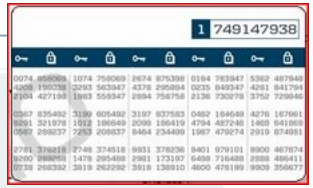
Towards User-Friendly Credential Transfer on Open Credential Platforms

Kari Kostiainen, N. Asokan
Nokia Research Center

Alexandra Afanasyeva
SUAI

ACNS 2011

Welcome to eBanking



Welcome Kari Timo Juhani Kostiainen

Please verify your identity:

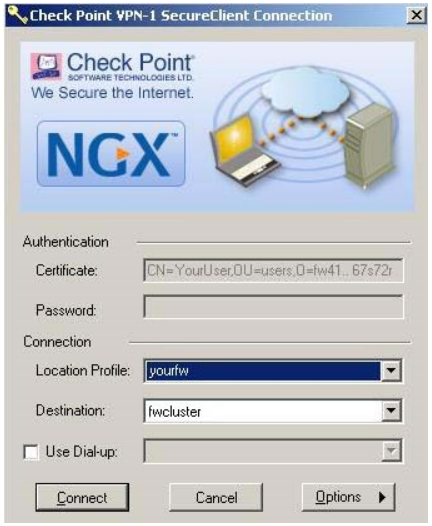
Security card number: 4600134200

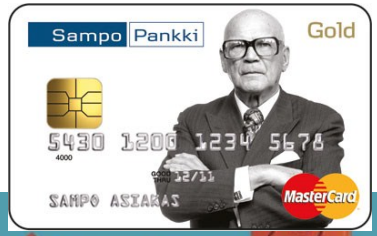
Key number: 6137

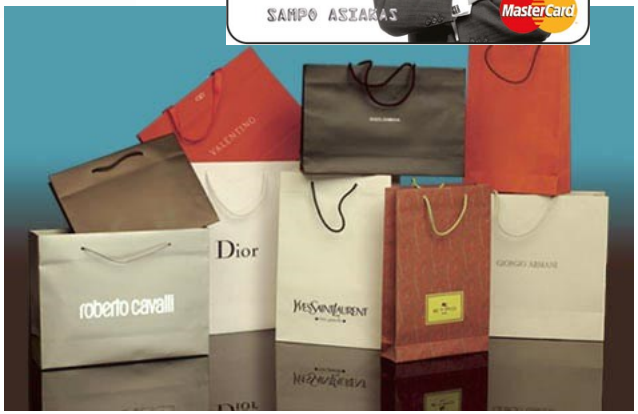
Security number:

OK Cancel

You have 23 numbers left on your security card.







NOKIA

Trusted execution environment (TrEE)

