



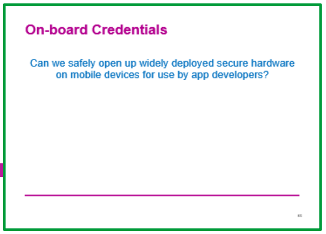
Channel Binding in protocol composition

Composing two secure authentication protocols carelessly can lead to a man-in-the-middle vulnerability

- Protocol composition can ease deployment
- Examples:
  - Server auth. using TLS + user auth. with password
  - Authentication for VPN access using legacy credentials
  - Bootstrapping a "local PKI"

Secure Device Pairing

How can the process of pairing two devices be made easy to use without compromising security or adding to cost?



On-board Credentials

Can we safely open up widely deployed secure hardware on mobile devices for use by app developers?

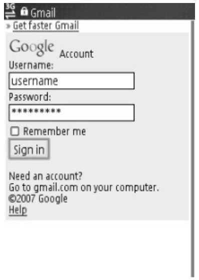
On-board Credentials

Can we safely open up widely deployed secure hardware on mobile devices for use by app developers?

Authentication on the Internet

Username/password rules the Internet

- Cheap, easy-to-deploy, portable
- Annoying, vulnerable (phishing, dictionary attacks, password-stealing trojans…)



Gmail

» Get faster Gmail

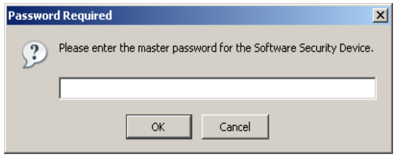
Google Account

Username: username

Password: *********

□ Remember me

Need an account?
Go to gmail.com on your computer.
©2007 Google
Help

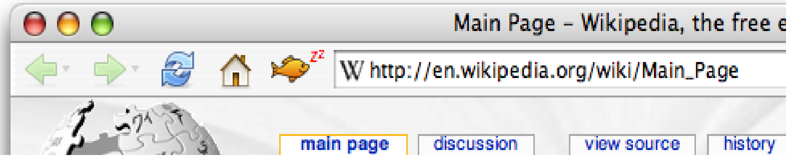


Password Required

Please enter the master password for the Software Security Device.

