LEHRSTUHL 14
SOFTWARE ENGINEERING

Willkommen zur Vorlesung

Methodische Grundlagen des Software-Engineering

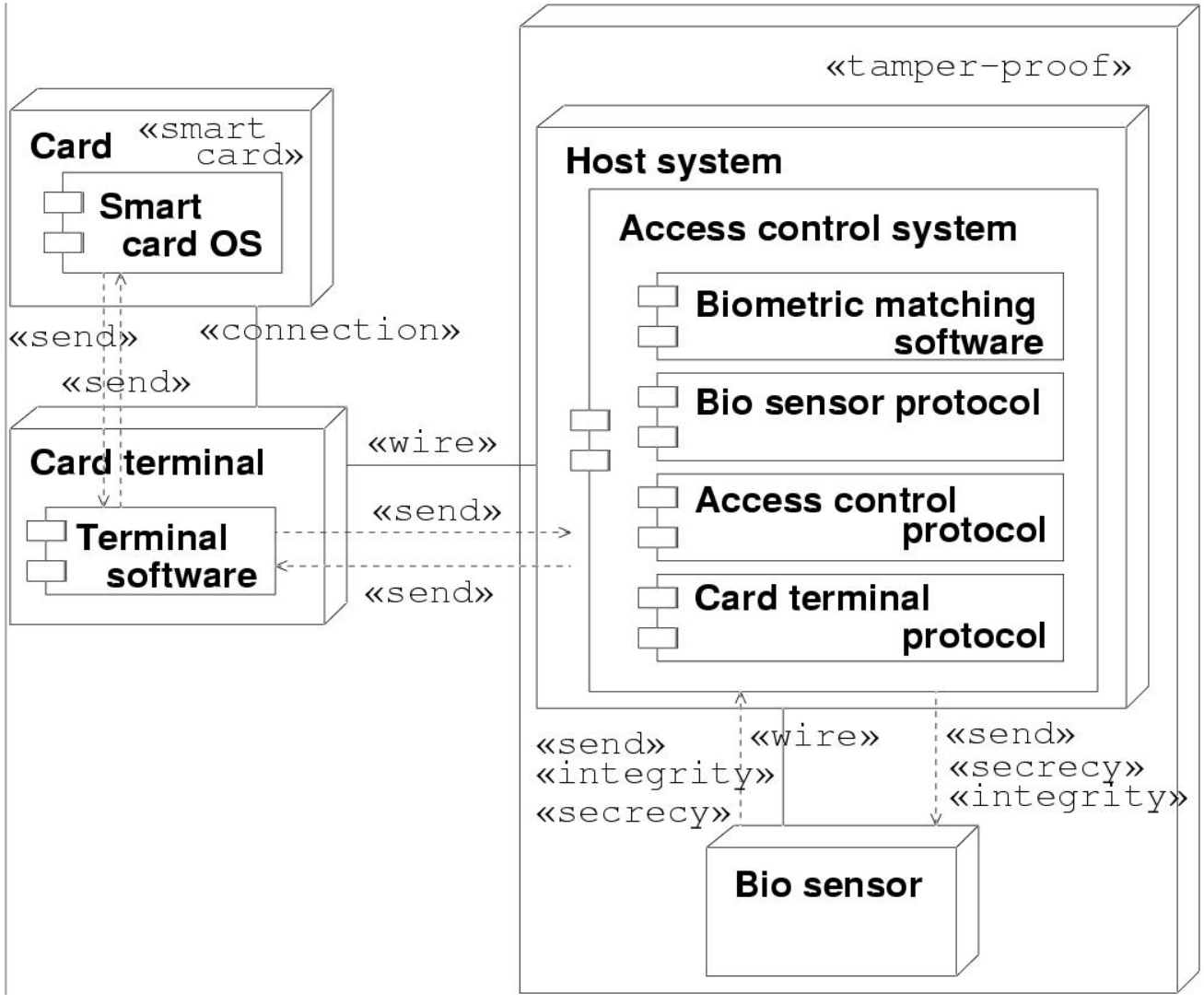
im Sommersemester 2011

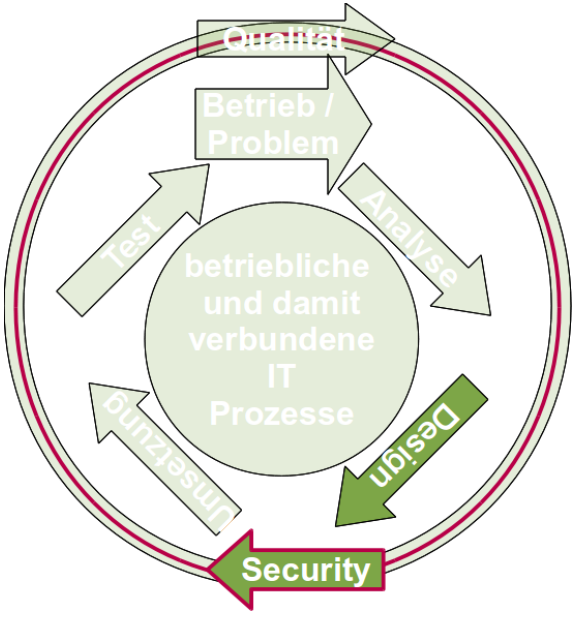
Prof. Dr. Jan Jürjens

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

24. Biometrische Authentisierung: Sicherheitsanalyse

Einordnung
Biometrische Authentifizierung mit UMLsec

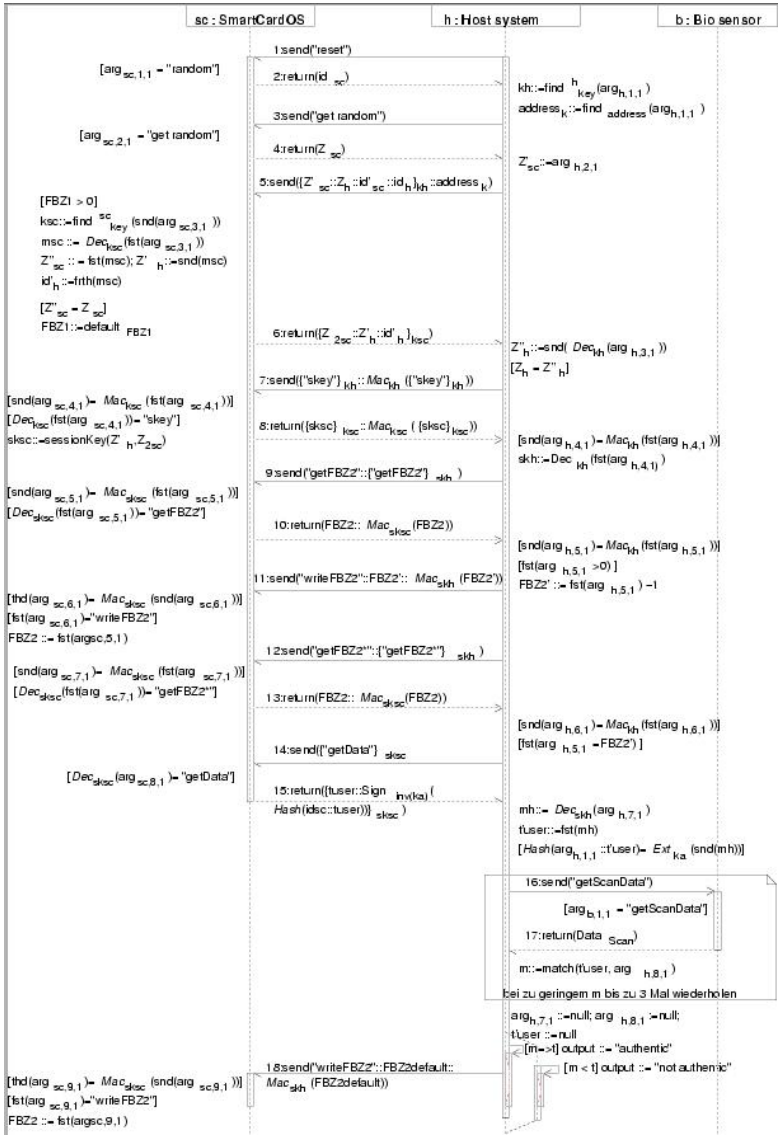




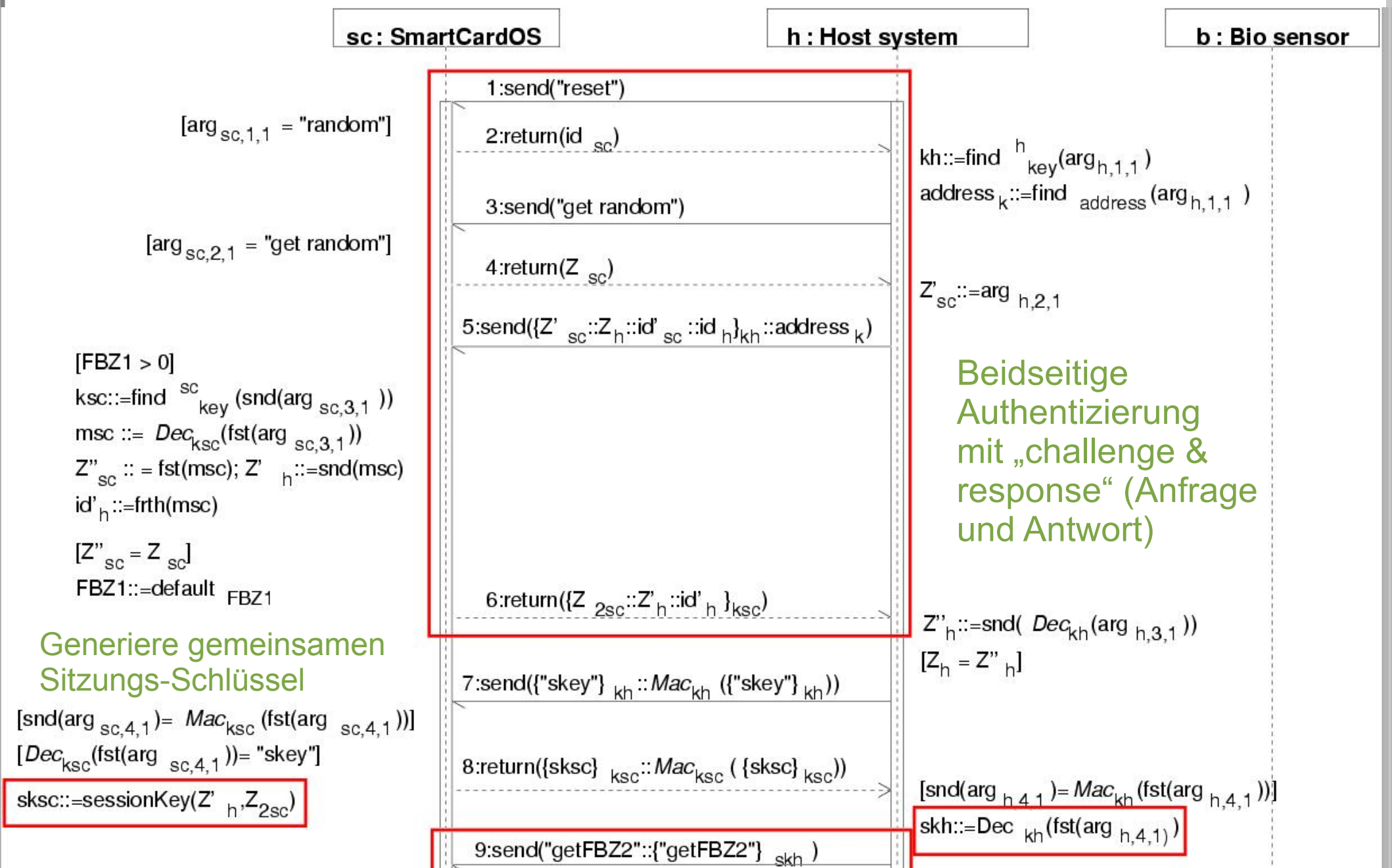
