Willkommen zur Vorlesung

Softwarearchitekturen im Finanz- und Versicherungsbereich

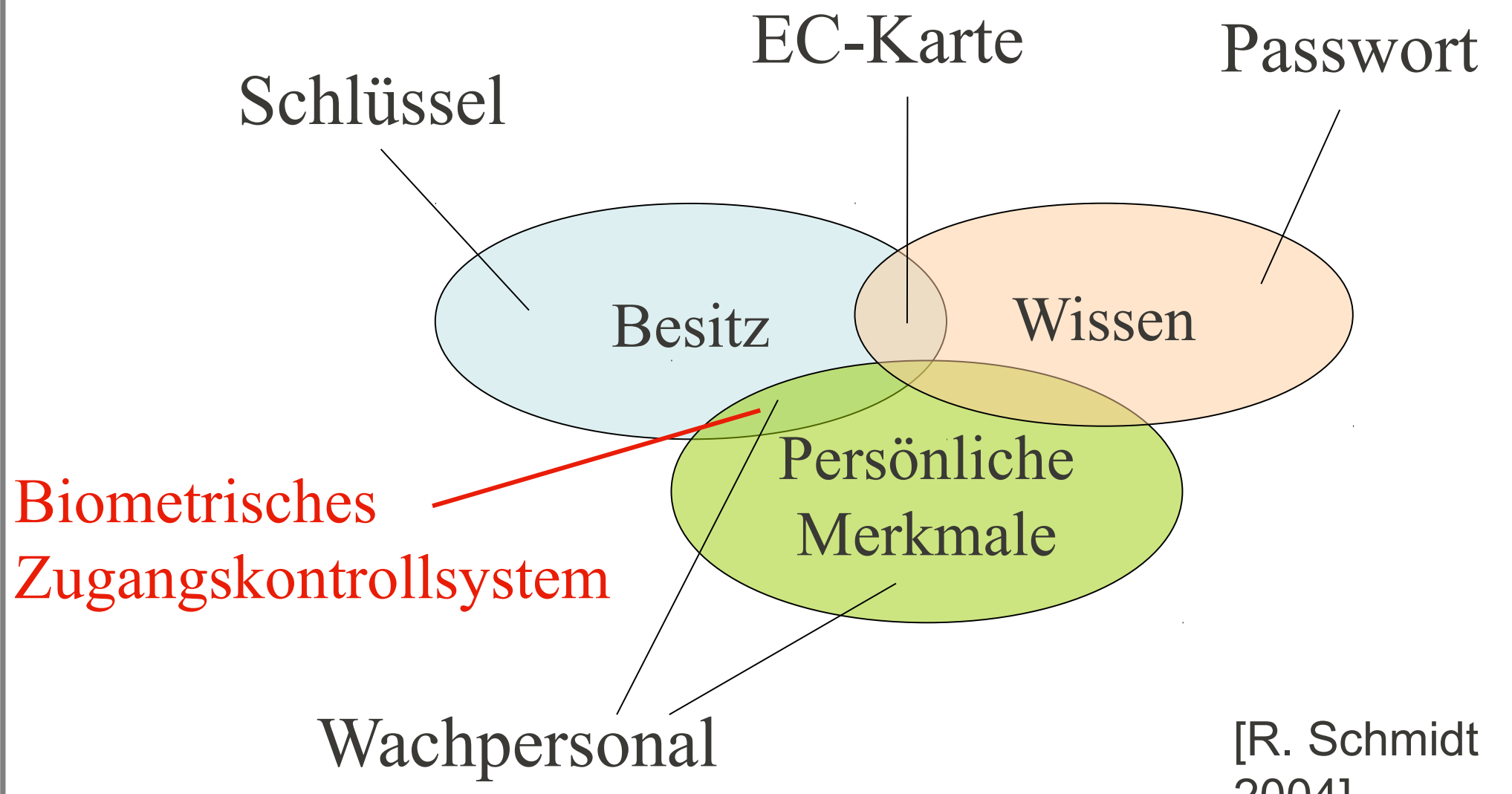
im Sommersemester 2010

Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

8.Biometrische Authentisierung

Zugangskontrollmechanismen



[R. Schmidt 2004]

Zugangskontrolle: Beispiele

Ziel: Sicherer Schutz vor unberechtigtem Zugang / Zutritt

Beispiele:

Gegenstand	Schutzmaßnahme	Kategorie
Raum / Tür	Schlüssel Wachmann	Besitz Persönliche Merkmale
Computer	Passwort	Wissen
Konto	Karte mit PIN	Besitz + Wissen

Zugangskontrolle: Probleme

Schwachstellen

Schutzmaßnahme	Problem
Schlüssel Wachmann	verloren / gestohlen worden Bestechlich / 24 h Einsatz?
Passwort	Am Arbeitsplatz notiert, an „Administrator“ verraten
Karte mit PIN	PIN auf Karte notiert

Loesung (?): Biometrie

Einige persönliche Merkmale: Handgeo-metrie, Augennetzhaut, Augeniris, Venen, Unterschrift, Fingerabdruck, Stimmen, Gesicht, DNA, Geruch...

Identifikation: 1 : n Vergleich,
Verifikation: 1 : 1 Vergleich

Biometrisches System führt biometrische Verifikation / Identifikation durch.

Aktuell: USA-Einreise; Reisepass.

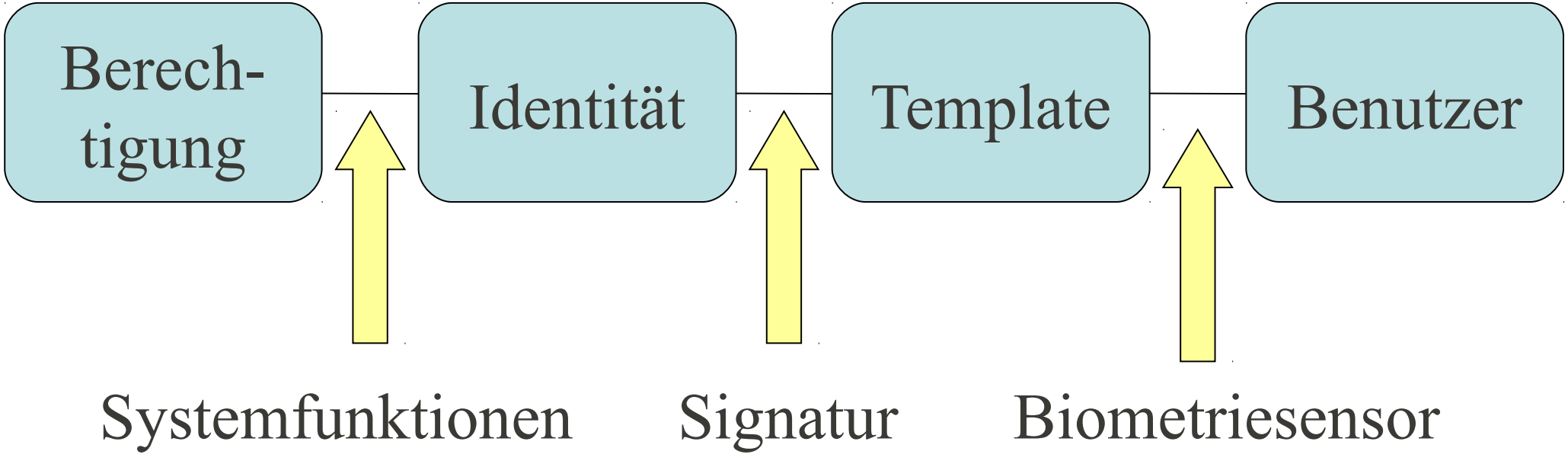
Effizienz biometrischer Verfahren

[Liu, Silverman 2001]

	Usability	Kosten	Geschwindig.	Genauigkeit	Sicherheitsanf.	Akzeptanz
Gesichtserkenn.	o	--	--	+	o	o
Fingerabdruck	+	+	o	+	-	+
Handgeometrie	+	-	-	+	o	o
Iris-Scan	o	--	--	++	--	o
Sprache	+	++	+	+	o	+
Untersch.	+	++	+	+	o	o

Biometrisches System realisiert

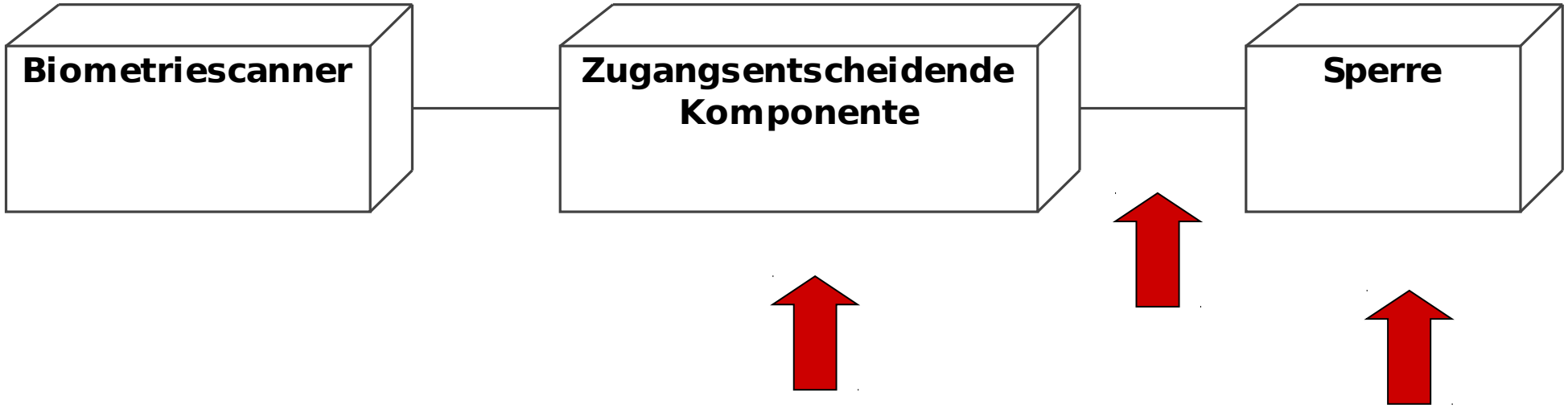
Verknüpfungskette:



Funktionsweise

- Scannen biometrischer Daten, Template extrahieren.
- Vergleich mit gespeichertem Referenztemplate.
- Bei ausreichender Übereinstimmung entsperren.

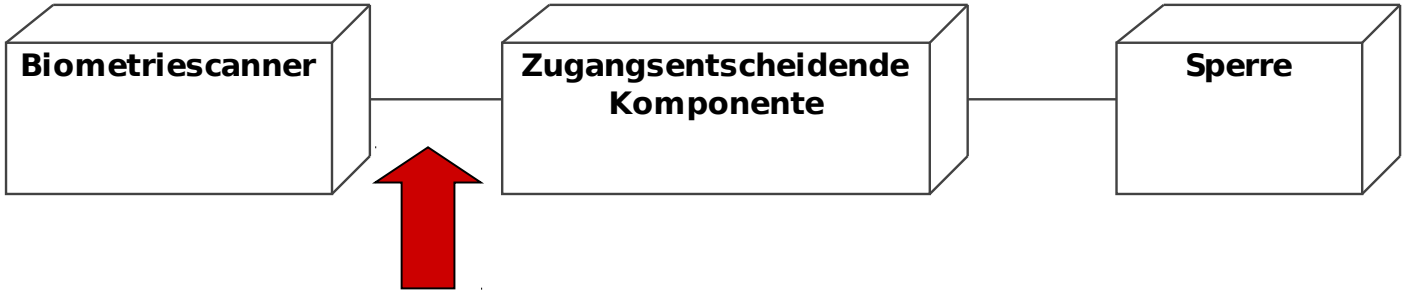
Angriffspunkte I



Angriff: Aufbrechen und / oder elektrisches Signal einspeisen.

➔ Physikalischer Schutz notwendig.