Attempts to improve usability and security

- Password-managers
- Single Sign-On
- Better protocols



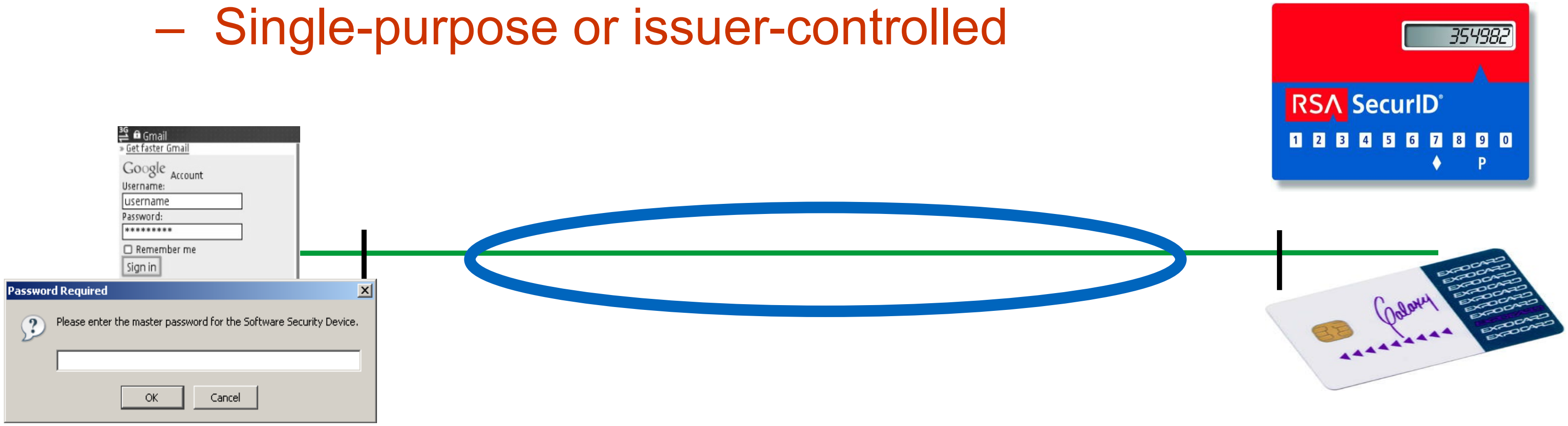
Main Page - Wikipedia, the free e

http://en.wikipedia.org/wiki/Main_Page

Hardware tokens

Deployed for specific-services

- More secure, sometimes more intuitive
- More expensive, usually no trusted path to user,
- Single-purpose or issuer-controlled



Gmail

» Get faster Gmail

Google Account

Username: username

Password: *********

□ Remember me

Sign in

Password Required

Please enter the master password for the Software Security Device.

OK Cancel

SW-only credentials

HW credentials

Trusted hardware is widely deployed

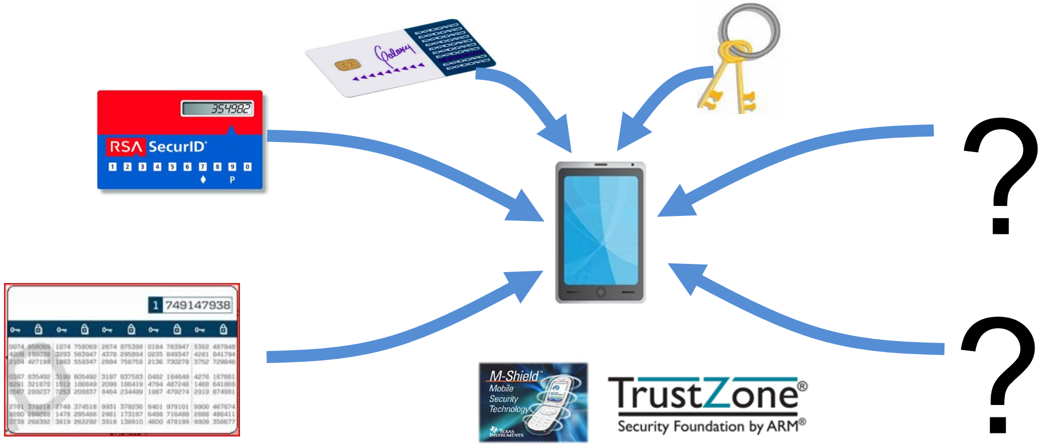
- Trusted Execution Environments on smartphones have been available for years
  - Introduced for manufacturer and operator needs
  - Not accessible for app developers

TrustZone®
Security Foundation by ARM®

[EKA14] "The Untapped Potential of Trusted Execution Environments on Mobile Devices", IEEE S&P Magazine, Jul-Aug 2014