Einordnung
Einführung UML / UMLsec

- Business Prozesse
- Qualitätsmanagement
- Testen
- Sicherheit
- Sicheres Software Design
  - Einführung UML / UMLsec
  - Kryptographische Protokolle
  - Biometrische Authentifizierung
  - Elektronische Geldbörse
  - Weitere Anwendungsbeispiele

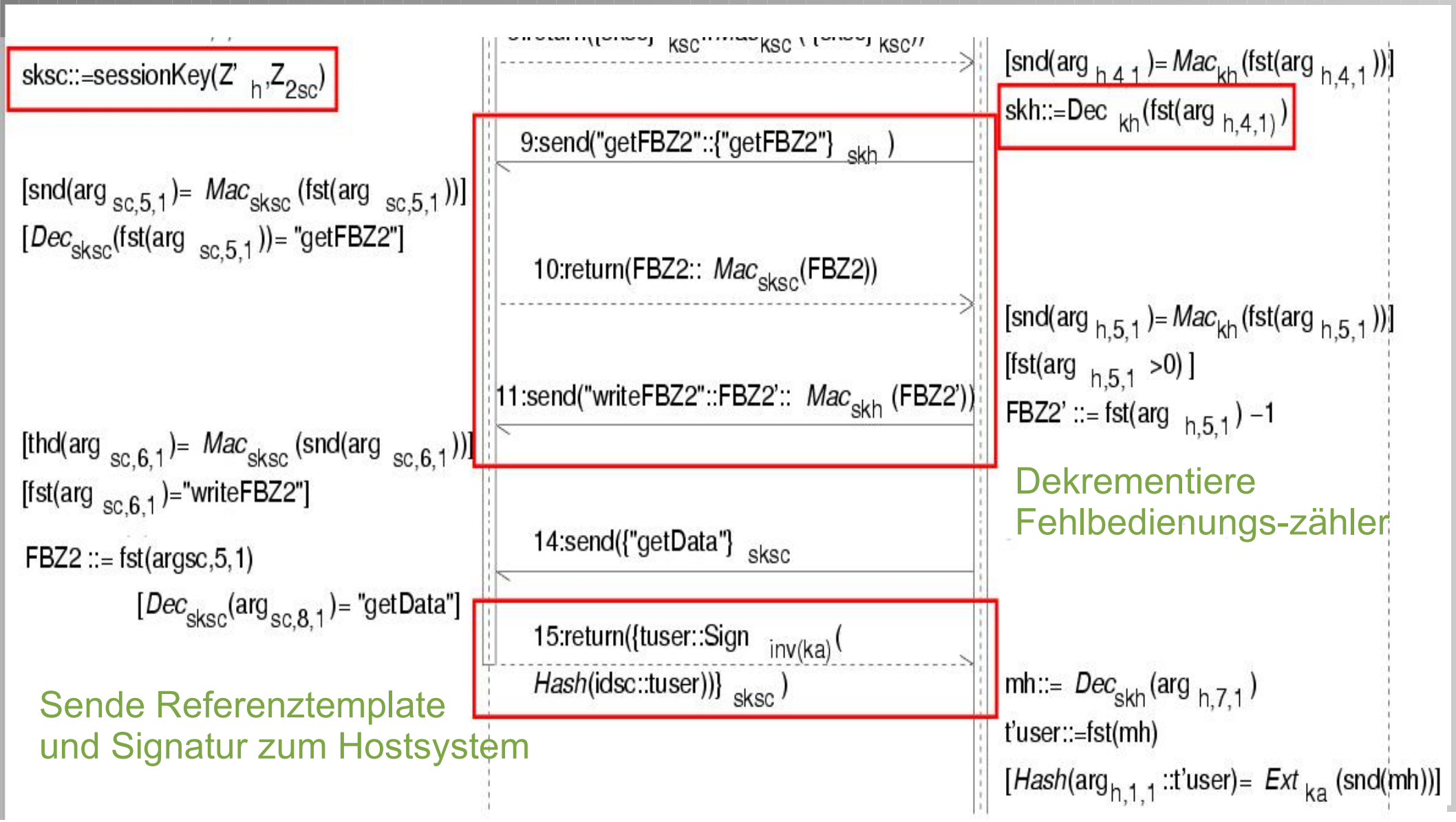
Protokoll


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::idh}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)

