Bewijzen van authenticatieprotocollen in een trace model

Palm, Niek P.

Award date:
2004

TECHNISCHE UNIVERSITEIT EINDHOVEN

Faculteit Wiskunde en Informatica

AFSTUDEERVERSLAG

Bewijzen van authenticatieprotocollen in een trace model

door
N.P. Palm

Afstudeerdocent: dr. S. Mauw
Afstudeerbegeleider: ir. C.J.F. Cremers

augustus 2004

Voorwoord

Dit verslag is een onderdeel van mijn afstudeerproject waarmee ik mijn studie Technische Informatica aan de Technische Universiteit Eindhoven afrondt.

Het afstudeerproject heeft plaats gevonden aan de Technische Universiteit Eindhoven bij de vakgroep Formele Methoden van de faculteit Wiskunde en Informatica. In dit project heb ik mij bezig gehouden met het formeel verifiëren van veiligheidseisen in security-protocollen.

In de eerste plaats wil ik mijn afstudeerdocent dr. S. Mauw bedanken voor zijn begeleiding van dit project. Ik wil ir. C.J.F. Cremers bedanken voor zijn adviezen, inzichten en begeleiding die hij mij gegeven heeft gedurende het afgelopen jaar. Tevens wil ik graag dr. ir. T. Verhoeff en dr. E.P. de Vink bedanken voor het plaatsnemen in mijn afstudeercommissie.

Tenslotte wil ik mijn familie en vrienden bedanken voor hun steun zowel tijdens het afstudeerproject als tijdens mijn studie.

Niek Palm

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

Hoofdstuk 1

Inleiding

Dit hoofdstuk beschrijft op welke wijze dit veslag is opgebouwd. Voor elk hoofdstuk wordt aangegeven wat in het betreffende hoofdstuk behandeld wordt.

In hoofdstuk 2 beschrijven we de opdracht en het kader waar binnen deze opdracht gesteld is. Tevens geven we doelen aan die we voor dit project gesteld hebben.

Hoofdstuk 3 beschrijft uitgebreid het model waar we gebruik van maken om security-protocollen te beschrijven en te bewijzen. Het model dat we in dit hoofdstuk beschrijven is volledig gebaseerd op het model dat in "Operational semantics of security protocols" [5] beschreven wordt.

In hoofdstuk 4 wordt in detail in gegaan op het bewijzen van non-injectieve synchronisatie voor security-protocollen. De eerste paragraaf bevat een analyse van de protocollen die in SPORE [14] opgenomen zijn. Hier wordt overzichtelijk aangegeven welke protocollen wel en welke niet in aanmerkingen komen voor non-injectieve synchronisatie. In paragraaf 4.2 geven we aan op welke manier de bewijzen opgebouwd zijn die later volgen en welke lemma's we hiervoor gebruiken. De paragrafen 4.3 t/m 4.8 behandelen elk het bewijs voor non-injectieve synchronisatie van een protocol uit SPORE. Tot slot geven we in de laatste paragraaf van dit hoofdstuk een bewijs voor een protocol dat oorspronkelijk niet aan non-injectieve synchronisatie voldeed. Na een kleine wijziging in dit protocol aangebracht te hebben, was het echter wel mogelijk om non-injectieve synchronisatie af te leiden.

Hoofdstuk 5 bevat een korte evaluatie van de opdracht gevolgd door enkele conclusies. We sluiten het hoofdstuk af met enkele aanbevelingen voor toekomstig onderzoek op dit gebied.

In bijlage A hebben we letterlijk de protocolbeschrijvingen uit SPORE opgenomen van de protocollen die we in hoofdstuk 4 bewijzen. In bijlage B is een lijst van alle definities opgenomen, die in hoofdstuk 3 reeds besproken zijn.

Hoofdstuk 2

Probleemstelling

Dit hoofdstuk beschrijft de opdracht en het kader waarbinnen deze opdracht gesteld is. Tevens geven we enkele doelen aan die voor dit project gesteld zijn.

2.1 Kader

Security protocollen zijn bekend om hun fouten, deze fouten zijn helaas moeilijk te vinden. Echter een fout in een werkend protocol zal snel uitgebuit worden en kan aanleiding geven tot grote schade. Het is dan ook essentieel dat security protocollen op een formele manier worden geverifiëerd. In de praktijk is het mogelijk om security protocollen partieel te verifiëren met behulp van een modelchecker, dit geeft geen garantie dat het protocol correct is.

In "Defining Authentication in a Trace Model"[6] wordt een trace model beschreven voor security protocollen. In hetzelfde artikel worden verschillende eigenschappen voor deze protocollen formeel gedefinieerd. Op basis van deze definities zou formele verificatie van security protocollen mogelijk moeten zijn.

2.2 Opdracht

Ga voor de protocollen uit SPORE [14] na of er non-injectieve synchronisatie geclaimd mag worden en verifieer de protocollen formeel waarvoor non-injectieve synchronisatie geldt. Aan de hand van deze bewijzen zal getracht worden een algemeen patroon te ontdekken.

Voor deze opdracht zijn de volgende doelen te onderscheiden.

- Ervaring opdoen met handmatig bewijzen van correctheid in security protocollen.
- Evalueren van de semantiek, i.h.b. de geschiktheid van correctheidsbewijzen.

- Ontwikkelen van bewijsprincipes zoals axioma's, afleidingregels, reductieregels, patronen, metastellingen etc.
- Het correct bewijzen van geschikte protocollen uit SPORE.
- Formuleren en eventueel implementeren van de bewijsprincipes.
- Validatie d.m.v. verificatie van een serieus protocol.

Hoofdstuk 3

Het model

Dit hoofdstuk beschrijft uitgebreid het model en de semantiek die de basis vormen voor de bewijzen van authenticatieprotocollen in hoofdstuk 4. De operationele semantiek die dit model vastlegt, wordt uitgebreid beschreven in "Operational semantics of security protocols" [5].

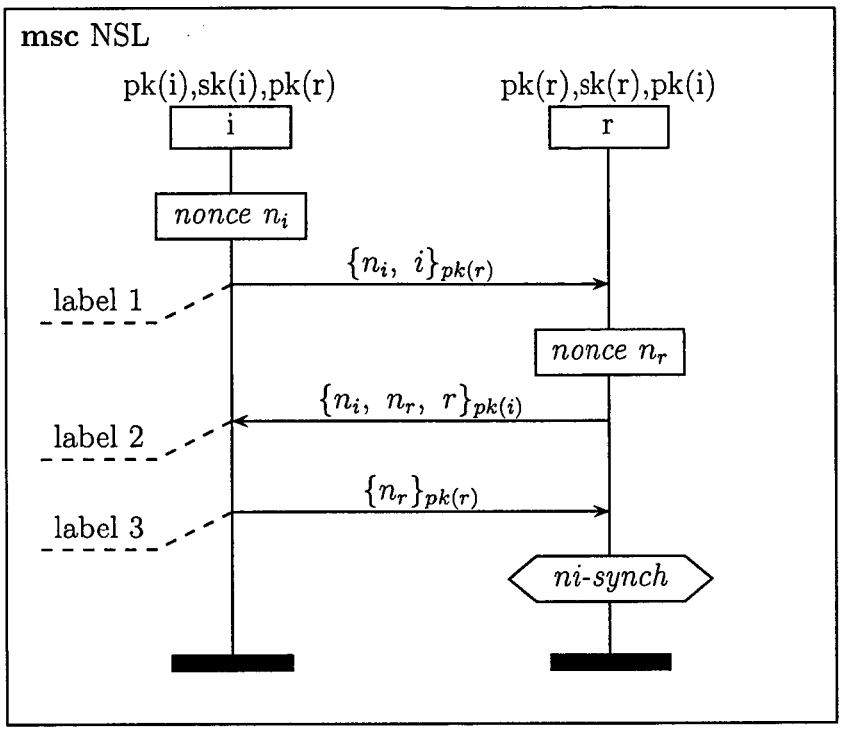
3.1 Inleiding

Een protocol is een abstracte beschrijving van de voorgeschreven manier waarop twee of meer partijen met elkaar dienen te communiceren. Deze partijen noemen we de rollen van een protocol. Voor een security-protocol onderscheiden we extra eisen zoals het geheimhouding van bepaalde informatie of authenticatie.

We zullen de uitleg van het model af en toe verduidelijken met een voorbeeld. We maken gebruik van één van de bekendste protocollen namelijk het Needham Schroeder Lowe [9], [14] protocol. Aan dit protocol ligt het Needham Schroeder protocol [13] ten grondslag, ontwikkeld door Roger Needham en Michael Schroeder in 1978. Het Needham Schroeder protocol bewerkstelligt authenticatie tussen de beide partijen. In 1989 wordt het Needham Schroeder protocol correct bewezen met behulp van de BAN logica [1]. Echter in 1995 wordt door Gravin Lowe een aanval [9] gevonden op dit protocol. Lowe geeft een oplossing voor deze aanval, hetgeen resulteert in het Needham Schroeder Lowe protocol. In Figuur 3.1 wordt het Needham Schroeder Lowe protocol op een schematische manier beschreven. We zullen het Needham Schroeder Lowe protocol kortweg aanduiden als het NSL protocol. Later zullen we voor dit protocol een bewijs geven.

Voor de schematische weergave maken we gebruik van een *Message Sequence Diagram* (MSC) [12]. Met behulp van deze diagrammen zijn we goed in staat om een protocol te beschrijven. De stokken stellen de rollen in een protocol voor waarbij de tijd gemodelleerd wordt op de verticale as. De bovenkant kan gezien worden als het moment waarop het protocol gestart wordt en de onderkant van een stok is het moment dat het protocol

voltooid wordt. Boven de stok staat aangegeven wat de initiële kennis van de rol is. In de rechthoek bovenaan de stok staat de naam van de rol aangegeven. Met behulp van de pijlen geven we de berichten aan die de rollen onderling uitwisselen. De communicatie loopt in dezelfde richting als de pijlen. Boven de pijl staat het patroon van het bericht dat uitgewisseld wordt. Met de rechthoeken binnen de stok geven we aan op welk moment lokale constanten gecreëerd worden. Tot slot maken we gebruik van een zeshoeken om aan te geven dat een rol een bepaalde eigenschap claimt.



Figuur 3.1: Lowe's fixed version of Needham-Schroder Public Key

De operationele semantiek waarvan we hier gebruik maken is gelijktijdig met dit project ontwikkeld en staat beschreven in "Operational semantics of security protocols" [5]. De semantiek die wij hier gebruiken zal slechts op kleine punten verschillen, dit heeft twee belangrijke oorzaken. Ten eerste is dit een gevolg van bepaalde keuzes die wij hier gemaakt hebben, zoals het gebruik van een specifiefiek intrudermodel. Tevens is een deel van de semantiek in het artikel pas later ontwikkeld wat ook enkele verschillen oplevert. Toch kunnen veel definities overgenomen worden uit het artikel. Een lijst van de definites die overgenomen zijn, is in appendix B opgenomen.

We onderscheiden in een security-protocol vijf componenten. Hierbij maken we onderscheid tussen de volgende componenten. In de protocolspecificatie leggen we het gedrag van het protocol vast. Het agentmodel beschrijft het protocol op het uitvoeringsniveau. Vervolgens

beschrijven we het communicatie- en intrudermodel waar we de communicatie van het systeem vastleggen en beschrijven op welke manier een intruder het systeem kan beïnvloeden. Als vierde component beschouwen we de veiligheidseisen waaraan security-protocollen kunnen voldoen. Als laatste component onderscheiden wij de cyptografische primitieven waar we vastleggen welke eigenschappen we van de encryptiemethoden verwachten. In de volgende paragrafen zullen we het model aan de hand van deze vijf componenten beschrijven.

3.2 Protocolspecificatie

De protocolspecificatie beschrijft het gedrag van de rollen in een protocol. We onderscheiden in de meeste gevallen drie verschillende rollen. De initiatorrol (I), deze rol initialiseert het protocol. De responderrol (R), de reagerende rol in het protocol. Tot slot wordt in veel protocollen gebruik gemaakt van een serverrol (S). Deze rol zal in veel gevallen sessiesleutels distribueren.

We definiëren rollen als een lijst van acties. Deze acties hebben berichten als parameters. Deze berichten zullen we definiëren met behulp van een termsysteem. We introduceren de volgende termen als basiselementen van het termsysteem.

element	**omschrijving**	**gebruikelijke notatie voor de elementen**
Role	de rollen	i, r, s
Var	de variabelen	X, Y
Const	de constanten	nr, ni, kir
Func	functies	$K(s,r), K(i,r), PK(i), SK(i)$

We kunnen nu het termsysteem voor de roltermen definiëren. Met deze roltermen zijn we in staat om de berichten samen te stellen. We definiëren eerst de basisroltermen. Vervolgens definiëren we de roltermen door middel van de basisroltermen, tupeling en encryptie.

Definitie 1 *Roltermen*

$$
\begin{aligned}
\mathit{BasicRoleTerm} \quad &:= \quad < \mathit{Role} > \\
&\quad\ \mid \mathit{Func}(< \mathit{RoleTerm}^* >) \\
&\quad\ \mid < \mathit{Const} > \\
&\quad\ \mid < \mathit{Var} >
\end{aligned}
$$

$$
\begin{aligned}
\mathit{RoleTerm} \quad &:= \quad < \mathit{BasicRoleTerm} > \\
&\quad\ \mid (< \mathit{RoleTerm} >, < \mathit{RoleTerm} >) \\
&\quad\ \mid \{< \mathit{RoleTerm} >\}_{< \mathit{RoleTerm} >}
\end{aligned}
$$

De protocollen die wij zullen bestuderen maken gebruik van encryptie. We maken hierbij onderscheid tussen twee encryptiemethoden: symmetrische en asymmetrische encryptie. Voor de asymmetrische sleutels introduceren we de partiële functie *inversekey* : $BasicRoleTerm \rightarrow BasicRoleTerm$. Het domein van deze functie bepaalt welke roltermen een asymmetrische sleutel representeren en relateert deze aan zijn inverse sleutel. We eisen wel dat $\forall_{bt \in dom(inversekey)} inversekey(inversekey(bt)) = bt$. We zullen verder aannemen dat pk en sk functies zijn van ariteit 1 zodanig dat $\forall_{r \in Role} inversekey(pk(r)) = sk(r)$

De termen die niet in het domein van *inversekey* functie voorkomen beschouwen we als een potentiële symmetrische sleutels. We introduceren voor alle roltermen de inverse functie $\_^{-1} : RoleTerm \rightarrow RoleTerm$. We definiëren deze als volgt.

Definitie 2 *Inverse sleutel*

$$t^{-1} = \begin{cases} inversekey(t) & t \in dom(inversekey) \\ t & anders \end{cases}$$

Een term die achtereenvolgens geëncrypt is met een sleutel en zijn inverse sleutel levert de oorspronkelijke term weer op. De geëncrypte termen reduceren dus op de volgende manier.

Definitie 3 *Encryptie reductie*
Voor een willekeurige rolterm t en een basisrolterm k geldt.

$$t \equiv \{\{t\}_k\}_{k^{-1}}$$

We willen eenvoudig kunnen redeneren over termen die bevat zijn als term in een andere term. Hiervoor introduceren we het begrip bevat ($\sqsubseteq$), we definiëren deze relatie als volgt.

Definitie 4 *Bevat* $\sqsubseteq$
Voor willekeurige termen s en t

identiteit $t \sqsubseteq t$

projectie $t \sqsubseteq (s,t)$
$s \sqsubseteq (s,t)$

encryptie $t \sqsubseteq \{t\}_s$

Naast termen die verzonden en ontvangen worden bevatten de rollen in een protocol initiële kennis. Deze kennis bestaat gebruikelijk uit rollen waarmee de betreffende rol berichten uitwisselt en enkele sleutels. Voor de initiële kennis definiëren we: $RoleKnow = \mathcal{P}(RoleTerm)$.

Voorbeeld

De initiële kennis van de initiatorrol van het NSL protocol bestaat uit de rol zelf en de rollen waarmee de initiator communiceert, dus i en r. Verder bestaat deze kennis uit de lokale constante ni en de sleutels $pk(i)$, $sk(i)$ en $pk(r)$. De initiële kennis voor de initiator ziet er dan als volgt uit.

$$\{i, r, ni, pk(i), sk(i), pk(r)\}$$

De rollen in het systeem kunnen drie verschillende acties uitvoeren. Een rol kan een $send_l(i, r, t)$ uitvoeren. Een rol i stuurt een term t naar een rol r. De tweede actie die een rol kan uitvoeren is een $read_l(i, r, t)$. Een rol r leest een term t gestuurd door een rol i. Tot slot kan een rol een $claim_l(i, c)$ uitvoeren. Een rol i claimt lokaal de eigenschap c uit de verzameling *Claim*. Voor alle acties is een label l noodzakelijk. We gebruiken hiervoor de vrameling *Label*. Door dit label is het mogelijk om dezelfde acties in een protocolspecificatie van elkaar te kunnen onderscheiden. We definiëren de verzameling acties (*events*) als volgt.

Definitie 5 *Event*

$$\begin{aligned} Event \;=\; \{ & send_\ell(i, r, t), read_\ell(i, r, t), claim_\ell(i, c) \mid \\ & \ell \in Label, i, r \in Role, t \in RoleTerm, c \in Claim\} \end{aligned}$$

We zijn nu in staat op de rolspecificatie te definiëren. De rolspecificatie bestaat uit een omschrijving van het gedrag van de rol d.w.z. de mogelijke acties die uitgevoerd kunnen worden en de initiële kennis van de rol. Het is niet mogelijk dat de initiële kennis variabele bevat.

Definitie 6 *Rolspecificatie*

$$RoleSpec = RoleKnow \times Event^*$$

Voorbeeld

We zullen aan de hand van het NSL protocol dat we geschetst hebben in Figuur 3.1 laten zien hoe de rolspecificatie voor de initiatorrol er uit komt te zien. We hebben in het vorige voorbeeld al laten zien wat de initiële kennis is. We voegen hier nog de acties die de initiatorrol kan uitvoeren aan toe.

$$\begin{aligned} nsl(i) \;=\; & \big(\{i, r, ni, pk(i), sk(i), pk(r)\}, \\ & send_1(i, r, \{n_i, i\}_{PK(r)}) \cdot \\ & read_2(r, i, \{n_i, X, r\}_{PK(i)}) \cdot \\ & send_3(i, r, \{X\}_{PK(r)})\big) \end{aligned}$$

De protocolspecificatie is het gedrag van alle rollen in een protocol. Hiervoor beelden we de rollen af op een rolspecificatie. We definiëren de protocolspecificatie als volgt.

Definitie 7 *Protocolspecificatie*

$$ProtSpec = Role \rightarrow RoleSpec$$

3.3 Agentmodel

De agenten executeren de rollen zoals beschreven in de protocolspecificatie. We nemen aan dat de agenten het protocol ten alle tijden alleen op de manier executeren zoals dit voorgeschreven wordt. De executie van een rol door een agent noemen we een run. Een agent kan parallel één of meer instanties van diverse protocollen runnen.

Om het protocol op het runniveau te beschrijven hebben we extra termen nodig. We introduceren de basiselementen *Agent* en *Runid*. De agenten zijn de instanties van een rol die het protocol runnen. Binnen de verzameling van agenten beschouwen we een bijzondere verzameling van betrouwbare agenten, deze verzameling duiden we aan met $Agent_T$. De runidentifier bindt de nonces aan de run waarin ze gemaakt zijn op een symbolische manier. Verder helpen ze de runs te onderscheiden van elkaar. De basiselementen *Func* en *Const* vervullen de dezelfde rol als bij de roltermen.

Net als we voor de protocolspecificatie roltermen gedefiniëerd hebben, kunnen we nu de runtermen definiëren voor het agentmodel.

Definitie 8 *Runtermen*

$$
\begin{aligned}
\mathit{BasicRunTerm} &:= < \mathit{Agent} > \\
&\mid \mathit{Func}(< \mathit{RunTerm}^* >) \\
&\mid < \mathit{Const} > \# < \mathit{Runid} >
\end{aligned}
$$

$$
\begin{aligned}
\mathit{RunTerm} &:= < \mathit{BasicRunTerm} > \\
&\mid (< \mathit{RunTerm} >, < \mathit{RunTerm} >) \\
&\mid \{< \mathit{RunTerm} >\}_{<\mathit{RunTerm}>}
\end{aligned}
$$

We hebben een functie nodig die de abstracte specificatie op een run afbeeldt. Hiervoor definiëren we de functie *Inst*. Deze functie legt een instantie, $(rid, \rho, \sigma) \in \mathit{Inst}$ als volgt vast. De *rid* legt de betreffende run vast. De ρ beeldt de rollen op agenten af en de σ zorgt voor de afbeeldingen van de variabelen op waarde. We definiëren de functie *Inst* als volgt.

Definitie 9 *Instantiatie functie*

$$\mathit{Inst} = \mathit{RID} \times (\mathit{Role} \rightarrow \mathit{Agent}) \times (\mathit{Var} \rightarrow \mathit{RunTerm})$$

Voorbeeld

We geven twee voorbeelden van instantiaties die kunnen voorkomen bij de executie van het NSL protocol.

$$
\begin{aligned}
\big(1, \{i \mapsto a, r \mapsto b\}, \emptyset\big) \quad \big(\{ni, i\}_{pk(r)}\big) &= \{ni\sharp 1, a\}_{pk(b)} \\
\big(2, \{i \mapsto c, r \mapsto d\}, \{X \mapsto ni\sharp 1\}\big) \quad \big(\{X, nr, r\}_{pk(i)}\big) &= \{ni\sharp 1, nr\sharp 2, d\}_{pk(c)}
\end{aligned}
$$

Een run is een geïnstantieerde rol. Deze is te beschrijven met een instantiatiefunctie en verzameling acties (events). We definiëren een run dan ook op de volgende manier.

Definitie 10 *Run*

$$\mathit{Run} = \mathit{Inst} \times \mathit{Event}^*$$

3.4 Communicatie- en intrudermodel

Het communicatiemodel legt vast op welke manier berichten die verzonden worden bij de ontvanger terecht komen. Het beschrijft dus op welke manier een bericht door het netwerk getransporteerd wordt.

Het intrudermodel legt vast op welke manier de intruder een systeem kan beïnvloeden. Wij hebben hier gekozen voor het Dolev-Yao intrudermodel [7]. De keuze voor dit model is gemaakt omdat het Dolev-Yao het sterkste intrudermodel is. Indien een bepaalde eigenschap in dit model geldt zal het ook in een zwakker model gelden. In het Dolev-Yao model vervult de intruder de rol van het netwerk. De communicatie van het systeem wordt dus volledig door de intruder bepaald.

Een bericht dat verzonden is, wordt door de intruder in zijn kennis opgenomen. Een bericht kan alleen gelezen worden door een agent als het bericht in de kennis van de intruder voorkomt. De intruder heeft de mogelijkheid om berichten naar willekeur te injecteren, berichten uit het systeem te nemen en berichten te modificeren naar gelang zijn kennis dit toelaat.

De kennis van de intruder duiden we aan als M. Deze kennis is opgebouwd uit alle mogelijk runtermen die de intruder heeft kunnen creëren en ontvangen. We definiëren de kennis van de intruder op de volgende manier.

Definitie 11 *Intruder kennis (M)*

$$M = \mathcal{P}(RunTerm)$$

Een intruder is in staat om berichten die niet geëncrypt zijn te lezen en te ontleden. Berichten die geëncrypt zijn kan een intruder alleen lezen als deze in het bezit is van de juiste sleutel om de berichten te decrypten. Een intruder heeft ook de mogelijkheid om uit zijn kennis zelf berichten te creëren en te encrypten. We definiëren de transitieve afsluiting voor een intruder M als volgt.

