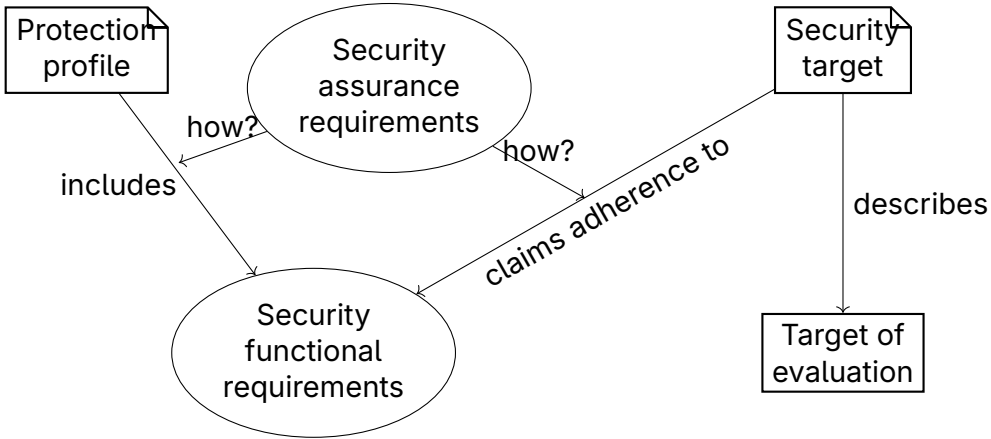
A Comparison-Based Methodology for the Security Assurance of Novel Systems

Jelizaveta Vakarjuk, Peeter Laud

Security assurance

- The system aims to do the right things
- The system implements these things in the right way
- The system has been carefully designed / implemented / deployed
- The assurance procedures have been carefully executed

Notions from *Common Criteria*



Updating a system

- New version of a system ⇒ new ToE ⇒ new certification needed
- New version of certified system may fulfill SFRs in novel ways

Updating a system

- New version of a system $\Rightarrow$ new ToE $\Rightarrow$ new certification needed
- New version of certified system may fulfill SFRs in novel ways

Proposed approach

- Do not repeat the whole conformance checking of the new ToE against the claimed SFRs
- Compare the new ToE against the old one
  - Show that new ToE is **at least as secure as** the old one

Denote: Old ToE: T°. New ToE: $T^{\bullet}$

Comparison methodology

- List all conceivable and inconceivable *weaknesses* of both systems
  - Considering also inconceivable weaknesses should make the process of collecting them all more mechanical
  - Let W° and $W^\bullet$ be the sets of weaknesses of T° and $T^\bullet$, respectively
    - May, and probably do significantly intersect
  - Exploiting a weakness should have “similar” effects against T° and $T^\bullet$
- Show that for each $\mathbf{w}^\bullet \subseteq W^\bullet$, there exists $\mathbf{w}^\circ \subseteq W^\circ$, such that $\mathbf{w}^\circ$ is **at least as bad as** $\mathbf{w}^\bullet$.
  - Denote $\mathbf{w}_1 \prec_D \mathbf{w}_2$, if $\mathbf{w}_1$ is no more difficult to exploit than $\mathbf{w}_2$
  - Denote $\mathbf{w}_1 \prec_S \mathbf{w}_2$, if effects of exploiting $\mathbf{w}_1$ are no worse than those of $\mathbf{w}_2$
  - **at least as bad as**: $\mathbf{w}^\circ \prec_D \mathbf{w}^\bullet$ and $\mathbf{w}^\bullet \prec_S \mathbf{w}^\circ$