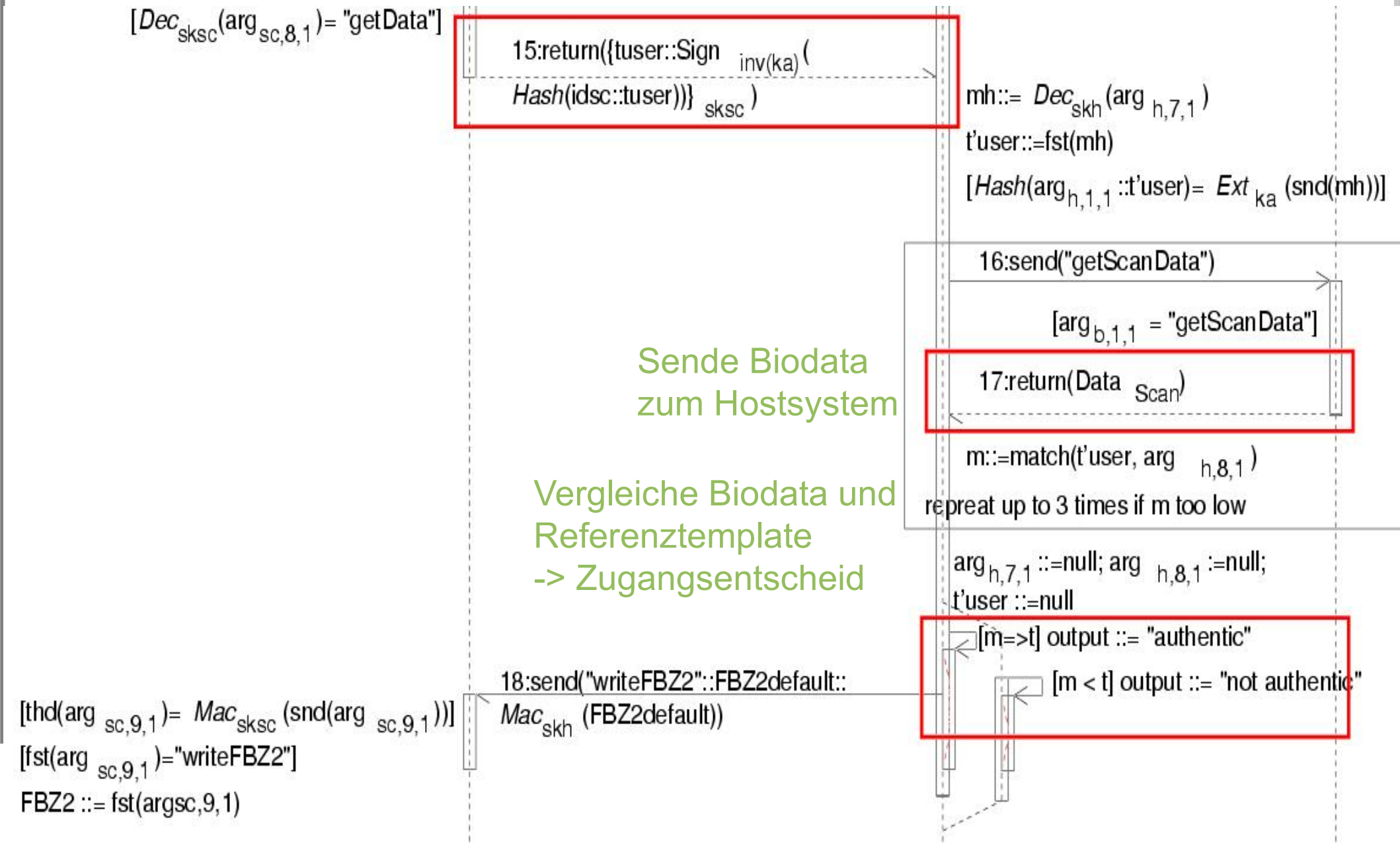
Authentifizierungsprotokoll Teil 1



Authentifizierungsprotokoll Teil 2



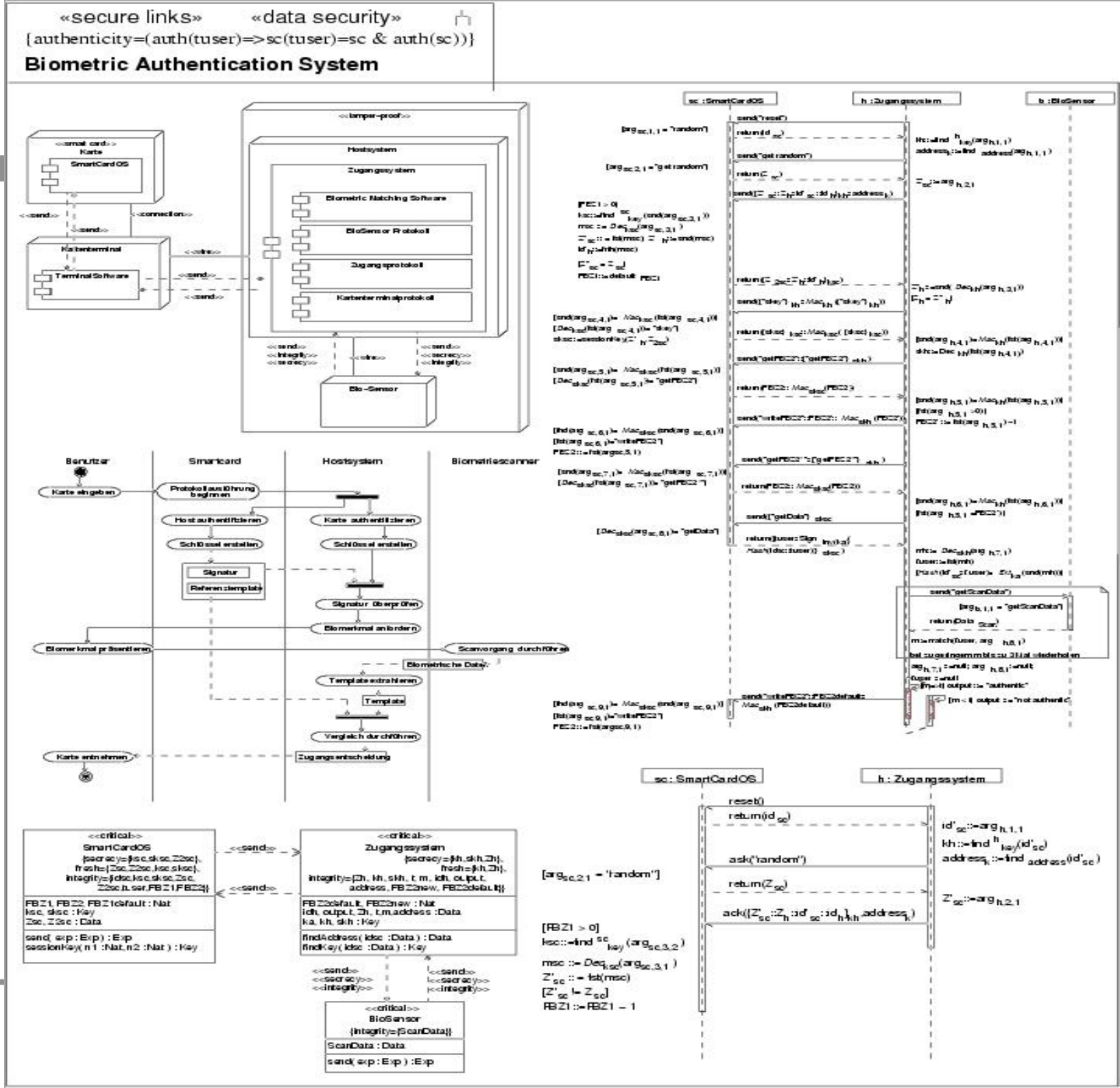
Authentifizierungsprotokoll Teil 3



Sende Biodata zum Hostsystem

Vergleiche Biodata und Referenztemplate -> Zugangsentscheid

Übersicht



technische universität dortmund

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 legitimer Smartcard
- Administrator: stellt Smartcards aus
- System: durch das biometrische System geschützter Bereich

Bedrohungen

- Benutzer: Angreifer richtet unter der Identität des Benutzers Schaden an.
- Administrator: Wird 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 Berechtigung ein biometrisches Template

Sicherheitsziele

- Benutzer: Nur er darf Zugang erhalten (und zwar nur unter seiner eigenen Identität).
- 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: $\mathcal{K}_A \cap \{\mathsf{t_{user}}\} = \emptyset.$