TrustZone®
Security Foundation by ARM®

NOKIA

Credential transfer

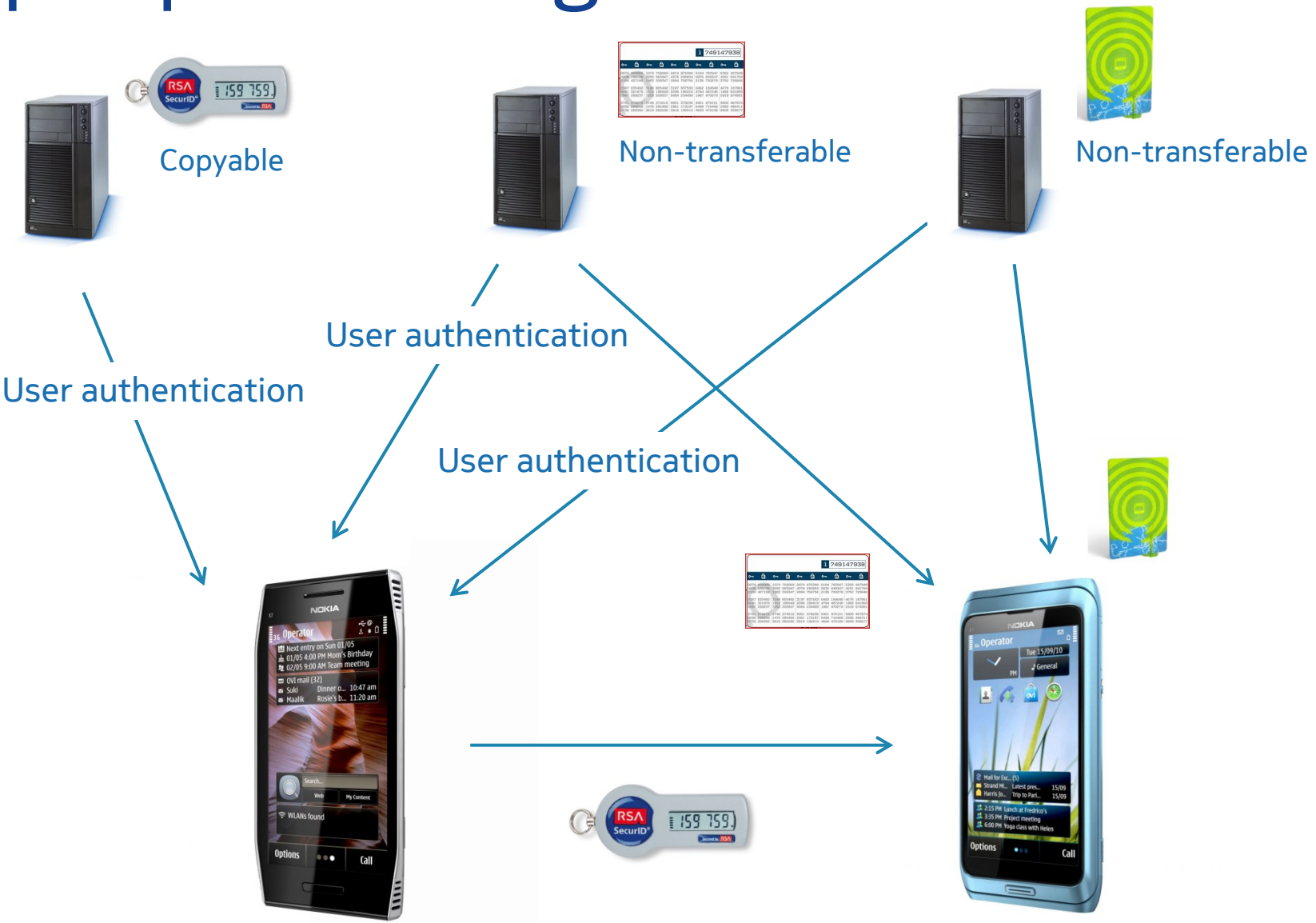
Outline

1. Credential transfer problem
2. Credential transfer protocol
3. Analysis

Credential transfer problem

Closed provisioning

Open provisioning



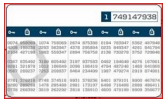
Late user binding

User binding

User binding

Device certificate

Enc(credentials)



NOKIA

Temporal disconnection

Requirements

- Usability
  - No additional user interaction
- Security
  - Credential secrecy
  - “Credential fidelity”

Credential transfer protocol

Protocol overview

Provisioner (P)

PK_{TS}/SK_{TS}, $Cert_{TS}$

Trusted Server (TS)

OS

HSM

2. Provisioning and user binding

3. Credential backup and "identity verification delegation"

5. Identity verification, credential recovery and "provisioning delegation"

6. Automatic re-provisioning

Source device (S)

OS

TrEE

PK_S/SK_S, $Cert_S$

1. User identity installation

4. User identity installation

Target device (T)