Example

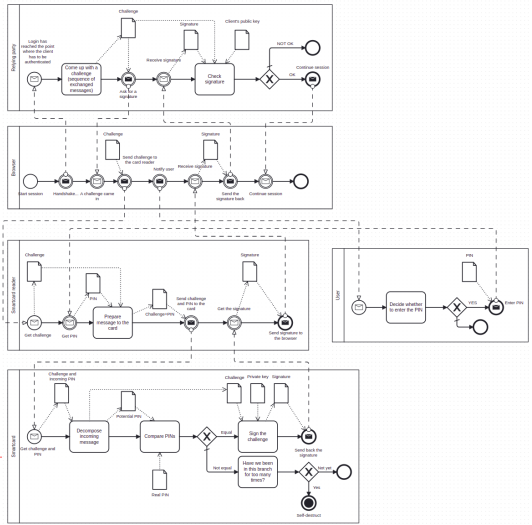
Authentication with a smartcard

- Sign a challenge
  - Sent from browser via PKCS #11 API
- Private key inside the chip
- Activated with a PIN
  - Enter from PINpad
  - Or, enter from computer keyboard

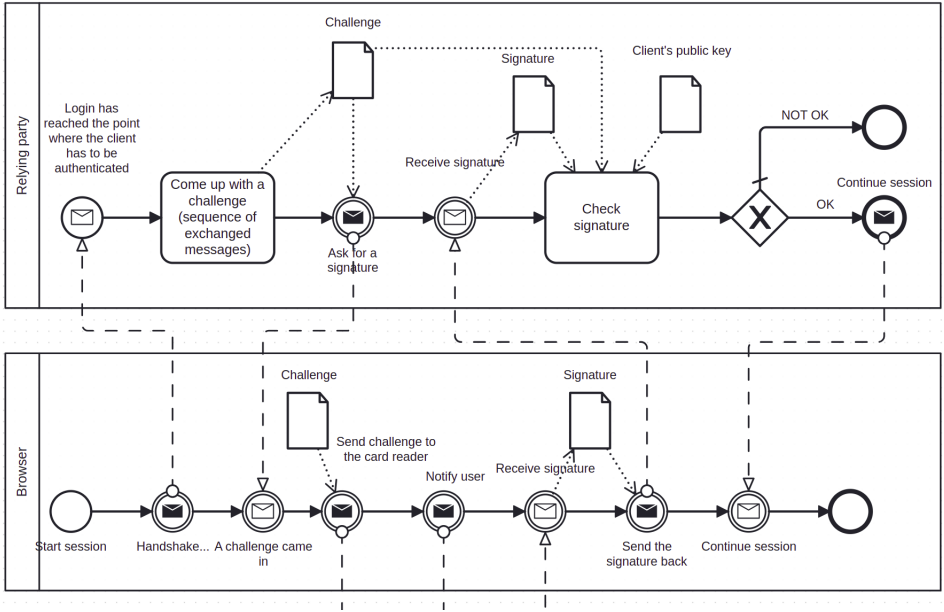
Authentication with a phone

- Sign a challenge
  - Sent from browser through Identity Provider via SMS
- Private key inside the SIM card
- Activated with a PIN
  - Enter from phone's keypad