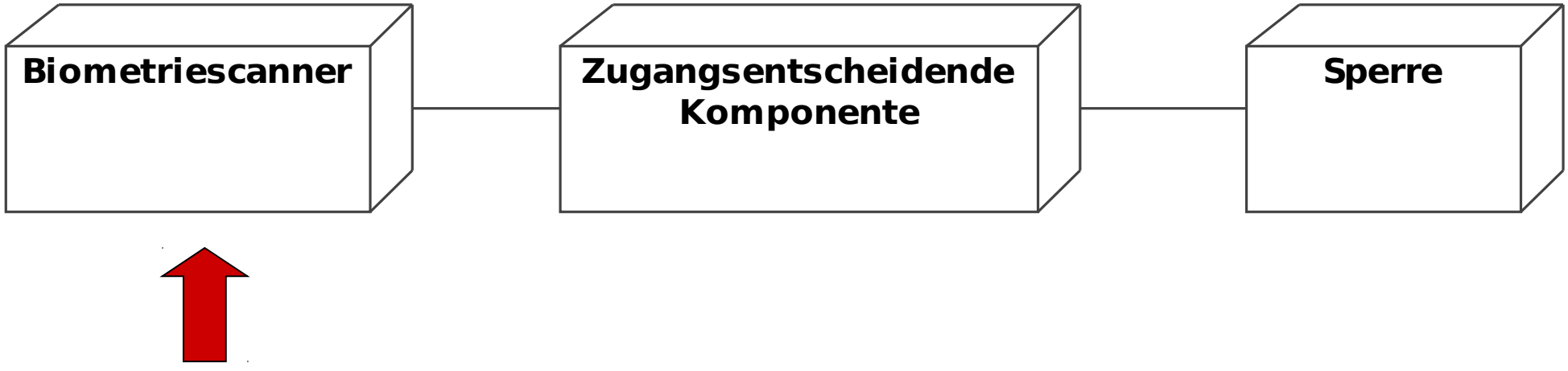
Angriffspunkte II



Angriff: Zugangsberechtigtes Template ablauschen und einspeisen.

➔ Physikalischer Schutz, oder Schutz durch Kryptographie.

Angriffspunkte III



Angriff: Imitation von Körperteilen, zum Beispiel Silikonfinger.

➔ Qualität des Biometriescanners erhöhen (Lebenderkennung).

Problem: Ewiger Wettlauf ?

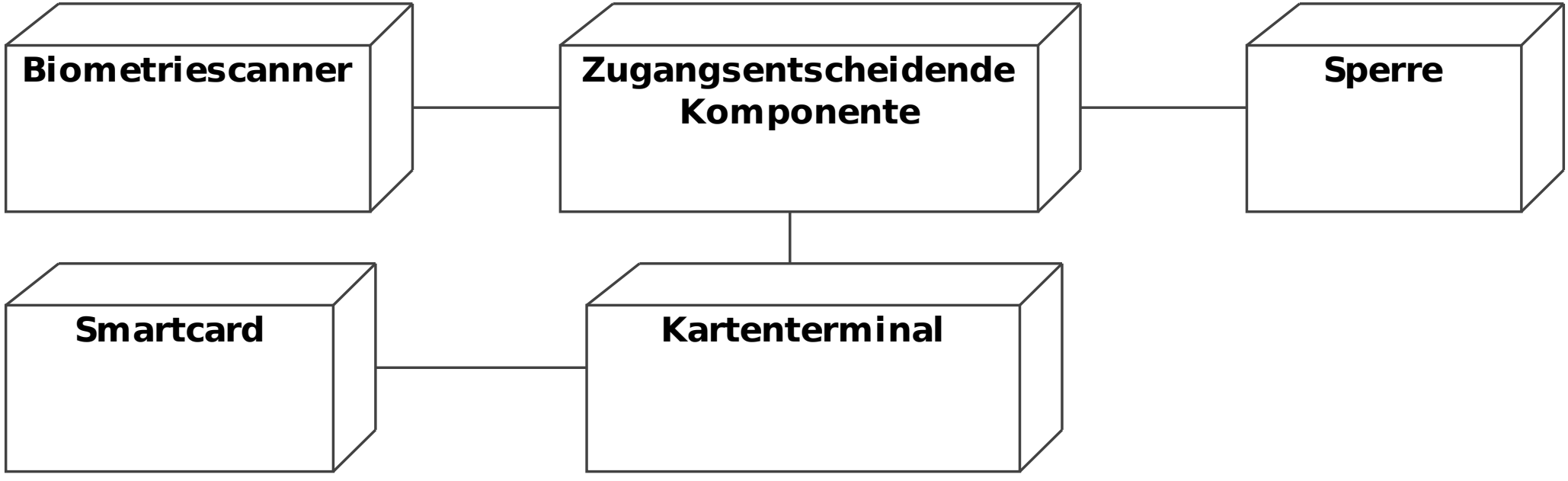
Einschränkungen

Realisierbarer Modus: Identifikation (1:n)

Notwendig: zentrale Speicherung biometrischer Datensätze. Probleme:

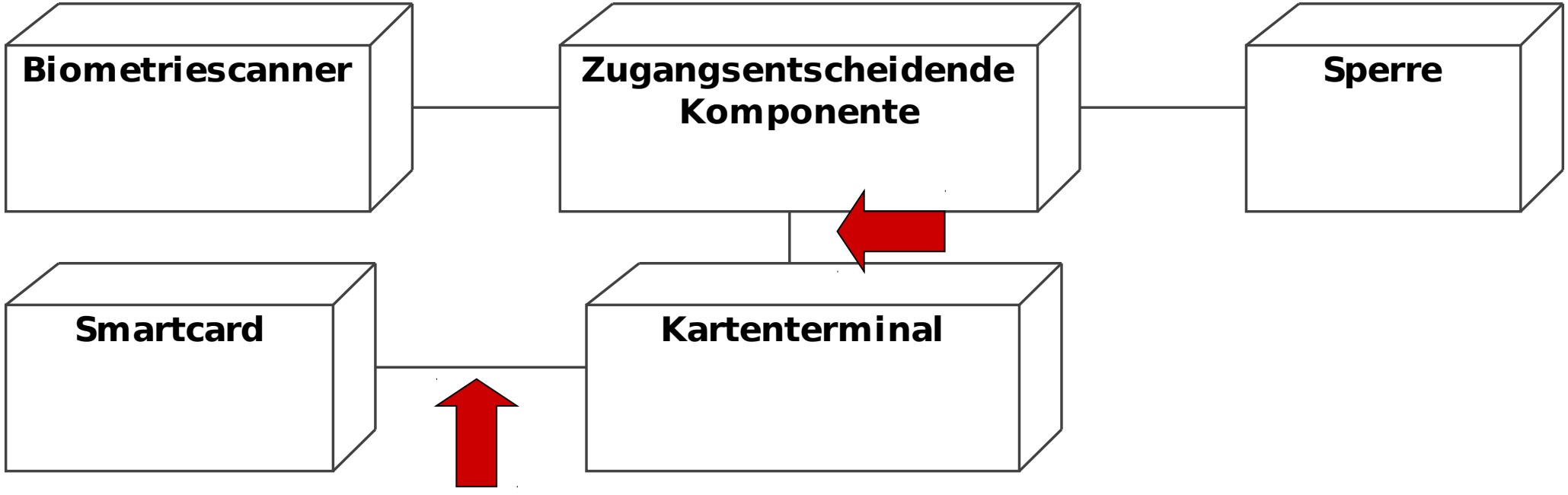
- mehrere Referenzdatensätze betrachten
- Speicherung persönlicher Merkmale unterliegt Datenschutz

System mit personifizierter Smartcard



- Referenztemplate auf Smartcard gespeichert.
- Besitzer der Smartcard trägt Verantwortung für seine biometrischen Daten: Datenschutz.
- Realisierbarer Modus: Verifikation (1:1).

Angriffspunkte IV

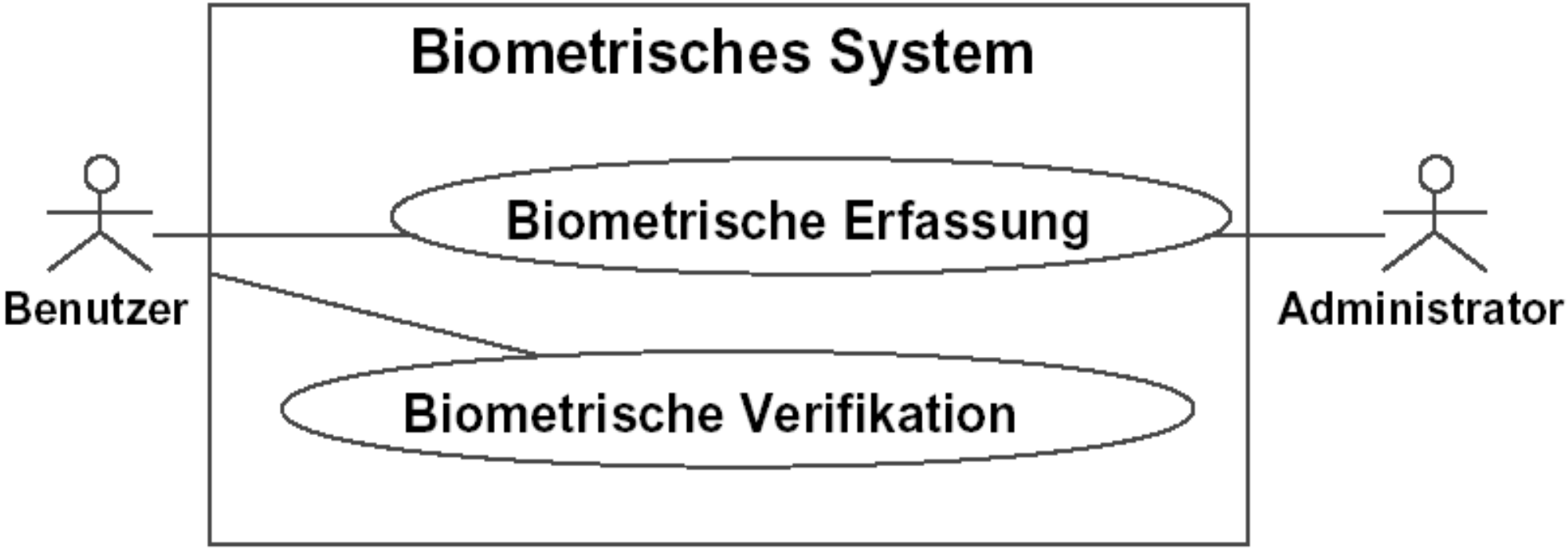


Angriff: Vergleich zwischen gespeichertem Reference-Template und aktuellem Wert manipulieren.