Definitie 12 *Transitieve afsluiting van M*
Als een intruder een paar van termen kent dan kent deze ook de termen afzonderlijk van elkaar.

$$(s,t) \in M \Leftrightarrow s \in M \wedge t \in M$$

Een intruder heeft de mogelijkheid om termen te encrypten.

$$(s,t) \in M \Rightarrow \{s\}_t \in M$$

Een intruder kan een term decrypten mits de intruder in het bezit is van de inverse van de sleutel waarmee de term versleuteld is.

$$(\{s\}_t \in M, t^{-1} \in M) \Rightarrow s \in M$$

De mogelijkheid bestaat dat termen in de kennis van de intruder voorkomen die versleuteld zijn en waarvan de intruder niet de decryptiesleutel kent. De intruder kan dan de versleutelde term niet herkennen. Bijvoorbeeld de term $\{t\}_k$ komt in de kennis van de intruder voor maar de term k^{-1} komt niet in zijn kennis voor. De intruder kan in dit geval de term t niet herkennen. Het is mogelijk dat de intruder op een later tijdstip alsnog de decryptiesleutel (k^{-1}) leert, zodat hij de term t nu wel kan herkennen. Vanwege deze eigenschap willen we makkelijk kunnen redeneren over subtermen die in de intruderkennis voorkomen. We introduceren de relatie *bevat als subterm* voor de intruderkennis.

Definitie 13 *Bevat als subterm*
Voor een willekeurige term t, t' en een intruder M

$$t \lhd M \equiv (\exists_{t' \in M} : t \sqsubseteq t')$$

3.5 Toestand

In deze paragraaf beschrijven we de toestand van het systeem en op welke manier een toestandsovergang plaats vindt. De toestand in ons model wordt vastgelegd door de agenten die een protocol runnen en door de intruder die bepaald welke berichten agenten te lezen krijgen. We definiëren de toestand van het systeem op de volgende manier.

Definitie 14 *Toestand*

$$State = \mathcal{P}(RunTerm) \times \mathcal{P}(Run)$$

We kunnen het gedrag van het systeem definiëren met behulp van een transitiesysteem. Het transitiesysteem beschrijft op welke manieren het systeem van de ene toestand kan overgaan in een ander toestand. Elke transitie is gelabeld met een element uit de verzameling *Transitionlabels*.

Definitie 15 *Transitionlabel*

$$Transitionlabel ::= (Inst, Event) \mid create(Run)$$

De initiële lokale toestand van een run definiëren we met de functie $runsof : ProtSpec \rightarrow \mathcal{P}(runs)$. Initieel heeft elke run een runidentifier rid. Elke runidentifier dient uniek te zijn. Ten tweede worden de rollen op agenten afgebeeld. Voor deze afbeeldingen van agenten op rollen gebruiken we de hulpfunctie $roles : RoleTerm \rightarrow \mathcal{P}(Roles)$, die aangeeft welke rollen voorkomen in een rolterm. Tot slot kan de toestand van een run initieel geen variabele bevatten, dus $\sigma = \emptyset$.

Definitie 16 *Initiële lokale toestand*

$$runsof(p) = \{((rid, \rho, \emptyset), p(r)) \mid rid \in RID, r \in dom(p) \ \wedge \ roles(p(r)) = dom(\rho)\}$$

Voorbeeld

We bekijken wederom het voorbeeld van het NSL protocol om een voorbeeld van een initiële toestand van de initiatorrol te geven. Om het domein van ρ te bepalen moeten we bepalen welke rollen ervoor komen in dit protocol. $roles(nsl(i)) = \{i, r\}$. We kiezen ervoor om deze rollen respectievelijk op de agenten a en b af te beelden. Verder kiezen we als runidentifier 1. Dit resulteert in de volgende initiële toestand voor de initiatorrol.

$$(1, \{i \mapsto a, r \mapsto b\}, \emptyset)$$

We definiëren de functie F die runs van alle bestaande agenten en agenten die ooit bestaan hebben representeert. Voor $F \in \mathcal{P}(Run)$ gebruiken we $F[r'/r]$ om de substitutie van r met r' in F aan te geven. Verder definiëren we de verzameling actieve runidentifiers als volgt.

Definitie 17 *Actieve runidentifiers*

$$runids(F) = \{rid \mid ((rid, \rho, \sigma), event) \in F\}$$

De initiële kennis van de intruder duiden we aan als M_0. Initieel kan de kennis van de intruder geen constanten bevatten die gebonden zijn. We definiëren dit op de volgende manier.

Definitie 18 *Initiële kennis van de intruder (M_0)*

$$\forall_{c \in Const, rid, \rho, \sigma} M_0 \not\vdash (rid, \rho, \sigma)(c)$$

Tot slot zullen we de initiële toestand van het systeem definiëren. De initiële toestand van de intruder hebben we reeds gedefinieërd. Verder bevat het systeem initieel geen runs en daarom zal deze verzameling leeg zijn.

Definitie 19 *Initiële toestand*

$$s_0 = \langle M_0, \emptyset \rangle$$

3.5.1 Matchen

Om te bepalen of bij een read-actie een ontvangen bericht aan het verwachte patroon voldoet maken we gebruik van de *match* functie. We zullen eerst nader ingaan op de typering van berichten.

We maken gebruik van een sterke typering. Wat wil zeggen dat de agenten in ons systeem in staat zijn om te herkennen of een variabelen aan het verwachte type voldoet. Bijvoorbeeld: als een agent verwacht een nonce te ontvangen en hij ontvangt hiervoor in de plaats een term van een ander type dan zal deze niet geaccepteerd worden.

Een agent die een geëncrypt bericht ontvangt en niet de sleutel bezit om dit bericht de decrypten is niet in staat om het type van de termen in dit bericht af te leiden. Dit probleem doet zich bijvoorbeeld voor wanneer een agent een ticket ontvangt. We zullen hier later in paragraaf 4.9 dieper op ingaan.

Voor alle variabelen definiëren we een verzameling van runtermen die toegestaan zijn voor deze variabelen. We introduceren hiervoor de hulpfunctie $Type : Var \rightarrow RunTerm$. Vervolgens introduceren we het predikaat: *Welltyped* op $(Var \rightarrow RunTerm)$. Dit predikaat stelt ons in staat om te bepalen of een substitutie op de juiste manier getypeerd is.

Definitie 20 *Correct getypeerd*

$$Welltyped(\sigma) = \forall_{v \in dom(\sigma)} \big(\sigma(v) \in Type(v)\big)$$

Berichten uit de intruderkennis die aan agenten worden aangeboden, worden op basis van een patroon geaccepteerd. Voor deze acceptatie introduceren we het predikaat *match* : $Inst \times RoleTerm \times RunTerm \times Inst$. De match controleert of het inkomende bericht, het derde argument, voldoet aan het verwachte patroon. Dit verwachte patroon wordt gerepresenteerd door het tweede argument en geïnstantieerd door het eerste argument maar het is mogelijk dat er nog vrije variabele in voorkomen die gebonden dienen te worden. Deze

variabelen worden zo gebonden dat het inkomende bericht gelijk is aan de instantie van de rolterm. De resulterende instantiatie wordt gerepresenteerd door het vierde argument.

Definitie 21 *Match*

$$match(inst, pt, m, inst') \iff inst = (rid, \rho, \sigma) \wedge inst' = (rid, \rho, \sigma') \wedge \sigma \subseteq \sigma' \wedge dom(\sigma') = dom(\sigma) \cup vars(pt) \wedge Welltyped(\sigma') \wedge (rid, \rho, \sigma')(pt) = m$$

3.5.2 Afleidingregels

We zijn nu in staat om de afleidingregels voor het systeem vast te leggen. De regels zijn een vereenvoudiging van de regels in [5]. We kunnen het systeem met een viertal regels beschrijven.

Deze vier regels beschrijven de verandering van de toestand als gevolg van respectievelijk: het creëren van een run, het versturen van een bericht, het ontvangen van een bericht of het claimen van een veiligheidseis. De afleidingregels zijn terug te vinden in Tabel 3.1.

$$[create] \quad \frac{run = ((rid, \rho, \sigma), s) \in runsof(p), rid \notin runids(F)}{\langle M, F\rangle \xrightarrow{create(run)} \langle M, F \cup \{run\}\rangle}$$

$$[send] \quad \frac{run = (inst, send_\ell(m) \cdot s) \in F}{\langle M, F\rangle \xrightarrow{(inst, send_\ell(m))} \langle M \cup \{inst(m)\}, F[(inst, s)/run]\rangle}$$

$$[read] \quad \frac{run = (inst, read_\ell(pt) \cdot s) \in F, m \in M, Match(inst, pt, m, inst')}{\langle M, F\rangle \xrightarrow{(inst', read_\ell(pt))} \langle M, F[(inst', s)/run]\rangle}$$

$$[claim] \quad \frac{run = (inst, claim_\ell(r, c) \cdot s) \in F}{\langle M, F\rangle \xrightarrow{(inst, claim_\ell(r,c))} \langle M, F[(inst, s)/run]\rangle}$$

Tabel 3.1: SOS-regels

3.5.3 Trace

De veiligheidseisen die later aanbod komen zijn gebaseerd op traces. We definiëren een trace op basis van de afleidingregels uit Tabel 3.1. Een trace legt vast in welke volgorde de verschillende acties door diverse agenten zijn uitgevoerd binnen een protocol.

We voeren de volgende notaties in voor traces. Voor $\alpha = \alpha_0 \ldots \alpha_{n-1} \in \mathit{Transitionlabel}^*$ gebruiken $s_0 \xrightarrow{\alpha} s_n$ om te omschrijven dat $\exists_{s_1,\ldots s_{n-1}} s_0 \xrightarrow{\alpha_0} s_1 \ldots s_{n-1} \xrightarrow{\alpha_{n-1}} s_n$. We gebruiken $s \xrightarrow{\alpha}$ om te omschrijven $\exists_{s'} s \xrightarrow{\alpha} s'$. De verzameling van traces $Tr : \mathit{ProtSpec} \rightarrow \mathcal{P}(\mathit{Transitionlabel}^*)$ definiëren we nu als volgt.

Definitie 22 *Trace*
$\{a \in \mathit{Transitionlabel}^* \mid s_0 \xrightarrow{a}\}$, *zodanig dat* s_0 *de initiële toestand van het protocol is. Voor de trace* α, *gebruiken we* α_i *om aan te geven dat* i^{th} *actie label van trace* α.

We kunnen de toestand van de intruder op de volgende manier koppelen aan een trace. Als α_i een actie uit de trace α is, dan is M_i de intruder kennis direct voor het uitvoeren van α_i.

3.6 Veiligheidseisen

Voor authenticatieprotocollen bestaan diverse veiligheidseisen. Enkele voorbeelden hiervan zijn *secrecy*, *agreement* en *synchronisation*. In de literatuur zijn voor deze eigenschappen vele verschillende definities te vinden die vaak slechts in details van elkaar verschillen.

In [6] worden de voor de veiligheisen: *(non-injective) synchronisation en (non-injective) agreement* formeel gedefiniëerd. We weten uit dit artikel dat synchronisatie de andere eigenschappen impliceert en non-injectieve synchronisatie impliceert non-injectieve *agreement*. Verder bestaat het vermoeden dat uit synchronisatie relatief eenvoudig non-injectieve synchronisatie is af te leiden. Non-injectieve synchronisatie is dus een redelijke sterke eigenschap en als het meezit is met behulp van deze eigenschap synchronisatie afleiden. Om deze reden besteden we alleen aandacht aan non-injectieve synchronisatie.

Een agent die een veiligheidseis claimt dient een eerlijk ofwel betrouwbare agent te zijn. Ook de agenten waarmee deze agent meent te communiceren dienen te vertrouwen te zien. Dus als in een run van een agent de actie $(\mathit{rid}, \rho, \sigma, \mathit{claim}_l(i, c))$ voorkomt dan nemen we aan dat $\mathit{rng}(\rho) \subseteq \mathit{Agent}_T$. Het heeft geen zin om claims van onbetrouwbare agenten te bekijken omdat dan zelfs de claim niet te vertrouwen zou zijn. Evenzo heeft het geen zin om aan te nemen dat de agenten waarmee men meent te communiceren niet te vertrouwen zouden zijn. We kunnen van onbetrouwbare agenten niet verwachten dat ze het protocol

netjes zullen volgen. Natuurlijk wil dit niet zeggen dat de agenten waarmee werkelijk gecommuniceerd wordt te vertrouwen zijn. Dit zullen we moeten aantonen.

Als een agent non-injectieve synchronisatie claimt dan wil zeggen dat alle acties die in dit protocol voor het moment van de claim voorkomen, plaats gevonden hebben zoals de protocolbeschrijving dit voorschrijft. Voor elke bericht dat gelezen is moet gelden dat het exact hetzelfde bericht verzonden is en dat het moment van verzending het moment van lezen voorafgegaan is.

De definitie van non-injectieve synchronisatie uit [5] is in drie stappen opgebouwd. Het eerste predikaat drukt uit dat voor een zeker label l een voorkomen $send_\ell$ en $read_\ell$ bij elkaar passen en dat de send-actie de read-actie voorafgegaan is. We gebruiken de functie *sendrole* en *readrole* om uit te drukken welke rol bij een label respectievelijk de afzender of ontvanger is. Verder definiëren we de projectiefunctie $runidof : Inst \rightarrow RID$ door $runidof(rid, \rho, \sigma) = rid$.

Definitie 23 *Eén label synchronisatie*
Voor alle traces α, $k \in N$, labels ℓ en run identifiers rid_1, rid_2, kunnen we het één-label synchronisatie predikaat 1L-SYNCH definiëren als.

$$
\begin{aligned}
&1L\text{-}SYNCH(\alpha, k, \ell, rid_1, rid_2) \iff \\
&\quad \exists_{i,j \in N, inst_1, inst_2 \in Inst, m_1, m_2 \in MSG} \\
&\qquad i < j < k \wedge \\
&\qquad \alpha_i = (inst_1, send_\ell(m_1)) \ \wedge \ runidof(inst_1) = rid_1 \wedge \\
&\qquad \alpha_j = (inst_2, read_\ell(m_2)) \ \wedge \ runidof(inst_2) = rid_2 \wedge \\
&\qquad inst_1(m_1) = inst_2(m_2)
\end{aligned}
$$

Het tweede predikaat generaliseert over de verzameling van labels. We introduceren hiervoor de functie *cast*. Deze functie beeldt een rol op een runidentifier af.

Definitie 24 *Multi label synchronisatie*
Voor alle traces α, $k \in N$, verzameling labels ℓ en de runindentifiers rid_1, rid_2, kunnen we het multi label synchronisatie predikaat ML-SYNCH als.

$$
\begin{aligned}
&ML\text{-}SYNCH(\alpha, k, L, cast) \iff \\
&\quad \forall_{\ell \in L} \ 1L\text{-}SYNCH\big(\alpha, k, \ell, cast(sendrole(\ell)), cast(readrole(\ell))\big)
\end{aligned}
$$

Als aan het predikaat $ML\text{-}SYNCH(\alpha, k, L, cast)$ voldaan is, kunnen we zeggen dat de acties behorende bij de verzameling labels L op een correcte manier in een trace α voorkomen voor een punt k met betrekking tot een instantie van *cast*.

Voor non-injectieve synchronisatie is de volgorde waarin de acties voorkomen van groot belang. Om de volgorde waarin acties binnen een rol behoren plaats te vinden definiëren we de ordeningsoperator ($\prec$).

Definitie 25 *Ordening*

Een event e dat voorafgegaan is aan een event e' in een rol r noteren we als volgt: $e \prec_r e'$.

Om de relevante verzameling van labels af te leiden die gecontroleerd dienen te worden voor een synchronisatie claim in een zeker protocol p definiëren we de verzameling $prec(p, cl)$.

Definitie 26 *Preceding label set*
De verzameling met de labels van alle acties die de claim gelabeld met label cl voorafgegaan zijn in een protocol p, definiëren we als.

$$prec(p, cl) = \{\ell \mid read_\ell(\_, \_, \_) \prec claim_{cl}(\_, \_)\}$$

We kunnen nu het predikaat voor non-injectieve synchronisatie definiëren. Hiervoor introduceren we de claim $nisynch \in Claim$.

Definitie 27 *Non-injectieve synchronisatie*
Een protocol p is correct met betrekking tot NI-SYNCH als het volgende geldig is voor alle traces $\alpha \in Tr(p)$.

$$\begin{aligned} \alpha_i = \ & (rid, \rho, \sigma, claim_\ell(r, nisynch)) \wedge rng(\rho) \subseteq Agent_T \\ & \Rightarrow \exists_{cast:Role \to RID}\ (cast(r) = rid \wedge ML\text{-}SYNCH(\alpha, i, prec(p, \ell), cast)) \end{aligned}$$

3.7 Cryptografische primitieven

We maken in ons model gebruik van de *black box* aanpak, wat betekent dat we ons niet bezig houden met de wiskundige details van de cryptografie. We maken alleen gebruik van de veronderstelde eigenschappen van cryptografische primitieven. We houden ons dus niet bezig met welke encryptiemethode gebruikt wordt. We gaan er tevens vanuit dat de gebruikte primitieven onfeilbaar zijn, wat wil zeggen dat het niet mogelijk is om bijvoorbeeld een sleutel te kraken of een waarde die onvoorspelbaar wordt geacht te voorspellen.

Voor de protocollen die wij bestuderen maken we gebruik van twee belangrijk cryptografische primitieven: symmetrische en asymmetrisch encryptie. Hashing laten we buiten beschouwing omdat de protocollen waarvan we in hoofdstuk 4 een bewijs geven, geen gebruik van hashing maken.

3.7.1 Symmetrische encryptie

Bij symmetrische encryptie wordt er slechts gebruik gemaakt van één sleutel, deze sleutel en zijn inverse sleutel zijn in dit geval hetzelfde. We duiden een symmetrische sleutel van twee rollen gebruikelijk aan als $K(i,r)$. We sluiten hier de mogelijkheid dat $K(i,r) = K(r,i)$ uit. We definiëren de gelijkheid van symmetrische sleutels als volgt.

Definitie 28 *Gelijkheid symmetrische sleutel*

$$\rho(K(i,r)) = \rho'(K(i,r)) \Rightarrow \rho(i) = \rho'(i) \wedge \rho(r) = \rho'(r)$$

3.7.2 Asymmetrische encryptie

Voor asymmetrische encryptie wordt gebruik gemaakt van twee sleutels. We onderscheiden hierbij een publieke sleutel pk en een geheime sleutel sk. De publieke sleutel mag iedereen kennen. Van de geheime sleutel gaan we ervan uit dat alleen de eigenaar deze sleutel kent. Dit levert een aantal belangrijke eigenschappen op.

1. Een bericht vercijferd met een geheime sleutel kan door iedereen ontcijferd worden maar kan slechts gemaakt zijn door degene die de geheime sleutel bezit.
2. Een bericht vercijferd met een publieke sleutel kan door iedereen gemaakt worden maar kan slechts ontcijferd worden door degene die de geheime sleutel bezit.

Hoofdstuk 4

Bewijzen van protocollen uit SPORE

In dit hoofdstuk bestuderen we de protocollen uit SPORE [14]. We maken eerst een korte analyse van de protocollen en vervolgens bewijzen we de protocollen waarvoor non-injectieve synchronisatie geclaimd mag worden.

4.1 Analyse van protocollen uit SPORE

SPORE is een collectie van security-protocollen. Op dit moment bevat SPORE 45 protocollen. We zullen deze 45 protocollen kort bestuderen en van elk protocol aangeven of er wel of geen aanval mogelijk is. Hierbij maken we onderscheid tussen de volgende type aanvallen. In Clark en Jacob [3] worden dezelfde type aanvallen onderscheiden.

Freshness attack (1): Bij een 'freshness attack' worden berichten of delen hiervan die al eens eerder zijn gestuurd opnieuw gebruikt om zo een agent om de tuin te leiden. In het geval dat een 'freshness attack' bestaat voor een zeker protocol dan zal er geen non-injectieve synchronisatie voor dit protocol gelden.

Type flaw attack (2): Een aanval van dit type maakt gebruik van het feit dat een ontvanger van een bericht ook een bericht zal accepteren dat niet conform de protocolbeschrijving is verstuurd. Een agent kan bijvoorbeeld een nonce accepteren op een plek waar een sleutel verwacht wordt. We maken in ons model gebruik van sterke typering, agenten zijn in staat om het type van termen te herkennen. Een aanval van dit type is dus niet mogelijk in ons model.

Parallel session attack (3): Bij een aanval van dit type wordt gebruik gemaakt van van de mogelijkheid om een parallelle sessie van het protocol te starten. Een bekend voorbeeld van een parallel session attack is de aanval op het Needham Schroeder protocol [13]. Indien dit type aanval op een protocol mogelijk is zal er geen non-injectieve synchronisatie optreden.

Voor de protocollen waarvoor geldt dat er geen aanval mogelijk is van het type 1 of 3, is het interessant om te bekijken of er non-injectieve synchronisatie optreedt. Echter er zijn nog twee eigenschappen die in de protocollen uit SPORE kunnen voorkomen die non-injectieve synchronisatie in de weg staan. Een protocol kan gebruik maken van een ticket of tijd. We zullen deze eigenschappen nader toelichten.

Tickets (a): Een ticket is een bericht of een deel van een bericht dat door een agent wordt ontvangen en niet leesbaar is voor deze agent. In de meeste gevallen zal dit door encryptie veroorzaakt worden. Deze agent stuurt op zijn beurt dit bericht in dezelfde vorm weer door naar een andere agent. De agent die deze berichten doorstuurt wordt slechts als tussenpersoon gebruikt. De tussenpersoon heeft dus niet de mogelijkheid om te controleren of hij de informatie heeft ontvangen en doorgestuurd zoals het protocol dat voorschrijft. Het is essentieel dat de agent die het bericht doorstuurt geen bewerking op dit bericht uitvoert. Een eventuele intruder kan ervoor zorgen dat zo'n ticket niet bij deze tussenpersoon terecht komt en omdat deze tussenpersoon het bericht niet kan controleren, kan een intruder elk willekeurig bericht aan deze tussenpersoon doen toekomen. Dit soort berichten zullen we aanduiden als een ticket. Indien een protocol een ticket bevat is het niet mogelijk dat er non-injectieve synchronisatie optreedt. In paragraaf 4.9 op pagina 84 zal uitvoerig op dit probleem in gegaan worden. Tevens zal voor het ticket-probleem in het "Kao Chow Authentication v.1" protocol een oplossing gepresenteerd worden.

Timing (b): In het model dat wij beschouwen is geen notie van tijd opgenomen Veel protocollen maken gebruik van 'timestamps' om de versheid van berichten te kunnen controleren. Voor de protocollen die op deze of een soortgelijke wijze gebruik maken van tijd, kunnen wij dus niets zeggen over non-injectieve synchronisatie.

In Tabel 4.1 zijn alle protocollen uit SPORE terug te vinden. De protocollen staan in dezelfde volgorde als waarin ze in SPORE opgenomen zijn. Per protocol zullen we aangeven welke aanval bekend is. Verder geven we per protocol aan of het protocol gebruik maakt van tickets en/of tijd.