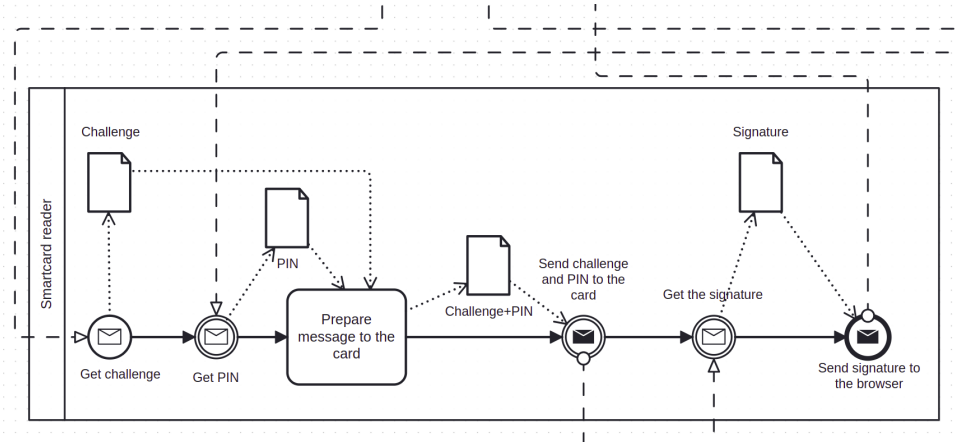
Authentication with smartcard

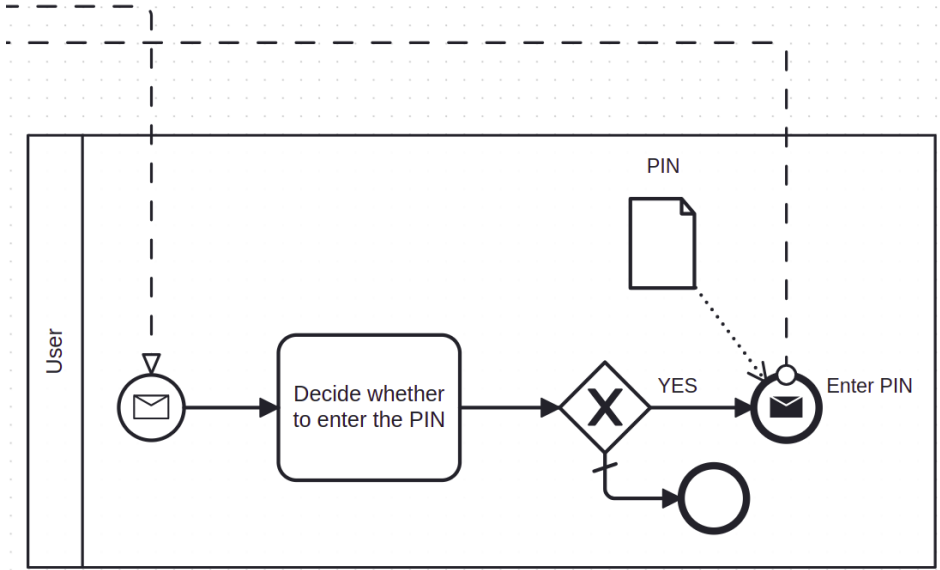


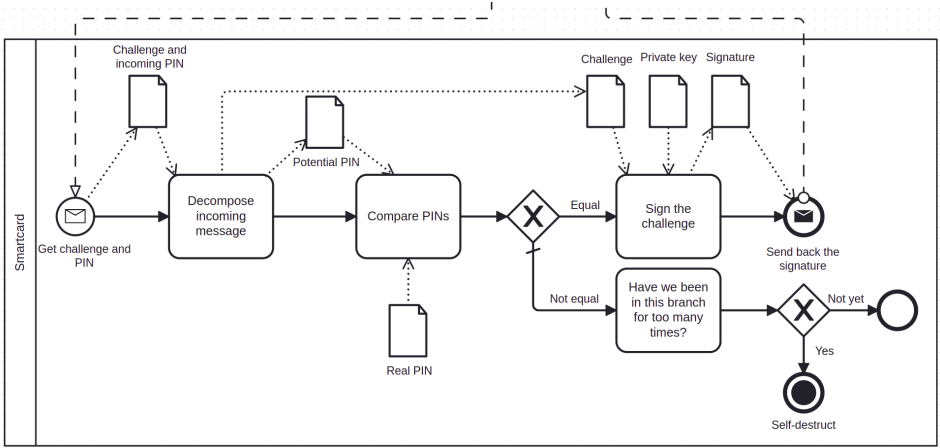
CYBERNETICA



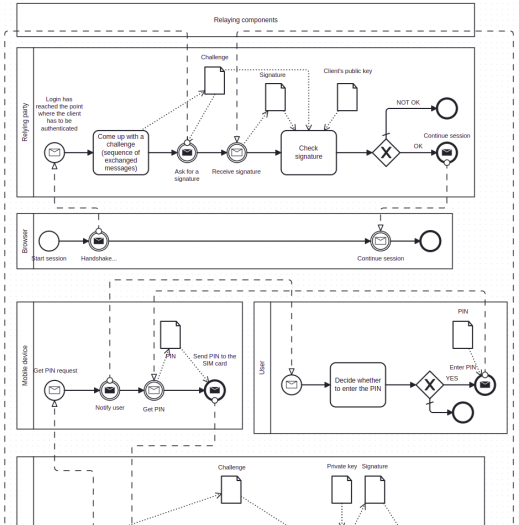
CYBERNETICA


Smartcard reader
Challenge
PIN
Prepare message to the card
Challenge+PIN
Send challenge and PIN to the card
Signature
Get the signature
Get challenge
Get PIN
Send signature to the browser