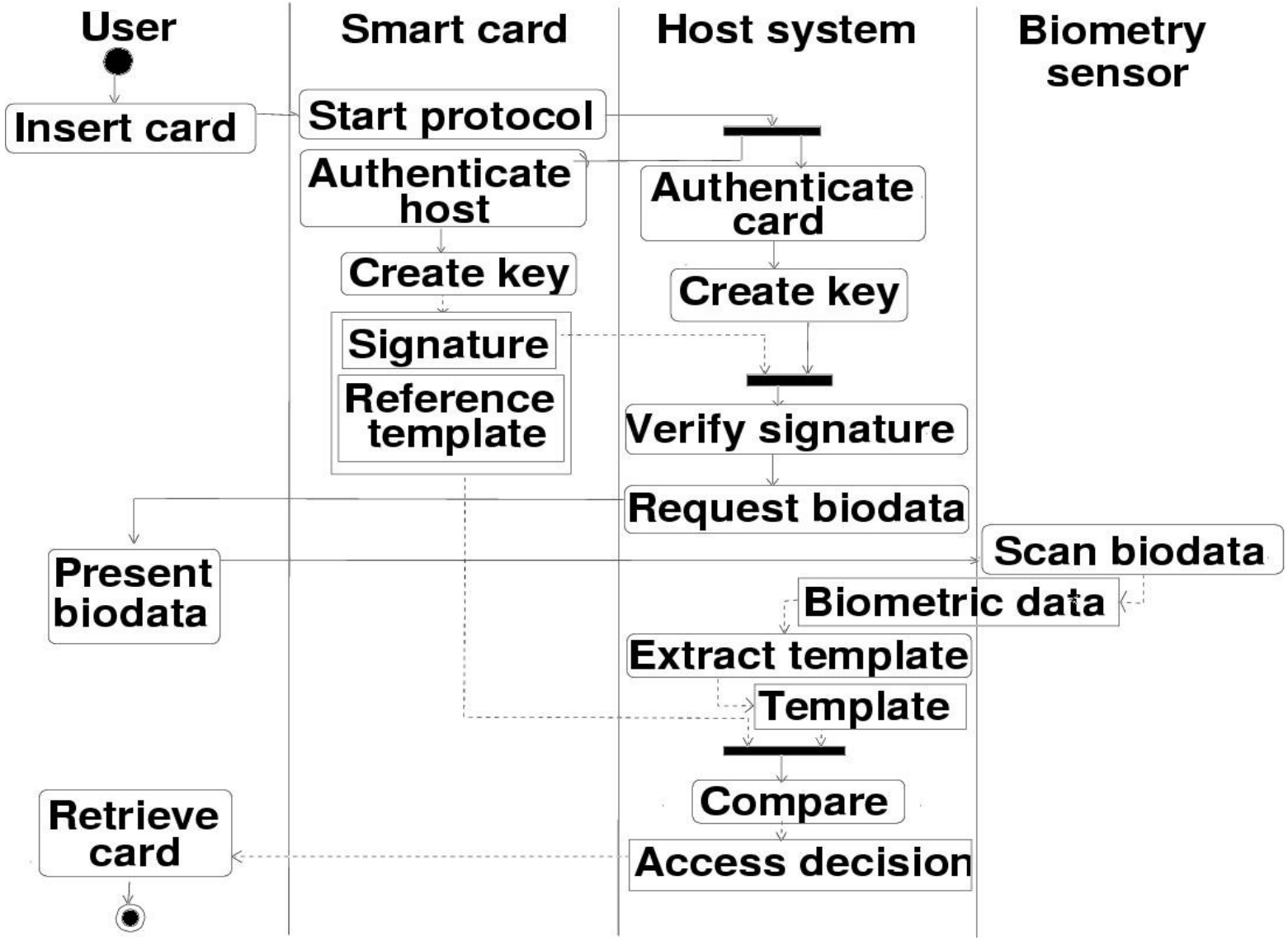
➔ Kryptographische Authentifikation.

Biometrisches Authentifikationssystem in industrieller Entwicklung.

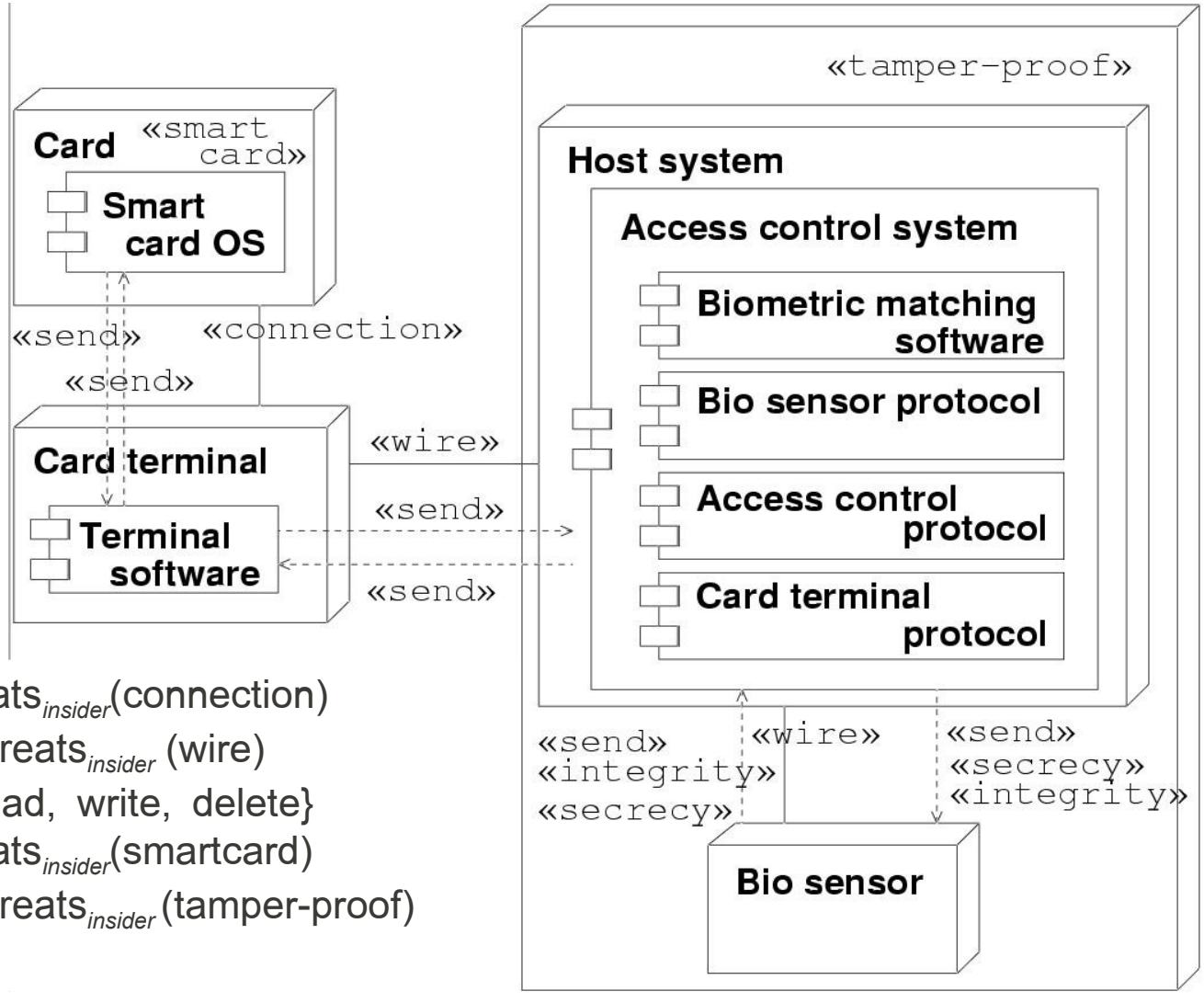
Anwendungsfälle



Use case: Biometric verification

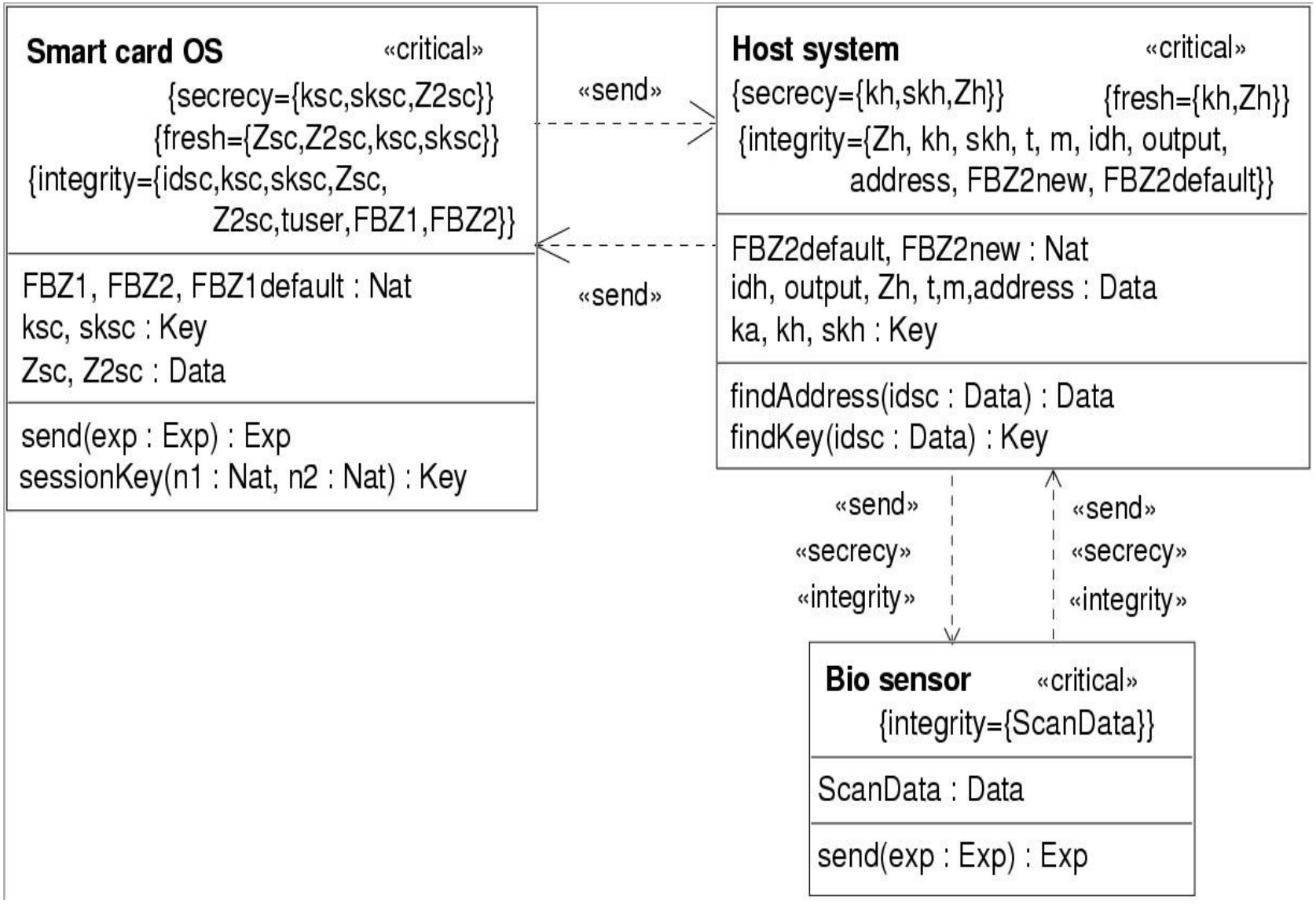


Architecture

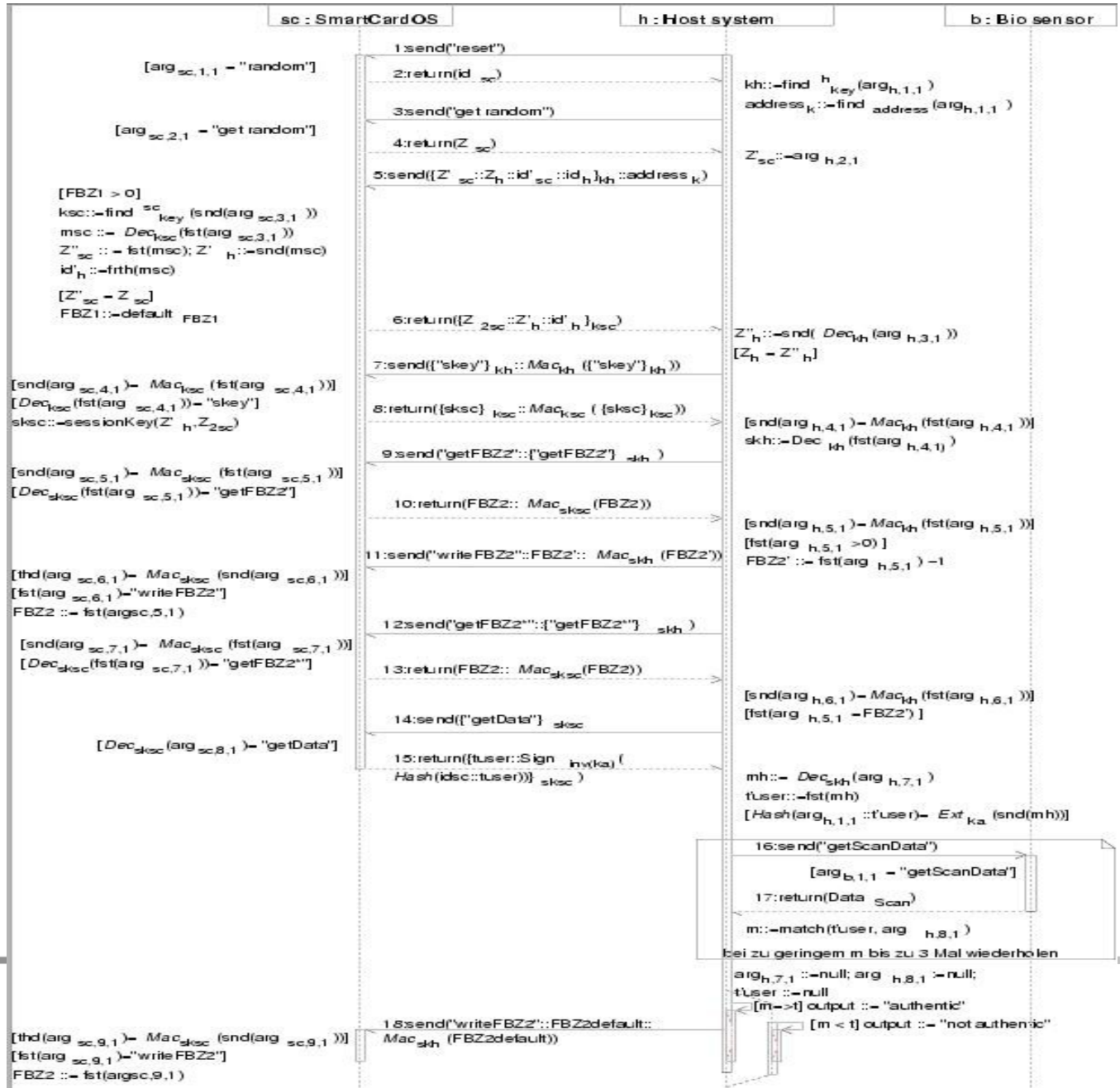


$Threats_{insider}$(connection)
= $Threats_{insider}$ (wire)
= {read, write, delete}
$Threats_{insider}$(smartcard)
= $Threats_{insider}$ (tamper-proof)
= Ø