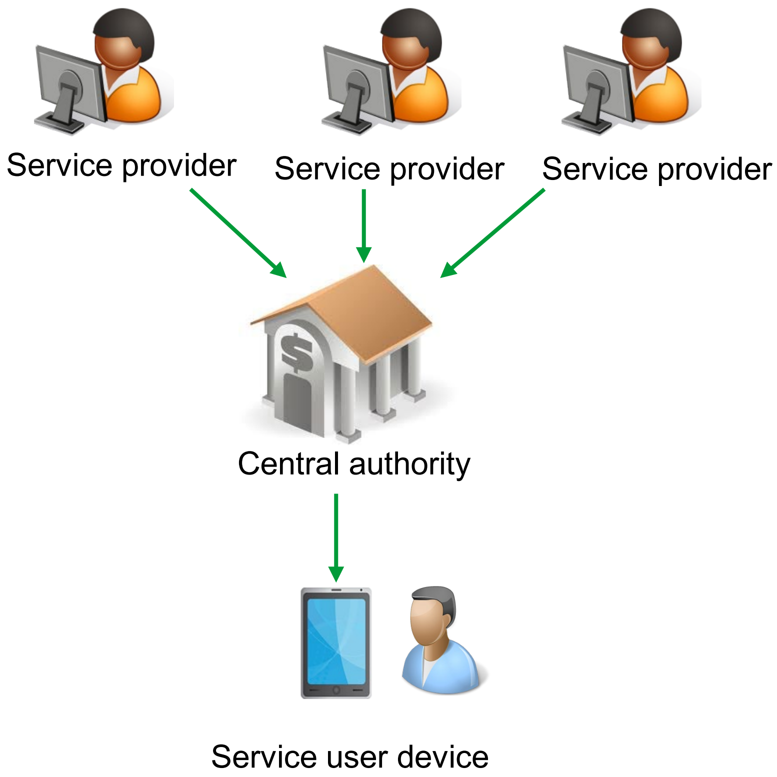
On-board Credentials

An **open** credential platform that leverages existing mobile TEEs


354982
RSA SecurID
1 749147938
?
?
M-Shield Mobile Security Technology
TrustZone Security Foundation by ARM

Secure yet inexpensive

Centralized vs. open provisioning



Service provider

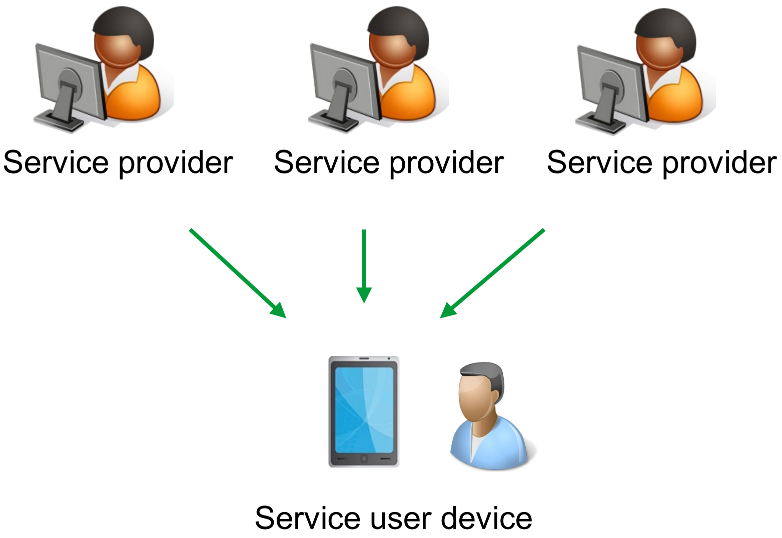
Service provider

Service provider

Central authority

Service user device

Centralized provisioning
(smart cards)