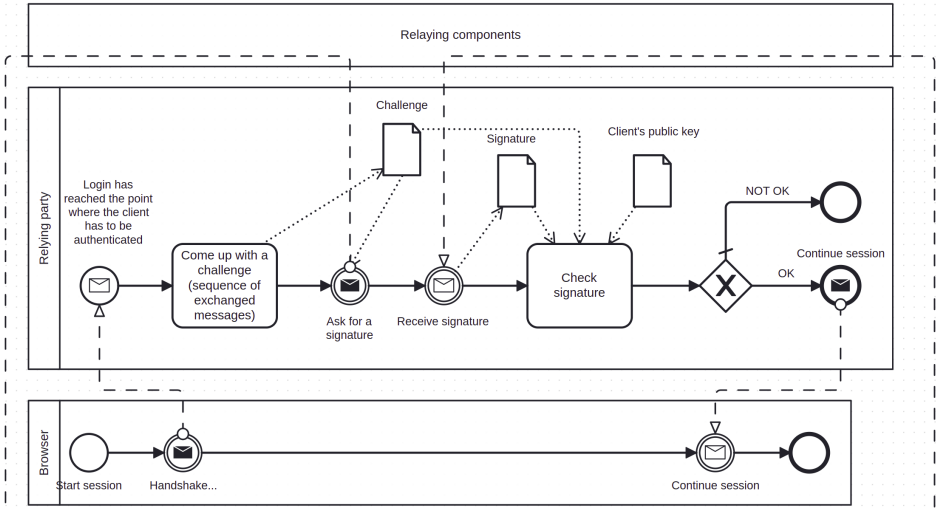

User
Decide whether to enter the PIN
X
YES
PIN
Enter PIN

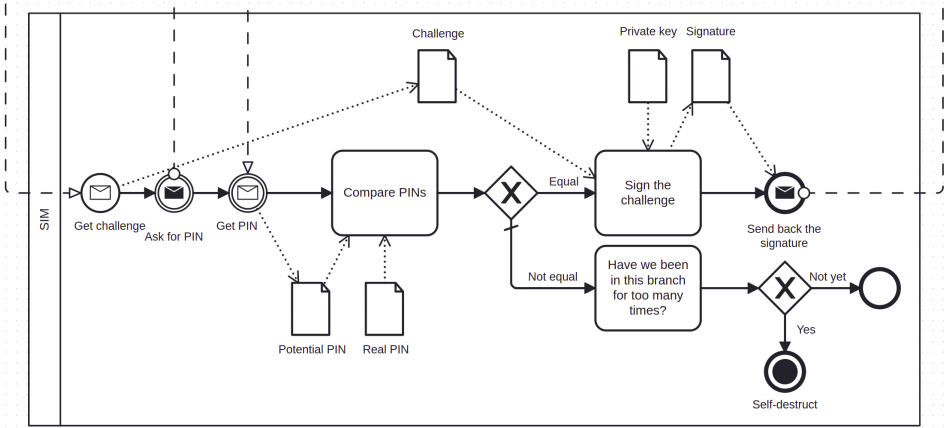

Smartcard
Challenge and incoming PIN
Potential PIN
Challenge
Private key
Signature
Get challenge and PIN
Decompose incoming message
Compare PINs
Equal
Sign the challenge
Send back the signature
Not equal
Have we been in this branch for too many times?
Not yet
Yes
Real PIN
Self-destruct