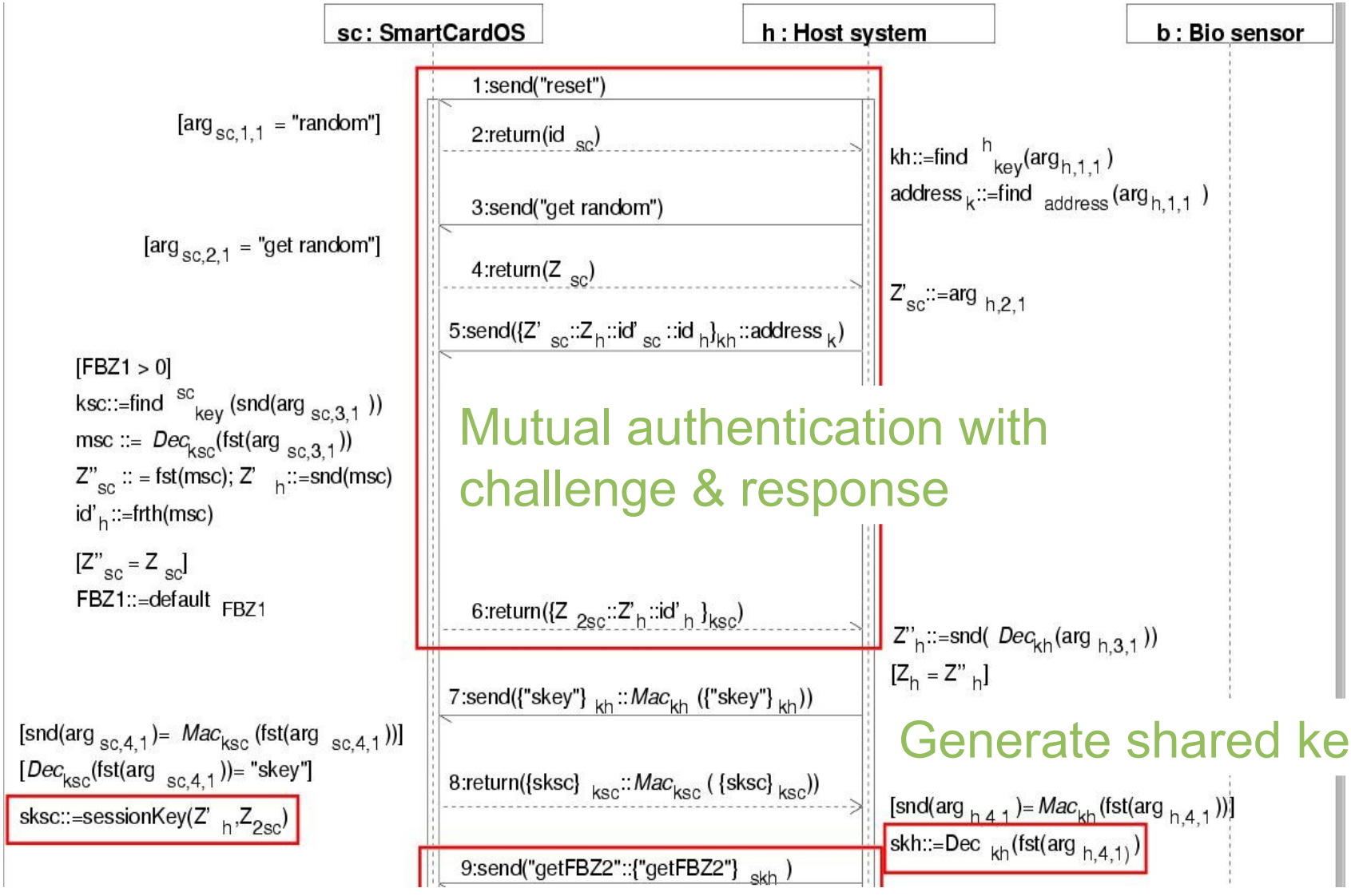
Class diagram



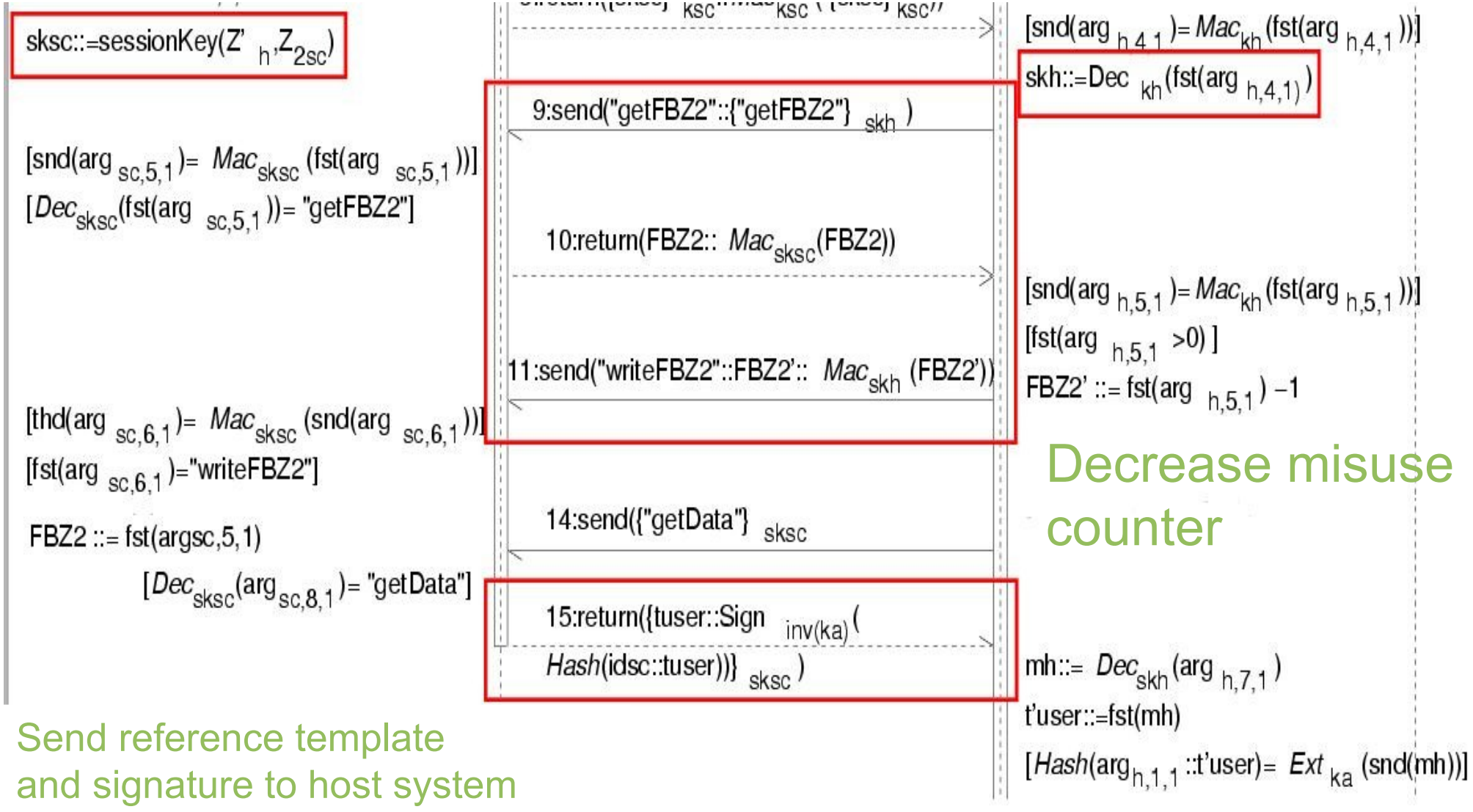
Protocol


sc : SmartCardOS
h : Host system
b : Bio sensor
1:send("reset")
[arg sc,1,1 = "random"]
2:return(id sc)
kh::=find h key(arg h,1,1)
address k::=find address(arg h,1,1)
3:send("get random")
[arg sc,2,1 = "get random"]
4:return(Z sc)
Z'sc::=arg h,2,1
5:send({Z' sc::Zh::id'sc::id h}kh::address k)
[FBZ1 > 0]
ksc::=find sc key(snd(arg sc,3,1))
msc ::= Dec ksc(fst(arg sc,3,1))
Z''sc ::= fst(msc); Z' h::=snd(msc)
id'h ::=frth(msc)
[Z''sc = Z sc]
FBZ1::=default FBZ1
6:return({Z 2sc::Z'h::id'h}ksc)
Z''h::=snd(Dec kh(arg h,3,1))
[Zh = Z''h]
7:send({"skey"}kh::Mac kh({"skey"}kh))
[snd(arg sc,4,1)= Mac ksc(fst(arg sc,4,1))]
[Dec ksc(fst(arg sc,4,1))= "skey"]
sksc::=sessionKey(Z' h,Z2sc)
8:return({sksc}ksc::Mac ksc({sksc}ksc))
[snd(arg h,4,1)= Mac kh(fst(arg h,4,1))]
skh::=Dec kh(fst(arg h,4,1))
9:send("getFBZ2"::{"getFBZ2"}skh)
[snd(arg sc,5,1)= Mac sksc(fst(arg sc,5,1))]
[Dec sksc(fst(arg sc,5,1))= "getFBZ2"]
10:return(FBZ2:: Mac sksc(FBZ2))
[snd(arg h,5,1)= Mac kh(fst(arg h,5,1))]
[fst(arg h,5,1 >0)]
FBZ2' ::= fst(arg h,5,1) -1
11:send("writeFBZ2"::FBZ2':: Mac skh(FBZ2'))
[thd(arg sc,6,1)= Mac sksc(snd(arg sc,6,1))]
[fst(arg sc,6,1)="writeFBZ2"]
FBZ2 ::= fst(argsc,5,1)
12:send("getFBZ2"'::{"getFBZ2"'}skh)
[snd(arg sc,7,1)= Mac sksc(fst(arg sc,7,1))]
[Dec sksc(fst(arg sc,7,1))= "getFBZ2"']
13:return(FBZ2:: Mac sksc(FBZ2))
[snd(arg h,6,1)= Mac kh(fst(arg h,6,1))]
[fst(arg h,5,1 =FBZ2')]
14:send({"getData"}sksc
[Dec sksc(arg sc,8,1)= "getData"]
15:return({tuser::Sign inv(ka)(Hash(idsc::tuser))}sksc)
mh::= Dec skh(arg h,7,1)
tuser::=fst(mh)
[Hash(arg h,1,1::tuser)= Ext ka(snd(mh))]
16:send("getScanData")
[arg b,1,1 = "getScanData"]
17:return(Data Scan)
m::=match(tuser, arg h,8,1)
bei zu geringem m bis zu 3 Mal wiederholen
arg h,7,1 ::=null; arg h,8,1 :=null;
tuser ::= null
[m->t] output ::= "authentic"
[m < t] output ::= "not authentic"
18:send("writeFBZ2"::FBZ2default::Mac skh(FBZ2default))
[thd(arg sc,9,1)= Mac sksc(snd(arg sc,9,1))]
[fst(arg sc,9,1)="writeFBZ2"]
FBZ2 ::= fst(argsc,9,1)