Protocol	1	2	3	a	b	Opmerkingen
Andrew Secure RPC	+					
BAN modified Andrew Secure RPC						Bewijs in paragraaf 4.6
BAN concrete Andrew Secure RPC			+			
Lowe modified BAN concrete Andrew Secure RPC						Bewijs in paragraaf 4.5
CAM					+	

1. *Freshness attack*
2. *Type flaw attack*
3. *Parallel session attack*

a. *Protocol maakt gebruik van tickets*
b. *Protocol maakt gebruik van timing*

Protocol	1	2	3	a	b	Opmerkingen
CCITT X.509 (1)					+	
CCITT X.509 (1c)					+	
CCITT X.509 (3)			+		+	
BAN modified version of CCITT X.509 (3)						Bewijs in paragraaf 4.7
Denning-Sacco shared key			+			
Lowe modified Denning-Sacco shared key					+	
Diffie Helman						De aanval uit [14] is niet in één van deze klassen in te delen. Het betreft een imitatie aanval.
GJM						Protocol is erg complex voor een handmatig bewijs. Tevens wordt er een aanval op het protocolbeschreven in [15].
Gong						Protocol is erg complex voor een handmatig bewijs. Er is geen aanval bekend.
Kao Chow Authentication v.1				+		Een oplossing voor het ticket probleem wordt in paragraaf 4.9 gepresenteerd.
Kao Chow Authentication v.2				+		
Kao Chow Authentication v.3				+		
Kerberos V5					+	
KSL	+			+	+	Timestamp is waarschijnlijk door een nonce te vervangen.
Lowe modified KSL				+	+	Timestamp is waarschijnlijk door een nonce te vervangen.
Neumann Stubblebine	+			+		Tevens is er ook nog een *plaintext* aanval mogelijk
Hwang modified version of Neumann Stubblebine				+	+	Timestamps mogelijk te vervangen door nonces.
Needham-Schroeder Public Key			+			
Lowe's fixed version of Needham-Schroder Public Key						Bewijs in paragraaf 4.3
Needham Schroeder Symmetric Key						Bewijs in paragraaf 4.8
Amended Needham Schroeder Symmetric Key				+		
Otway Rees		+		+		

1. *Freshness attack*
2. *Type flaw attack*
3. *Parallel session attack*

a. *Protocol maakt gebruik van tickets*
b. *Protocol maakt gebruik van timing*

Protocol	1	2	3	a	b	Opmerkingen
SK3				+		De aanval is niet relevant aangezien er aangenomen wordt dat een nonce voorspelbaar is.
SmartRight view-only						Geen non-injectieve synchronisatie omdat VoR in bericht 3 elk willekeurig nummer kan zijn.
SPLICE/AS					+	Diverse aanvallen mogelijk maar niet te klassificeren
Hwang and Chen modified SPLICE/AS					+	Aanval niet te classificeren
Clark and Jacob modified Hwang and Chen modified SPLICE/AS	+				+	
TMN	+		+			
Wide Mouthed Frog	+				+	
Lowe modified Wide Mouthed Frog					+	
Woo and Lam Mutual Authentication			+	+		
Woo and Lam Pi		+		+		
Woo and Lam Pi 1		+		+		
Woo and Lam Pi 2		+		+		
Woo and Lam Pi 3		+		+		
Woo and Lam Pi f						Het is niet mogelijk om non-injectieve synchronisatie aan te tonen voor dit protocol. Indien er communicatie met zichzelf plaatsvindt, is het niet mogelijk om onderscheid te maken tussen de berichten 3 en 5.
Yahalom				+		
BAN simplified version of Yahalom	+					
Lowe's modified version of Yahalom						Bewijs in paragraaf 4.4
Paulson's strengthened version of Yahalom				+		

1. *Freshness attack*
2. *Type flaw attack*
3. *Parallel session attack*

a. *Protocol maakt gebruik van tickets*
b. *Protocol maakt gebruik van timing*

Tabel 4.1: Overzicht van SPORE protocollen

Van de protocollen waarvoor een 'freshness attack' of een 'parallel session attack' bestaat weten we dat er geen non-injectieve synchronisatie geldt. Tevens weten we dat de protocollen die gebruik maken van een ticket niet non-injectief synchroniseren. Een 'type flaw

attack' is niet mogelijk in ons model dus een aanval van dit type is niet relevant. Van de protocollen die gebruik maken van tijd kunnen we niets zeggen omdat ons model dit niet toe staat. In sommige protocollen komen veel communicaties voor of wordt gebruik gemaakt van lastige primitieven. We hebben hier als doel gesteld om een patroon in de bewijzen te ontdekken en daarom besteden we geen aandacht aan deze protocollen. Bij deze protocollen hebben we opgemerkt dat deze te complex zijn voor een handmatig bewijs. Na deze korte analyse houden we het volgende lijstje protocollen over die mogelijkerwijs non-injectief synchroniseren.

protocol	paragraaf	pagina
Lowe's fixed version of Needham-Schroder Public Key	4.3	30
Lowe's modified version of Yahalom	4.4	40
Lowe modified BAN concrete Andrew Secure RPC	4.5	51
BAN modified Andrew Secure RPC	4.6	58
BAN modified version of CCITT X.509 (3)	4.7	65
Needham Schroeder Symmetric Key	4.8	72

Tabel 4.2: SPORE protocollen die mogelijkerwijs non-injectief synchroniseren

Zoals in Tabel 4.1 te zien is, geldt er voor het 'Kao Chow Authentication v.1' protocol geen non-injectieve synchronisatie omdat het protocol gebruikt maakt van een ticket. We zullen in paragraaf 4.9 op pagina 84 laten zien dat dit protocol door een kleine wijziging gerepareerd kan worden zodat er wel non-injectieve synchronisatie geclaimd mag worden.

4.2 Bewijsmethode

In de paragrafen 4.3 t/m 4.9 geven we de bewijzen voor non-injectieve synchronisatie van diverse protocollen. We zullen eerst kort toelichten op welke manier de bewijzen opgebouwd zijn en welke lemma's hiervoor gebruikt worden.

4.2.1 Opbouw

We beginnen elke paragraaf met een schematische weergave van het protocol d.m.v. een MSC. De protocolbeschrijvingen uit SPORE waaruit deze MSC's gegenereerd zijn, zijn in bijlage A te vinden. In de MSC's zijn ook de veronderstelde claims opgenomen. We hebben gekozen om voor één rol non-injectieve synchronisatie te bewijzen. Het bewijzen van andere rollen in een protocol zou in veel gevallen min of meer op dezelfde wijze verlopen. Tevens hebben we ons zelf als doel gesteld om zoveel mogelijk protocollen uit SPORE te bewijzen. Het bewijzen van meerdere rollen zou omwille van de beschikbare tijd dit doel in de weg staan. Tot slot vergde het bewijzen van een protocol, zeker in het begin, erg veel tijd. Enige afwisseling bij het bewijzen en ervaringen met andere procollen opdoen wordt als zeer wenselijk beschouwd.

Vervolgens genereren we uit deze schematische weergave van het protocol de rolbeschrijvingen. Deze rolbeschrijvingen stellen ons in staat om te zien welke berichten door de rollen verstuurd en verwacht worden.

Nadat we het protocol uitvoerig hebben beschreven kunnen we overgaan tot het bewijzen van non-injectieve synchronisatie voor dit protocol. We nemen aan dat een bepaalde rol non-injectieve synchronisatie claimt. Meestal zullen we ervoor kiezen om deze claim na de laatste read-actie in een protocol te plaatsen. Vervolgens zullen we met behulp van de definities en lemma's laten zien dat deze claim correct is. Voor veel bewijzen maken we gebruik van één of meer deelbewijzen om het overzicht in het bewijs te bewaren. De deelbewijzen zijn aan het einde van iedere paragraaf opgenomen en worden in elke paragraaf opnieuw genummerd. De deelbewijzen worden eerst als een stelling geformuleerd zodat makkelijk terug te zoeken is welke conclusie aan de hand van het deelbewijs getrokken mag worden. Vervolgens wordt het deelbewijs volledig uitgeschreven. We gebruiken Romeinse cijfers om aan vergelijkingen te refereren die we in het bewijs hebben afgeleid. Deze Romeinse cijfers zijn in de rechter kantlijn te opgenomen.

4.2.2 Lemma's

De lemma's die nodig zijn om voor de diverse protocollen het bewijs te leveren worden in deze paragraaf beschreven. De bewijzen van de lemma's worden niet gegeven omdat deze niet direct relevant zijn voor het doel van de opdracht en het ons hiervoor aan tijd ontbreekt.

Het eeerste lemma geeft de mogelijk om af te leiden welke acties een actie in een run voorafgegaan zijn. De functie $role(e)$ drukt uit welke rol de event e uitgevoerd heeft.

Lemma 1 *Event*
Als een event e in een run heeft plaats gevonden dan hebben alle events voorafgaande aan event e in deze run ook plaats gevonden.

$$\forall_{\alpha \in T(p), e \in Event, (rid,\rho,\sigma) \in Inst, i \in N,} : \alpha_i = (rid, \rho, \sigma)(e) \Rightarrow$$
$$\forall_{e' \in Event : e' \prec_{role(e)} e} : \exists_{j \in N : j<i, \sigma' : \sigma' \subseteq \sigma} : \alpha_j = (rid, \rho, \sigma')(e')$$

Het volgende lemma is specifiek voor het model waarvan wij hier gebruiken. Het lemma is een direct gevolg van de keuze voor het Dolev-Yao model. Met behulp van dit lemma kunnen we afleiden dat als een bericht ontvangen wordt dit bericht op hetzelfde moment in de kennis van de intruder voorkomt.

Lemma 2 *Read*
Als een agent een bericht leest op een moment i, dan is dit bericht bekend bij een intruder op moment i.

$$\forall_{\alpha \in T(p), i \in N, (rid,\rho,\sigma) \in Inst, l \in Label, r, r' \in Role, m \in RoleTerm} :$$
$$\alpha_i = (rid, \rho, \sigma, read_l(r, r', m)) \Rightarrow (rid, \rho, \sigma)(m) \in M_i$$

De volgende lemma's (3 t/m 5) geven ons de mogelijkheid te concluderen wanneer er een send-actie plaats gevonden heeft als gevolg van bepaalde intruderkennis.

Lemma 3 *Publieke sleutel*
Stel er komt een term van de vorm $\{t\}_k$ voor in de kennis van de intruder, deze term is niet initieel bekend bij de intruder en de subterm t komt niet voor in de intruderkennis. Dan is er ooit een agent geweest die een bericht heeft gestuurd zodanig dat de term $\{t\}_k$ in dit bericht bevat is.

$$\forall_{\alpha \in T(p), j \in N, (rid,\rho,\sigma) \in Inst, t, k \in RoleTerm} :$$
$$(rid, \rho, \sigma)(\{t\}_k) \lhd M_j \wedge (rid, \rho, \sigma)(\{t\}_k) \in M_0 \wedge (rid, \rho, \sigma)(t) \notin M_j$$
$$\Rightarrow$$
$$\exists_{i \in N, i<j, (rid', \rho', \sigma') \in Inst, l \in Label, r, r' \in Role, m \in RoleTerm} :$$
$$\alpha_i = (rid', \rho', \sigma', send_l(r, r', m)) \wedge (rid, \rho, \sigma)(\{t\}_k) \sqsubseteq (rid', \rho', \sigma')(m)$$

Lemma 4 *Symmetrische of geheime sleutel*
Stel er komt een term van de vorm $\{t\}_k$ voor in de kennis van de intruder, deze term komt niet intieel voor in de kennis van de intruder en de sleutel k is geen element van de intruderkennis. Dan is er ooit een agent geweest die een bericht heeft gestuurd zodanig dat de term $\{t\}_k$ in dit bericht bevat is.

$$\forall_{\alpha \in T(p), j \in N, (rid,\rho,\sigma) \in Inst, t,k \in RoleTerm} :$$
$$(rid,\rho,\sigma)(\{t\}_k) \lhd M_j \wedge (rid,\rho,\sigma)(\{t\}_k) \in M_0 \wedge (rid,\rho,\sigma)(k) \not\lhd M_j$$
$$\Rightarrow$$
$$\exists_{i \in N, i<j, (rid',\rho',\sigma') \in Inst, l \in Label, r,r' \in Role, m \in RoleTerm} :$$
$$\alpha_i = (rid',\rho',\sigma', send_l(r,r',m)) \wedge (rid,\rho,\sigma)(\{t\}_k) \sqsubseteq (rid',\rho',\sigma')(m)$$

Lemma 5 *Basistermen*
Stel de basisterm t komt voor in de kennis van de intruder en deze basisterm komt intieel niet voor in de intruderkennis. Dan is er ooit een agent geweest die een bericht heeft verstuurd zodanig dat de term t bevat is in dit bericht.

$$\forall_{\alpha \in T(p), j \in N, (rid,\rho,\sigma) \in Inst, t \in BasicRoleTerm} :$$
$$(rid,\rho,\sigma)(t) \lhd M_j \wedge (rid,\rho,\sigma)(t) \not\lhd M_0$$
$$\Rightarrow$$
$$\exists_{i \in N, i<j, (rid',\rho',\sigma') \in Inst, l \in Label, r,r' \in Role, m \in RoleTerm} :$$
$$\alpha_i = (rid',\rho',\sigma', send_l(r,r',m)) \wedge (rid,\rho,\sigma)(t) \sqsubseteq (rid',\rho',\sigma')(m)$$

Het volgende lemma geeft de mogelijkheid om gelijkheid van runs af te leiden bij gelijkheid van twee lokale constanten.

Lemma 6 *Gelijke lokale constante*
Als twee instanties van een lokale constante gelijk zijn, zijn de functies die deze instanties vastleggen hetzelfde.

$$\forall_{\alpha \in Tr(p), i,j \in N, (rid,\rho,\sigma),(rid',\rho',\sigma') \in Inst, c \in Const, l,l' \in Label, m,m' \in RoleTerm} :$$
$$\alpha_i = (rid,\rho,\sigma, event_l(m)) \wedge \alpha_j = (rid',\rho',\sigma', event_{l'}(m')) \wedge$$
$$c \sqsubseteq m \wedge c \sqsubseteq m' \wedge (rid,\rho,\sigma)(c) = (rid',\rho',\sigma')(c)$$
$$\Rightarrow$$
$$rid = rid' \wedge \rho = \rho' \wedge ((i \leq j \wedge \sigma \subseteq \sigma') \vee (j \leq i \wedge \sigma' \subseteq \sigma))$$

Het laatste lemma geeft de mogelijkheid om af te leiden dat als een lokale constante voor het eerst verzonden wordt, elke andere actie waarin deze term voorkomt op een later moment plaats heeft gevonden.

Lemma 7 *Eerste voorkomen lokale constante*
De functie FirstSend *geeft voor een rol r en een constante c het label l waarvoor geldt dat de $send_l$ de eerste send is waarbij de constante c wordt verzonden.*

$$\begin{aligned} FirstSend(r, c) \quad \equiv \quad & (\textbf{\textit{min}} \prec_i : l \in Label, r' \in Role, m \in RoleTerm \\ & send_l(r, r', m) \wedge c \sqsubseteq m : l) \end{aligned}$$

Als een lokale constante voor de eerste keer wordt verzonden dan wil dat zeggen dat elk read-actie waarin dezelfde lokale constante voorkomt op een later moment heeft plaats gevonden.

$$\begin{aligned} & \forall_{i,j \in N,(rid,\rho,\sigma),(rid',\rho',\sigma') \in Inst,c \in Const,r,r',s,s' \in Role,m,m' \in RoleTerm} : \\ & \quad \alpha_i = (rid, \rho, \sigma, send_{FirstSend(r,c)}(r, s, m)) \\ & \quad \wedge \alpha_j = (rid', \rho', \sigma', read_l(r', s', m')) \wedge (rid, \rho, \sigma)(c) \sqsubseteq (rid', \rho', \sigma')(m') \\ & \Rightarrow \\ & \quad i < j \end{aligned}$$

4.3 Needham Schroeder Lowe

Het eerste protocol dat we bestuderen, is het Needham Schroeder Lowe protocol ofwel het NSL protocol. Dit protocol hebben we als voorbeeld gebruikt in hoofdstuk 3 waar we de semantiek beschreven hebben. Het NSL protocol bewerkstelligt authenticatie tussen twee partijen. We hebben dit protocol reeds schematisch met een MSC in Figuur 3.1 op pagina 6 beschreven.

4.3.1 Rolbeschrijving

Uit de MSC in Figuur 3.1 kunnen we de rolbeschrijvingen voor het NSL protocol genereren. De rolbeschrijving geeft op een abstracte manier weer hoe een protocolrun wordt uitgevoerd.

Globalen:

$$\begin{aligned}&\mathit{func}\ \ PK(r) : \mathit{PublicKey};\\ &\mathit{func}\ \ PK(i) : \mathit{PublicKey}\end{aligned}$$

Initiatorrol (I)

$$\begin{aligned}&\mathit{Initiator}(i,r) =\\ &\quad \mathit{const}\ \ n_i : \mathit{nonce};\\ &\quad \mathit{var}\ \ X : \mathit{nonce};\\ &\quad \mathit{send}_1(i,r,\{n_i,i\}_{PK(r)}) \cdot\\ &\quad \mathit{read}_2(r,i,\{n_i,X,r\}_{PK(i)}) \cdot\\ &\quad \mathit{send}_3(i,r,\{X\}_{PK(r)})\end{aligned}$$

Responderrol (R)

$$\begin{aligned}&\mathit{Responder}(r,i) =\\ &\quad \mathit{const}\ \ n_r : \mathit{nonce};\\ &\quad \mathit{var}\ \ Y : \mathit{nonce};\\ &\quad \mathit{read}_1(i,r,\{Y,i\}_{PK(r)}) \cdot\\ &\quad \mathit{send}_2(r,i,\{Y,n_r,r\}_{PK(i)}) \cdot\\ &\quad \mathit{read}_3(i,r,\{n_r\}_{PK(r)}) \cdot\\ &\quad \mathit{claim}_4(r,\mathit{ni\text{-}synch})\end{aligned}$$

4.3.2 Bewijs

We zullen bewijzen dat de claim voor non-injectieve synchronisatie van de responderrol correct is. Ten behoeve van de eenvoud van het bewijs is een deel van het bewijs als deelbewijs opgenomen in paragraaf 4.3.3 op pagina 35.

$$
\begin{array}{ll}
 & NI\text{-}SYNCH(p,4) \\
\equiv & \{ \text{ definitie } \} \\
 & \forall_{\alpha \in T(p), k4 \in N, (rid^k, \rho^k, \sigma^{k4}) \in Inst} : \\
 & \quad \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, claim_4(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \Rightarrow \\
 & \quad\quad \exists_{cast:Role \to RID}\ cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k4, prec(p,4), cast)
\end{array}
$$

We veronderstellen dat een instantie van de responderrol non-injectieve synchronisatie claimt en dat de agenten waarmee de responderrol meent te communiceren te vertrouwen zijn. We laten zien dat uit deze aanname non-injectieve synchronisatie is af te leiden.

$$
\begin{array}{ll}
 & \text{Stel: } \alpha \in T(p), k4 \in N, (rid^k, \rho^k, \sigma^{k4}) \in Inst \\
 & \quad \text{zodat: } \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, claim_4(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \\
\Rightarrow & \{ \text{ Lemma 1, kies: } \sigma^{k1}, \sigma^{k2}, \sigma^{k3} \in Inst, \text{ zodat: } \sigma^{k1} \subseteq \sigma^{k2} \subseteq \sigma^{k3} \subseteq \sigma^{k4} \\
 & \quad \text{en kies: } k1, k2, k3 \in N, \text{ zodat: } k1 < k2 < k3 < k4 \ \} \\
 & \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, read_1(i, r, \{Y, i\}_{PK(r)})) \wedge \\
 & \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, send_2(r, i, \{Y, n_r, r\}_{PK(i)})) \wedge \\
 & \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(i, r, \{n_r\}_{PK(r)}))
\end{array}
$$

We hebben een instantie van de responderrol verondersteld en zijn run geconstrueerd. Nu zullen we bekijken of we een instantie van een initiatorrol kunnen afleiden uit het voorkomen van een $read_3$ in de run rid^k.

$$
\begin{array}{ll}
\Rightarrow & \{ \text{ verzwakking } \} \\
 & \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(i, r, \{n_r\}_{PK(r)})) \\
\Rightarrow & \{ \text{ Lemma 2 } \} \\
 & (rid^k, \rho^k, \sigma^{k3})(\{n_r\}_{PK(r)}) \in M_{k3}
\end{array}
$$

$\Rightarrow$ { Deelbewijs 1: $(rid^k, \rho^k, \sigma^{k3})(n_r) \notin M_{k3}$, Lemma 3 }

$\exists_{j3 \in N, j3<k3, (rid^j,\rho^j,\sigma^{j3}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_l(a, b, m))$

$\wedge\ (rid^k, \rho^k, \sigma^{k3})(\{n_r\}_{PK(r)}) \sqsubseteq (rid^j, \rho^j, \sigma^{j3})(m)$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus een $send_3$ plaats gevonden hebben.

$\Rightarrow$ { protocolbeschrijving }

$\exists_{j3 \in N, j3<k3, (rid^j,\rho^j,\sigma^{j3}) \in Inst} : \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(i, r, \{X\}_{PK(r)})) \wedge$

$(rid^j, \rho^j, \sigma^{j3})(\{X\}_{PK(r)}) = (rid^k, \rho^k, \sigma^{k3})(\{n_r\}_{PK(r)})$

$\Rightarrow$ { kies: $j3 \in N, (rid^j, \rho^j, \sigma^{j3}) \in Inst$, zodat: $j3 < k3$

$\wedge\ (rid^j, \rho^j, \sigma^{j3})(\{X\}_{PK(r)}) = (rid^k, \rho^k, \sigma^{k3})(\{n_r\}_{PK(r)})$ } (I)

$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(i, r, \{X\}_{PK(r)}))$

$\Rightarrow$ { Lemma 1, kies: $j1, j2 \in N$, zodat: $j1 < j2 < j3$

en kies: $\sigma^{j1}, \sigma^{j2} \in Inst$, zodat: $\sigma^{j1} \subseteq \sigma^{j2} \subseteq \sigma^{j3}$ }

$\alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(i, r, \{n_i, i\}_{PK(r)})) \wedge$

$\alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(r, i, \{n_i, X, r\}_{PK(i)}))$

We hebben voor de initiatorrol een instantie en een run gevonden. Om synchronisatie te bewijzen moeten we laten zien dat voor alle labels de bijbehorende send- en read-acties van deze twee runs bij elkaar passen. We kunnen nog geen conclusies trekken over synchronisatie voor één van de labels. Daarom bekijken we nu welke conclusies we uit de $read_2$ uit de run rid^j kunnen afleiden.

$\Rightarrow$ { verzwakking }

$\alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(r, i, \{n_i, X, r\}_{PK(i)}))$

$\Rightarrow$ { Lemma 2 }

$(rid^j, \rho^j, \sigma^{j2})(\{n_i, X, r\}_{PK(i)}) \in M_{j2}$

$\Rightarrow$ { $(rid^k, \rho^k, \sigma^{k3})(n_r) = (rid^j, \rho^j, \sigma^{j3})(X)$, Deelbewijs 1, Lemma 3 }

$\exists_{i2 \in N, i2<j2, (rid^{kk},\rho^{kk},\sigma^{i2}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_l(a, b, m))$

$\wedge\ (rid^j, \rho^j, \sigma^{j2})(\{n_i, X, r\}_{PK(i)}) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{i2})(m)$

Op basis van de protocolbeschrijving kan de conclusie getrokken worden dat er maar één mogelijke match voor m is, namelijk die van een $send_2$.

$\Rightarrow$ { protocolbeschrijving en gelijkheid I }

$$\exists_{i2 \in N, i2<j2, (rid^{kk}, \rho^{kk}, \sigma^{i2}) \in Inst} : \alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_2(r, i, \{Y, n_r, r\}_{PK(i)}))$$
$$\wedge (rid^j, \rho^j, \sigma^{j2})(\{n_i, X, r\}_{PK(i)})$$
$$= (rid^{kk}, \rho^{kk}, \sigma^{i2})(\{Y, n_r, r\}_{PK(i)})$$
$$\wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(n_r)$$
$$= (rid^j, \rho^j, \sigma^{j2})(X)$$
$$= (rid^k, \rho^k, \sigma^{k4})(n_r)$$

$\Rightarrow$ { Lemma 6 }

$$(rid^j, \rho^j, \sigma^{j2})(\{n_i, X, r\}_{PK(i)}) = (rid^k, \rho^k, \sigma^{k2})(\{Y, n_r, r\}_{PK(i)}) \quad \text{(II)}$$

$\Rightarrow$ { gelijkheid van berichten }

$$(rid^k, \rho^k, \sigma^{i2})(i, r) = (rid^j, \rho^j, \sigma^{j2})(i, r)$$

Dankzij de laatste gelijkheid (II) weten we nu dat de afbeeldingen van de rollen i en r op agenten voor zowel de initiator- als de responderrol hetzelfde zijn. De identiteit van de responder (r) in bericht 2 is de toevoeging die Gravin Lowe aan het "Needham Schroeder Public Key" protocol [13] heeft gedaan, waardoor het NSL protocol is ontstaan. In tegenstelling tot het NSL protocol is voor het Needham Schroeder protocol een aanval mogelijk door het ontbreken van de identiteit van de responder. We hebben zojuist laten zien dat deze aanval niet meer mogelijk is op het NSL protocol.