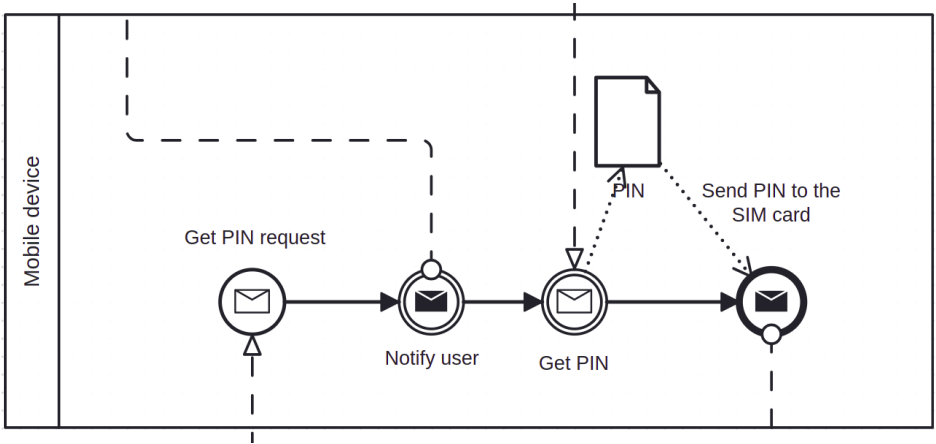
Authentication with SIM card

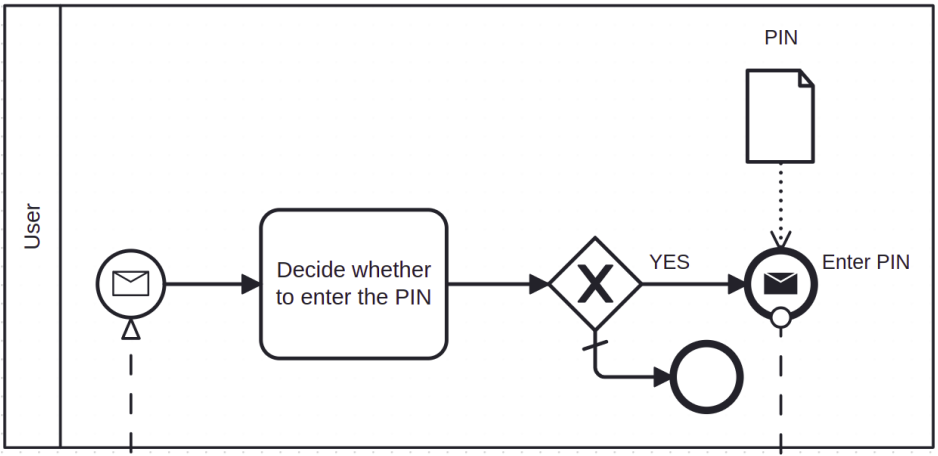


CYBERNETICA


Relaying components
Relying party
Login has reached the point where the client has to be authenticated
Come up with a challenge (sequence of exchanged messages)
Challenge
Ask for a signature
Receive signature
Signature
Client's public key
Check signature
NOT OK
OK
Continue session
Browser
Start session
Handshake...
Continue session


SIM
Get challenge
Ask for PIN
Get PIN
Challenge
Compare PINs
Potential PIN
Real PIN
Equal
Not equal
Private key
Signature
Sign the challenge
Send back the signature
Have we been in this branch for too many times?
Not yet
Yes
Self-destruct


Mobile device
Get PIN request
Notify user
Get PIN
PIN
Send PIN to the SIM card


User
Decide whether to enter the PIN
YES
PIN
Enter PIN

Possible weaknesses

- Relying party
- Browser
- User
- Card reader
- Mobile device
- Chip (Smartcard or SIM)
- Relaying parties