Service provider

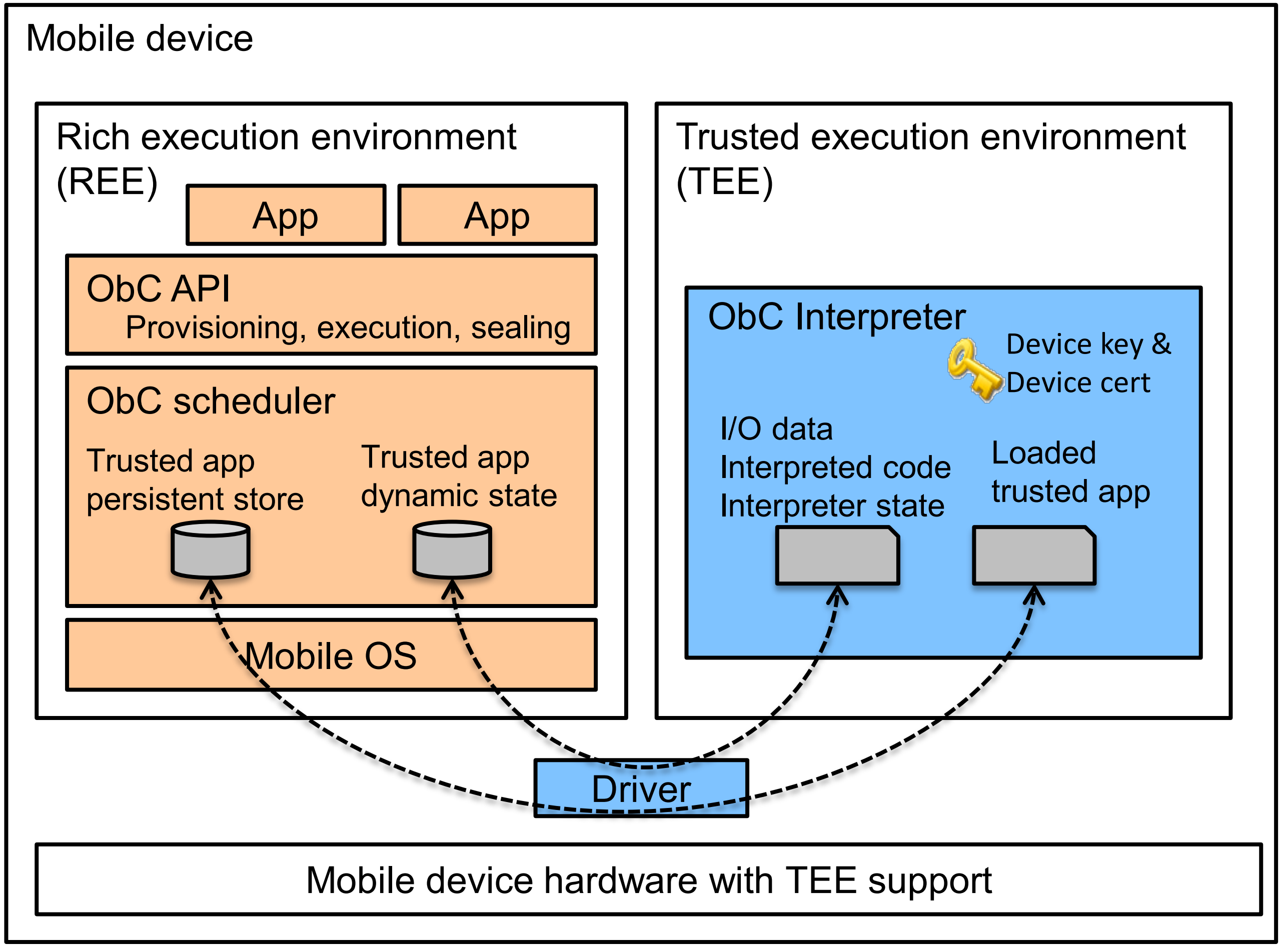
Service provider

Service provider

Service user device

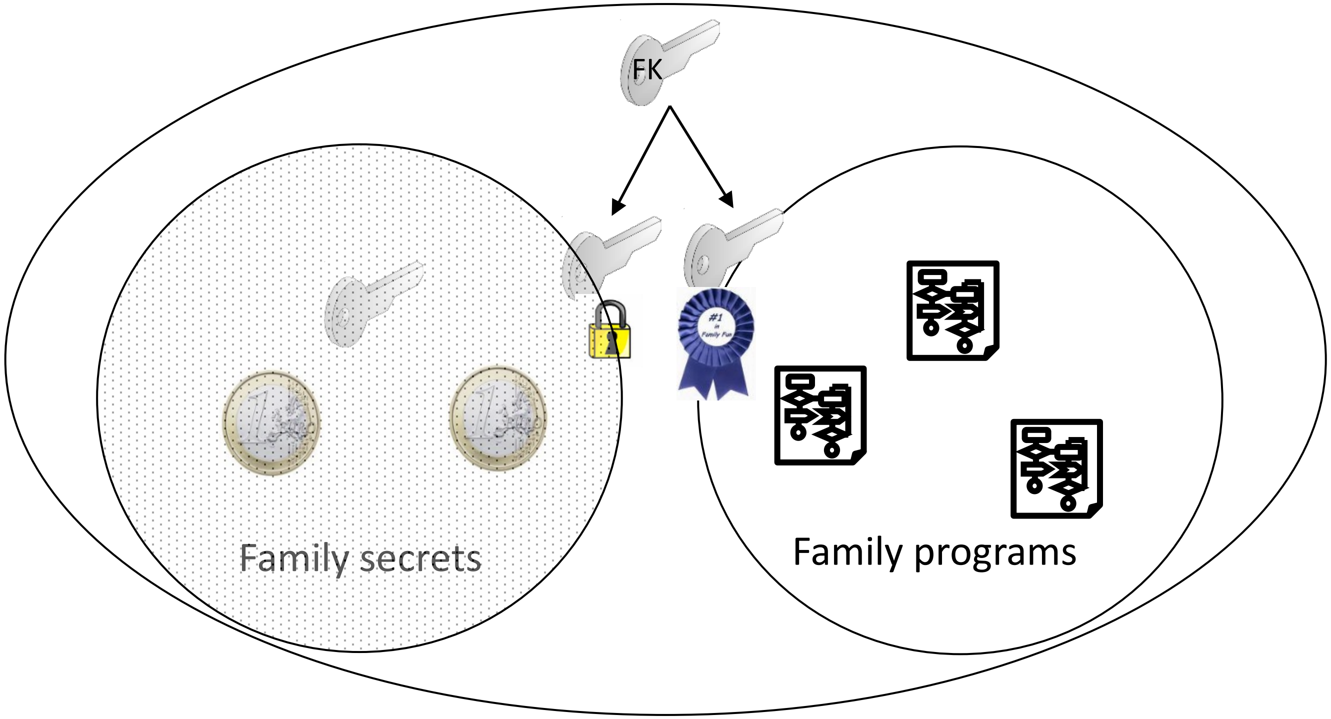
Open provisioning
(On-board Credentials)

On-board Credentials (ObC) architecture


Mobile device
Rich execution environment (REE)
App
App
ObC API
Provisioning, execution, sealing
ObC scheduler
Trusted app persistent store
Trusted app dynamic state
Mobile OS
Trusted execution environment (TEE)
ObC Interpreter
Device key &
Device cert
I/O data
Interpreted code
Interpreter state
Loaded trusted app
Driver
Mobile device hardware with TEE support