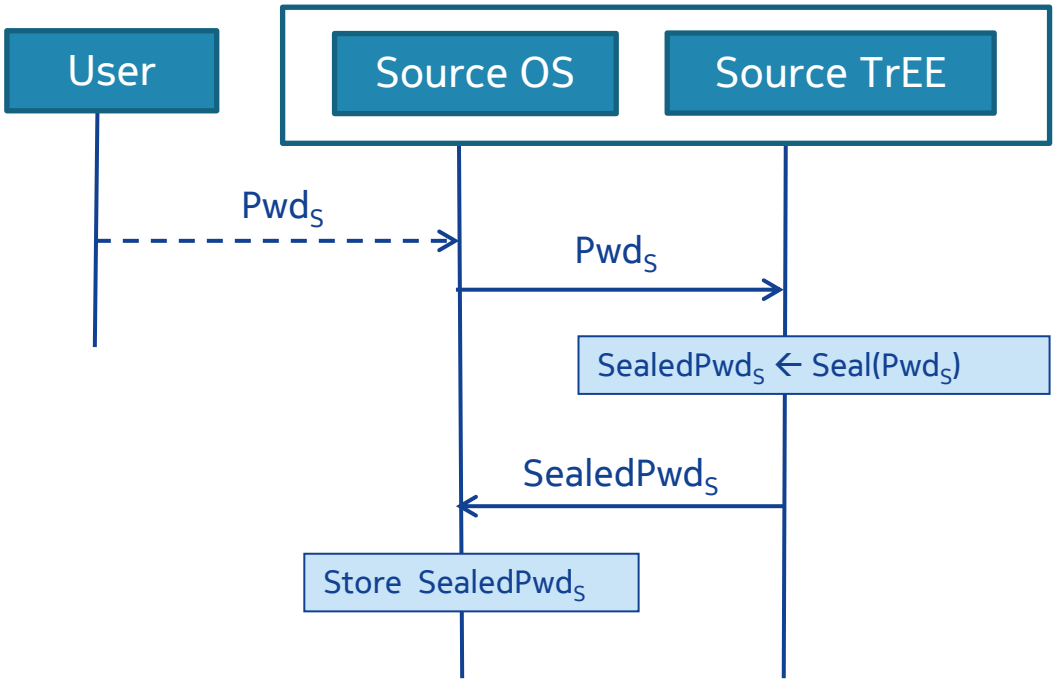
OS

TrEE

PK_T/SK_T, $Cert_T$

NOKIA

1. User identity installation



1. Trust on first use
2. Typical device login

2. Credential provisioning

Provisioner

Source OS

Source TrEE

$Cert_S$ + user auth.

Verify $Cert_S$ and extract PK_S
Map Cred to PK_S
ProvCred ← Enc(PK_S, Cred || policy)

ProvCred

ProvCred, $SealedPwd_S$

Cred || policy ← Dec(SK, ProvCred)
Pwd_S ← Unseal($SealedPwd_S$)
SealedCred ← Seal(Cred || policy || Pwd_S)

SealedCred

Store SealedCred

3. Credential backup to server

Source TrEE

Source OS

Trusted Server (TS)

$Cert_{TS}$

$Cert_{TS}$, SealedCred

Verify $Cert_{TS}$ and extract PK_{TS}
Cred || policy || Pwd_S ← Unseal(SealedCred)
Copyable:
  EncCred ← Enc(PK_{TS}, Cred || Pwd_S)
Non-transferable:
  SDT ← Sign(SK_S, PK_{TS}) || Enc(PK_{TS}, Pwd_S)

EncCred / SDT

$Cert_S$, EncCred / SDT

Store $Cert_S$ and EncCred / SDT

5. Credential recovery

User identity installation

Trusted Server (TS)

Target OS

Target TrEE

$SealedPwd_T$

$Pwd_T \leftarrow Unseal(SealedPwd_T)$
$PwdToken \leftarrow Enc(PK_{TS}, Pwd_T)$

$PwdToken$

$PwdToken, Cert_T$

Verify $Cert_T$ and extract PK_T
$Pwd_T \leftarrow Dec(SK_{TS}, PwdToken)$
Copyable:
  $Cred \,||\, Pwd_S \leftarrow Dec(SK_{TS}, EncCred)$
  Verify $Pwd_T = Pwd_S$
  $EncCred \leftarrow End(PK_T, Cred \,||\, Pwd_T)$
Non-transferable:
  Extract Pwd_S from SDT
  Verify $Pwd_T = Pwd_S$
  $DT \leftarrow Sign(SK_{TS}, PK_T) \,||\, Sign(SK_S, PK_{TS})$