Possible weaknesses

Relying party

(RP1) Affect the computation of the challenge

(RP2) Learn the challenge

(RP3) Modify the challenge while it is sent out

(RP4) Change the outcome of the signature check

(RP5) Accept the log-in, even if the signature does not check

Possible weaknesses

Browser (in smartcard authentication)

- (B1) Learn the challenge
- (B2) Modify the challenge while it is sent to the card reader
- (B3) Learn the signature
- (B4) Modify the signature while it is sent to the relying party

Possible weaknesses

User

(U1) Learn the PIN from the user

(U2) Change the PIN

Possible weaknesses

Card reader

(CR1) Learn the PIN that the user entered

(CR2) Learn the challenge

(CR3) Change the challenge that is sent to the smartcard

(CR4) Change the PIN that is sent to the smartcard

(CR5) Learn the signature

(CR6) Change the signature while it is sent to the browser

Possible weaknesses

Mobile device

(MD1) Learn the PIN that the user entered

(MD2) Change the PIN that is sent to the smartcard

Possible weaknesses

Chip (Smartcard or SIM)

(CH1) Learn the PIN
(CH2) Interfere with the PIN comparison procedure
(CH3) Make the decision of the PIN check take the other path
(CH4) Make the decision about counts of incorrect PINs take the other path
(CH5) Learn the private key
(CH6) Change the private key
(CH7) Change the challenge that enters the computation of the signature
(CH8) Learn the signature
(CH9) Change the signature that is sent back

Possible weaknesses

Parties relaying the messages for SIM card

(CN1) Learn the challenge
(CN2) Change the challenge
(CN3) Learn the signature
(CN4) Change the signature

Relationships between weaknesses

Smartcard only	Both	SIM card only
(B*), (CR*)	(MP*), (U*), (CH*)	(MD*), (CN*)

Relationships between weaknesses

Smartcard only	Both	SIM card only
(B*), (CR*)	(MP*), (U*), (CH*)	(MD*), (CN*)

$$\{(MD1)\} \prec \{(CR1)\}$$
$$\{(MD2)\} \prec \{(CR4)\}$$

$$\{(CN1)\} \prec \{(B1)\}$$
$$\{(CN2)\} \prec \{(B2)\}$$
$$\{(CN3)\} \prec \{(B3)\}$$
$$\{(CN4)\} \prec \{(B4)\}$$

Relationships between weaknesses

Smartcard only	Both	SIM card only
(B*), (CR*)	(MP*), (U*), (CH*)	(MD*), (CN*)

$$\{(\text{MD1})\} \prec \{(\text{CR1})\}$$
$$\{(\text{MD2})\} \prec \{(\text{CR4})\}$$

$$\{(\text{CN1})\} \prec \{(\text{B1}), (\text{RP2}), (\text{CR2})\}$$
$$\{(\text{CN2})\} \prec \{(\text{B2}), (\text{RP3}), (\text{CR3})\}$$
$$\{(\text{CN3})\} \prec \{(\text{B3}), (\text{CR5})\}$$
$$\{(\text{CN4})\} \prec \{(\text{B4}), (\text{RP4}), (\text{CR6})\}$$

Relationships between weaknesses

Smartcard only	Both	SIM card only
(B*), (CR*)	(MP*), (U*), (CH*)	(MD*), (CN*)

$$\{(\text{MD1})\} \prec \{(\text{CR1})\}$$
$$\{(\text{MD2})\} \prec \{(\text{CR4})\}$$

$$\{(\text{CN1})\} \prec \{(\text{B1}), (\text{RP2}), (\text{CR2})\}$$
$$\{(\text{CN2})\} \prec \{(\text{B2}), (\text{RP3}), (\text{CR3})\}$$
$$\{(\text{CN3})\} \prec \{(\text{B3}), (\text{CR5})\}$$
$$\{(\text{CN4})\} \prec \{(\text{B4}), (\text{RP4}), (\text{CR6})\}$$

“at least as bad” for given sets of weaknesses

The relationships above allow to establish it for all $\mathbf{w}^\bullet \subseteq W^\bullet$ (assuming monotonicity)

Not all sets of weaknesses...