Übersetzung in Logik erster Stufe II

Hier: Korrekte Nachrichtenreihenfolge nicht durch Smartcard erwzungen(!).

Daher leichte Variation der vorigen Formel (beachte Klammerung !):

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

gefolgt von $TR2$ gibt Prädikat

$PRED(TR1)=$

$$\forall msg_in.\ [knows(msg_in) \wedge cond(msg_in) \Rightarrow knows(msg_out)] \wedge PRED(TR2)$$

Authentifizierungsprotokoll Teil 2: Problem ?

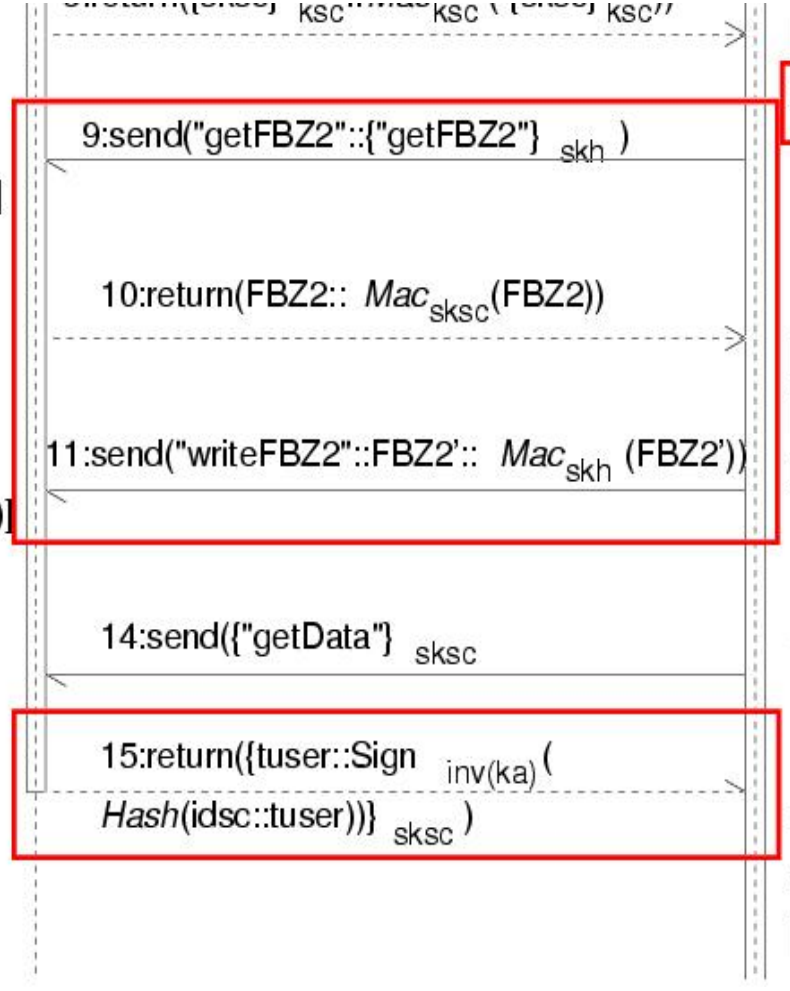
sksc::=sessionKey(Z' $_h$,Z_{2sc})

[snd(arg $_{sc,5,1}$)= Mac_{sksc} (fst(arg $_{sc,5,1}$))]
[Dec_{sksc}(fst(arg $_{sc,5,1}$))= "getFBZ2"]

[thd(arg $_{sc,6,1}$)= Mac_{sksc} (snd(arg $_{sc,6,1}$))]
[fst(arg $_{sc,6,1}$)="writeFBZ2"]

FBZ2 ::= fst(argsc,5,1)

[Dec_{sksc}(arg $_{sc,8,1}$)= "getData"]



[snd(arg $_{h,4,1}$)= Mac_{kh}(fst(arg $_{h,4,1}$))]
skh::=Dec $_{kh}$(fst(arg $_{h,4,1)}$)