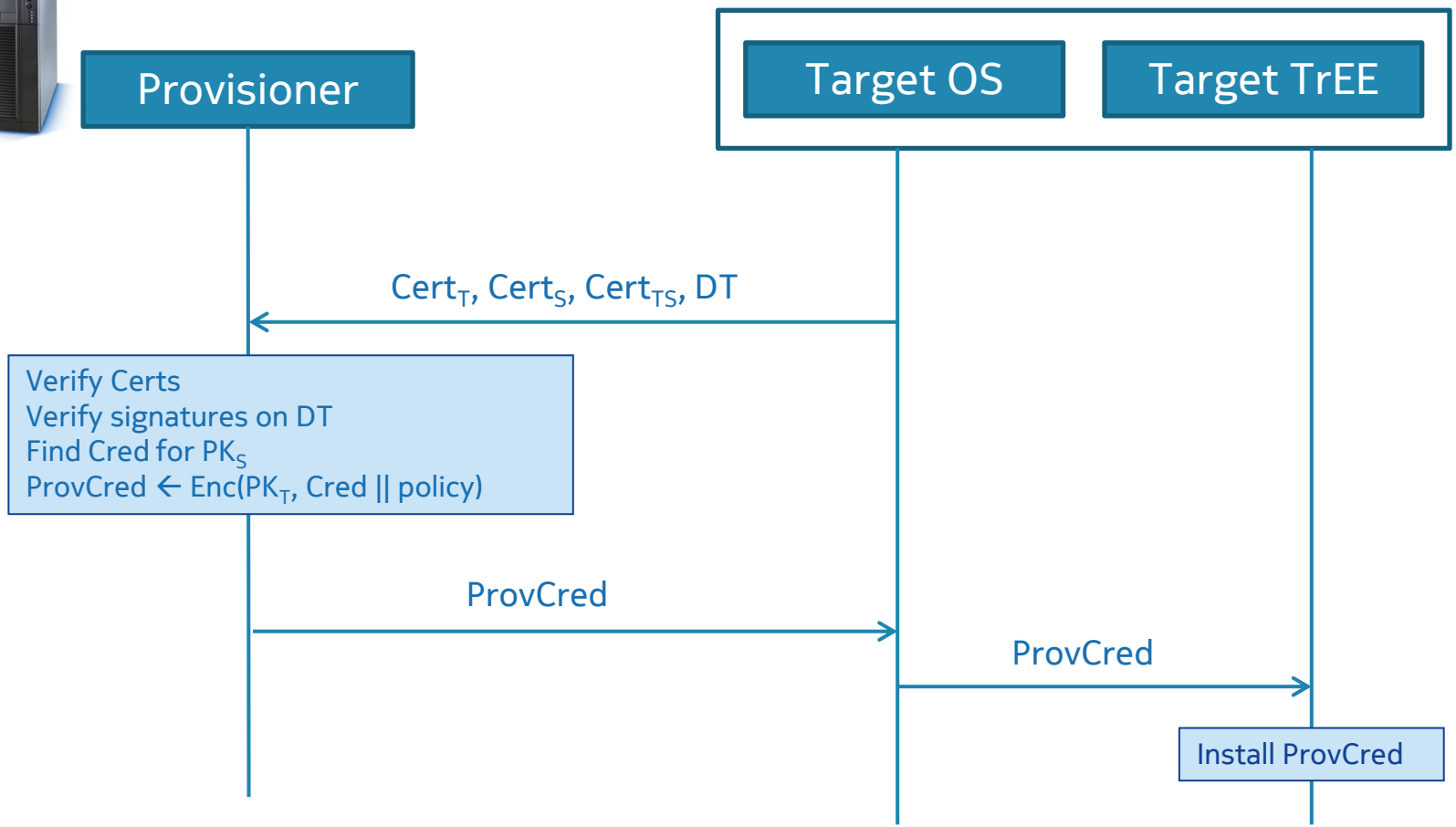
$Cert_S$, EncCred / DT

EncCred

Install EncCred

NOKIA

6. Credential re-provisioning



Analysis

Analysis

- Usability
  - Reusing typical device login
- Credential secrecy
  - Common public key mechanism
  - Trusted server (HSM)
- Credential fidelity
  - User identity password
    - Brute force (throttling)
    - Phishing (separate password)
    - Password change (“trusted UI” or secure connection to server)
- Protocol validated with AVISPA tool

Summary

- Credential transfer challenging
  - Open provisioning
  - Late user identity binding
  - Temporal disconnection
- Server-assisted credential transfer protocol
  - Can be implemented using existing devices