We kunnen nu eenvoudig laten zien dat voor alle labels, één label synchronisatie geldt. Per label vatten we de benodigde informatie samen. In sommige gevallen volgt nog een korte afleiding. We beginnen met het laten zien dat voor label 1, de run rid^k en rid^j synchroniseren.

- { herhaling }

$$\alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(i, r, m_1))$$
$$\wedge \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, read_l(i, r, m_2))$$
$$\wedge m_1 = \{n_i, i\}_{PK(r)}$$
$$\wedge m_2 = \{Y, i\}_{PK(r)}$$

$\Rightarrow$ { gelijkheid II, Lemma 7 }

$$(rid^j, \rho^j, \sigma^{j1})(m_1) = (rid^k, \rho^k, \sigma^{k1})(m_2) \wedge j1 < k1$$

$\Rightarrow$ { definitie }

$$1L\text{-}SYNCH(\alpha, k4, 1, rid^j, rid^k)$$

We hebben laten zien dat de send en read met label 1 één label synchroniseren. Vervolgens zullen bekijken of voor label 2 synchronisatie optreedt.

$\circ$ { herhaling }
$\alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, send_2(r, i, m_1))$
$\wedge\ \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(r, i, m_2))$
$\wedge\ m_1 = \{Y, n_r, r\}_{PK(i)}$
$\wedge\ m_2 = \{n_i, X, r\}_{PK(i)}$
$\Rightarrow$ { gelijkheid II, Lemma 7 }
$(rid^k, \rho^k, \sigma^{k2})(m_1) = (rid^j, \rho^j, \sigma^{j1})(m_2) \wedge k2 < j2$
$\Rightarrow$ { definitie }
$1L\text{-}SYNCH(\alpha, k4, 2, rid^k, rid^j)$

We hebben laten zien dat voor label 2 één label synchronisatie geldt. Tot slot zullen we bekijken of de send en read met label 3 synchroniseren.

$\circ$ { herhaling, gelijkheid I }
$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(i, r, m_1))$
$\wedge\ \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(i, r, m_2))$
$\wedge\ m_1 = \{X\}_{PK(r)}$
$\wedge\ m_2 = \{n_r\}_{PK(r)}$
$\wedge\ j3 < k3$
$\wedge\ (rid^j, \rho^j, \sigma^{j3})(m_1) = (rid^k, \rho^k, \sigma^{k3})(m_2)$
$\Rightarrow$ { definitie }
$1L\text{-}SYNCH(\alpha, k4, 3, rid^j, rid^k)$

We hebben tot slot aangetoond dat de send en read met label 3 synchroniseren. Voor alle labels van de acties die de claim-actie voorafgegaan, zijn hebben we laten zien dat ze synchroniseren. Het bewijs kan nu afgerond worden door het invullen van de definities. We kunnen de *cast* functie eenvoudig construeren uit de gevonden resultaten.

$\circ$ { We kiezen de functie *cast* als volgt: }
$cast(i) = rid^j \wedge cast(r) = rid^k$

$\Rightarrow$ { gevonden resultaten en definities }
$\forall_{\ell \in \{1,2,3\}}\ 1L\text{-}SYNCH(\alpha, 4, \ell, cast(sendrole(\ell)), cast(readrole(\ell)))$
$\equiv$ { definitie }
$\exists_{cast:Inst \to RID} : cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, 4, \{1,2,3\}, cast)$
$\Rightarrow$ { definitie }
$NI\text{-}SYNCH(p, 4)$

4.3.3 Deelbewijs

In de vorige paragraaf is gerefereerd aan Deelbewijs 1. Het deelbewijs garandeert de geheimhouding van de nonce n_r uit de run^k en moet in de context van het bewijs in de vorige paragraaf plaatst worden.

Deelbewijs 1

$$(rid^k, \rho^k, \sigma^{k3})(n_r) \notin M_{k3}$$

Het deelbewijs garandeert dat de nonce $(rid^k, \rho^k, \sigma^{k3})(n_r)$ op moment $k3$ niet bekend is bij andere agenten dan slechts de *trusted* agenten. Dit betekent tevens dat op elk moment voorafgaand aan moment $k3$ deze nonce geheim is. Voor het bewijs wordt stilzwijgend aangenomen dat de nonce n_r niet initiëel bekend is bij een intruder. Dit kan vrij eenvoudig hard gemaakt worden omdat deze nonce een lokale constante is, die in de betreffende run is gecreëerd.

Ad deelbewijs 1

Er is bekend dat $rng(\rho^k) \subseteq Agent_T$ uit het bewijs in paragraaf 4.3.2.

$(rid^k, \rho^k, \sigma^{k3})(n_r) \in M_{k3}$
$\Rightarrow$ { Lemma 5 }
$\exists_{i3 \in N, i3<k3, (rid,\rho,\sigma^{i3}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm,}$:
$\alpha_{i3} = (rid, \rho, \sigma^{i3}, send_l(a,b,m)) \wedge ((rid^k, \rho^k, \sigma^{k3})(n_r) \sqsubseteq (rid, \rho, \sigma^{i3})(m)$
$\wedge (rid^k, \rho^k, \sigma^{k3})(n_r) \notin M_{i3}$
$\Rightarrow$ { protocolbeschrijving, mogelijke matchen voor m }
$\exists_{i1<k3, (rid^{jj}, \rho^{jj}, \sigma^{i1})}$:
$\alpha_{i1} = (rid^{jj}, \rho^{jj}, \sigma^{i1})(send_1(i, r, \{n_i, i\}_{PK(r)}))$
$\wedge (rid^{jj}, \rho^{jj}, \sigma^{i1})(n_i) = (rid^k, \rho^k, \sigma^{k3})(n_r)$
$\wedge (rid^{jj}, \rho^{jj}, \sigma^{i1})(PK^{-1}(r)) \in M_{k3}$ (1.1)

$$
\begin{aligned}
&\vee \\
&\exists_{i2\in N, i2<k3, (rid^{kk},\rho^{kk},\sigma^{i2})\in Inst} : \\
&\quad \alpha_{i2} = (rid^{kk},\rho^{kk},\sigma^{i2}, send_2(r,i,\{Y,n_r,r\}_{PK(i)})) \wedge \\
&\quad\quad (((rid^{kk},\rho^{kk},\sigma^{i2})(n_r) = (rid^{k},\rho^{k},\sigma^{k3})(n_r) \\
&\quad\quad\quad \wedge (rid^{kk},\rho^{kk},\sigma^{i2})(PK^{-1}(i)) \in M_{k3}) \qquad (1.2a) \\
&\quad\quad \vee \\
&\quad\quad (((rid^{kk},\rho^{kk},\sigma^{i2})(Y) = (rid^{k},\rho^{k},\sigma^{k3})(n_r) \\
&\quad\quad\quad \wedge (rid^{kk},\rho^{kk},\sigma^{i2})(PK^{-1}(i)) \in M_{k3})) \qquad (1.2b) \\
&\vee \\
&\exists_{i3<k3, (rid^{jj},\rho^{jj},\sigma^{i3})} : \\
&\quad \alpha_{i3} = (rid^{jj},\rho^{jj},\sigma^{i3})(send_3(i,r,\{X\}_{PK(r)})) \\
&\quad\quad \wedge (rid^{jj},\rho^{jj},\sigma^{i3})(X) = (rid^{k},\rho^{k},\sigma^{k3})(n_r) \\
&\quad\quad \wedge (rid^{jj},\rho^{jj},\sigma^{i3})(PK^{-1}(r)) \in M_{k3} \qquad (1.3)
\end{aligned}
$$

De nonce n_r kan uit drie verschillende send-acties geleerd zijn. Uit de $send_2$ kan dit op twee verschillende manieren gebeurd zijn. We zullen alle gevallen hieronder apart bestuderen en laten zien dat elke mogelijkheid van het leren van de nonce uit een send tot een tegenspraak leidt.

We bestuderen eerst de mogelijkheid dat een intruder de nonce n_r heeft geleerd uit een $send_1$.

ad 1.1

$$
\begin{aligned}
&\exists_{i1<k3, (rid^{jj},\rho^{jj},\sigma^{i1})} : \\
&\quad\quad \alpha_{i1} = (rid^{jj},\rho^{jj},\sigma^{i1})(send_1(i,r,\{n_i,i\}_{PK(r)})) \\
&\quad\quad \wedge (rid^{jj},\rho^{jj},\sigma^{i1})(n_i) = (rid^{k},\rho^{k},\sigma^{k3})(n_r) \\
&\quad\quad \wedge (rid^{jj},\rho^{jj},\sigma^{i1})(PK^{-1}(r)) \in M_{k3} \\
\Rightarrow \quad &\{\ n_r = n_i \text{ is syntactisch onmogelijk } \} \\
&\mathit{Tegenspraak}
\end{aligned}
$$

De nonce n_r kan dus niet uit een $send_1$ zijn geleerd. We zullen bekijken of een intruder de nonce geleerd kan hebben uit een $send_2$. Als een intruder deze nonce heeft geleerd uit een send van het type 2 kan dit op twee manieren. De eerste mogelijkheid dat de nonce uit deze send is geleerd, is dat de nonce n_r gelijk is aan de nonce n_r in de $send_2$.

ad 1.2a

$$
\begin{array}{ll}
 & \exists_{i2\in N, i2<k3, (rid^{kk},\rho^{kk},\sigma^{i2})\in Inst} : \\
 & \quad \alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_2(r, i, \{Y, n_r, r\}_{PK(i)})) \\
 & \quad \wedge ((rid^{kk}, \rho^{kk}, \sigma^{i2})(n_r) = (rid^{k}, \rho^{k}, \sigma^{k3})(n_r) \\
 & \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(PK^{-1}(i)) \in M_{k3}) \\
\Rightarrow & \{ \text{ kies: } i2 \in N, (rid^{kk}, \rho^{kk}, \sigma^{i2}) \in Inst \text{ zodat: } i2 < k2 \\
 & \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(PK^{-1}(i)) \in M_{k3} \\
 & \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(n_r) = (rid^{k}, \rho^{k}, \sigma^{k3})(n_r) \} \\
 & \alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_2(r, i, \{Y, n_r, r\}_{PK(i)})) \\
\Rightarrow & \{ \text{ Lemma 6 } \} \\
 & rid^{k} = rid^{kk} \wedge \rho^{k} = \rho^{kk} \\
\Rightarrow & \{ rng(\rho^{k}) \subseteq Agent_T \wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(PK^{-1}(i)) \in M_{i3} \} \\
 & \textit{Tegenspraak}
\end{array}
$$

Vervolgens zullen we de tweede mogelijkheid bekijken. In dit geval is de variabele Y uit de $send_2$ gelijk aan de nonce n_r.

ad 1.2b

$$
\begin{array}{ll}
 & \exists_{i2\in N, i2<k3, (rid^{kk},\rho^{kk},\sigma^{i2})\in Inst} : \\
 & \quad \alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_2(r, i, \{Y, n_r, r\}_{PK(i)})) \\
 & \quad \wedge ((rid^{kk}, \rho^{kk}, \sigma^{i2})(Y) = (rid^{k}, \rho^{k}, \sigma^{k3})(n_r) \\
 & \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(PK^{-1}(i)) \in M_{k3}) \\
\Rightarrow & \{ \text{ kies: } i2 \in N, (rid^{kk}, \rho^{kk}, \sigma^{i3}) \in Inst, \text{ zodat: } i2 < k3 \wedge \\
 & \quad (rid^{kk}, \rho^{kk}, \sigma^{i2})(PK^{-1}(i)) \in M_{k3} \wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(Y) = (rid^{k}, \rho^{k}, \sigma^{k3})(n_r) \} \\
 & \alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i3}, send_2(r, i, \{Y, n_r, r\}_{PK(i)})) \\
\Rightarrow & \{ \text{ Lemma 1 } \} \\
 & \alpha_{i1} = (rid^{kk}, \rho^{kk}, \sigma^{i1}, read_1(i, r, \{Y, i\}_{PK(r)})) \\
\Rightarrow & \{ \text{ Lemma 2 } \} \\
 & (rid^{kk}, \rho^{kk}, \sigma^{i1})(\{Y, i\}_{PK(r)}) \in M_{i1} \\
\Rightarrow & \{ (rid^{kk}, \rho^{kk}, \sigma^{i1})(Y) \notin M_{k1}, \text{ Lemma 3 } \} \\
 & \exists_{h1\in N, h1<i1, (rid^{jj},\rho^{jj},\sigma^{h1})\in Inst} : \alpha_{h1} = (rid^{jj}, \rho^{jj}, \sigma^{h1}, send_l(a, b, m)) \\
 & \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{i1})(\{Y, i\}_{PK(r)}) \sqsubseteq (rid^{jj}, \rho^{jj}, \sigma^{h1})(m)
\end{array}
$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_1$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ kies: } h1 \in N, (rid^{jj}, \rho^{jj}, \sigma^{h1}) \in Inst, \text{ zodat: } h1 < i1 \wedge \\
& \quad (rid^{jj}, \rho^{jj}, \sigma^{h1})(\{n_i, i\}_{PK(r)}) = (rid^{kk}, \rho^{kk}, \sigma^{i1})(\{Y, i\}_{PK(r)}) \} \\
& \alpha_{h1} = (rid^{jj}, \rho^{jj}, \sigma^{h1}, send_1(i, r, \{n_i, i\}_{PK(r)})) \\
\Rightarrow \quad & \{ (rid^{k}, \rho^{k}, \sigma^{k3})(n_r) = (rid^{jj}, \rho^{jj}, \sigma^{h1})(n_i) \} \\
& \textit{Tegenspraak}
\end{aligned}
$$

In beide gevallen levert het leren van de nonce n_r uit een send van het type 2 een tegenspraak op. Het is dus niet mogelijk dat de nonce n_r uit een $send_2$ is geleerd. Tot slot bekijken we of het mogelijk is dat de nonce n_r uit een $send_3$ wordt geleerd.

ad 1.3

$$
\begin{aligned}
& \exists_{i3<k3,(rid^{jj},\rho^{jj},\sigma^{i3})} : \\
& \quad \alpha_{i3} = (rid^{jj}, \rho^{jj}, \sigma^{i3})(send_3(i, r, \{X\}_{PK(r)})) \\
& \quad \wedge (rid^{jj}, \rho^{jj}, \sigma^{i3})(X) = (rid^{k}, \rho^{k}, \sigma^{k3}, n_r) \\
& \quad \wedge (rid^{jj}, \rho^{jj}, \sigma^{i3})(PK^{-1}(r)) \in M_{k3} \\
\Rightarrow \quad & \{ \text{ kies: } i3 \in N, (rid^{jj}, \rho^{jj}, \sigma^{i3}), \text{ zodat: } i3 < k3 \wedge \\
& \quad (rid^{jj}, \rho^{jj}, \sigma^{i3})(X) = (rid^{k}, \rho^{k}, \sigma^{k3})(n_r) \wedge \qquad \text{(III)} \\
& \quad (rid^{i}, \rho^{i}, \sigma^{i3})(PK^{-1}(r)) \in M_{k3} \} \\
& \alpha_{i3} = (rid^{jj}, \rho^{jj}, \sigma^{i3}, send_3(i, r, \{X\}_{PK(r)})) \\
\Rightarrow \quad & \{ \text{ Lemma 1 } \} \\
& \alpha_{i2} = (rid^{jj}, \rho^{jj}, \sigma^{i2}, read_2(r, i, \{n_i, X, r\}_{PK(i)})) \wedge \\
& \alpha_{i1} = (rid^{jj}, \rho^{jj}, \sigma^{i1}, send_1(i, r, \{n_i, i\}_{PK(r)})) \\
\Rightarrow \quad & \{ \text{ Lemma 2 } \} \\
& (rid^{jj}, \rho^{jj}, \sigma^{i2})(\{n_i, X, r\}_{PK(i)}) \in M_{i2} \\
\Rightarrow \quad & \{ (rid^{jj}, \rho^{jj}, \sigma^{i2})(X) = (rid^{k}, \rho^{k}, \sigma^{k3})(n_r) \notin M_{k2}, \text{ Lemma 3 } \} \\
& \exists_{h2\in N, h2<i2, (rid^{jj},\rho^{jj},\sigma^{h1})\in Inst, a,b\in Role, m\in RoleTerm} : \\
& \quad \alpha_{h2} = (rid^{kk}, \rho^{kk}, \sigma^{h2}, send_2(a, b, m)) \\
& \quad \wedge (rid^{jj}, \rho^{jj}, \sigma^{i2})(\{n_i, X, r\}_{PK(i)}) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{h2})(m)
\end{aligned}
$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_2$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ protocolbeschrijving, gelijkheid III } \} \\
& \exists_{h2 \in N, h2<i2, (rid^{jj}, \rho^{jj}, \sigma^{h1}) \in Inst} : \\
& \quad \alpha_{h2} = (rid^{kk}, \rho^{kk}, \sigma^{h2}, send_2(a, b, \{Y, n_r, r\}_{PK(i)})) \\
& \quad \wedge (rid^{jj}, \rho^{jj}, \sigma^{i2})(\{n_i, X, r\}_{PK(i)}) = (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{Y, n_r, r\}_{PK(i)}) \\
& \qquad (rid^{kk}, \rho^{kk}, \sigma^{h2})(n_r) \\
& \quad = (rid^{jj}, \rho^{jj}, \sigma^{i2})(X) \\
& \quad = (rid^{k}, \rho^{k}, \sigma^{k2})(n_r) \\
\Rightarrow \quad & \{ \text{ Lemma 6 } \} \\
& \quad (rid^{jj}, \rho^{jj}, \sigma^{i2})(r) \\
& = (rid^{k}, \rho^{k}, \sigma^{k2})(r) \\
\Rightarrow \quad & \{ (rid^{k}, \rho^{k}, \sigma^{k2})(r) \in Agents_T \wedge (rid^{jj}, \rho^{jj}, \sigma^{i3})(PK^{-1}(r)) \in M_{k3} \} \\
& \textit{Tegenspraak}
\end{aligned}
$$

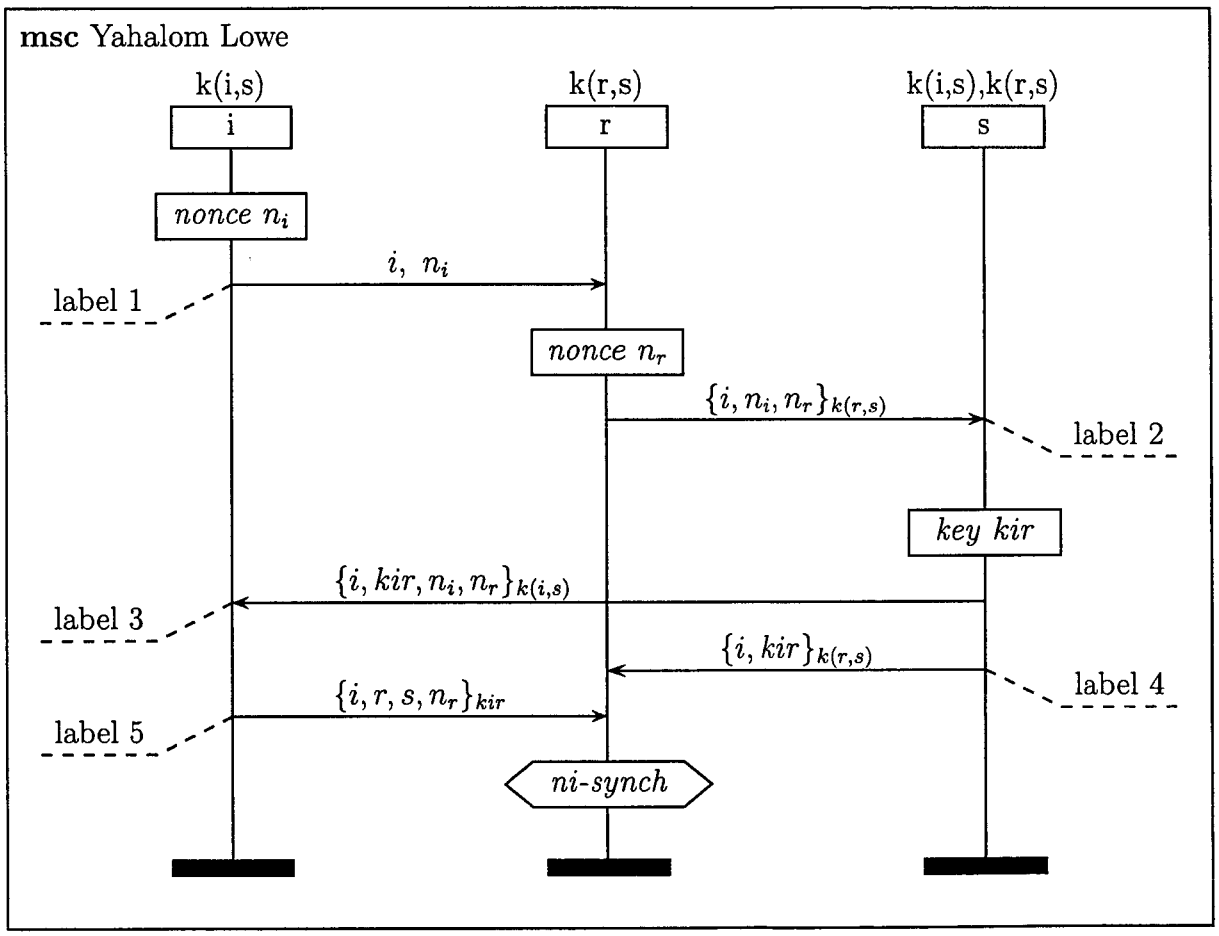
We hebben laten zien dat onder de aanname dat $(rid^k, \rho^k, \sigma^{k3})(n_r) \in M_{k3}$, alle gevallen waardoor de nonce kan zijn geleerd tot een tegenspraak leiden. Daarom kan de conclusie worden getrokken dat $(rid^k, \rho^k, \sigma^{k3})(n_r) \notin M_{k3}$.

4.4 Lowe's modified version of Yahalom

In deze paragraaf bestuderen we het "Lowe's modified version of Yahalom" protocol [11], [14]. Het protocol bewerkstelligt de uitwisseling van een verse symmetrische sleutel gecreëerd door een server. We zullen laten zien dat voor de responder na de laatste readactie non-injectieve synchronisatie geclaimd mag worden.

4.4.1 Protocolbeschrijving

In Figuur 4.1 is schematisch de protocolspecificatie weergeven van "Lowe's modified version of Yahalom". Voor het gemak zal dit protocol het Yahalom Lowe protocol genoemd worden.