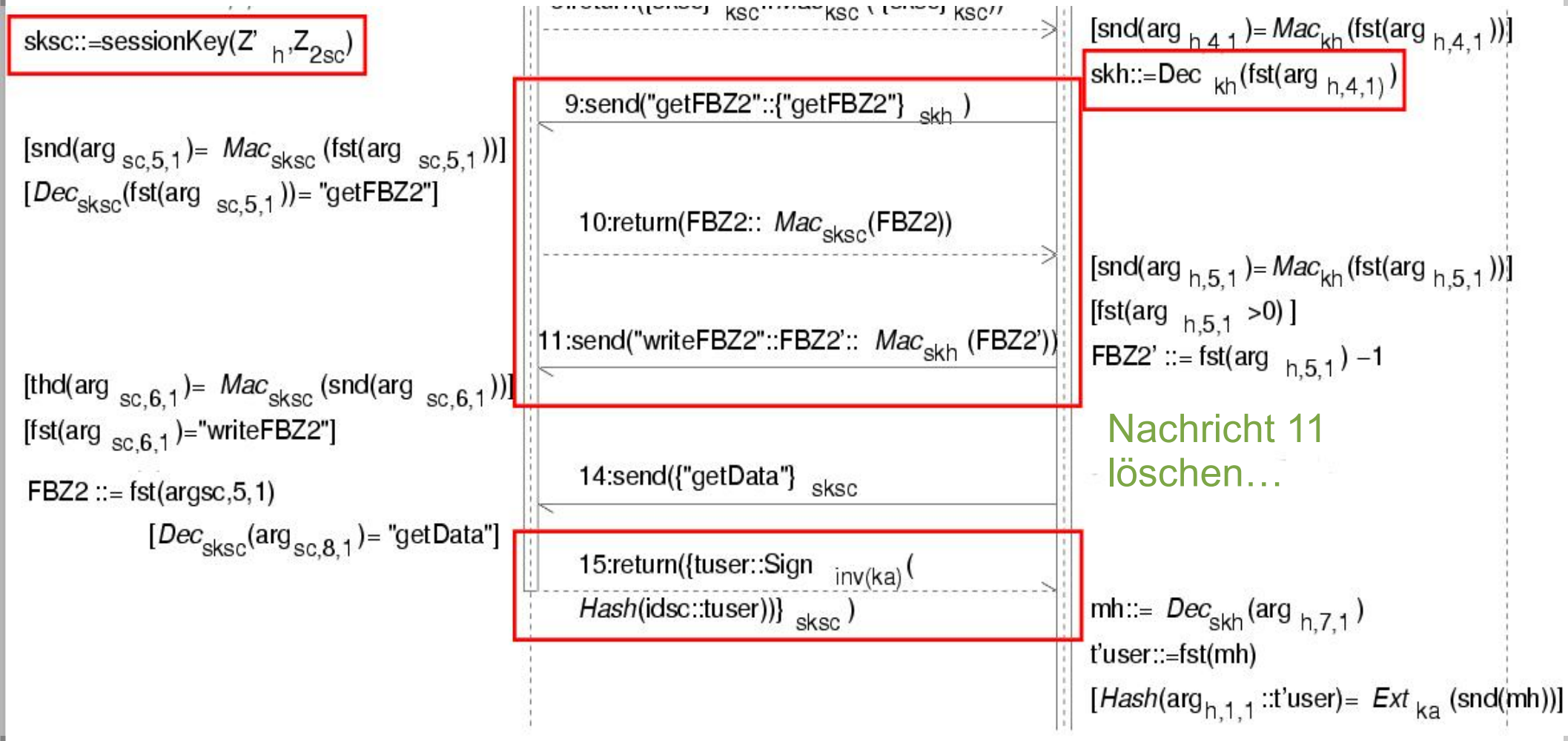
[snd(arg $_{h,5,1}$)= Mac_{kh}(fst(arg $_{h,5,1}$))]
[fst(arg $_{h,5,1}$ >0)]
FBZ2' ::= fst(arg $_{h,5,1}$) −1

Dekrementiere Fehlbedienungszähler

Nachrichtenreihenfolge?

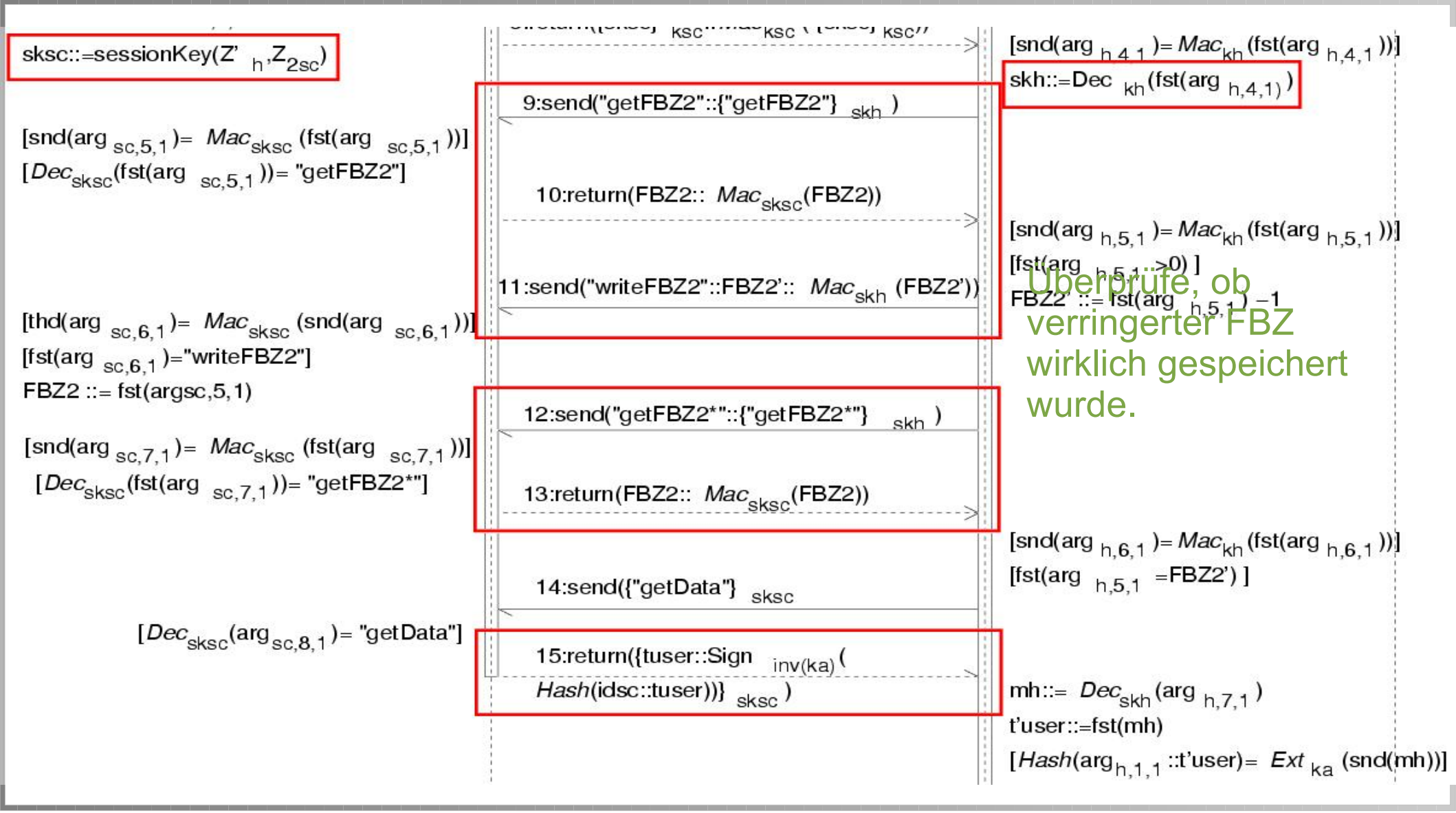
mh::= Dec_{skh}(arg $_{h,7,1}$)
t'user::=fst(mh)
[$Hash$(arg$_{h,1,1}$ ::t'user)= Ext_{ka} (snd(mh))]

Authentifizierungsprotokoll Teil 2: Problem.