- The ST document of T° did not consider every $\mathbf{w} \subseteq W^\circ$
  - There are considered threats, and assumptions about the operational environments
  - These determine, which sets of weaknesses are expected to be exploited, and exploited together
- We have $\mathcal{W}^\circ \subseteq 2^{W^\circ}$ and $\mathcal{W}^\bullet \subseteq 2^{W^\bullet}$: the considered sets of weaknesses
  - These may affect our treatment of (CN*)-weaknesses in particular

Propagation of security requirements

- $\{(MD1)\} \prec \{(CR1)\}$ and $\{(MD2)\} \prec \{(CR4)\}$
- I.e. PIN entry through mobile device must be at least as weakness-free as the entry through a card reader
  - Whether we accept this or not, depends on the card reader…
  - I would accept it, if the card reader actually uses computer's keyboard

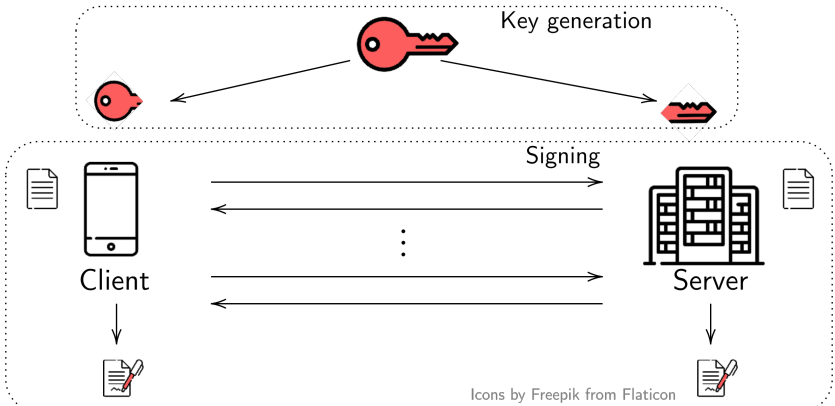
Sequences of systems

- What if T° and $T^{\bullet}$ are too different?
- Come up with intermediate systems!
- Similar to security proofs of cryptographic primitives
  - The “sequence of games” method
  - First game is algorithm + security definition. Last game is “obviously secure”
    - Steps $G_i \to G_{i+1}$ are simple to analyse
- The intermediate systems do not have to be “realistic”
  - They still must have well-defined behaviour and weaknesses

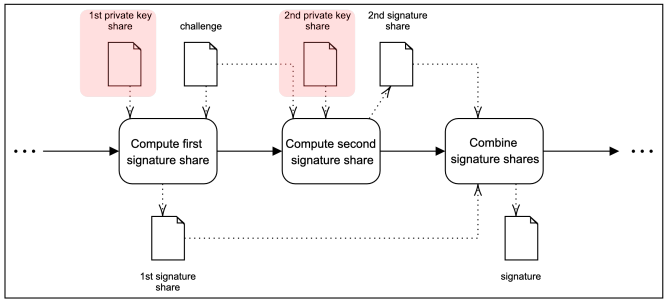
Example: SplitKey

- A threshold signing solution with two parties
  - First party: the smartphone. Keyshare encrypted with a PIN
  - Second party: a central server
  - Resulting signature: looks like a normal RSA signature
- Some measures for the server to detect that the phone could not protect its keyshare
- Used for authentication and signing
- Approx. 3.4 million users in EE+LV+LT
  - A separate deployment in IS. And in BE
- Cybernetica's technology. SK ID Solutions's service. smart-id.com