Figuur 4.1: Lowe's modified version of Yahalom

4.4.2 Rolbeschrijving

De rollen in het Yahalom Lowe protocol kunnen als volgt beschreven worden:

Globalen:

$$\begin{aligned}&\mathit{func}\ K(i,s) : \mathit{key};\\ &\mathit{func}\ K(r,s) : \mathit{key};\end{aligned}$$

Initiatorrol (I)

$$\begin{aligned}&\mathit{Initiator}(i,r,s) =\\ &\quad \mathit{const}\ n_i : \mathit{nonce};\\ &\quad \mathit{var}\ X : \mathit{nonce};\\ &\quad \mathit{var}\ KX : \mathit{key};\\ &\quad \mathit{send}_1(i,r,(i,n_i)) \cdot\\ &\quad \mathit{read}_3(r,i,\{i,KX,n_i,X\}_{K(i,s)}) \cdot\\ &\quad \mathit{send}_5(i,r,\{i,r,s,X\}_{KX})\end{aligned}$$

Responderrol (R)

$$\begin{aligned}&\mathit{Responder}(r,i,s) =\\ &\quad \mathit{const}\ n_r : \mathit{nonce};\\ &\quad \mathit{var}\ Y : \mathit{nonce};\\ &\quad \mathit{var}\ KY : \mathit{key};\\ &\quad \mathit{read}_1(i,r,(i,Y)) \cdot\\ &\quad \mathit{send}_2(r,s,\{i,Y,n_r\}_{K(r,s)}) \cdot\\ &\quad \mathit{read}_4(s,r,\{i,KY\}_{K(r,s)}) \cdot\\ &\quad \mathit{read}_5(i,r,\{i,r,s,n_r\}_{KY}) \cdot\\ &\quad \mathit{claim}_6(r,\mathit{ni\text{-}synch})\end{aligned}$$

Serverrol (S)

$$\begin{aligned}&\mathit{Server}(i,r,s) =\\ &\quad \mathit{const}\ kir : \mathit{key};\\ &\quad \mathit{var}\ P,Q : \mathit{nonce};\\ &\quad \mathit{read}_2(r,s,\{i,P,Q\}_{K(r,s)}) \cdot\\ &\quad \mathit{send}_3(s,i,\{i,kir,P,Q\}_{K(i,s)}) \cdot\\ &\quad \mathit{send}_4(s,r,\{i,kir\}_{K(r,s)})\end{aligned}$$

4.4.3 Bewijs

We zullen bewijzen dat de claim voor non-injectieve synchronisatie van de responderrol correct is. Ten behoeve van de eenvoud van het bewijs zal een deelbewijs gebruikt worden. Het deelbewijs is terug te vinden in paragraaf 4.4.4 op pagina 48. Indien de responderrol non-injectieve synchronisatie claimt betekent dit het volgende.

$$
\begin{aligned}
& NI\text{-}SYNCH(p,6) \\
\equiv\ & \{\ \text{definitie}\ \} \\
& \forall_{\alpha \in T(p), k6 \in N, (rid^k, \rho^k, \sigma^{k6}) \in Inst}\ : \\
& \quad \alpha_{k6} = (rid^k, \rho^k, \sigma^{k6})(claim_6(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \Rightarrow \\
& \quad\quad \exists_{cast:Role \to RID}\ cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k6, prec(p,6), cast)
\end{aligned}
$$

We veronderstellen dat een instantie van de responderrol non-injectieve synchronisatie claimt en dat de agenten waarmee de responderrol meent te communiceren te vertrouwen zijn. We laten zien dat uit deze aanname non-injectieve synchronisatie is af te leiden.

$$
\begin{aligned}
& \text{Stel:}\ \ \alpha \in T(p), k6 \in N, (rid^k, \rho^k, \sigma^{k6}) \in Inst \\
& \quad \text{zodat:}\ \ \alpha_{k6} = (rid^k, \rho^k, \sigma^{k6}, claim_6(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \\
\Rightarrow\ & \{\ \text{Lemma 1, kies:}\ \ \sigma^{k1}, \sigma^{k2}, \sigma^{k4}, \sigma^{k5}\ \text{zodat:}\ \ \sigma^{k1} \subseteq \sigma^{k2} \subseteq \sigma^{k4} \subseteq \sigma^{k5} \subseteq \sigma^{k6} \\
& \quad \text{en kies:}\ \ k1, k2, k4, k5 \in N\ \text{zodat:}\ \ k1 < k2 < k4 < k5 < k6\ \} \\
& \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, read_1(i, r, (i, Y))) \wedge \\
& \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, send_2(r, s, \{i, Y, n_r\}_{K(r,s)})) \wedge \\
& \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, read_4(s, r, \{i, KY\}_{K(r,s)})) \wedge \\
& \alpha_{k5} = (rid^k, \rho^k, \sigma^{k5}, read_5(i, r, \{i, r, s, n_r\}_{KY}))
\end{aligned}
$$

Nu we een run voor de responderrol hebben geconstrueerd zullen we bekijken of hieruit de runs voor de andere rollen in het protocol af te leiden zijn. Als eerste zullen we bekijken of we uit de $read_4$ in run rid^k een instantie van de serverrol kunnen afleiden. We beschouwen eerst de $read_4$ omdat het bestaan van een instantie van de serverrol van belang is voor deelbewijs 1. Dit deelbewijs is later nodig voor het afleiden van de initiatorrol.

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{verzwakking}\ \} \\
& \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, read_4(s, r, \{i, KY\}_{K(r,s)})) \\
\Rightarrow\ & \{\ \text{Lemma 2}\ \} \\
& (rid^k, \rho^k, \sigma^{k4})(\{i, KY\}_{K(r,s)}) \in M_{k4}
\end{aligned}
$$

$\Rightarrow$ { $(rid^k, \rho^k, \sigma^{k4})(r,s) \subseteq Agent_T \Rightarrow (rid^k, \rho^k, \sigma^{k4})(K(r,s)) \notin M_{k4}$, Lemma 4 }

$\exists_{i4 \in N, i4<k4, (rid^i, \rho^i, \sigma^{i4}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{i4} = (rid^i, \rho^i, \sigma^{i4}, send_l(a,b,m))$

$\wedge\, (rid^k, \rho^k, \sigma^{k4})(\{i, KY\}_{K(r,s)}) \sqsubseteq (rid^i, \rho^i, \sigma^{i4})(m)$

Op basis van de protocolbeschrijving kan de conclusie getrokken worden dat er maar één mogelijke match voor m is, namelijk die van een $send_4$.

$\Rightarrow$ { protocolbeschrijving }

$\exists_{i4 \in N, i4<k4, (rid^i, \rho^i, \sigma^{k4}) \in Inst}$:

$\alpha_{i4} = (rid^i, \rho^i, \sigma^{i4}, send_4(s,r,\{i,kir\}_{K(r,s)}))$

$\wedge\, (rid^k, \rho^k, \sigma^{k4})(\{i, KY\}_{K(r,s)}) = (rid^i, \rho^i, \sigma^{i4})(\{i,kir\}_{K(r,s)})$

$\Rightarrow$ { kies: $i4 \in N, (rid^i, \rho^i, \sigma^{i4}) \in Inst$, zodat: $i4 < k4$

$\wedge\, (rid^k, \rho^k, \sigma^{k4})(\{i, KY\}_{K(r,s)}) = (rid^i, \rho^i, \sigma^{i4})(\{i,kir\}_{K(r,s)})$ } (I)

$\alpha_{i4} = (rid^i, \rho^i, \sigma^{i4}, send_4(s,r,\{i,kir\}_{K(r,s)}))$

$\Rightarrow$ { Lemma 1, kies: $i2, i3 \in N$, zodat: $i2 < i3 < i4$,

en kies: $\sigma^{i2}, \sigma^{i3} \in Inst$, zodat: $\sigma^{i2} \subseteq \sigma^{i3} \subseteq \sigma^{i4}$ }

$\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, read_2(r,s,\{i,P,Q\}_{K(r,s)})) \wedge$

$\alpha_{i3} = (rid^i, \rho^i, \sigma^{i3}, send_3(s,i,\{i,kir,P,Q\}_{K(i,s)}))$

We hebben een instantie en een run voor de serverrol gevonden. We bekijken nu of we uit het voorkomen van een $read_5$ in de run rid^k (responderrun) een instantie voor de initiatorrol kunnen afleiden.

$\Rightarrow$ { verzwakking }

$\alpha_{k5} = (rid^k, \rho^k, \sigma^{k5}, read_5(i,r,\{i,r,s,n_r\}_{KY}))$

$\Rightarrow$ { Lemma 2 }

$(rid^k, \rho^k, \sigma^{k5})(\{i,r,s,n_r\}_{KY}) \in M_{k5}$

$\Rightarrow$ { Deelbewijs 1, Lemma 4 }

$\exists_{j5 \in N, j5<k5, (rid^j, \rho^j, \sigma^{j5}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{j5} = (rid^j, \rho^j, \sigma^{j5}, send_l(a,b,m))$

$\wedge\, (rid^k, \rho^k, \sigma^{k5})(\{i,r,s,n_r\}_{KY}) \sqsubseteq (rid^j, \rho^j, \sigma^{j5})(m)$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_5$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{protocolbeschrijving}\ \} \\
& \exists_{j5\in N, j5<k5, (rid^j,\rho^j,\sigma^{j5})\in Inst}: \\
& \quad \alpha_{j5} = (rid^j, \rho^j, \sigma^{j5}, send_5(i, r, \{i, r, s, X\}_{KX})) \\
& \quad \wedge (rid^j, \rho^j, \sigma^{j5})(\{i, r, s, X\}_{KX}) = (rid^k, \rho^k, \sigma^{k5})(\{i, r, s, n_r\}_{KY}) \\
\Rightarrow\ & \{\ \text{kies: } j5 \in N, (rid^j, \rho^j, \sigma^{j5}) \in Inst, \text{ zodat: } j5 < k5 \\
& \quad \wedge (rid^j, \rho^j, \sigma^{j5})(\{i, r, s, X\}_{KX}) = (rid^k, \rho^k, \sigma^{k5})(\{i, r, s, n_r\}_{KY})\ \} \qquad \text{(II)} \\
& \alpha_{j5} = (rid^j, \rho^j, \sigma^{j5}, send_5(i, r, \{i, r, s, X\}_{KX})) \\
\Rightarrow\ & \{\ \text{Lemma 1, kies: } j1, j3 \in N \text{ zodat: } j1 < j3 < j5, \\
& \quad \text{en kies: } \sigma^{j1}, \sigma^{j3} \in Inst, \text{ zodat: } \sigma^{j1} \subseteq \sigma^{j3} \subseteq \sigma^{j5}\ \} \\
& \alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(i, r, (i, n_i))) \wedge \\
& \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, read_3(s, i, \{i, KX, n_i, X\}_{K(i,s)}))
\end{aligned}
$$

Tot slot hebben we een instantie voor de initiatorrol gevonden. We hebben nu voor de drie rollen die in dit protocol voorkomen een instantie en een bijbehorende run gevonden. We zullen nog moeten laten zien dat de bij elkaar passende send- en readacties dezelfde berichten en agenten bevatten en dat de volgorde waarin ze voorkomen in orde is. Als eerst zullen we laten zien dat de send en read met label 5, synchroniseren.

$$
\begin{aligned}
& \alpha_{j5} = (rid^j, \rho^j, \sigma^{j5}, send_5(m_1)) \\
\wedge\ & \alpha_{k5} = (rid^k, \rho^k, \sigma^{k5}, read_5(m_2)) \\
\wedge\ & (rid^j, \rho^j, \sigma^{j5})(m_1) = (rid^k, \rho^k, \sigma^{k5})(m_2) \\
\wedge\ & m_1 = (i, r, \{i, r, s, X\}_{KX}) \wedge m_2 = (i, r, \{i, r, s, n_r\}_{KY}) \\
\wedge\ & j5 < k5 \\
\Rightarrow\ & \{\ \text{definitie NI-SYNCH}\ \} \\
& 1L\text{-}SYNCH(\alpha, k6, 5, rid^j, rid^k)
\end{aligned}
$$

We hebben eenvoudig laten zien de send en read met label 5 synchroniseren. Vervolgens zullen we bekijken of voor label 4 synchronisatie optreedt.

$$
\begin{aligned}
& \alpha_{i4} = (rid^i, \rho^i, \sigma^{i4}, send_4(m_1)) \\
\wedge\ & \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, read_4(m_2)) \\
\wedge\ & (rid^i, \rho^i, \sigma^{i4})(m_1) = (rid^k, \rho^k, \sigma^{k4})(m_2)
\end{aligned}
$$

$$\wedge\, m_1 = (s, r, \{i, kir\}_{K(r,s)}) \wedge m_2 = (s, r, \{i, KY\}_{K(r,s)})$$
$$\wedge\, j4 < k4$$
$$\Rightarrow \quad \{ \text{ definitie NI-SYNCH } \}$$
$$1L\text{-}SYNCH(\alpha, k6, 4, rid^i, rid^k)$$

Wederom hebben we redelijk eenvoudig laten zien dat de read en send met label 4 synchroniseren. We zullen vervolgens bekijken of dit ook het geval is voor label 3.

$$\Rightarrow \quad \{ \text{ verzwakking } \}$$
$$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, read_3(s, i, \{i, KX, n_i, X\}_{K(i,s)}))$$
$$\Rightarrow \quad \{ \text{ Lemma 2 } \}$$
$$(rid^j, \rho^j, \sigma^{j3})(s, i, \{i, KX, n_i, X\}_{K(i,s)}) \in M_{j3}$$
$$\Rightarrow \quad \{ (rid^k, \rho^k, \sigma^{k5})(i, r, s) = (rid^j, \rho^j, \sigma^{j5})(i, r, s) \Rightarrow$$
$$rng(\rho^j) \subseteq Agent_T \text{ dus } (rid^j, \rho^j, \sigma^{j3})(K(i, s)) \notin M_{j4}, \text{ Lemma 4 } \}$$
$$\exists_{h \in N, h<j3, (rid^{kk}, \rho^{kk}, \sigma^h) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm} :$$
$$\alpha_h = (rid^{ii}, \rho^{ii}, \sigma^h, send_l(a, b, m))$$
$$\wedge\, (rid^j, \rho^j, \sigma^{j3})(\{i, KX, n_i, X\}_{K(i,s)}) \sqsubseteq (rid^{ii}, \rho^{ii}, \sigma^h)(m)$$

Op basis van de protocolbeschrijving en typering kan de conclusie getrokken worden dat er maar één mogelijke match voor m is, namelijk die van een $send_3$.

$$\Rightarrow \quad \{ \text{ protocolbeschrijving en gelijkheden I en II } \}$$
$$\exists_{h \in N, h<j3, (rid^{ii}, \rho^{ii}, \sigma^h) \in Inst,} :$$
$$\alpha_h = (rid^{ii}, \rho^{ii}, \sigma^h, send_3(s, i, \{i, kir, P, Q\}_{K(i,s)}))$$
$$\wedge\, (rid^j, \rho^j, \sigma^{j3})(\{i, KX, n_i, X\}_{K(i,s)}) = (rid^{ii}, \rho^{ii}, \sigma^h)(\{i, kir, P, Q\}_{K(i,s)})$$
$$\wedge\, (rid^{ii}, \rho^{ii}, \sigma^h)(kir)$$
$$= (rid^j, \rho^j, \sigma^{j3})(KX)$$
$$= (rid^k, \rho^k, \sigma^{k5})(KY)$$
$$= (rid^i, \rho^i, \sigma^{i4})(kir)$$
$$\Rightarrow \quad \{ \text{ Lemma 6 } \}$$
$$\alpha_{i3} = (rid^i, \rho^i, \sigma^{i3}, send_3(m_1))$$
$$\wedge\, \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, read_3(m_2))$$
$$\wedge\, (rid^i, \rho^i, \sigma^{i3})(m_1) = (rid^j, \rho^j, \sigma^{j3})(m_2) \qquad \text{(III)}$$
$$\wedge\, m_1 = (s, i, \{i, kir, P, Q\}_{K(i,s)}) \wedge m_2 = (s, i, \{i, KX, n_i, X\}_{K(i,s)})$$
$$\Rightarrow \quad \{ \text{ Lemma 7, } (rid^i, \rho^i, \sigma^{i3})(kir) \sqsubseteq (rid^i, \rho^i, \sigma^{i3})(m_1) \}$$
$$i3 < j3$$

$\Rightarrow$ { definitie NI-SYNCH }
$1L\text{-}SYNCH(\alpha, k6, 3, rid^i, rid^j)$

We zien dat het bewijs voor synchronisatie voor label 3 beduidend ingewikkelder is dan het bewijs voor de labels 4 en 5. Het bewijs voor synchronisatie van label 2 zal van dezelfde omvang zijn als het bewijs voor label 3.

$\Rightarrow$ { verzwakking }
$\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, read_2(r, s, \{i, P, Q\}_{K(r,s)}))$
$\Rightarrow$ { Lemma 2 }
$(rid^i, \rho^i, \sigma^{i2})(\{i, P, Q\}_{K(r,s)}) \in M_{i2}$
$\Rightarrow$ { $(rid^k, \rho^k, \sigma^{k4})(r, s) = (rid^i, \rho^i, \sigma^{i4})(r, s) \wedge rng(\rho^k) \subseteq Agent_T \Rightarrow$
$(rid^i, \rho^i, \sigma^{i4})(K(r, s)) \notin M_{k4}$, Lemma 4 }
$\exists_{h \in N, h<i2, (rid^{kk}, \rho^{kk}, \sigma^h) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:
$\alpha_h = (rid^{kk}, \rho^{kk}, \sigma^h, send_l(a, b, m)) \wedge$
$(rid^i, \rho^i, \sigma^{i2})(\{i, P, Q\}_{K(r,s)}) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^h)(m)$

Op basis van de protocolbeschrijving kan de conclusie worden getrokken dat er maar één mogelijke match voor m is, namelijk die van een $send_2$.

$\Rightarrow$ { protocolbeschrijving en gelijkheden II en III }
$\exists_{h \in N, h<i2, (rid^{kk}, \rho^{kk}, \sigma^h) \in Inst} : \alpha_h = (rid^{kk}, \rho^{kk}, \sigma^h, send_2(r, s, \{i, Y, n_r\}_{K(r,s)})) \wedge$
$(rid^i, \rho^i, \sigma^{i2})(\{i, P, Q\}_{K(r,s)}) = (rid^{kk}, \rho^{kk}, \sigma^h)(\{i, Y, n_r\}_{K(r,s)})$
$\wedge (rid^{kk}, \rho^{kk}, \sigma^h)(n_r)$
$= (rid^i, \rho^i, \sigma^{i2})(Q)$
$= (rid^j, \rho^j, \sigma^{j3})(X)$
$= (rid^k, \rho^k, \sigma^{k5})(n_r)$
$\Rightarrow$ { Lemma 6 }
$\alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, send_2(m_1))$
$\wedge \alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, read_2(m_2))$
$\wedge (rid^k, \rho^k, \sigma^{k2})(m_1) = (rid^i, \rho^i, \sigma^{i2})(m_2)$ (IV)
$\wedge m_1 = (r, s, \{i, Y, n_r\}_{K(r,s)}) \wedge m_2 = (r, s, \{i, P, Q\}_{K(r,s)})$
$\Rightarrow$ { Lemma 7, $(rid^k, \rho^k, \sigma^{k2})(n_r) \sqsubseteq (rid^k, \rho^k, \sigma^{k2})(m_1)$ }
$k2 < i2$

$\Rightarrow$ { definitie NI-SYNCH }
$1L\text{-}SYNCH(\alpha, k6, 2, rid^i, rid^k)$

Nu we ook weten dat de send en read met label 2 synchroniseren, moeten wij alleen nog laten zien dat ook de send en read met label 1 synchroniseren. We hebben voor de meeste termen in de voorafgaande bewijzen al laten zien dat gelijkheid voor de verschillende instanties van roltermen geldt. Hierdoor zal het laatste stuk bewijs redelijk eenvoudig verlopen.

$\Rightarrow$ { gelijkheden III en IV }
$(rid^i, \rho^i, \sigma^{i2})(i, P)$
$= (rid^j, \rho^j, \sigma^{j3})(i, n_i)$
$= (rid^k, \rho^k, \sigma^{k2})(i, Y)$
$\Rightarrow$ { gelijkheid van termen }
$\alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(m_1))$
$\wedge\, \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, read_1(m_2))$
$\wedge\, (rid^j, \rho^j, \sigma^{j1})(m_1) = (rid^k, \rho^k, \sigma^{k1})(m_2)$
$\wedge\, m_1 = (i, r, (i, n_i)) \wedge m_2 = (i, r, (i, Y))$
$\Rightarrow$ { Lemma 7, $(rid^j, \rho^j, \sigma^{j1})(n_i) \sqsubseteq (rid^j, \rho^j, \sigma^{j1})(m_1)$ }
$j1 < k1$
$\Rightarrow$ { definitie NI-SYNCH }
$1L\text{-}SYNCH(\alpha, k6, 1, rid^j, rid^k)$

We hebben tot slot aangetoond dat de send en read met label 1 synchroniseren. Voor alle labels van de acties die de claimactie voorafgegaan zijn, hebben we laten zien dat ze synchroniseren. Het bewijs kan nu afgerond worden door het invullen van de definities. We kunnen de *cast* functie eenvoudig construeren uit de gevonden resultaten.

- { We kiezen de functie *cast* als volgt: }
$cast(i) = rid^j \wedge cast(r) = rid^k \wedge cast(s) = rid^i$

$\Rightarrow$ { gevonden resultaten en definities }
$\forall_{\ell \in \{1,2,3,4,5\}}\ 1L\text{-}SYNCH(\alpha, k6, \ell, cast(sendrole(\ell)), cast(readrole(\ell)))$
$\equiv$ { definitie }
$\exists_{cast:Inst \to RID} : cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k6, \{1,2,3,4,5\}, cast)$

$\Rightarrow$ { definitie }
$NI\text{-}SYNCH(p, 6)$

4.4.4 Deelbewijs

Het volgende deelbewijs is nodig voor het bewijs voor non-injectieve synchronisatie van het Yahalom Lowe protocol.

Deelbewijs 1

$(rid^k, \rho^k, \sigma^{k5})(KY) \notin M_{k5}$

Ad deelbewijs 1

We willen laten zien dat de intruder de sessiesleutel van de responder niet geleerd kan hebben. We hebben onafhankelijk van dit deelbewijs al laten zien dat $(rid^k, \rho^k, \sigma^{k5})(KY) = (rid^i, \rho^i, \sigma^{i4})(kir)$. Omdat kir een lokale constante is, mogen we aannemen dat de bewuste sessiesleutel niet initieel bekend is bij een intruder.