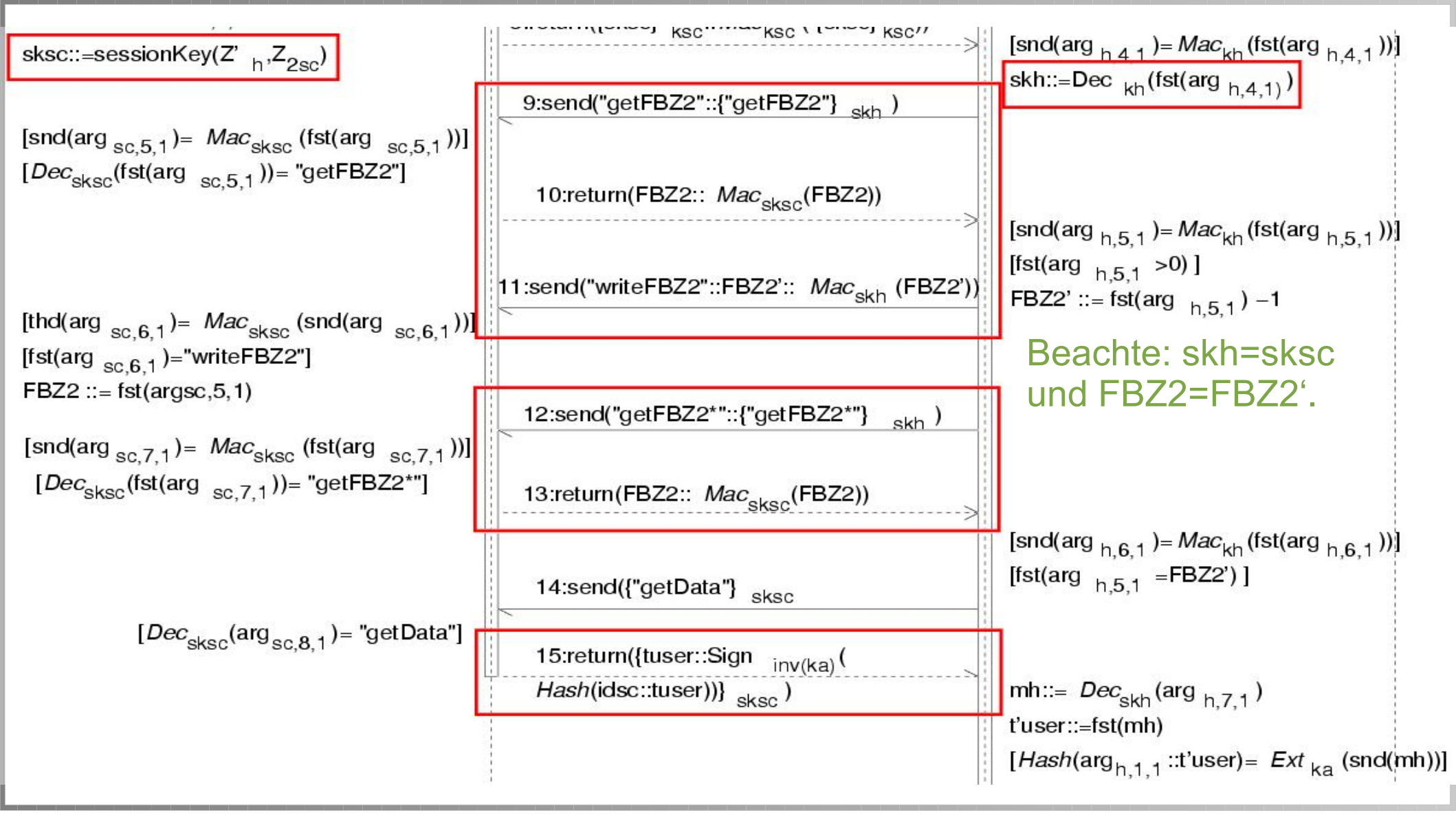
Nachricht 11 löschen…

Authentifizierungsprotokoll Teil 2: Verbesserung



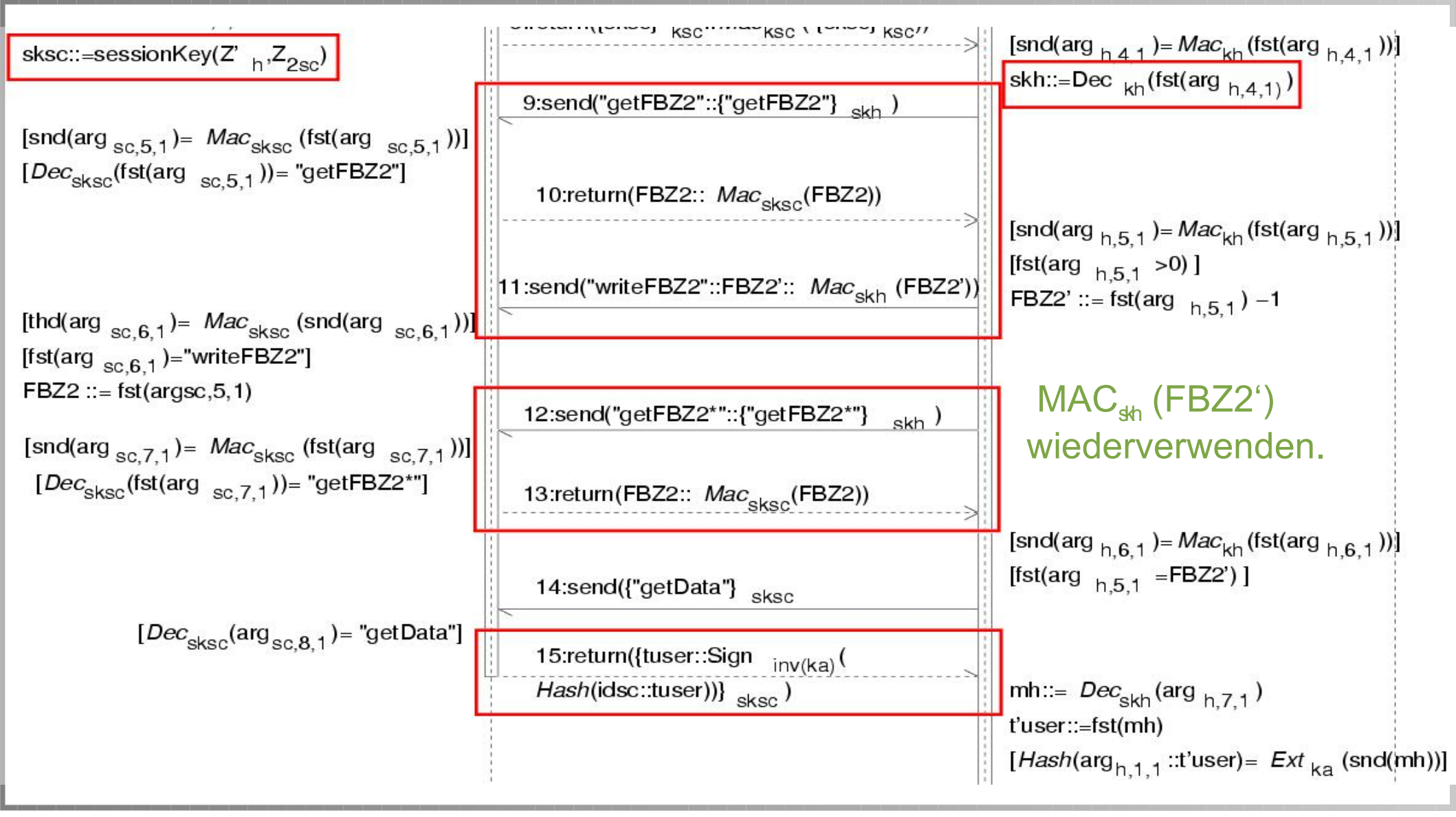
Überprüfe, ob verringerter FBZ wirklich gespeichert wurde.

Authentifizierungsprotokoll Teil 2: Verbesserung?