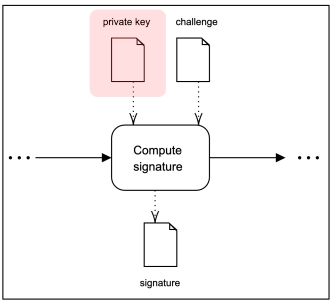
Threshold signature



Comparison step



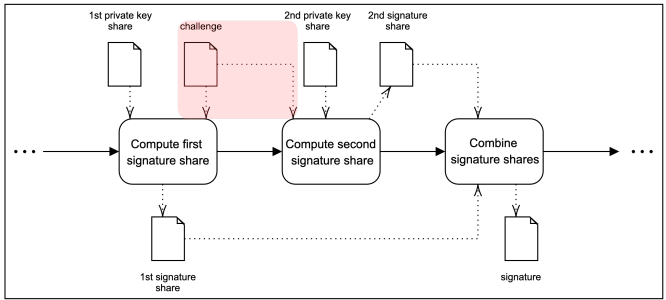
System $T^{\bullet}$



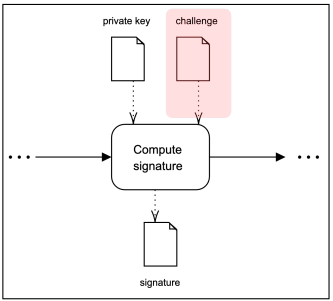
System T°

($T^{\bullet}$.learn 1st key share) AND ($T^{\bullet}$.learn 2nd key share) $\Leftrightarrow$ T°.learn private key

Comparison step



System $T^{\bullet}$



System T°

($T^{\bullet}$.change challenge for 1st signature share) OR
($T^{\bullet}$.change challenge for 2nd signature share) $\Leftrightarrow$ T°.change challenge

Intermediate systems

- $T^{\bullet} \prec T_1 \prec T_2 \prec T^{\circ}$

T_1

- All crypto happens in server
- Both keyshares stored in server
- First keyshare encrypted with PIN
- Phone sends PIN to server

T_2

- Server computes a non-threshold signature
- Phone sends PIN to server
- Server compares PIN with stored PIN

Propagating back the security requirements

- T° has certain protection mechanisms in place
- Phone $\leftrightarrow$ server channel in $T^{\bullet}$ must be no weaker than (CR*)
- Private key in T° is protected somehow. At least the same kind of protection has to be available to at least one share of the key in $T^{\bullet}$
- etc.
- With intermediate systems, these requirements may propagate all the way to $T^{\bullet}$
  - but may also become trivial in some intermediate system

Example: voting

- We have tried the proposed method to compare socio-technical systems
- T°: vote by mail
  - Exists in EE since 1998
- $T^{\bullet}$: internet voting (in EE)
  - Exists in EE since 2005
- A difficulty: the specification of T° is not too detailed

Representing sets of weaknesses and mappings between them

- Let $\mathbb{B} = \{\text{true}, \text{false}\}$
- $\mathbf{w} \subseteq W$ represented as assignment $W \to \mathbb{B}$
- $\mathcal{W} \subseteq 2^W$ represented as boolean function $(W \to \mathbb{B}) \to \mathbb{B}$
  - Boolean functions can be represented as boolean formulas, or BDDs, or…
- Mapping $f : 2^{W^\bullet} \to 2^{W^\circ}$ is thought as a relation $R_f \subseteq W^\bullet \times W^\circ$
  - Requires a separate argument of R_f being *serial*
- $\prec$ can be expressed as implication

Current status

- We have done a couple of examples by hand. The trustworthiness of manual analysis is so-so...
- Methodology needs evaluation / acceptance from certification bodies
- Needs tool support to
  - collect the conceivable and inconceivable weaknesses from the description of systems;
  - compare the sets of attacks;
  - propagate security requirements

Discussion

- Discussion

cybernetica
Cybernetica
cybernetica_ee
Cybernetica