We weten uit gelijkheid I dat $(rid^k, \rho^k, \sigma^{k4})(i, r, s) = (rid^i, \rho^i, \sigma^{i4})(i, r, s)$ omdat we verondersteld hebben dat $rng(\rho^k) \subseteq Agent_T$ mogen we concluderen dat $rng(\rho^i) \subseteq Agent_T$.

$$
\begin{aligned}
& (rid^k, \rho^k, \sigma^{k5})(KY) \in M_{k5} \\
\Rightarrow\ & \{ \text{ Lemma 5 } \} \\
& \exists_{g<k5,(rid^{kk},\rho^{kk},\sigma^g)\in Inst,l\in Label,a,b\in Role,m\in RoleTerm,} : \\
& \quad \alpha_g = (rid^{kk}, \rho^{kk}, \sigma^g, send_l(a, b, m)) \\
& \quad \wedge ((rid^k, \rho^k, \sigma^{k5})(KY) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^g)(m) \\
& \quad \wedge (rid^k, \rho^k, \sigma^{k5})(KY) \not\triangleleft M_g \\
\Rightarrow\ & \{ \text{ protocolbeschrijving, mogelijke matchen voor m } \} \\
& \exists_{g\in N,g<k5,(rid^{ii},\rho^{ii},\sigma^g)\in Inst} : \\
& \quad \alpha_g = (rid^{ii}, \rho^{ii}, \sigma^g, send_3(s, i, \{i, kir, P, Q\}_{K(i,s)})) \\
& \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^g)(kir) = (rid^k, \rho^k, \sigma^{k5})(KY) \\
& \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^g)(K(i, s)) \in M_{k5} \qquad (1.1) \\
& \vee \\
& \quad \alpha_g = (rid^{ii}, \rho^{ii}, \sigma^g, send_4(s, r, \{i, kir\}_{K(r,s)})) \\
& \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^g)(kir) = (rid^k, \rho^k, \sigma^{k5})(KY) \\
& \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^g)(K(r, s)) \in M_{k5} \qquad (1.2)
\end{aligned}
$$

Het bewijs voor de geheimhouding van de sleutels KY valt uiteen in twee gevallen. Omwille van het overzicht zullen de gevallen apart van elkaar beschouwd worden. In beide gevallen wordt aangenomen dat de send die de bewuste sleutel bevat, de eerste send is die deze sleutel bevat. Dit betekent dat de instantie van KY niet in de intruder kennis kan voorkomen voor de bewuste send heeft plaatsgevonden

We bekijken eerst de mogelijkheid dat de intruder de sessiesleutel uit een $send_3$ geleerd heeft.

ad 1.1

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ kies: } g3 < k5 \in N, (rid^{ii}, \rho^{ii}, \sigma^{g3}) \\
& \quad \text{zodat: } (rid^{ii}, \rho^{ii}, \sigma^{g3})(kir) = (rid^{k}, \rho^{k}, \sigma^{k5})(KY) \} \\
& \alpha_{g3} = (rid^{ii}, \rho^{ii}, \sigma^{g3}, send_3(s, i, \{i, kir, P, Q\}_{K(i,s)})) \\
& \wedge (rid^{ii}, \rho^{ii}, \sigma^{g3})(K(i,s)) \in M_{k5} \\
\Rightarrow \quad & \{ \text{ gelijkheid I } \} \\
& (rid^{ii}, \rho^{ii}, \sigma^{g3})(kir) \\
& = (rid^{k}, \rho^{k}, \sigma^{k5})(KY) \\
& = (rid^{i}, \rho^{i}, \sigma^{i3})(kir) \\
\Rightarrow \quad & \{ \text{ Lemma 6 } \} \\
& rid^{i} = rid^{ii} \wedge \rho^{i} = \rho^{ii} \\
\Rightarrow \quad & \{ (rid^{ii}, \rho^{ii}, \sigma^{g3})(K(i,s)) \in M_{k5} \wedge (rid^{i}, \rho^{i}, \sigma^{i5})(i,s) \subseteq Agent_T \} \\
& \textit{Tegenspraak}
\end{aligned}
$$

We hebben laten zien dat de sessiesleutel niet uit een $send_3$ geleerd kan zijn. Blijft over de mogelijkheid dat een intruder deze sleutel uit een $send_4$ geleerd heeft.

ad 1.2

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ kies: } g4 < k5 \in N, (rid^{ii}, \rho^{ii}, \sigma^{g4}) \\
& \quad \text{zodat: } (rid^{ii}, \rho^{ii}, \sigma^{g4})(kir) = (rid^{k}, \rho^{k}, \sigma^{k5})(KY) \} \\
& \alpha_{g4} = (rid^{ii}, \rho^{ii}, \sigma^{g4}, send_4(s, r, \{i, kir\}_{K(r,s)})) \wedge (rid^{ii}, \rho^{ii}, \sigma^{g4})(K(r,s)) \in M_{k5} \\
\Rightarrow \quad & \{ \text{ gelijkheid van termen en gelijkheid I } \} \\
& (rid^{ii}, \rho^{ii}, \sigma^{g4})(kir) \\
& = (rid^{k}, \rho^{k}, \sigma^{k5})(KY) \\
& = (rid^{i}, \rho^{i}, \sigma^{i4})(kir) \\
\Rightarrow \quad & \{ \text{ Lemma 6 } \} \\
& rid^{i} = rid^{ii} \wedge \rho^{i} = \rho^{ii}
\end{aligned}
$$

$\Rightarrow$ $\{ (rid^{ii}, \rho^{ii}, \sigma^{g3})(K(r,s)) \in M_{k5} \wedge (rid^{i}, \rho^{i}, \sigma^{i5})(r,s) \subseteq Agent_T \}$
Tegenspraak

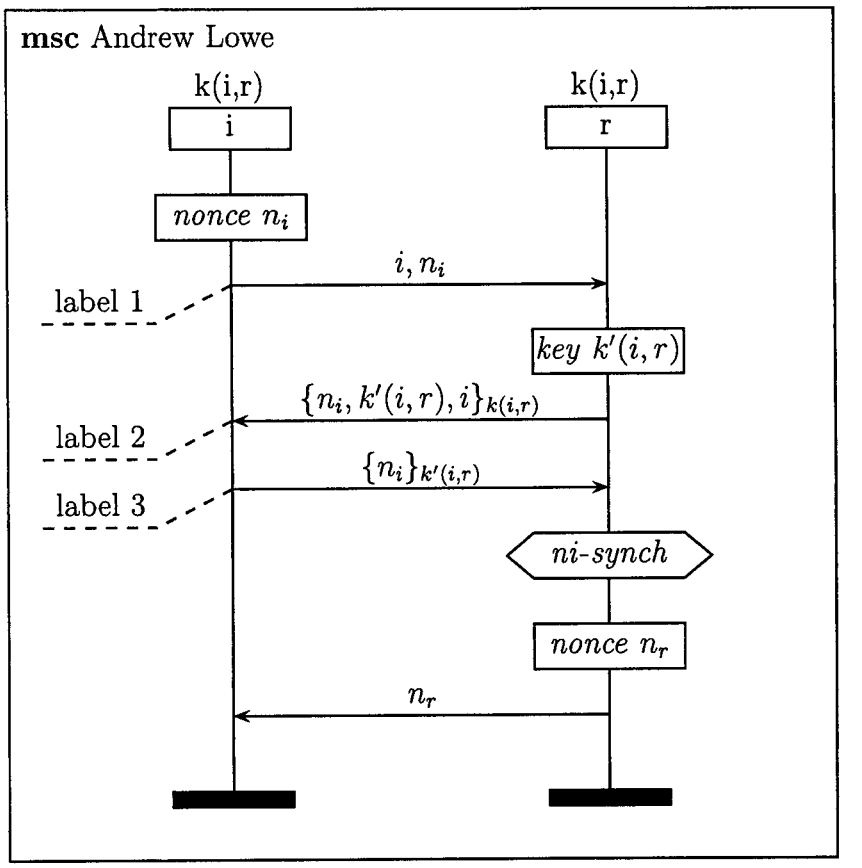
Een intruder kan de sessiesleutel niet geleerd hebben uit een send met label 3 of 4. De aanname dat de sleutel bij een intruder bekend zou zijn leidt dus tot een tegenspraak. We mogen dus aannemen dat de sessiesleutel geheim blijft.

4.5 Lowe modified BAN concrete Andrew Secure RPC

In deze paragraaf bestuderen we het "Lowe modified BAN concrete Andrew Secure RPC" protocol [10], [14]. Het protocol bewerkstelligt de uitwisseling van een verse symmetrische sleutel. We zullen laten zien dat voor de responderrol na de laatste read-actie non-injectieve synchronisatie geclaimd mag worden.

4.5.1 Protocolbeschrijving

In Figuur 4.2 is schematisch de protocolspecificatie weergeven van "Lowe modified BAN concrete Andrew Secure RPC". Voor het gemak zal dit protocol het Andrew Lowe protocol worden genoemd.



Figuur 4.2: Lowe modified BAN concrete Andrew Secure RPC

4.5.2 Rolbeschrijving

De rollen in het Andrew Lowe protocol kunnen als volgt beschreven worden:

Globalen:

$$\mathit{func}\ \ K(i,r) : \mathit{key};$$

Initiatorrol (I)

$$\begin{aligned}
&\mathit{Initiator}(i,r,s) = \\
&\quad \mathit{const}\ \ n_i : \mathit{nonce}; \\
&\quad \mathit{var}\ \ X : \mathit{nonce}; \\
&\quad \mathit{var}\ \ KX : \mathit{key}; \\
&\quad \mathit{send}_1(i,r,(i,n_i)) \cdot \\
&\quad \mathit{read}_2(r,i,\{n_i, KX, i\}_{K(i,r)}) \cdot \\
&\quad \mathit{send}_3(i,r,\{n_i\}_{KX}) \cdot \\
&\quad \mathit{read}_5(r,i,(X))
\end{aligned}$$

Responderrol (R)

$$\begin{aligned}
&\mathit{Responder}(r,i,s) = \\
&\quad \mathit{const}\ \ n_r : \mathit{nonce}; \\
&\quad \mathit{const}\ \ k'ir : \mathit{key}; \\
&\quad \mathit{var}\ \ Y : \mathit{nonce}; \\
&\quad \mathit{read}_1(i,r,(i,Y)) \cdot \\
&\quad \mathit{send}_2(r,i,\{Y, k'ir, i\}_{K(i,r)}) \cdot \\
&\quad \mathit{read}_3(i,r,\{Y\}_{k'ir}) \cdot \\
&\quad \mathit{claim}_4(r, \mathit{ni\text{-}synch}) \\
&\quad \mathit{send}_5(r,i,n_r) \cdot
\end{aligned}$$

4.5.3 Bewijs

We zullen bewijzen dat de claim voor non-injectieve synchronisatie van de responderrol correct is. Ten behoeve van de eenvoud van het bewijs zal een deelbewijs gebruikt worden. Het deelbewijs is terug te vinden in paragraaf 4.5.4 op pagina 56. Indien de responderrol non-injectieve synchronisatie claimt betekent dit het volgende.

$$
\begin{aligned}
& NI\text{-}SYNCH(p,4) \\
\equiv\ & \{\ \text{definitie}\ \} \\
& \forall_{\alpha \in T(p), k4 \in N, (rid^k, \rho^k, \sigma^{k4}) \in Inst:} \\
& \quad \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4})(claim_4(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \Rightarrow \\
& \quad\quad \exists_{cast:Role \to RID}\ cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k4, prec(p,4), cast)
\end{aligned}
$$

We veronderstellen dat een instantie van de responderrol non-injectieve synchronisatie claimt en dat de agenten waarmee de responder meent te communiceren te vertrouwen zijn. We laten zien dat uit deze aanname non-injectieve synchronisatie is af te leiden.

$$
\begin{aligned}
& \text{Stel:}\ \ \alpha \in T(p), k4 \in N, (rid^k, \rho^k, \sigma^{k4}) \in Inst \\
& \quad \text{zodat:}\ \ \alpha_{k4} = (rid^k, \rho^k, \sigma^{k5}, claim_4(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \\
\Rightarrow\ & \{\ \text{Lemma 1, Kies:}\ \ \sigma^{k1}, \sigma^{k2}, \sigma^{k3},\ \text{zodat:}\ \ \sigma^{k1} \subseteq \sigma^{k2} \subseteq \sigma^{k3} \subseteq \sigma^{k4} \\
& \quad \text{en kies:}\ \ k1, k2, k3 \in N\ \text{zodat:}\ \ k1 < k2 < k3 < k4\ \} \\
& \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, read_1(i, r, (i, Y))) \wedge \\
& \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, send_2(r, i, \{Y, k'ir, i\}_{K(i,r)})) \wedge \\
& \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(i, r, \{Y\}_{k'ir}))
\end{aligned}
$$

We hebben een instantie en run van de responderrol geconstrueerd. Nu zullen we bekijken of we uit de $read_3$ in run rid^k een instantie van de initiatorrol kunnen afleiden.

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{verzwakking}\ \} \\
& \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(i, r, \{Y\}_{k'ir})) \\
\Rightarrow\ & \{\ \text{Lemma 2}\ \} \\
& (rid^k, \rho^k, \sigma^{k3})(\{Y\}_{k'ir}) \in M_{k3} \\
\Rightarrow\ & \{\ \text{Deelbewijs 1, Lemma 4}\ \} \\
& \exists_{j3 \in N, j3 < k3, (rid^j, \rho^j, \sigma^{j3}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}: \\
& \quad \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_l(a, b, m)) \\
& \quad \wedge (rid^k, \rho^k, \sigma^{k3})(\{Y\}_{k'ir}) \sqsubseteq (rid^j, \rho^j, \sigma^{j3})(m)
\end{aligned}
$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_3$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{protocolbeschrijving}\ \} \\
& \exists_{j3\in N, j3<k3, (rid^j,\rho^j,\sigma^{j3})\in Inst} : \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(r, s, \{n_i\}_{KX})) \wedge \\
& (rid^k, \rho^k, \sigma^{k3})(\{Y\}_{k'ir}) = (rid^j, \rho^j, \sigma^{j3})(\{n_i\}_{KX}) \\
\Rightarrow\ & \{\ \text{kies: } j3 \in N, (rid^j, \rho^j, \sigma^{j3}) \in Inst, \text{ zodat: } j3 < k3 \wedge \\
& (rid^k, \rho^k, \sigma^{k3})(\{Y\}_{k'ir}) = (rid^j, \rho^j, \sigma^{j3})(\{n_i\}_{KX})\ \} \qquad \text{(I)} \\
& \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(i, r, \{n_i\}_{KX})) \\
\Rightarrow\ & \{\ \text{Lemma 1, kies: } \sigma^{j1}, \sigma^{j2}, \text{ zodat: } \sigma^{j1} \subseteq \sigma^{j2} \subseteq \sigma^{j3} \\
& \text{en kies: } j1, j2 \in N, \text{ zodat: } j1 < j2 < j3\ \} \\
& \alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(i, r, (i, n_i))) \wedge \\
& \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(r, i, \{n_i, KX, i\}_{K(i,r)}))
\end{aligned}
$$

We hebben een instantie voor de initiatorrol en een bijbehorende run gevonden. We zullen nu proberen af te leiden dat het bericht dat de initiator ontvangt middels een $read_2$, afkomstig is uit de run rid^k van de responderrol.

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{verzwakking}\ \} \\
& \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(r, i, \{n_i, KX, i\}_{K(i,r)})) \\
\Rightarrow\ & \{\ \text{Lemma 2}\ \} \\
& (rid^j, \rho^j, \sigma^{j2})(\{n_i, KX, i\}_{K(i,r)}) \in M_{j2} \\
\Rightarrow\ & \{\ \text{gelijkheid I, Deelbewijs 1}\ \} \\
& (rid^j, \rho^j, \sigma^{j2})(K(i, r)) \notin M_{j2} \\
\Rightarrow\ & \{\ \text{Lemma 4}\ \} \\
& \exists_{i2\in N, i2<j2, (rid^{kk},\rho^{kk},\sigma^{i2})\in Inst, l\in Label, a,b\in Role, m\in RoleTerm} : \\
& \alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_l(a, b, m)) \wedge \\
& (rid^j, \rho^j, \sigma^{j2})(\{n_i, KX, i\}_{K(i,r)}) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{i2})(m)
\end{aligned}
$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_2$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{protocolbeschrijving, gelijkheid I}\ \} \\
& \exists_{i2\in N, i2<j2, (rid^{kk},\rho^{kk},\sigma^{i2})\in Inst} : \\
& \alpha_i = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_2(r, i, \{Y, k'ir, i\}_{K(i,r)})) \\
& \wedge (rid^j, \rho^j, \sigma^{j2})(\{n_i, KX, i\}_{K(i,r)}) = (rid^{kk}, \rho^{kk}, \sigma^{i2})(\{Y, k'ir, i\}_{K(i,r)}) \\
& \wedge (rid^k, \rho^k, \sigma^{k3})(k'ir) = (rid^j, \rho^j, \sigma^{j3})(KX) = (rid^{kk}, \rho^{kk}, \sigma^{i2})(k'ir)
\end{aligned}
$$

$\Rightarrow$ { Lemma 6 }

$(rid^k, \rho^k, \sigma^{k2})(\{Y, k'ir, i\}_{K(i,r)}) = (rid^j, \rho^j, \sigma^{j2})(\{n_i, KX, i\}_{K(i,r)})$ (II)

We weten nu genoeg om eenvoudig te laten zien dat voor alle labels synchronisatie optreedt. We zullen als eerste laten zien dat de $send_3$ en $read_3$ in respectievelijk run rid^j en rid^k bij elkaar horen.

- { herhalingen en gelijkheid I }

$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(m_1))$
$\wedge\ \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(m_2))$
$\wedge\ (rid^j, \rho^j, \sigma^{j3})(m_1) = (rid^k, \rho^k, \sigma^{k3})(m_2)$
$\wedge\ m_1 = (i, r, \{n_i\}_{KX}) \wedge m_2 = (i, r, \{Y\}_{k'ir})$
$\wedge\ j3 < k3$

$\Rightarrow$ { definitie NI-SYNCH }

$1L\text{-}SYNCH(\alpha, k4, 3, rid^j, rid^k)$

We hebben laten zien dat voor label 3 synchronisatie optreedt. Vervolgens bekijken we of de $send_2$ en $read_2$ uit respectievelijk run rid^k en rid^j synchroniseren.

- { herhalingen en gelijkheid II }

$\alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, send_2(m_1))$
$\wedge\ \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(m_2))$
$\wedge\ (rid^k, \rho^k, \sigma^{k2})(m_1) = (rid^j, \rho^j, \sigma^{j2})(m_2)$
$\wedge\ m_1 = (r, i, \{Y, k'ir, i\}_{K(i,r)}) \wedge m_2 = (r, i, \{n_i, KX, i\}_{K(i,r)})$

$\Rightarrow$ { Lemma 7, $(rid^k, \rho^k, \sigma^{k2})(k'ir)$ }

$k2 < j2$

$\Rightarrow$ { definitie NI-SYNCH }

$1L\text{-}SYNCH(\alpha, k4, 2, rid^k, rid^j)$

Voor label 2 kunnen we concluderen dat er synchronisatie optreedt. Tot slot moeten we laten zien dat voor label 1 synchronisatie optreedt.

$\circ$ { herhalingen en gelijkheid II }

$\alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(m_1))$

$\wedge\ \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, read_1(m_2))$

$\wedge\ (rid^j, \rho^j, \sigma^{j1})(m_1) = (rid^k, \rho^k, \sigma^{k1})(m_2)$

$\wedge\ m_1 = (i, r, (i, n_i)) \wedge m_2 = (i, r, (i, Y))$

$\Rightarrow$ { Lemma 7, $(rid^j, \rho^j, \sigma^{j1})(n_i)$ }

$j1 < k1$

$\Rightarrow$ { definitie NI-SYNCH }

$1L\text{-}SYNCH(\alpha, k4, 1, rid^j, rid^k)$

We hebben tot slot aangetoond dat de send en read met label 1 synchroniseren. Voor alle labels van de acties die de claim-actie voorafgegaan zijn, hebben we laten zien dat ze synchroniseren. Het bewijs kan nu afgerond worden door het invullen van de definities. We kunnen de *cast* functie eenvoudig construeren uit de gevonden resultaten.

$\circ$ { We kiezen de functie *cast* als volgt: }

$cast(i) = rid^j \wedge cast(r) = rid^k$

$\Rightarrow$ { gevonden resultaten en definities }

$\forall_{\ell \in \{1,2,3\}}\ 1L\text{-}SYNCH(\alpha, k4, \ell, cast(sendrole(\ell)), cast(readrole(\ell)))$

$\equiv$ { *Definitie* }

$\exists_{cast:Inst \to RID} : cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k4, \{1,2,3\}, cast)$

$\Rightarrow$ { *Definitie* }

$NI\text{-}SYNCH(p, k4)$

4.5.4 Deelbewijs

We hebben in het bewijs in de vorige paragraaf aangenomen dat de verse sleutel gemaakt door de responderrol geheim blijft. We zullen nu het bewijs hiervoor gaan leveren.

Deelbewijs 1

$(rid^k, \rho^k, \sigma^{k3})(k'ir) \notin M_{k3}$

Ad deelbewijs 1

Aangezien $(rid^k, \rho^k, \sigma^{k3})(k'ir)$ een constante van de run rid^k is, kan deze niet door een intruder gemaakt zijn zodanig dat deze hetzelfde is. Dit betekent dat er aangenomen mag worden dat $(rid^k, \rho^k, \sigma^{k3})(k'ir)$ niet initieel in de kennis een intruder aanwezig is.

$$
\begin{array}{ll}
 & (rid^k, \rho^k, \sigma^{k3})(k'ir) \in M_{k3} \\
\Rightarrow & \{ \text{ Lemma 5 } \} \\
 & \exists_{g<k5,(rid,\rho,\sigma^g)\in Inst,l\in Label,a,b\in Role,m\in RoleTerm} : \\
 & \quad \alpha_g = (rid, \rho, \sigma^g, send_l(a,b,m)) \\
 & \quad \wedge ((rid^k, \rho^k, \sigma^{k3})(k'ir) \sqsubseteq (rid, \rho, \sigma^g)(m) \\
 & \quad \wedge (rid^k, \rho^k, \sigma^{k3})(k'ir) \not\triangleleft M_g \\
\Rightarrow & \{ \text{ protocolbeschrijving, mogelijke matchen } \} \\
 & \exists_{g\in N,g<k5,(rid^{kk},\rho^{kk},\sigma^g)\in Inst} : \\
 & \quad \alpha_g = (rid^{kk}, \rho^{kk}, \sigma^g, send_2(r,i,\{Y,k'ir,r\}_{K(i,s)})) \\
 & \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^g)(k'ir) = (rid^k, \rho^k, \sigma^{k5})(k'ir) \\
 & \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^g)(K(i,s)) \in M_{k3} \\
\Rightarrow & \{ \text{ Lemma 6 } \} \\
 & (rid^k, \rho^k, \sigma^{k2})(K(i,s)) \in M_{k3} \\
\Rightarrow & \{ \ rng(\rho^k) \subseteq Agent_T \ \} \\
 & \textit{Tegenspraak}
\end{array}
$$

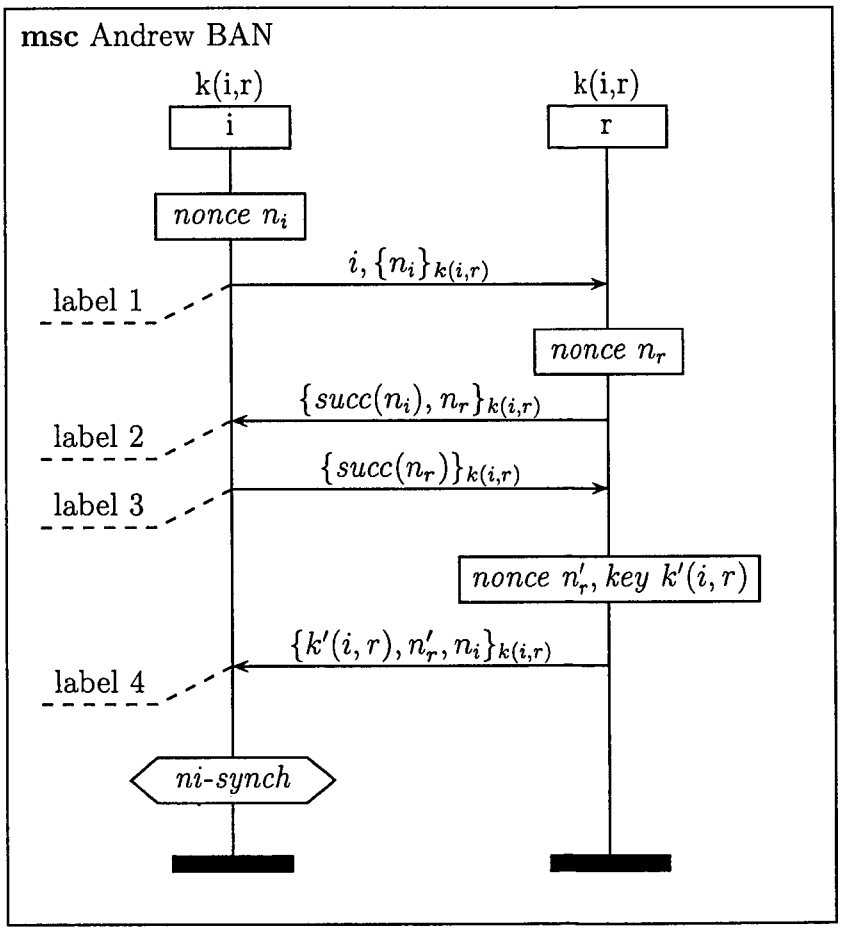
We hebben laten zien dat de aanname dat de sleutel $(rid^k, \rho^k, \sigma^{k3})(k'ir)$ bij een intruder bekend is tot een tegenspraak lijdt. Dus mogen we aannemen dat deze sleutel onbekend is voor een intruder.

4.6 BAN modified Andrew Secure RPC

In deze paragraaf bestuderen we het "BAN modified Andrew Secure RPC" [2], [14] protocol. Het protocol bewerkstelligt de uitwisseling van een verse symmetrische sleutel. We zullen in tegenstelling tot de vorige bewijzen voor dit protocol laten zien dat voor de initiatorrol non-injectieve synchronisatie geclaimd mag worden. We hebben deze keuze gemaakt omdat de laatste read-actie in het protocol in de rolbeschrijving van de initiatorrol voorkomt.

4.6.1 Protocolbeschrijving

In Figuur 4.3 is schematisch de protocolspecificatie weergeven van "BAN modified Andrew Secure RPC". Voor het gemak zal dit protocol het Andrew BAN protocol worden genoemd.