Authentication Protocol Part 1



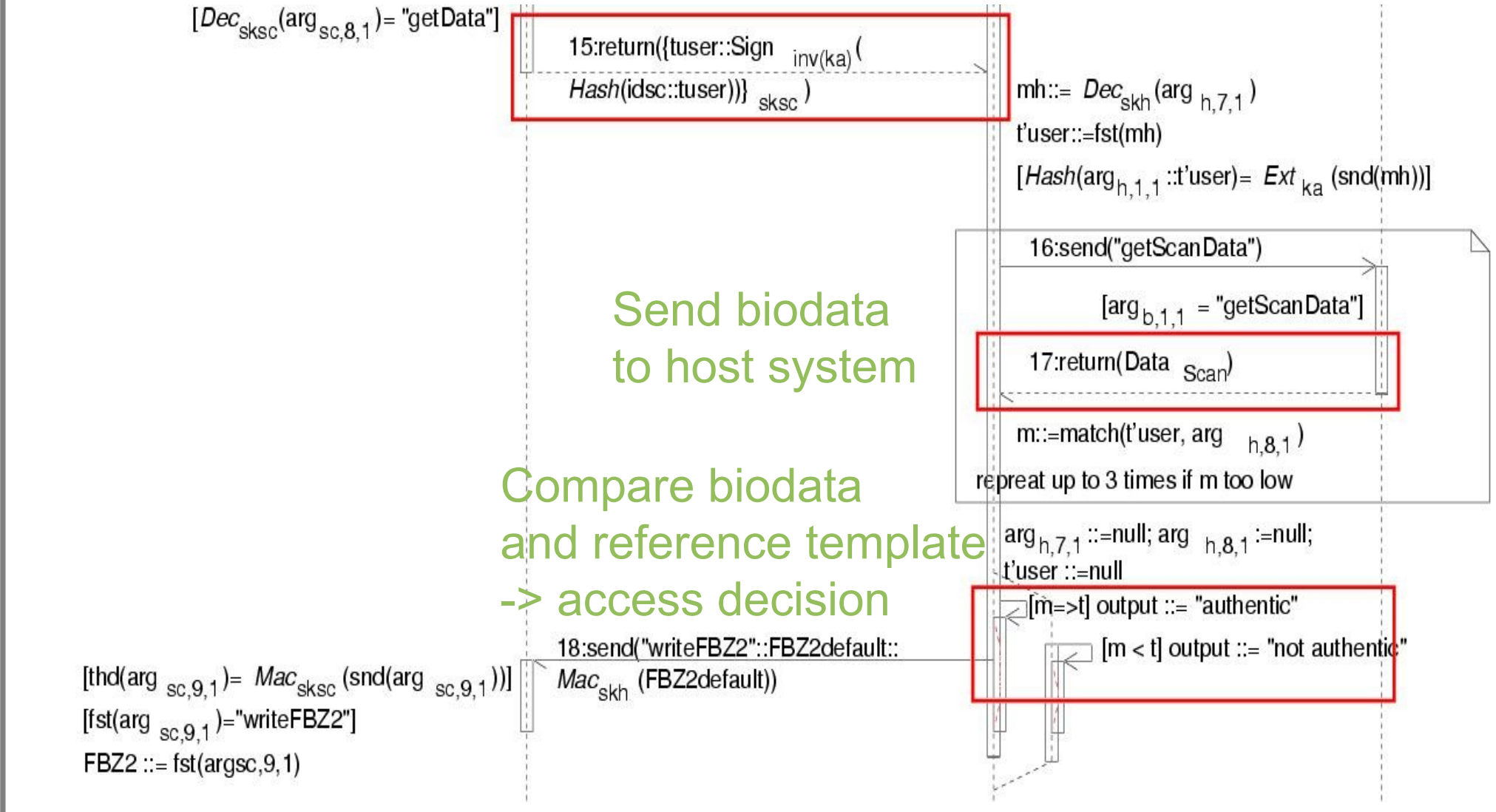
Authentication Protocol Part 2



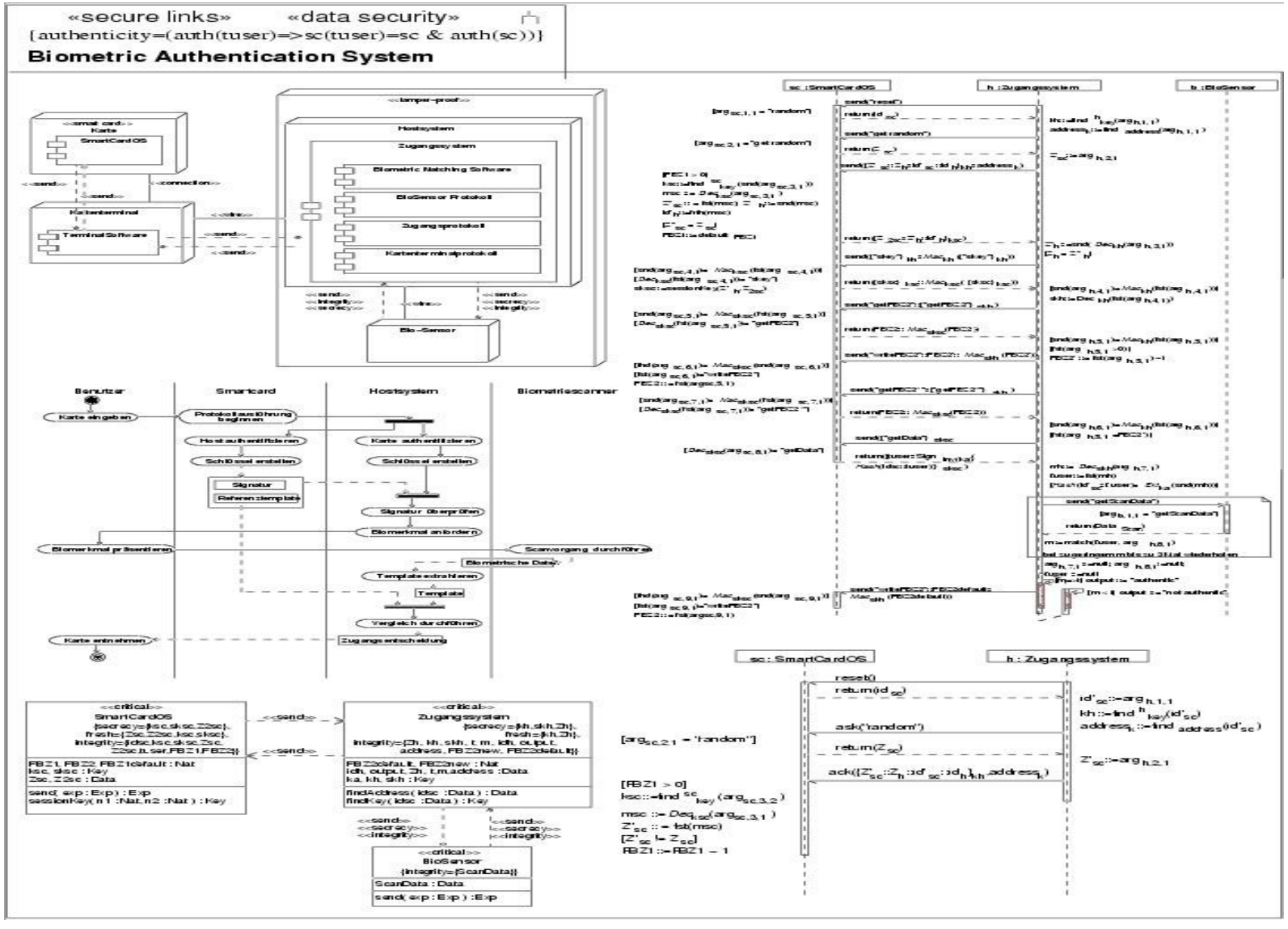
Decrease misuse counter

Send reference template and signature to host system

Authentication Protocol Part 3



Big Picture


«secure links» «data security»
{authenticity=(auth(tuser)=>sc(tuser)=sc & auth(sc))}
Biometric Authentication System
Benutzer
Smartcard
Hostsystem
Biometriescanner
Karte eingeben
Protokollaushandlung beginnen
Host authentifizieren
Schlüssel erstellen
Signatur
Referenztemplate
Karte authentifizieren
Schlüssel erstellen
Signatur überprüfen
Biomerkmal anfordern
Biomerkmal präsentieren
Scanvorgang durchführen
Biometrische Daten
Template extrahieren
Template
Vergleich durchführen
Karte entnehmen
Zugangsentscheidung
SmartCardOS
Zugangssystem
BioSensor
sc : SmartCardOS
h : Zugangssystem
h : BioSensor
reset()
return(id_sc)
ask('random')
return(Z_sc)
ack([Z'_sc::Z_h::id'_sc::id_h,k_h,address_k])
[arg_sc,2,1 = 'random']
[RBZ1 > 0]
RBZ1 := RBZ1 − 1

Sicherheitsanalyse

Mögliches unerwünschtes Verhalten:

- Zugangberechtigte Person erhält keinen Zutritt
- Zugangberechtigte Person erhält Zutritt unter fremder Identität
- Person ohne Zugangsberechtigung erhält Zutritt

Rollen:

- Benutzer: Besitzer von legitimierter Smartcard
- Administrator: stellt Smartcards aus
- System: durch das biometrische System geschützter Bereich

Bedrohungen

- Benutzer: Angreifer richtet unter Identität des Benutzers Schaden an.
- Administrator: Beschuldigt, einer unberechtigten Person eine Smartcard angefertigt zu haben.
- System:
  1. Unberechtigte Person hat Zutritt erhalten.
  2. Schuldiger ist im Schadensfall nicht eindeutig zu identifizieren.
- Datenschutz: Ein Angreifer erhält ohne Zustimmung ein biometrisches Template

Sicherheitsziele

- Benutzer: Nur er darf (nur) unter seiner Identität Zugang erhalten.
- Administrator: Nur er darf in der Lage sein eine personifizierte Smartcard erstellen, die im System erfolgreich Zugang erhält.
- System: Nur zugangsberechtigte Personen erhalten nachweisbar Zugang.
- Datenschutz: Vertraulichkeit des biometrischen Templates muss gewährt sein.

Sicherheitsziele formalisiert

- Sicherheit des Benutzers: Aus output = authentic folgt für $x_i = \mathsf{arg}_{\mathsf{h},1,1_\mathsf{i}}$: es existieren Werte a, b, c so, dass $\mathsf{arg}_{\mathsf{h},5,1_\mathsf{i}} = \{a :: \mathcal{S}ign_{\mathsf{k_a}^{-1}}(\mathcal{H}ash(x_i :: b))\}_c$

- Sicherheit des Administrators: $\mathcal{K}_A \cap \{\mathsf{k_a^{-1}}\} = \emptyset.$

- Sicherheit des Systems: Aus output = authentic folgt für $x_i = \mathsf{arg}_{\mathsf{h},1,1_\mathsf{i}}$: es existieren Werte a, b, c so, dass $\mathsf{arg}_{\mathsf{h},5,1_\mathsf{i}} = \{a :: \mathcal{S}ign_{\mathsf{k_a}^{-1}}(\mathcal{H}ash(x_i :: b))\}_c$

Datenschutzanforderung: $\mathcal{K}_A \cap \{\mathsf{t_{user}}\} = \emptyset.$

Translation to First-order Logic II

Message order not enforced by smart card (!).