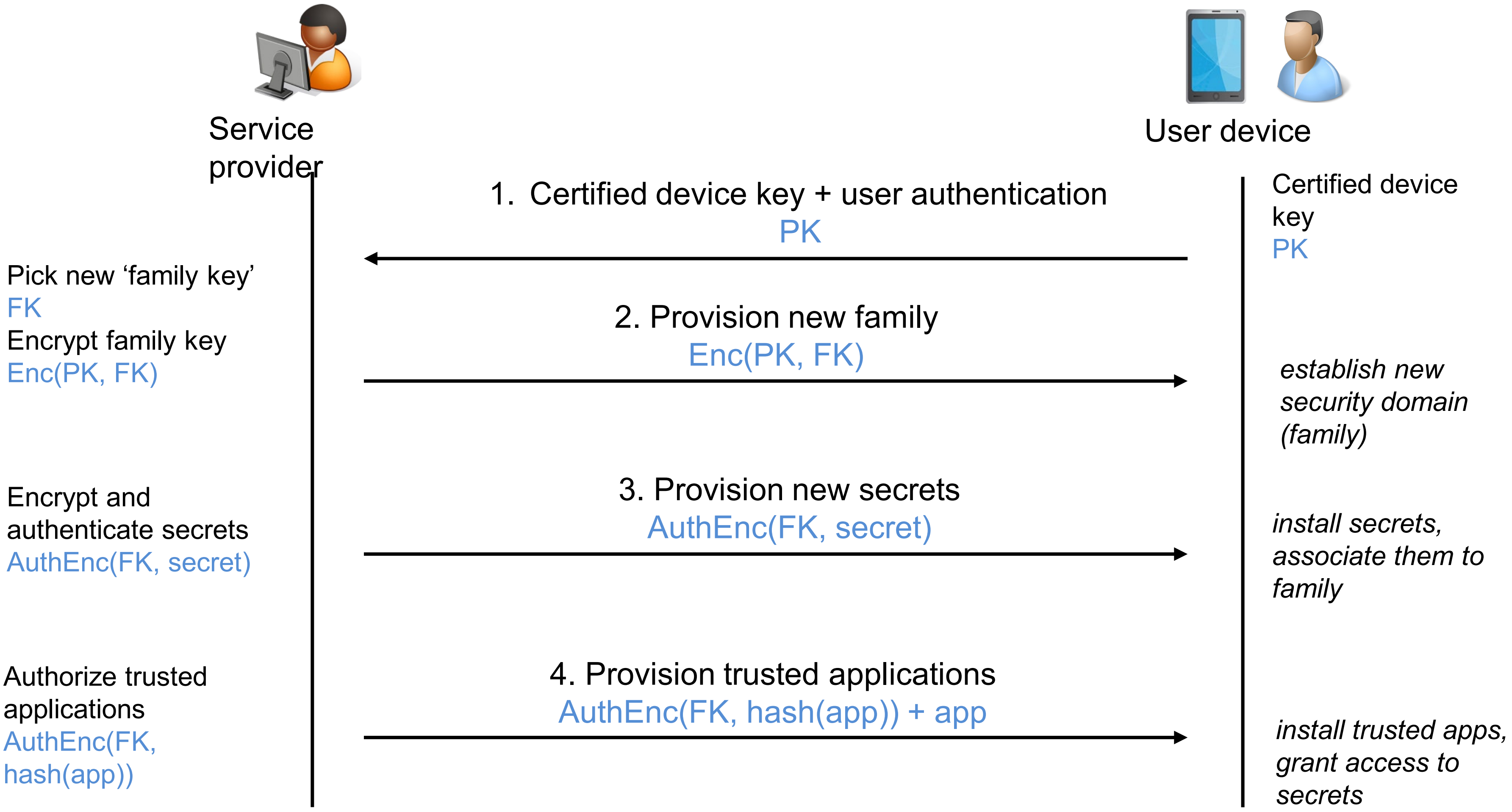
ObC Provisioning (1/2)

Basic Idea: the notion of a **family** of credential secrets and credential programs endorsed to use them



Principle of same-origin policy

Open provisioning model



Service provider

User device

Certified device key
PK

1. Certified device key + user authentication
PK

Pick new 'family key'
FK
Encrypt family key
Enc(PK, FK)

2. Provision new family
Enc(PK, FK)

establish new security domain (family)

Encrypt and authenticate secrets
AuthEnc(FK, secret)

3. Provision new secrets
AuthEnc(FK, secret)

install secrets, associate them to family

Authorize trusted applications
AuthEnc(FK, hash(app))

4. Provision trusted applications
AuthEnc(FK, hash(app)) + app

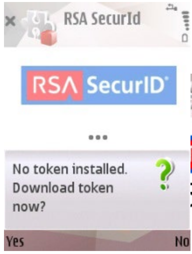
install trusted apps, grant access to secrets

[KEAR09] "On-board Credentials with Open Provisioning". ASIACCS 2009.

Ekberg. Securing Software Architectures for Trusted Processor Environments. Dissertation, Aalto University 2013.
Kostiainen. On-board Credentials: An Open Credential Platform for Mobile Devices. Dissertation, Aalto University 2012.