Figuur 4.3: BAN modified Andrew Secure RPC

4.6.2 Rolbeschrijving

De rollen in het Andrew BAN protocol kunnen als volgt beschreven worden:

Globalen:

$$func\ K(i,r) : key;$$

Initiatorrol (I)

$$Initiator(i,r,s) =$$
$$const\ n_i : nonce;$$
$$var\ X : nonce;$$
$$var\ X' : nonce;$$
$$var\ KX : key;$$
$$send_1(i,r,(i,\{n_i\}_{K(i,r)})) \cdot$$
$$read_2(r,i,\{succ(n_i),X\}_{K(i,r)}) \cdot$$
$$send_3(i,r,\{succ(X)\}_{K(i,r)}) \cdot$$
$$read_4(r,i,\{KX,X',n_i\}_{K(i,r)}) \cdot$$
$$claim_5(i,ni\text{-}synch)$$

Responderrol (R)

$$Responder(r,i,s) =$$
$$const\ n_r : nonce;$$
$$const\ k'ir : key;$$
$$var\ Y : nonce;$$
$$read_1(i,r,(i,\{Y\}_{K(i,r)})) \cdot$$
$$send_2(r,i,\{succ(Y),n_r\}_{K(i,r)}) \cdot$$
$$read_3(i,r,\{succ(n_r\}_{K(i,r)}) \cdot$$
$$send_4(r,i,\{K'(i,r),n'_r,Y\}_{K(i,r)})$$

4.6.3 Bewijs

We zullen bewijzen dat de claim voor non-injectieve synchronisatie van de initiatorrol correct is. Voor dit bewijs maken wij geen gebruik van deelbewijzen. Indien de initiatorrol non-injectieve synchronisatie claimt betekent dit het volgende.

$$
\begin{aligned}
& NI\text{-}SYNCH(p,5) \\
\equiv\ & \{\ \text{definitie}\ \} \\
& \forall_{\alpha \in T(p), k5 \in N, (rid^k, \rho^k, \sigma^{k5}) \in Inst}\ : \\
& \quad \alpha_{k5} = (rid^k, \rho^k, \sigma^{k5})(claim_5(i, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \Rightarrow \\
& \quad\quad \exists_{cast:Role \to RID}\ cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k5, prec(p,5), cast)
\end{aligned}
$$

We veronderstellen dat een instantie van de initiatorrol non-injectieve synchronisatie claimt en dat de agenten waarmee de initiator meent te communiceren te vertrouwen zijn. We laten zien dat uit deze aanname non-injectieve synchronisatie is af te leiden.

$$
\begin{aligned}
& \text{Stel:}\ \ \alpha \in T(p), k5 \in N, (rid^k, \rho^k, \sigma^{k5}) \in Inst \\
& \quad \text{zodat:}\ \ \alpha_{k5} = (rid^k, \rho^k, \sigma^{k5}, claim_5(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \\
\Rightarrow\ & \{\ \text{Lemma 1, kies:}\ \ \sigma^{k1}, \sigma^{k2}, \sigma^{k3}, \sigma^{k4},\ \text{zodat:}\ \ \sigma^{k1} \subseteq \sigma^{k2} \subseteq \sigma^{k3} \subseteq \sigma^{k4} \subseteq \sigma^{k5} \\
& \quad \text{en kies:}\ \ k1, k2, k3, k4 \in N,\ \text{zodat:}\ \ k1 < k2 < k3 < k4 < k5\ \} \\
& \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, send_1(i, r, (i, \{n_i\}_{K(i,r)}))) \wedge \\
& \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, read_2(r, i, \{succ(n_i), X\}_{K(i,r)})) \wedge \\
& \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, send_3(i, r, \{succ(X)\}_{K(i,r)})) \wedge \\
& \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, read_4(r, i, \{KX, X', n_i\}_{K(i,r)}))
\end{aligned}
$$

We hebben een instantie en run voor de initiatorrol geconstrueerd. Nu zullen we bekijken of we uit de $read_4$ in run rid^k een instantie voor de responderrol kunnen afleiden.

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{verzwakking}\ \} \\
& (rid^k, \rho^k, \sigma^{k4})(\{KX, X', n_i\}_{K(i,r)}) \in M_{k4} \\
\Rightarrow\ & \{\ \text{Lemma 4,}\ \ rng(\rho^k) \subseteq Agent_T\ \} \\
& \exists_{j4 \in N, j4 < k4, (rid^j, \rho^j, \sigma^{j4}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}\ : \\
& \quad \alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_l(a, b, m)) \wedge \\
& \quad (rid^k, \rho^k, \sigma^{k4})(\{KX, X', n_i\}_{K(i,r)}) \sqsubseteq (rid^j, \rho^j, \sigma^{j4})(m)
\end{aligned}
$$

Volgens de protocol beschrijving is er maar een mogelijke match en er moet dus wel een $send_4$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ protocolbeschrijving } \} \\
& \exists_{j4 \in N, j4 < k4, (rid^j, \rho^j, \sigma^{j4}) \in Inst} : \alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_4(r, i, \{kir, n'_r, Y\}_{K(i,r)})) \\
& \quad \wedge (rid^k, \rho^k, \sigma^{k4})(\{KX, X', n_i\}_{K(i,r)}) = (rid^j, \rho^j, \sigma^{j4})(\{kir, n'_r, Y\}_{K(i,r)}) \\
\Rightarrow \quad & \{ \text{ kies: } (rid^j, \rho^j, \sigma^{j4}) \in Inst \text{ zodat:} \\
& \quad (rid^k, \rho^k, \sigma^{k4})(\{KX, X', n_i\}_{K(i,r)}) = (rid^j, \rho^j, \sigma^{j4})(\{kir, n'_r, Y\}_{K(i,r)}) \ \} \qquad \text{(I)} \\
& \alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_4(i, r, \{kir, n'_r, Y\}_{K(i,r)})) \\
\Rightarrow \quad & \{ \text{ Lemma 1, kies: } \sigma^{j1}, \sigma^{j2}, \sigma^{j3}, \text{ zodat: } \sigma^{j1} \subseteq \sigma^{j2} \subseteq \sigma^{j3} \subseteq \sigma^{j4} \\
& \quad \text{en kies: } j1, j2, j3 \in N, \text{ zodat: } j1 < j2 < j3 < j4 \ \} \\
& \alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, read_1(i, r, (i, \{Y\}_{K(i,r)}))) \wedge \\
& \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, send_2(r, i, \{succ(Y), n_r\}_{K(i,r)})) \wedge \\
& \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, read_3(i, r, \{succ(n_r)\}_{K(i,r)}))
\end{aligned}
$$

We hebben een instantie en run voor de responderrol gevonden. We weten dat de $send_4$ en $read_4$ uit respectievelijk run rid^j en rid^k overeenkomen. We zullen vervolgens bekijken of we ook kunnen laten zien dat de overige acties uit beide runs overeenkomen.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ verzwakking } \} \\
& \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, read_3(i, r, \{succ(n_r)\}_{K(i,r)})) \\
\Rightarrow \quad & \{ \text{ Lemma 2 } \} \\
& (rid^j, \rho^j, \sigma^{j3})(\{succ(n_r)\}_{K(i,r)}) \in M_{j3} \\
\Rightarrow \quad & \{ \text{ Lemma 4 } \} \\
& \exists_{i3 \in N, i3 < j3, (rid^{kk}, \rho^{kk}, \sigma^{i3}), l \in Label, a, b \in Role, m \in RoleTerm} : \\
& \quad \alpha_{i3} = (rid^{kk}, \rho^{kk}, \sigma^{i3}, send_l(a, b, m)) \\
& \quad \wedge (rid^j, \rho^j, \sigma^{j3})(\{succ(n_r)\}_{K(i,r)}) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{i3})(m) \\
\Rightarrow \quad & \{ \text{ mogelijke matchen } \} \\
& \exists_{i1 \in N, i1 < j3, (rid^{jj}, \rho^{jj}, \sigma^{i3}) \in Inst} : \\
& \quad \alpha_{i1} = (rid^{jj}, \rho^{jj}, \sigma^{i1}, send_1(i, r, (i, \{n_i\}_{K(i,r)}))) \wedge \\
& \quad (rid^j, \rho^j, \sigma^{j3})(\{succ(n_r)\}_{K(i,r)}) = (rid^{kk}, \rho^{kk}, \sigma^{i1})(\{n_i\}_{K(i,r)}) \\
& \vee \\
& \exists_{i3 \in N, i3 < j3, (rid^{kk}, \rho^{kk}, \sigma^{i3}) \in Inst} : \\
& \quad \alpha_{i3} = (rid^{kk}, \rho^{kk}, \sigma^{i3}, send_3(r, i, \{succ(X)\}_{K(i,r)})) \wedge \\
& \quad (rid^j, \rho^j, \sigma^{j3})(\{succ(n_r)\}_{K(i,r)}) = (rid^{kk}, \rho^{kk}, \sigma^{i3})(\{succ(X)\}_{K(i,r)})
\end{aligned}
$$

Indien de runs gelijk zijn waarin n_i en $succ(n_r)$ gemaakt zijn, is het syntactische onmogelijk dat deze waarde gelijk aan elkaar zijn. Indien de runs verschillend zijn, is de kans dat deze waarde gelijk zijn te verwaarlozen omdat wij van nonces verwachten dat de waarde random zijn en dus onvoorspelbaar ver uit elkaar liggen.

$\Rightarrow$ { protocolbeschrijving }

$\exists_{i3 \in N, i3<j3, (rid^{kk}, \rho^{kk}, \sigma^{i3})} : \alpha_{i3} = (rid^{kk}, \rho^{kk}, \sigma^{i3}, send_3(r, i, \{succ(X)\}_{K(i,r)})) \wedge$
$(rid^{j}, \rho^{j}, \sigma^{j3})(\{succ(n_r)\}_{K(i,r)}) = (rid^{kk}, \rho^{kk}, \sigma^{i3})(\{succ(X)\}_{K(i,r)})$

$\Rightarrow$ { kies: $(rid^{kk}, \rho^{kk}, \sigma^{i3}) \in Inst$, zodat:
$(rid^{j}, \rho^{j}, \sigma^{j3})(\{succ(n_r)\}_{K(i,r)}) = (rid^{kk}, \rho^{kk}, \sigma^{i3})(\{succ(X)\}_{K(i,r)})$ } (II)

$\alpha_{i3} = (rid^{kk}, \rho^{kk}, \sigma^{i3}, send_3(i, r, \{succ(X)\}_{K(i,r)}))$

$\Rightarrow$ { Lemma 1, kies: σ^{i1}, σ^{i2} zodat: $\sigma^{i1} \subseteq \sigma^{i2} \subseteq \sigma^{i3}$
en kies: $i1, i2 \in N$, zodat: $i1 < i2 < i3$ }

$\alpha_{i1} = (rid^{kk}, \rho^{kk}, \sigma^{k1}, send_1(i, r, (i, \{n_i\}_{K(i,r)}))) \wedge$
$\alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{k2}, read_2(r, i, \{succ(n_i), X\}_{K(i,r)}))$

We hebben een instantie en run gevonden die net als de run rid^k de rol van initiator lijkt te vervullen. We zullen proberen te laten zien dat de runs rid^k en rid^{kk} dezelfde zijn.

$\Rightarrow$ { verzwakking }

$\alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{k2}, read_2(r, i, \{succ(n_i), X\}_{K(i,r)}))$

$\Rightarrow$ { Lemma 2 }

$(rid^{kk}, \rho^{kk}, \sigma^{i2})(\{succ(n_i), X\}_{K(i,r)}) \in M_{i2}$

$\Rightarrow$ { Lemma 4 }

$\exists_{g2 \in N, g2<i2, (rid^{jj}, \rho^{jj}, \sigma^{g2}), l \in Label, a,b \in Role, m \in RoleTerm}$:
$\alpha_{g2} = (rid^{jj}, \rho^{jj}, \sigma^{g2}, send_l(a, b, m)) \wedge$
$(rid^{kk}, \rho^{kk}, \sigma^{i2})(\{succ(n_i), X\}_{K(i,r)}) \sqsubseteq (rid^{jj}, \rho^{jj}, \sigma^{g2})(m)$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_2$ plaats hebben gevonden.

$\Rightarrow$ { protocolbeschrijving, gelijkheid II }

$\exists_{g2 \in N, g2<i2, (rid^{jj}, \rho^{jj}, \sigma^{g2})}$:
$\alpha_{g2} = (rid^{jj}, \rho^{jj}, \sigma^{g2}, send_2(r, i, \{succ(Y), n_r\}_{K(i,r)}))$
$\wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(\{succ(n_i), X\}_{K(i,r)}) = (rid^{jj}, \rho^{jj}, \sigma^{g2})(\{succ(Y), n_r\}_{K(i,r)})$
$\wedge (rid^{j}, \rho^{j}, \sigma^{j3})(succ(n_r)) = (rid^{kk}, \rho^{kk}, \sigma^{i3})(succ(X))$

$$
\begin{aligned}
\Rightarrow \quad & \{\ \textit{Lemma}\ 6\ \} \\
& (rid^{kk}, \rho^{kk}, \sigma^{i2})(\{succ(n_i), X\}_{K(i,r)}) = (rid^{j}, \rho^{j}, \sigma^{j2})(\{succ(Y), n_r\}_{K(i,r)}) \\
\Rightarrow \quad & \{\ \text{gelijkheid I}\ \} \\
& (rid^{kk}, \rho^{kk}, \sigma^{i2})(n_i) = (rid^{j}, \rho^{j}, \sigma^{j2})(Y) = (rid^{k}, \rho^{k}, \sigma^{k4})(n_i) \\
\Rightarrow \quad & \{\ \textit{Lemma}\ 6\ \} \\
& rid^{k} = rid^{kk} \wedge \rho^{k} = \rho^{kk} \wedge (\sigma^{k} \sqsubseteq \sigma^{kk} \vee \sigma^{kk} \sqsubseteq \sigma^{k})
\end{aligned}
$$

We hebben laten zien dat de runs rid^k en rid^{kk} gelijk aan elkaar zijn. Het is nu voor alle labels vrij eenvoudig af te leiden dat er één label synchronisatie optreedt tussen de runs rid^k en rid^j. We zullen als eerste laten zien dat de $send_4$ en $read_4$ in respectievelijk run rid^j en rid^k synchroniseren.

$$
\begin{aligned}
& \alpha_{j4} = (rid^{j}, \rho^{j}, \sigma^{j4}, send_4(m_1)) \\
& \wedge\, \alpha_{k4} = (rid^{k}, \rho^{k}, \sigma^{k4}, read_4(m_2)) \\
& \wedge\, (rid^{j}, \rho^{j}, \sigma^{j4})(m_1) = (rid^{k}, \rho^{k}, \sigma^{k4})(m_2) \\
& \wedge\, m_1 = (r, i, \{kir, n'_r, Y\}_{K(i,r)}) \wedge m_2 = (r, i, \{KX, X', n_i\}_{K(i,r)}) \\
& \wedge\, j4 < k5 \\
\Rightarrow \quad & \{\ \text{definitie NI-SYNCH}\ \} \\
& \textit{1L-SYNCH}(\alpha, k5, 4, rid^{j}, rid^{k})
\end{aligned}
$$

We hebben laten zien dat voor label 3 synchronisatie optreedt. Vervolgens zullen we bekijken of we voor de $send_3$ en $read_3$ in respectievelijk run rid^k en rid^j synchronisatie kunnen afleiden.

$$
\begin{aligned}
& \alpha_{k3} = (rid^{k}, \rho^{k}, \sigma^{k3}, send_3(m_1)) \\
& \wedge\, \alpha_{j3} = (rid^{j}, \rho^{j}, \sigma^{j3}, read_3(m_2)) \\
& \wedge\, (rid^{k}, \rho^{k}, \sigma^{k3})(m_1) = (rid^{j}, \rho^{j}, \sigma^{j3})(m_2) \\
& \wedge\, m_1 = (i, r, \{succ(X)\}_{K(i,r)}) \wedge m_2 = (i, r, \{succ(n_r)\}_{K(i,r)}) \\
\Rightarrow \quad & \{\ (rid^{j}, \rho^{j}, \sigma^{j3})(K(i,r)) \notin M_{j3}\ \} \\
& k3 < j3 \\
\Rightarrow \quad & \{\ \text{definitie NI-SYNCH}\ \} \\
& \textit{1L-SYNCH}(\alpha, k5, 3, rid^{k}, rid^{j})
\end{aligned}
$$

Voor label 3 hebben we laten zien dat de runs rid^k en rid^j synchroniseren. Vervolgens zullen we bekijken of voor label 2 synchronisatie optreedt.

$$
\begin{aligned}
& \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, send_2(m_1)) \\
& \wedge\ \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, read_2(m_2)) \\
& \wedge\ (rid^j, \rho^j, \sigma^{j2})(m_1) = (rid^k, \rho^k, \sigma^{k2})(m_2) \\
& \wedge\ m_1 = (r, i, \{succ(Y), n_r\}_{K(i,r)}) \wedge m_2 = (r, i, \{succ(n_i), X\}_{K(i,r)}) \\
\Rightarrow\ & \{\ \text{Lemma 7}\ \} \\
& j2 < k2 \\
\Rightarrow\ & \{\ \text{definitie NI-SYNCH}\ \} \\
& \mathit{1L\text{-}SYNCH}(\alpha, k5, 2, rid^j, rid^k)
\end{aligned}
$$

We hebben laten zien dat voor label 2 synchronisatie tussen de runs rid^k en rid^j optreedt. Tenslotte laten we zien dat dit voor label 1 ook het geval is.

$$
\begin{aligned}
& \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, send_1(m_1)) \\
& \wedge\ \alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, read_1(m_2)) \\
& \wedge\ (rid^k, \rho^k, \sigma^{k1})(m_1) = (rid^j, \rho^j, \sigma^{j1})(m_2) \\
& \wedge\ m_1 = (i, r, (i, \{n_i\}_{K(i,r)})) \wedge m_2 = (i, r, (i, \{Y\}_{K(i,r)})) \\
\Rightarrow\ & \{\ \text{Lemma 7}\ \} \\
& k1 < j1 \\
\Rightarrow\ & \{\ \text{Definitie NI-SYNCH}\ \} \\
& \mathit{1L\text{-}SYNCH}(\alpha, k4, 1, rid^k, rid^j)
\end{aligned}
$$

We hebben tot slot aangetoond dat de send en read met label 1 synchroniseren. Voor alle labels van de acties die de claim-actie voorafgegaan zijn, hebben we laten zien dat ze synchroniseren. Het bewijs kan nu afgerond worden door het invullen van de definities. We kunnen de *cast* functie eenvoudig construeren uit de gevonden resultaten.

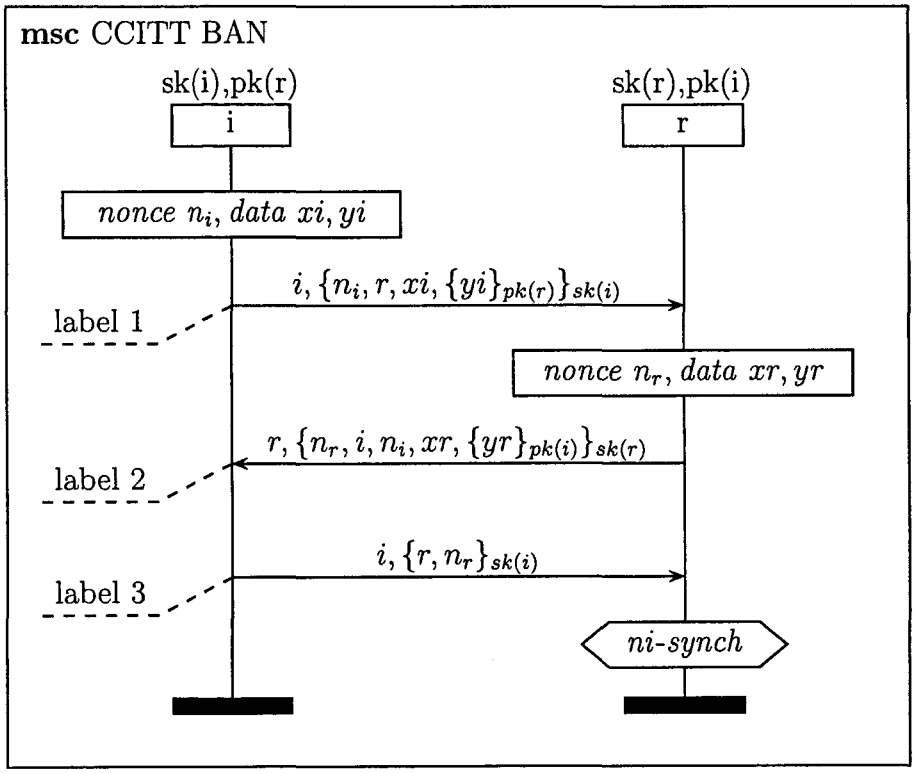
$$
\begin{aligned}
\circ\ & \{\ \text{We kiezen de functie } \mathit{cast} \text{ als volgt:}\ \} \\
& \mathit{cast}(i) = rid^k \wedge \mathit{cast}(r) = rid^j \\
\Rightarrow\ & \{\ \text{gevonden resultaten en definities}\ \} \\
& \forall_{\ell \in \{1,2,3,4\}}\ \mathit{1L\text{-}SYNCH}(\alpha, k5, \ell, \mathit{cast}(\mathit{sendrole}(\ell)), \mathit{cast}(\mathit{readrole}(\ell)))\ ) \\
\equiv\ & \{\ \text{definitie}\ \} \\
& \exists_{\mathit{cast}:\mathit{Inst}\to\mathit{RID}} : \mathit{cast}(r) = rid^k \wedge \mathit{ML\text{-}SYNCH}(\alpha, k5, \{1,2,3,4\}, \mathit{cast}) \\
\Rightarrow\ & \{\ \text{definitie}\ \} \\
& \mathit{NI\text{-}SYNCH}(p, 5)
\end{aligned}
$$

4.7 BAN modified version of CCITT X.509

In deze paragraaf bestuderen we het "BAN modified version of CCITT X.509" [2], [14] protocol. Het protocol bewerkstelligt het oversturen van vertrouwelijk informatie tussen twee agenten door gebruik te maken van asymmetrische encryptie.

4.7.1 Protocolbeschrijving

In Figuur 4.4 is schematisch de protocolspecificatie weergeven van "BAN modified version of CCITT X.509". Voor het gemak zal dit protocol het CCITT BAN protocol genoemd worden.