Connection from smart card

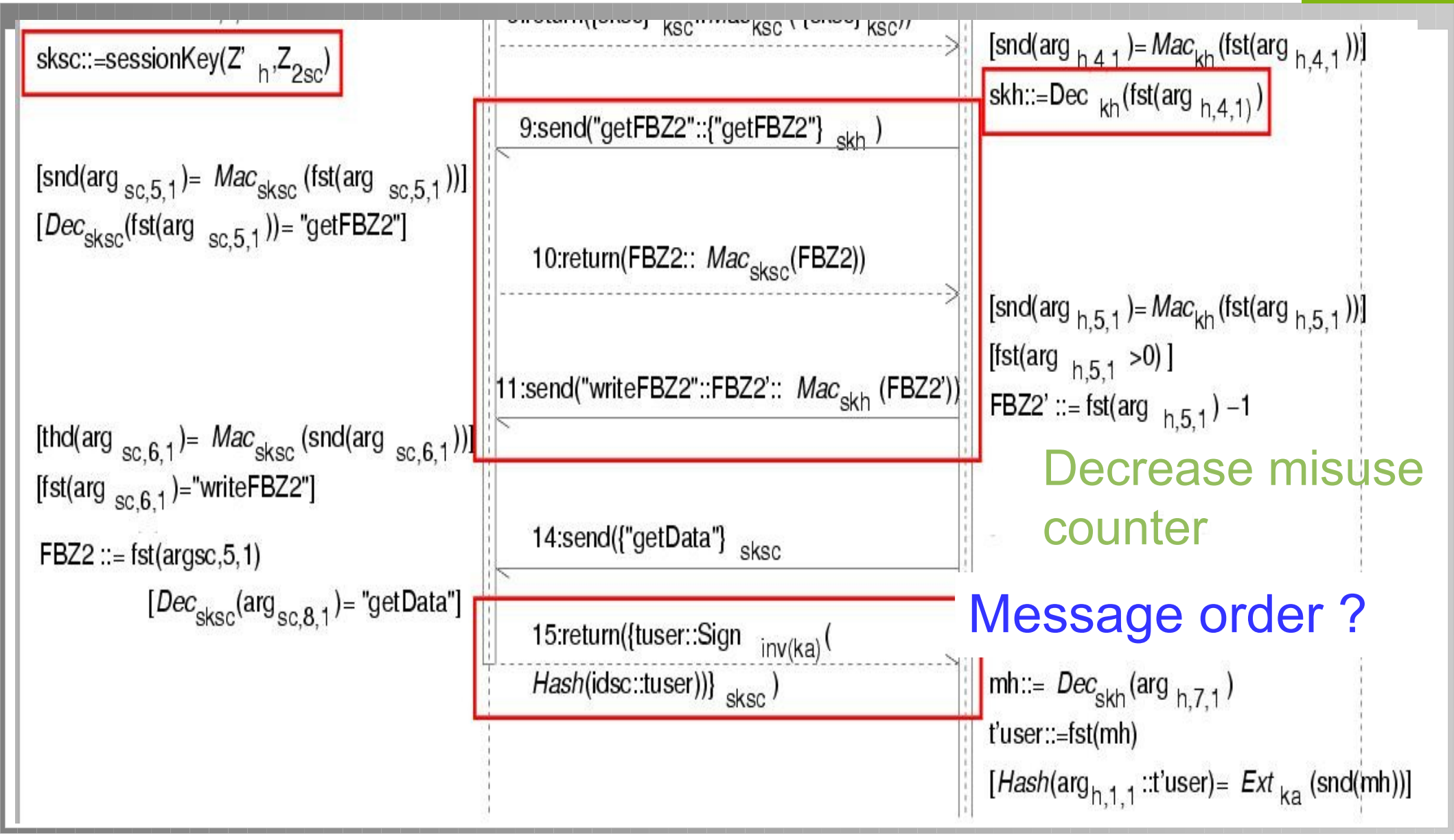
TR1=(in(msg_in),cond(msg_in),out(msg_out))

followed by *TR2* gives predicate

PRED(TR1)=

$$\forall msg\_in.\ [knows(msg\_in) \wedge cond(msg\_in) \Rightarrow knows(msg\_out)] \wedge PRED(TR2)$$

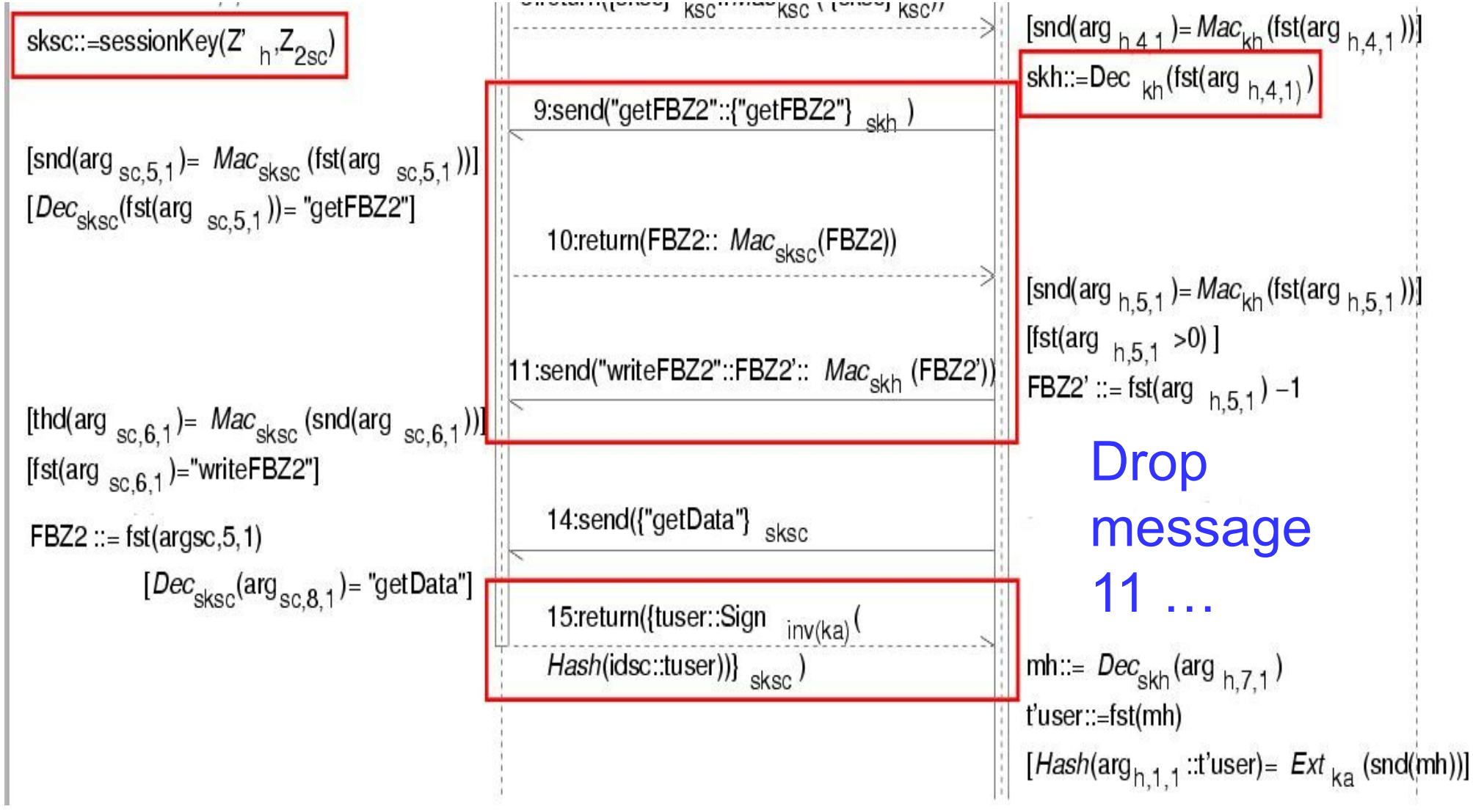
Authent. Protocol Pt. 2: Problem ?



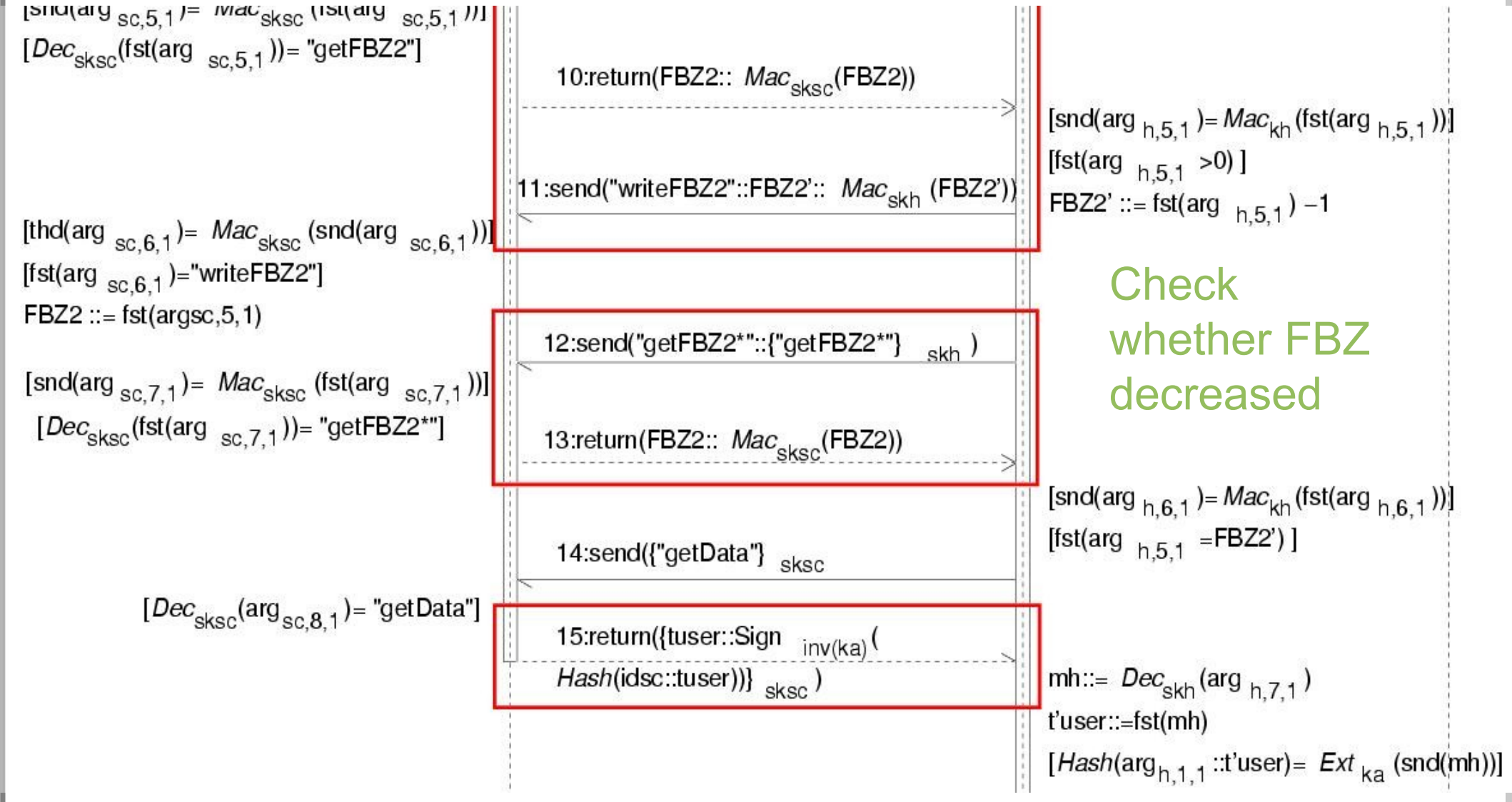
Decrease misuse counter

Message order ?