ObC: the aftermath

- Initial prototypes ca. 2008
  - RSA SecurID, SoftSIM
- (Silently) deployed in recent Lumia devices
  - Used for, e.g., [MirrorLink](#) attestation, LIRR ticketing trial
- Stumbling blocks:
  - “who takes liability?” “avoid stepping on toes”
- Related standardization
  - Global Platform device committee
  - Open provisioning is elusive



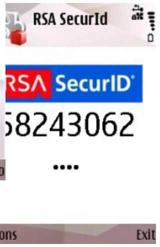
RSA SecurId

RSA SecurID

...

No token installed. Download token now?

Yes No



RSA SecurId

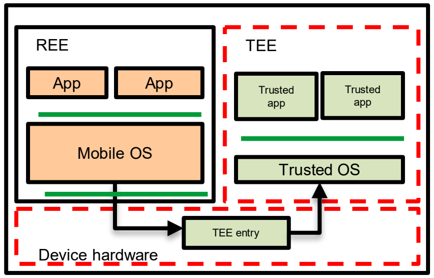
RSA SecurID

;8243062

....

Options Exit

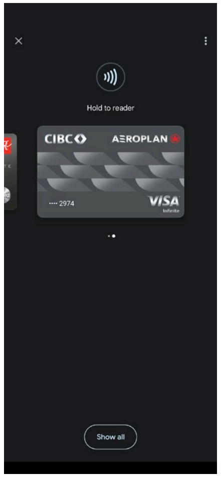
https://www.newsday.com/long-island/transportation/lirr-tests-smartphone-payment-system-u04362

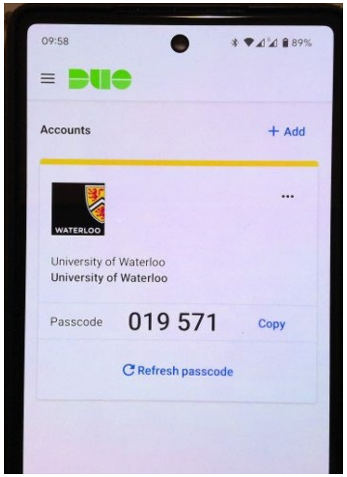


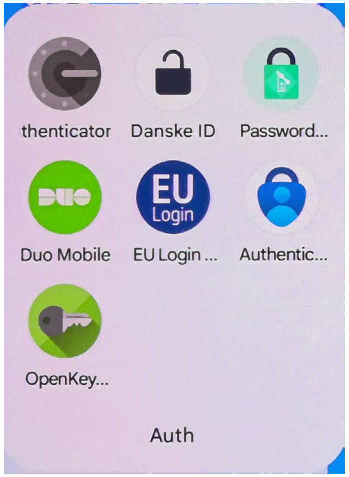
GLOBALPLATFORM™

[GP12] “[A New Model: The Consumer-Centric Model and How It Applies to the Mobile Ecosystem](#)”

“On-board Credentials” on my phone







ObC: Lessons Learned

- Address pain points - builds credibility with stakeholders
- Politics can suffocate a good idea
- Standardization can make a good idea see light of day
- (Tech transfer) Impact ↛ Capturing researcher interest

Lessons Learned

https://asokan.org/asokan/research

- How to choose the “right” problems?
  - Don’t just guess security requirements; Ask stakeholders
  - Desiderata for deployment and research can be different
  - “90-10 rule” applies to deploying security
- How to identify “good” results?
  - Negative results are useful for security practitioners
  - Capturing researcher interest ↛ (Tech transfer) Impact
  - (Tech transfer) Impact ↛ Capturing researcher interest
- How to find paths to deployment?
  - Address pain points - builds credibility with stakeholders
  - (Standardization) Politics can suffocate a good idea
  - Standardization can make a good idea see light of day