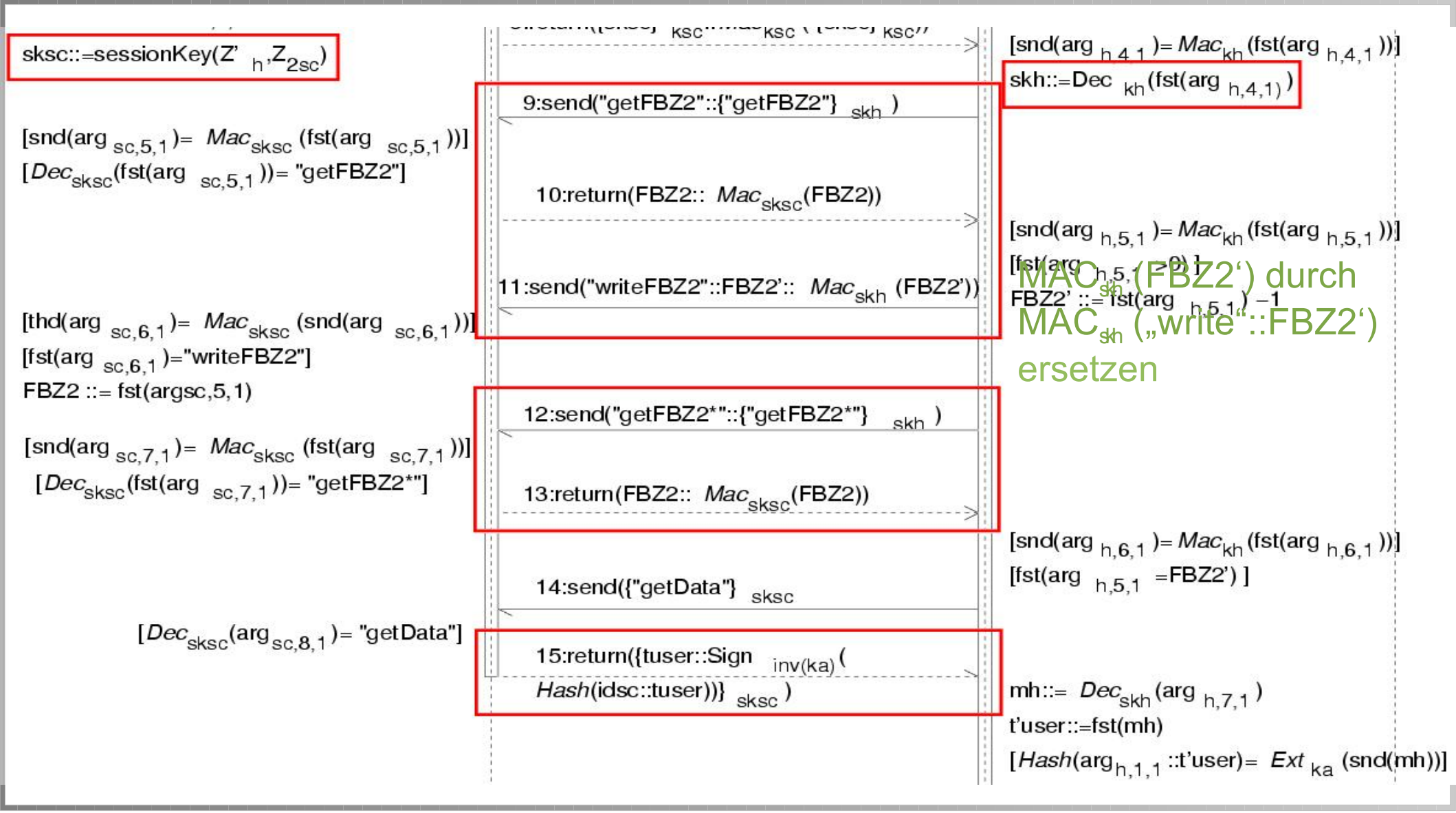
Beachte: skh=sksc und FBZ2=FBZ2'.

Authentifizierungsprotokoll Teil 2: Problem

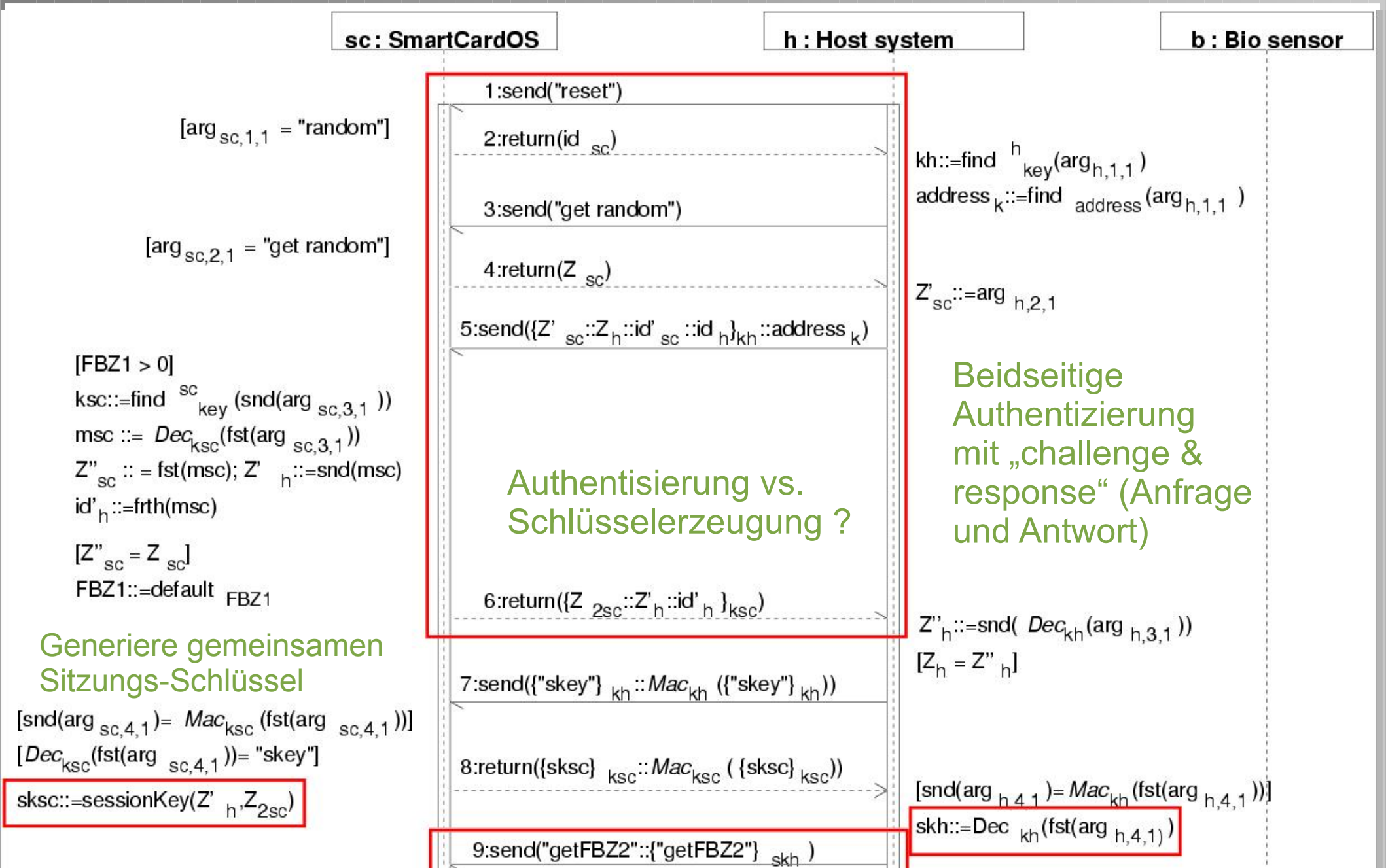


MAC_{skh} (FBZ2') wiederverwenden.