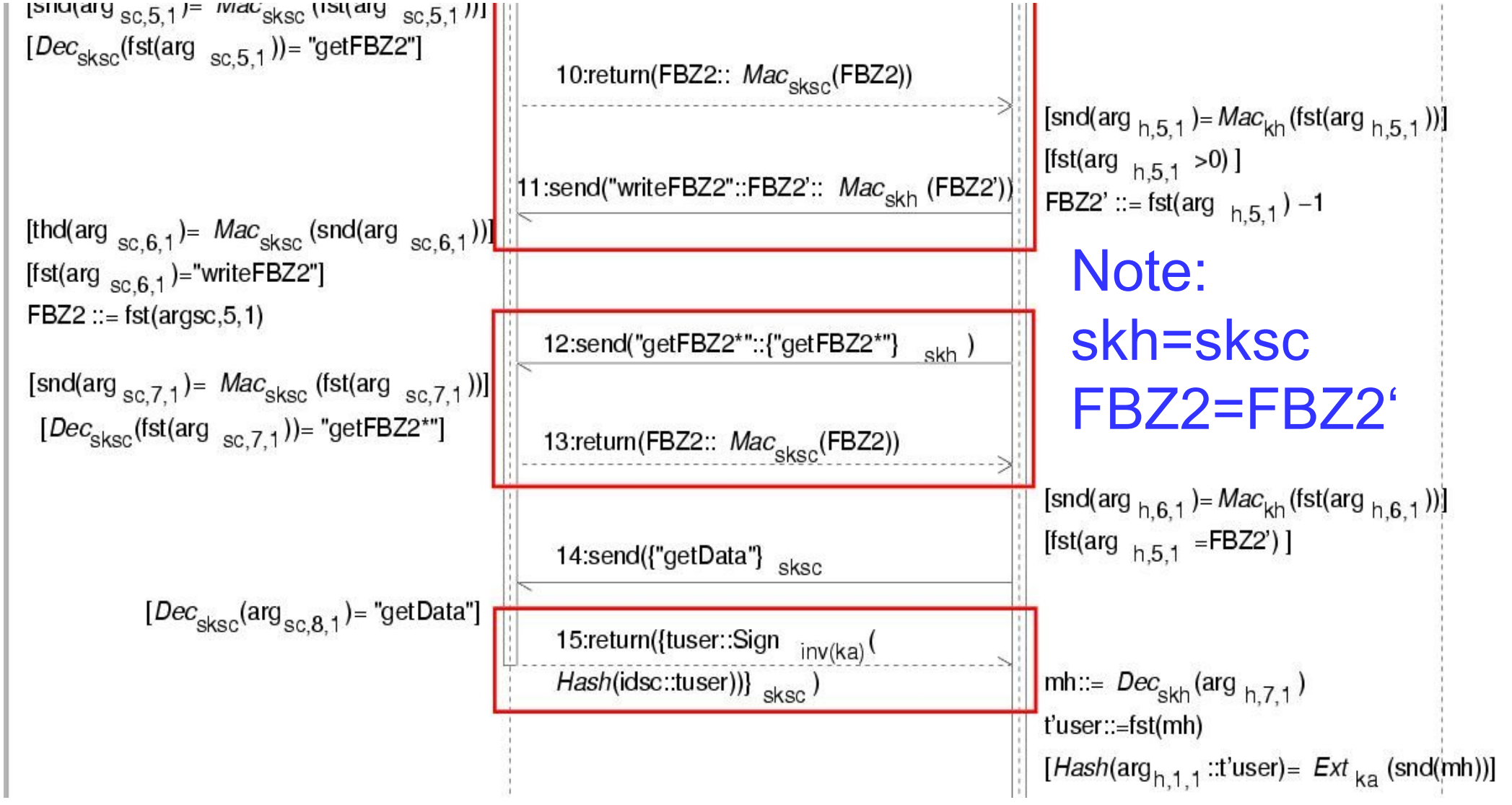
Authent. Protocol Pt. 2: Problem.



Authent. Protocol Pt. 2: Improvement

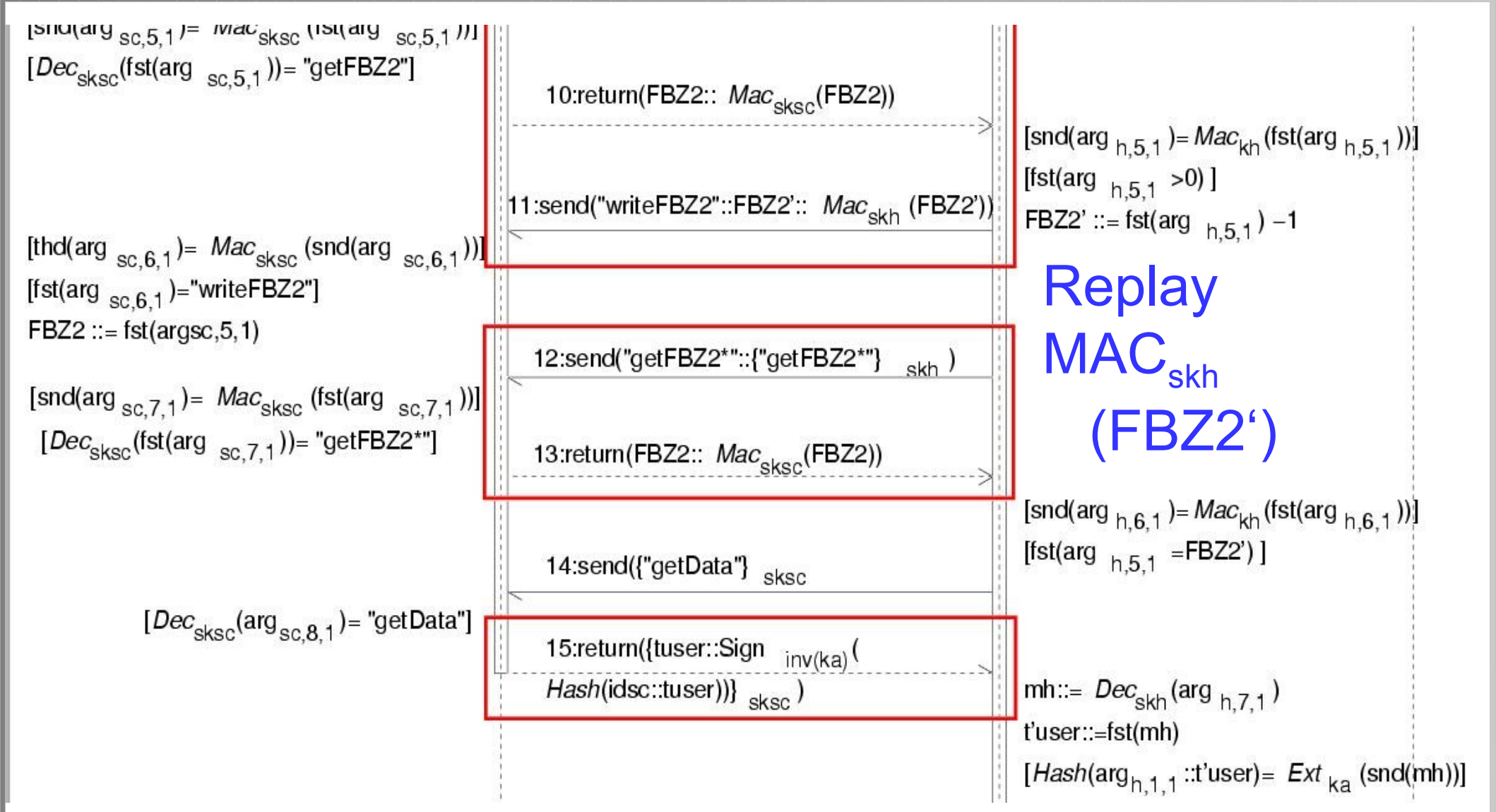


Authent. Prot. Pt. 2: Improvement ?

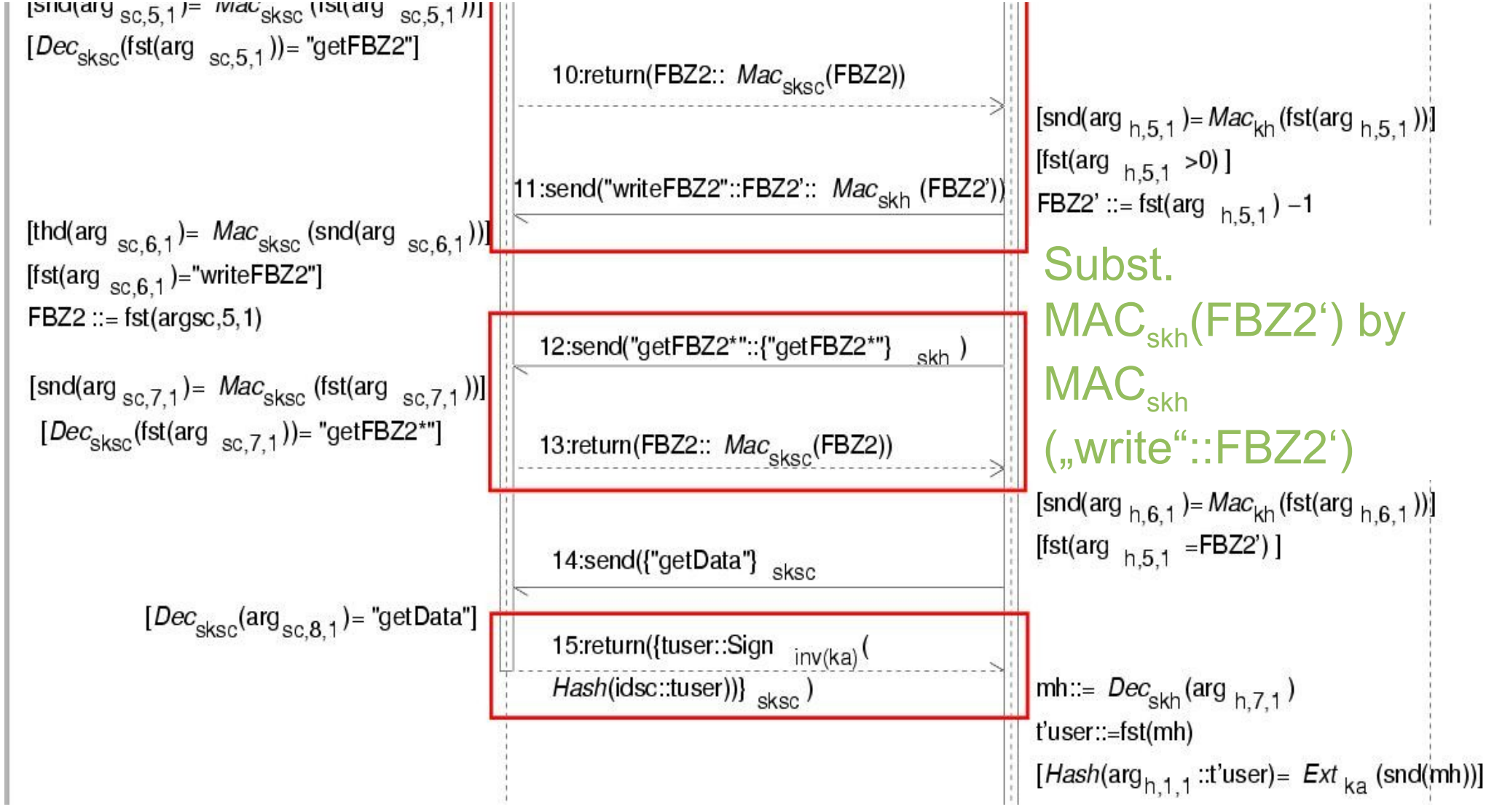


Note:
skh=sksc
FBZ2=FBZ2'

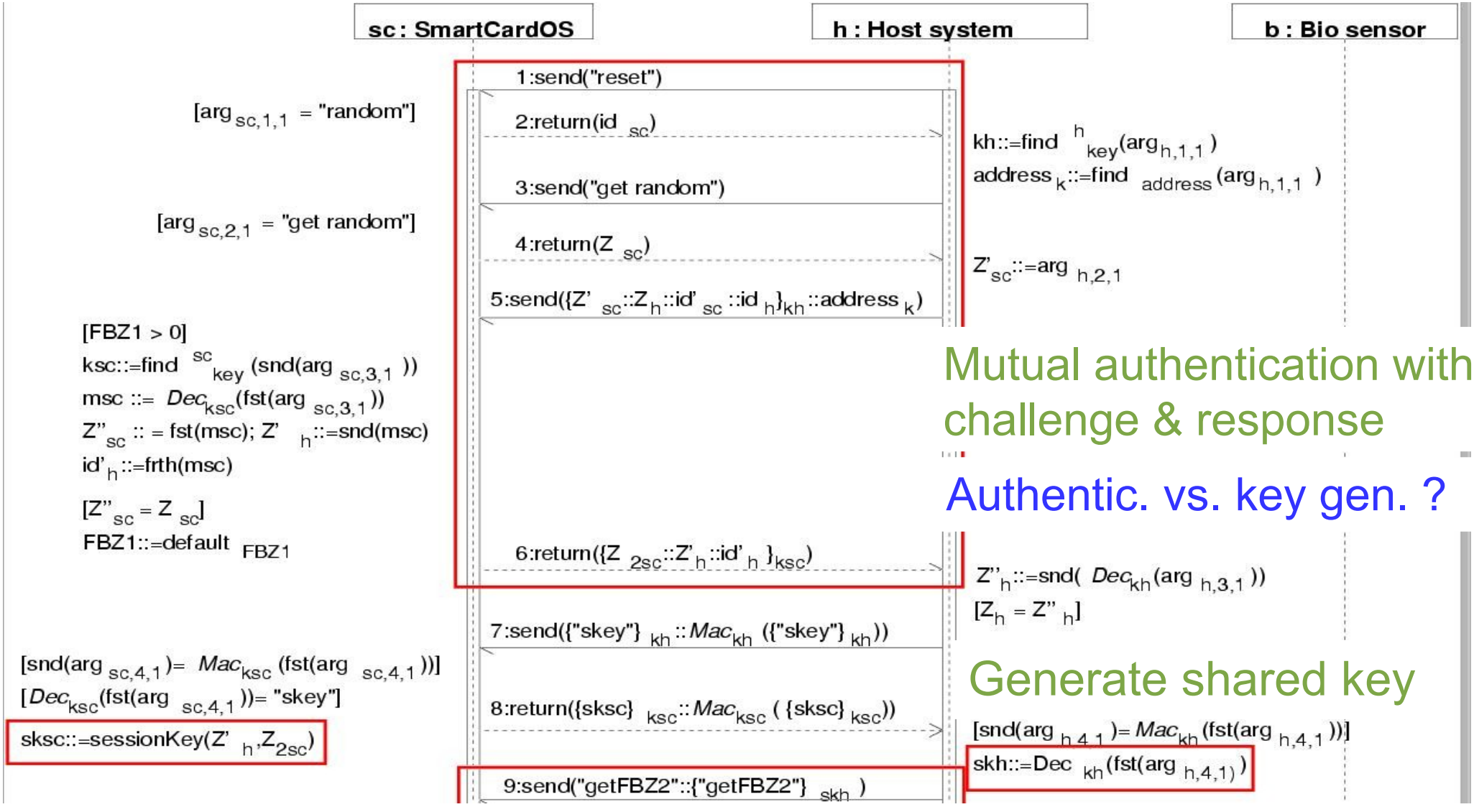
Authent. Prot. Pt. 2: Problem



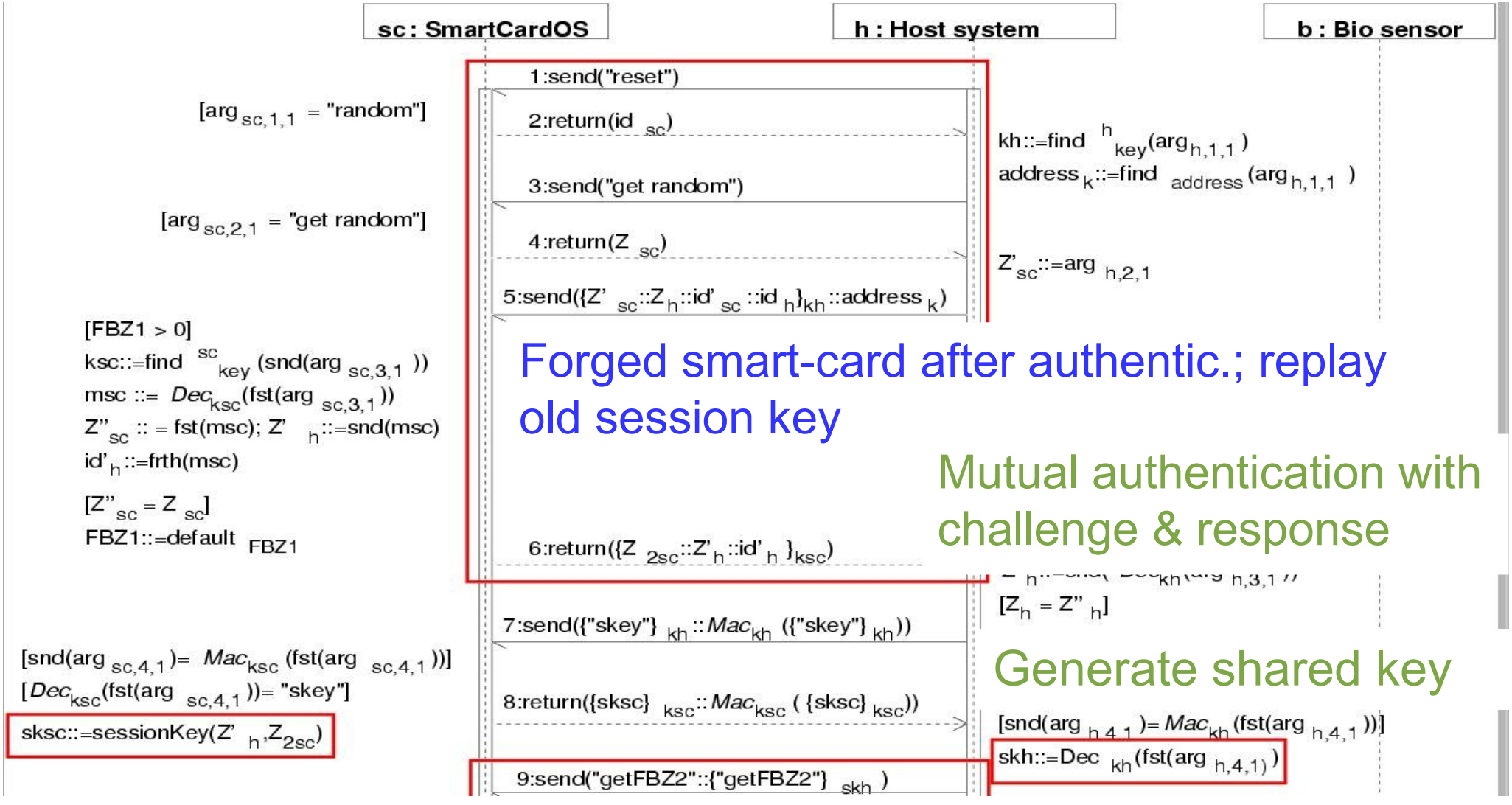
Authent. Prot. Pt. 2: Improvement (?)



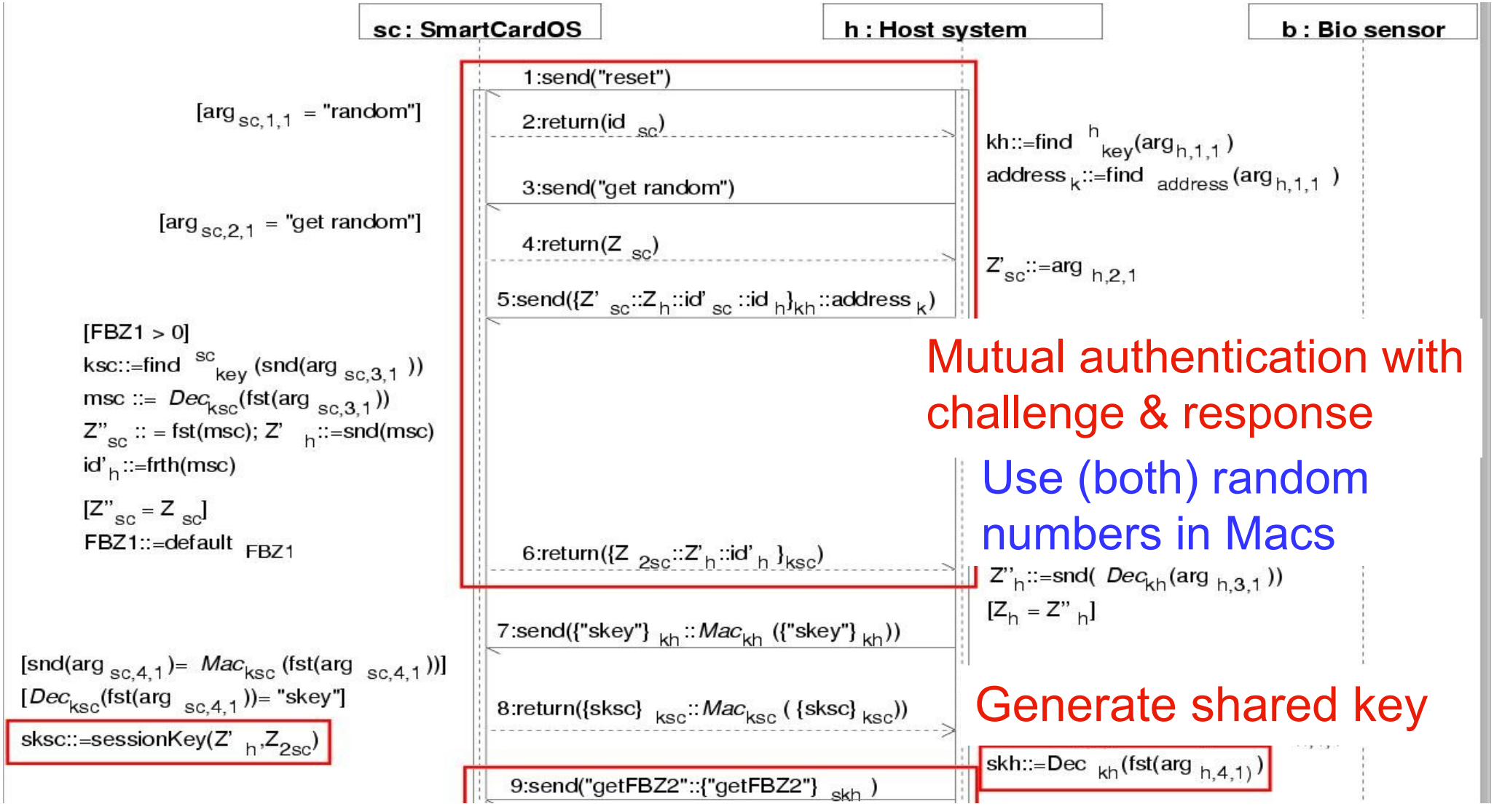
Authentic. Protocol Part 1: Problem ?



Authentic. Protocol Part 1: Problem.



Authentic. Protocol Part 1: Improvement (?)



Aufgabe 8.1

Diese Aufgabe bezieht sich auf folgende Ausarbeitung (insbes. Abb. 1 auf S. 7): http://ls14-www.cs.tu-dortmund.de/main2/jj/umlsectool/applications/biometrics/Documentation/Ausarbeitung_Biosys.pdf

- Karte-Host Authent. (Nachrichten 3-8):

a) Aendere die Spezifikation dieses Teiles durch Entfernen der Zufallszahl r_host . [2 P.]

b) Welcher Angriff ist nun moeglich ? (Angriffablauf als Pfeildiagramm) [3 P.]

c) Wie a), nur r_icc statt h_host entfernen. [2 P.]

d) Angriff (wie b) ? [3 P.]

Aufgabe 8.1

- Biodaten-Austausch (Nachrichten 11-19):

e) Welcher Angriff ist moeglich, wenn der Angreifer den symmetrischen Schluessel K_C=K_H kennt ?

(Pfeildiagramm) [4 P.]

f) Welcher Angriff ist moeglich, wenn der Angreifer den Sitzungs-Schluessel sk kennt ? (Pfeildiagramm) [4 P.]

Bonus: Finde einen weiteren Angriff gegen das Protokoll. [+ 20 P. ☺]