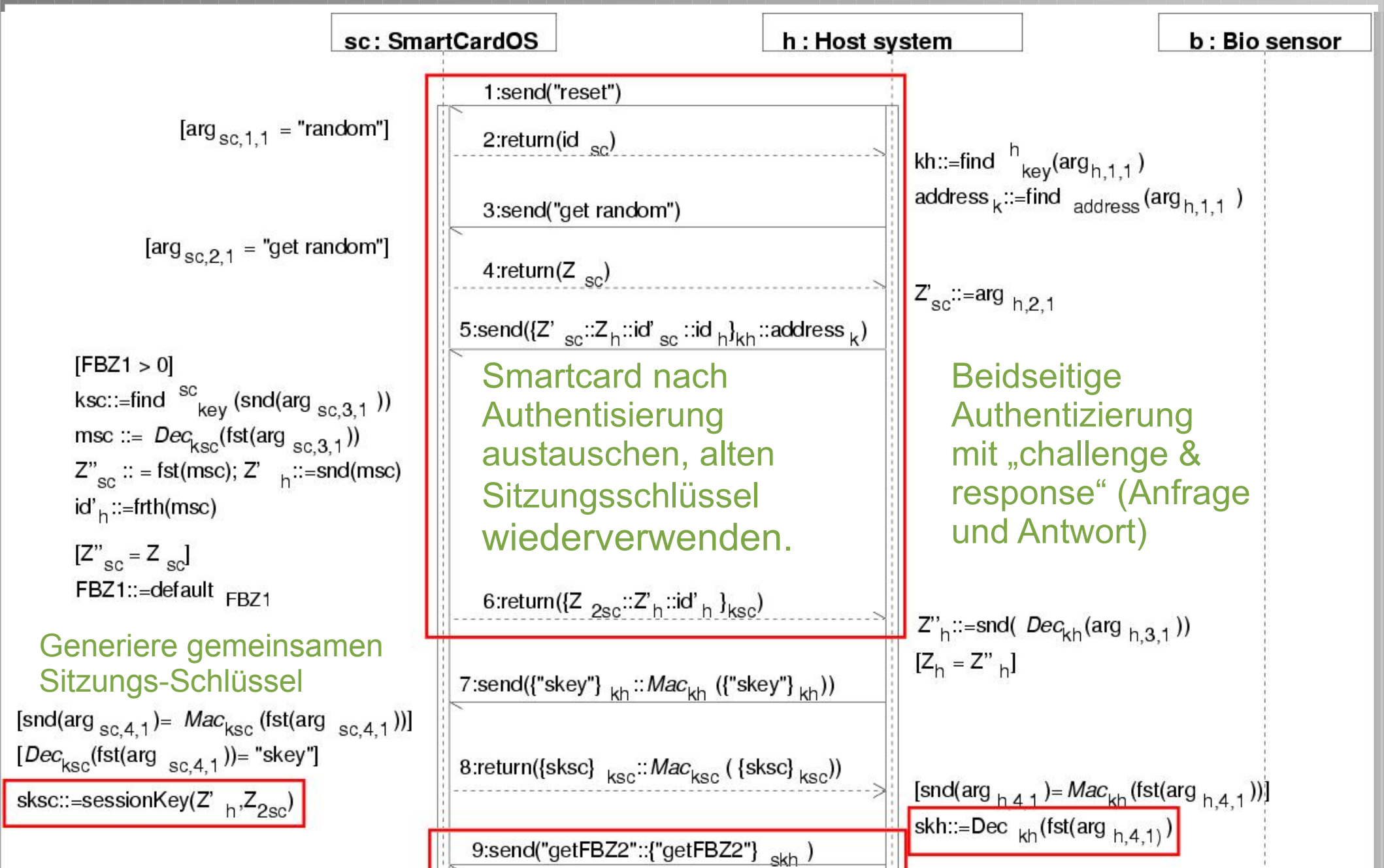
Authentifizierungsprotokoll Teil 2: Verbesserung (?)



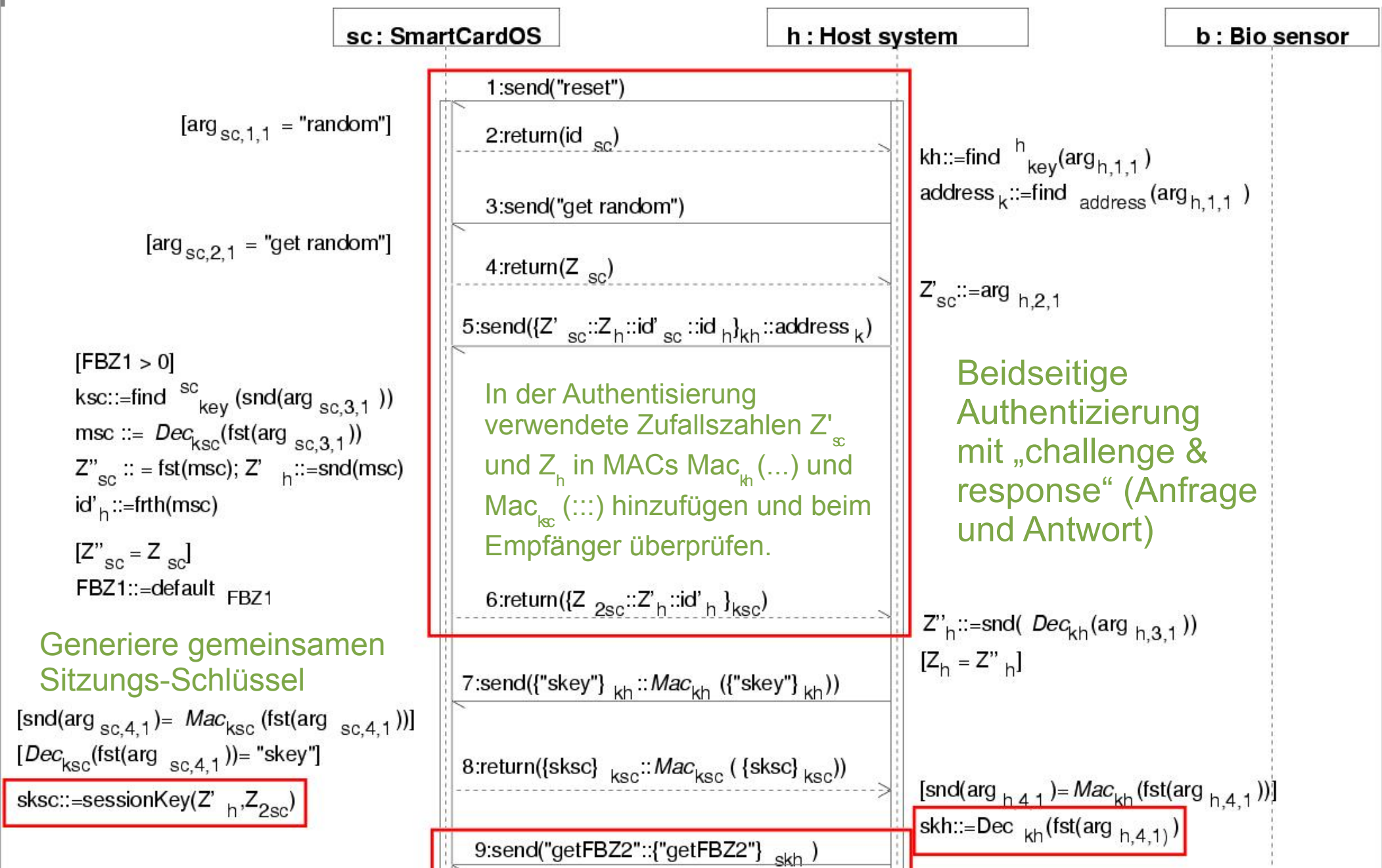
Authentifizierungsprotokoll Teil 1: Problem ?



Authentifizierungsprotokoll Teil 1: Problem



Authentifizierungsprotokoll Teil 1: Verbesserung



Zusammenfassung

- Biometrieprotokoll
- Schwachstellen-Analyse
- Verbesserungen