Figuur 4.4: BAN modified version of CCITT X.509

4.7.2 Rolbeschrijving

De rollen in het CCITT BAN protocol kunnen als volgt beschreven worden:

Globalen:

$$\begin{aligned}&\mathit{func}\ \ PK(r) : \mathit{PublicKey};\\ &\mathit{func}\ \ PK(i) : \mathit{PublicKey}\end{aligned}$$

Initiatorrol (I)

$$\begin{aligned}&\mathit{Initiator}(i, r, s) =\\ &\quad \mathit{const}\ \ n_i : \mathit{nonce};\\ &\quad \mathit{const}\ \ xi, yi : \mathit{data};\\ &\quad \mathit{var}\ \ P : \mathit{nonce};\\ &\quad \mathit{var}\ \ PX, PY : \mathit{data};\\ &\quad \mathit{send}_1(i, r, (i, \{n_i, r, xi, \{yi\}_{PK(r)}\}_{SK(i)})) \cdot\\ &\quad \mathit{read}_2(r, i, (r, \{P, i, n_i, PX, \{PY\}_{PK(i)}\}_{SK(r)})) \cdot\\ &\quad \mathit{send}_3(i, r, (i, \{r, P\}_{SK(i)})) \cdot\end{aligned}$$

Responder rol (R)

$$\begin{aligned}&\mathit{Responder}(r, i, s) =\\ &\quad \mathit{const}\ \ n_r : \mathit{nonce};\\ &\quad \mathit{const}\ \ xr, yr : \mathit{data};\\ &\quad \mathit{var}\ \ Q : \mathit{nonce};\\ &\quad \mathit{var}\ \ QX, QY : \mathit{data};\\ &\quad \mathit{read}_1(i, r, (i, \{Q, r, QX, \{QY\}_{PK(r)}\}_{SK(i)})) \cdot\\ &\quad \mathit{send}_2(r, i, (r, \{n_r, i, Q, xr, \{yr\}_{PK(i)}\}_{SK(r)})) \cdot\\ &\quad \mathit{read}_3(i, r, (i, \{r, n_r\}_{SK(i)})) \cdot\\ &\quad \mathit{claim}_4(r, \mathit{ni\text{-}synch})\end{aligned}$$

4.7.3 Bewijs

We zullen bewijzen dat de claim voor non-injectieve synchronisatie van de responderrol correct is. Voor dit bewijs maken wij geen gebruik van deelbewijzen. Indien de responderrol non-injectieve synchronisatie claimt betekent dit het volgende.

$$
\begin{aligned}
& NI\text{-}SYNCH(p,4) \\
\equiv\ & \{\ \text{definitie}\ \} \\
& \forall_{\alpha \in T(p), k4 \in N, (rid^k, \rho^k, \sigma^{k4}) \in Inst} : \\
& \quad \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4})(claim_4(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \Rightarrow \\
& \quad\quad \exists_{cast:Role \to RID}\ cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k4, prec(p,4), cast)
\end{aligned}
$$

We veronderstellen dat een instantie van de responderrol non-injectieve synchronisatie claimt en dat de agenten waarmee de responder meent te communiceren te vertrouwen zijn. We laten zien dat uit deze aanname non-injectieve synchronisatie is af te leiden.

$$
\begin{aligned}
& \text{Stel:}\ \ \alpha \in T(p), k4 \in N, (rid^k, \rho^k, \sigma^{k4}) \in Inst \\
& \quad \text{zodat:}\ \ \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, claim_4(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \\
\Rightarrow\ & \{\ \text{Lemma 1, kies:}\ \ \sigma^{k1}, \sigma^{k2}, \sigma^{k3},\ \text{zodat:}\ \ \sigma^{k1} \subseteq \sigma^{k2} \subseteq \sigma^{k3} \subseteq \sigma^{k4} \\
& \quad\ \text{en kies:}\ \ k1, k2, k3 \in N,\ \text{zodat:}\ \ k1 < k2 < k3 < k4\ \} \\
& \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, read_1(i, r, (i, \{Q, r, QX, \{QY\}_{PK(r)}\}_{SK(i)}))) \wedge \\
& \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, send_2(r, i, (r, \{n_r, i, Q, xr, \{yr\}_{PK(i)}\}_{SK(r)}))) \wedge \\
& \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(i, r, (i, \{r, n_r\}_{SK(i)})))
\end{aligned}
$$

We hebben een instantie en run voor de responderrol geconstrueerd. Nu zullen we bekijken of we uit de $read_3$ in run rid^k een instantie voor de initiatorrol kunnen afleiden.

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{verzwakking}\ \} \\
& \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(i, r, (i, \{r, n_r\}_{SK(i)}))) \\
\Rightarrow\ & \{\ \text{Lemma 2}\ \} \\
& (rid^k, \rho^k, \sigma^{k3})(i, \{r, n_r\}_{SK(i)}) \in M_{k3} \\
\Rightarrow\ & \{\ \text{Lemma 4}\ \} \\
& \exists_{j3 \in N, j3 < k3, (rid^j, \rho^j, \sigma^{j3}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm} : \\
& \quad \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_l(a,b,m)) \\
& \quad \wedge (rid^k, \rho^k, \sigma^{k3})(\{r, n_r\}_{SK(i)}) \sqsubseteq (rid^j, \rho^j, \sigma^{j3})(m)
\end{aligned}
$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_3$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ protocolbeschrijving } \} \\
& \exists_{j3 \in N, j3 < k3, (rid^j, \rho^j, \sigma^{j3}) \in Inst} : \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(r, i, (i, \{r, P\}_{SK(i)}))) \\
& \quad \wedge (rid^k, \rho^k, \sigma^{k3})(\{r, n_r\}_{SK(i)}) = (rid^j, \rho^j, \sigma^{j3})(\{r, P\}_{SK(i)}) \\
\Rightarrow \quad & \{ \text{ kies: } j3 \in N, (rid^j, \rho^j, \sigma^{j3}) \in Inst, \text{ zodat: } j3 < k3 \wedge \\
& \quad (rid^k, \rho^k, \sigma^{k3})(\{r, n_r\}_{SK(i)}) = (rid^j, \rho^j, \sigma^{j3})(\{r, P\}_{SK(i)}) \ \} \qquad \text{(I)} \\
& \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(i, r, (i, \{r, P\}_{SK(i)}))) \\
\Rightarrow \quad & \{ \text{ Lemma 1, kies: } \sigma^{j1}, \sigma^{j2}, \text{ zodat: } \sigma^{j1} \subseteq \sigma^{j2} \subseteq \sigma^{j3} \\
& \quad \text{en kies: } j1, j2 \in N \text{ zodat: } j1 < j2 < j3 \ \} \\
& \alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(r, i, (i, \{n_i, r, xi, \{yi\}_{PK(r)}\}_{SK(i)}))) \wedge \\
& \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(i, r, (r, \{P, i, n_i, PX, \{PY\}_{PK(i)}\}_{SK(r)})))
\end{aligned}
$$

We hebben een instantie voor de initiatorrol en een bijbehorende run gevonden. We weten dat de $send_3$ en $read_3$ uit respectievelijk run rid^j en rid^k overeenkomen. Vervolgens zullen we proberen te laten zien dat de overige acties uit deze twee runs ook overeenkomen.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ verzwakking } \} \\
& \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(i, r, (r, \{P, i, n_i, PX, \{PY\}_{PK(i)}\}_{SK(r)}))) \\
\Rightarrow \quad & \{ \text{ Lemma 2 } \} \\
& (rid^j, \rho^j, \sigma^{j2})((r, \{P, i, n_i, PX, \{PY\}_{PK(i)}\}_{SK(r)})) \in M_{j2} \\
\Rightarrow \quad & \{ \text{ Lemma 4, } (rid^j, \rho^j, \sigma^{j2})(r) \in Agent_T \text{ vanwege gelijkheid I } \} \\
& \exists_{i2 \in N, i2 < j2, (rid^{kk}, \rho^{kk}, \sigma^{i2}) \in Inst, l \in Label, a, b \in Role, m \in RoleTerm} : \\
& \quad \alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_l(a, b, m)) \\
& \quad \wedge (rid^j, \rho^j, \sigma^{j2})(\{P, i, n_i, PX, \{PY\}_{PK(i)}\}_{SK(r)}) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{i2})(m)
\end{aligned}
$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_2$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ protocolbeschrijving en gelijkheid I } \} \\
& \exists_{i2 \in N i2 < j2, (rid^{kk}, \rho^{kk}, \sigma^{i2}) \in Inst} : \\
& \quad \alpha_{i2} = (rid^{kk}, \rho^{kk}, \sigma^{i2}, send_2(r, i, (i, \{n_r, i, Q, xr, \{yr\}_{PK(i)}\}_{SK(r)}))) \\
& \quad \wedge (rid^j, \rho^j, \sigma^{j2})(\{P, i, n_i, PX, \{PY\}_{PK(i)}\}_{SK(r)}) \\
& \quad = (rid^{kk}, \rho^{kk}, \sigma^{i2})(\{n_r, i, Q, xr, \{yr\}_{PK(i)}\}_{SK(r)})
\end{aligned}
$$

$$
\begin{aligned}
& \wedge (rid^{kk}, \rho^{kk}, \sigma^{i2})(n_r) \\
& = (rid^{j}, \rho^{j}, \sigma^{j2})(P) \\
& = (rid^{k}, \rho^{k}, \sigma^{k3})(n_r) \\
\Rightarrow \quad & \{ \text{ Lemma 6 } \} \\
& (rid^{j}, \rho^{j}, \sigma^{j2})(\{P, i, n_i, PX, \{PY\}_{PK(i)}\}_{SK(r)}) \\
& = (rid^{k}, \rho^{k}, \sigma^{i2})(\{n_r, i, Q, xr, \{yr\}_{PK(i)}\}_{SK(r)}) \qquad \text{(II)}
\end{aligned}
$$

We weten nu dat de send en read met label 2 hetzelfde bericht respectievelijk versturen en ontvangen. Vervolgens zullen we bekijken of dit ook het geval is voor de send en read met label 1.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ verzwakking } \} \\
& \alpha_{k1} = (rid^{k}, \rho^{k}, \sigma^{k1}, read_1(i, r, (i, \{Q, r, QX, \{QY\}_{PK(r)}\}_{SK(i)}))) \\
\Rightarrow \quad & \{ \text{ Lemma 2 } \} \\
& (rid^{k}, \rho^{k}, \sigma^{k1})((i, \{Q, r, QX, \{QY\}_{PK(r)}\}_{SK(i)})) \in M_{k1} \\
\Rightarrow \quad & \{ \text{ Lemma 4, } (rid^{k}, \rho^{k}, \sigma^{k1})(Q) = (rid^{j}, \rho^{j}, \sigma^{j2})(n_i) \text{ vanwege II } \} \\
& \exists_{i1 \in N, i1 < j1, (rid^{jj}, \rho^{jj}, \sigma^{i1}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm} : \\
& \quad \alpha_{i1} = (rid^{kk}, \rho^{kk}, \sigma^{i1}, send_l(a, b, m)) \\
& \quad \wedge (rid^{k}, \rho^{k}, \sigma^{k1})(\{Q, r, QX, \{QY\}_{PK(r)}\}_{SK(i)}) \sqsubseteq (rid^{jj}, \rho^{jj}, \sigma^{i1})(m)
\end{aligned}
$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_1$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ protocolbeschrijving en gelijkheid II } \} \\
& \exists_{i1 \in N i1 < j1, (rid^{kk}, \rho^{kk}, \sigma^{i1}) \in Inst} : \\
& \quad \alpha_{i1} = (rid^{kk}, \rho^{kk}, \sigma^{i1}, send_1(r, i, (r, \{n_i, r, xi, \{yi\}_{PK(r)}\}_{SK(i)}))) \\
& \quad \wedge (rid^{k}, \rho^{k}, \sigma^{k1})(\{Q, r, QX, \{QY\}_{PK(r)}\}_{SK(i)}) \\
& \quad = (rid^{jj}, \rho^{jj}, \sigma^{i1})(\{n_i, r, xi, \{yi\}_{PK(r)}\}_{SK(i)}) \\
& \quad \wedge (rid^{jj}, \rho^{jj}, \sigma^{j1})(n_i) \\
& \quad = (rid^{k}, \rho^{k}, \sigma^{k1})(Q) \\
& \quad = (rid^{j}, \rho^{j}, \sigma^{j2})(n_i) \\
\Rightarrow \quad & \{ \text{ Lemma 6 } \} \\
& (rid^{k}, \rho^{k}, \sigma^{i1})(\{Q, r, QX, \{QY\}_{PK(r)}\}_{SK(i)}) \\
& = (rid^{j}, \rho^{j}, \sigma^{j1})(\{n_i, r, xi, \{yi\}_{PK(r)}\}_{SK(i)})
\end{aligned}
$$

We weten genoeg om eenvoudig te laten zien dat voor alle labels synchronisatie optreedt. We zullen als eerste laten zien dat de $send_3$ en $read_3$ in respectievelijk run rid^j en rid^k synchroniseren.

$$
\begin{aligned}
& \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(m_1)) \\
\wedge\ & \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(m_2)) \\
\wedge\ & (rid^j, \rho^j, \sigma^{j3})(m_1) = (rid^k, \rho^k, \sigma^{k3})(m_2) \\
\wedge\ & m_1 = (i, r, (r, \{r, P\}_{SK(i)})) \wedge m_2 = (i, r, (i, \{r, n_r\}_{SK(i)})) \\
\wedge\ & j3 < k3 \\
\Rightarrow\ & \{\ \text{definitie NI-SYNCH}\ \} \\
& 1L\text{-}SYNCH(\alpha, k4, 3, rid^j, rid^k)
\end{aligned}
$$

We hebben laten zien dat voor label 3 synchronisatie optreedt. Vervolgens zullen we bekijken of we voor de $send_2$ en $read_2$ in respectievelijk run rid^k en rid^j synchronisatie optreedt.

$$
\begin{aligned}
& \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, send_2(m_1)) \\
\wedge\ & \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(m_2)) \\
\wedge\ & (rid^k, \rho^k, \sigma^{k2})(m_1) = (rid^j, \rho^j, \sigma^{j2})(m_2) \\
\wedge\ & m_1 = (r, i, (r, \{n_r, i, Q, xr, \{yr\}_{PK(i)}\}_{SK(r)})) \\
\wedge\ & m_2 = (r, i, (r, \{P, i, n_i, PX, \{PY\}_{PK(i)}\}_{SK(r)})) \\
\Rightarrow\ & \{\ \text{Lemma 7, } (rid^k, \rho^k, \sigma^{k2})(n_r)\ \} \\
& k2 < j2 \\
\Rightarrow\ & \{\ \text{definitie NI-SYNCH}\ \} \\
& 1L\text{-}SYNCH(\alpha, k4, 2, rid^k, rid^j)
\end{aligned}
$$

Voor label 2 kunnen we concluderen dat er synchronisatie optreedt. Tot slot moeten we laten zien dat voor label 1 synchronisatie optreedt.

$$
\begin{aligned}
& \alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(m_1)) \\
\wedge\ & \alpha_{k1} = (rid^k, \rho^k, \sigma^{k1}, read_1(m_2)) \\
\wedge\ & (rid^j, \rho^j, \sigma^{j1})(m_1) = (rid^k, \rho^k, \sigma^{k1})(m_2) \\
\wedge\ & m_1 = (i, r, (i, \{n_i, r, xi, \{yi\}_{PK(r)}\}_{SK(i)})) \\
\wedge\ & m_2 = (i, r, (i, \{Q, r, QX, \{QY\}_{PK(r)}\}_{SK(i)}))
\end{aligned}
$$

$\Rightarrow$ { Lemma 7, $(rid^j, \rho^j, \sigma^{j1})(n_i)$ }
$j1 < k1$
$\Rightarrow$ { definitie NI-SYNCH }
$1L\text{-}SYNCH(\alpha, k4, 1, rid^j, rid^k)$

We hebben tot slot aangetoond dat de send en read met label 1 synchroniseren. Voor alle labels van de acties die de claim-actie voorafgegaan zijn, hebben we laten zien dat ze synchroniseren. Het bewijs kan nu afgerond worden door het invullen van de definities. We kunnen de *cast* functie eenvoudig construeren uit de gevonden resultaten.

- { We kiezen de functie *cast* als volgt: }
  $cast(i) = rid^j \wedge cast(r) = rid^k$

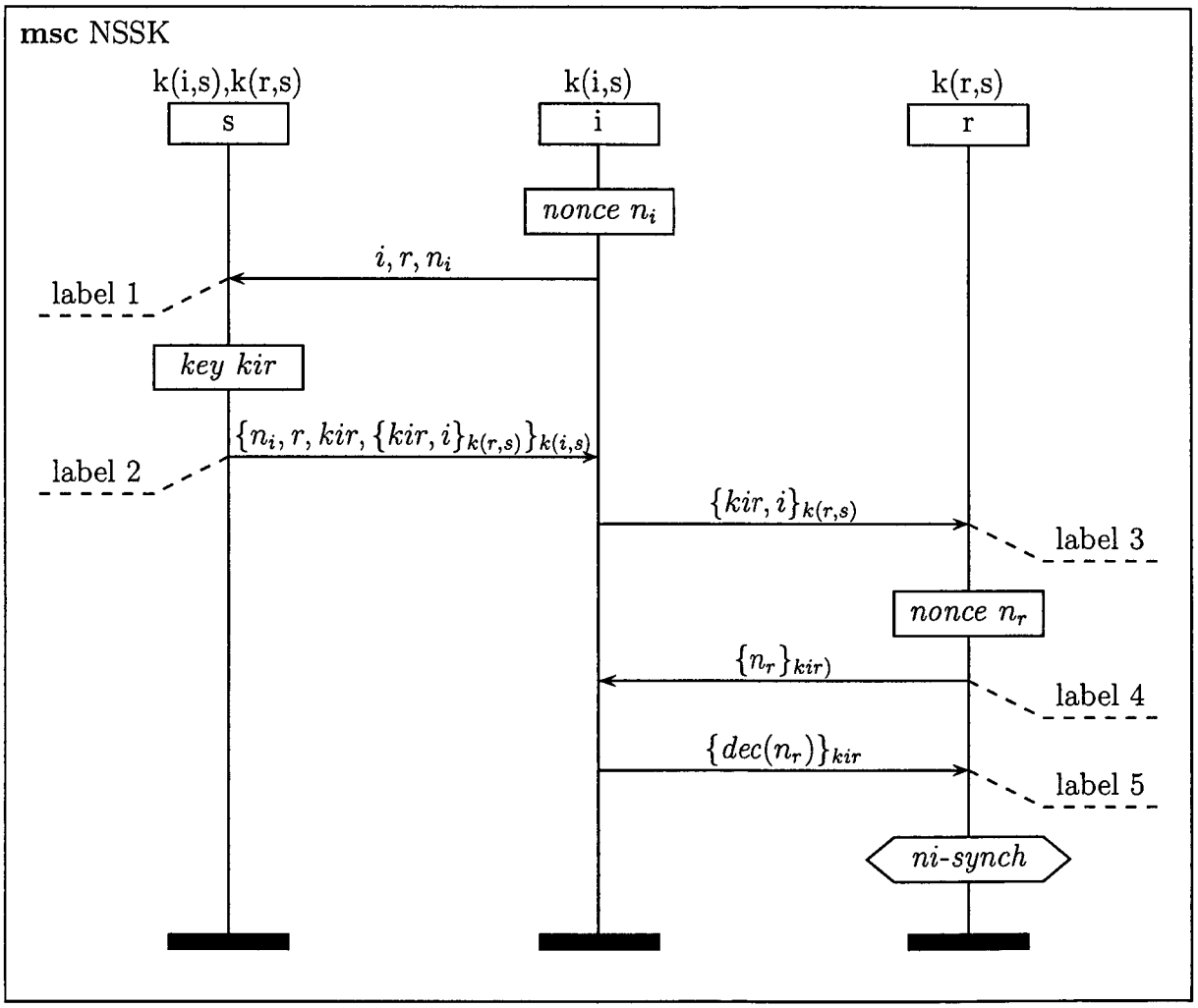
$\Rightarrow$ { gevonden resultaten en definities }
$\forall_{\ell \in \{1,2,3\}}\ 1L\text{-}SYNCH(\alpha, 4, \ell, cast(sendrole(\ell)), cast(readrole(\ell)))$
$\equiv$ { definitie }
$\exists_{cast:Inst \to RID} : cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k4, \{1,2,3\}, cast)$
$\Rightarrow$ { definitie }
$NI\text{-}SYNCH(p, 4)$

4.8 Needham Schroeder Symmetric Key

In deze paragraaf bestuderen we het "Needham Schroeder Symmetric Key" protocol [13], [14]. Het protocol bewerkstelligt de uitwisseling van een verse symmetrische sleutel gecreëerd door een server en die alleen bekend is bij de agenten die het protocol runnen. We zullen laten zien dat voor de responderrol na de laatste read-actie non-injectieve synchronisatie geclaimd mag worden.

4.8.1 Protocolbeschrijving

In Figuur 4.5 is schematisch de protocolspecificatie weergeven van "Needham Schroeder Symmetric Key". Voor het gemak zal dit protocol het NSSK protocol worden genoemd.



Figuur 4.5: Needham Schroeder Symmetric Key

4.8.2 Rolbeschrijving

De rollen in het NSSK protocol kunnen als volgt beschreven worden:

Globalen:

$func\ K(i,s) : key;$
$func\ K(r,s) : key;$

Initiatorrol (I)

$Initiator(i,r,s) =$
$const\ n_i : nonce;$
$var\ X : nonce;$
$var\ KX : key;$
$var\ TX : roleterm;$
$send_1(i,s,(i,r,n_i)) \cdot$
$read_2(s,i,\{n_i,r,KX,\ TX\}_{K(i,s)}) \cdot$
$send_3(i,r,\ TX) \cdot$
$read_4(r,i,\{X\}_{KX}) \cdot$
$send_5(i,r,\{dec(KX)\}_{KX})$

Responderrol (R)

$Responder(r,i,s) =$
$const\ n_r : nonce;$
$var\ KY : key;$
$read_3(i,r,\{KY,i\}_{K(r,s)}) \cdot$
$send_4(r,i,\{n_r\}_{KY}) \cdot$
$read_5(i,r,\{dec(n_r)\}_{KY}) \cdot$
$claim_6(r,\ ni\text{-}synch)$

Serverrol (S)

$Server(i,r,s) =$
$const\ kir : key;$
$var\ P : nonce;$
$read_1(i,s,(i,r,P)) \cdot$
$send_2(s,i,\{P,r,kir,\{kir,i\}_{K(r,s)}\}_{K(i,s)})$

4.8.3 Bewijs

We zullen bewijzen dat de claim voor non-injectieve synchronisatie van de responderrol correct is. Ten behoeve van de eenvoud van het bewijs zal een deelbewijs gebruikt worden. Het deelbewijs is terug te vinden in paragraaf 4.8.4 op pagina 81. Indien de responderrol non-injectieve synchronisatie claimt betekent dit het volgende.

$$
\begin{aligned}
& \mathit{NI\text{-}SYNCH}(p, 6) \\
\equiv \quad & \{\ \textit{definitie}\ \} \\
& \forall_{\alpha \in T(p), k6 \in N, rid^k, \rho^k, \sigma^{k6} \in \mathit{Inst}} : \\
& \quad \alpha_k = (rid^k, \rho^k, \sigma^{k6})(\mathit{claim}_6(r, \mathit{ni\text{-}synch})) \wedge \mathit{rng}(\rho^k) \subseteq \mathit{Agent}_T \Rightarrow \\
& \quad\quad \exists_{cast:Role \to RID}\ cast(r) = rid^k \wedge \mathit{ML\text{-}SYNCH}(\alpha, k6, \mathit{prec}(p, 6), cast)
\end{aligned}
$$

We veronderstellen dat een instantie van de responderrol non-injectieve synchronisatie claimt en dat de agenten waarmee de responder meent te communiceren te vertrouwen zijn. We laten zien dat uit deze aanname non-injectieve synchronisatie is af te leiden.

$$
\begin{aligned}
& \text{Stel:}\ \ \alpha \in T(p), k6 \in N, (rid^k, \rho^k, \sigma^{k6}) \in \mathit{Inst} \\
& \quad \text{zodat:}\ \ \alpha_{k6} = (rid^k, \rho^k, \sigma^{k6}, \mathit{claim}_6(r, \mathit{ni\text{-}synch})) \wedge \mathit{rng}(\rho^k) \subseteq \mathit{Agent}_T \\
\Rightarrow \quad & \{\ \text{Lemma 1, kies:}\ \ \sigma^{k3}, \sigma^{k4}, \sigma^{k5},\ \text{zodat:}\ \ \sigma^{k3} \subseteq \sigma^{k4} \subseteq \sigma^{k5} \subseteq \sigma^{k6} \\
& \quad\ \text{en kies:}\ \ k3, k4, k5 \in N,\ \text{zodat:}\ \ k3 < k4 < k5 < k6\ \} \\
& \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, \mathit{read}_3(i, r, \{KY, i\}_{K(r,s)})) \wedge \\
& \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, \mathit{send}_4(r, i, \{n_r\}_{KY})) \wedge \\
& \alpha_{k5} = (rid^k, \rho^k, \sigma^{k5}, \mathit{read}_5(i, r, \{\mathit{dec}(n_r)\}_{KY}))
\end{aligned}
$$

We hebben een instantie voor de responderrol gevonden. Nu zullen we bekijken, of we uit de $read_3$ in run rid^k een instantie voor de initiatorrol kunnen afleiden.

$$
\begin{aligned}
\Rightarrow \quad & \{\ \text{verzwakking}\ \} \\
& \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, \mathit{read}_3(i, r, \{KY, i\}_{K(r,s)})) \\
\Rightarrow \quad & \{\ \text{Lemma 2}\ \} \\
& (rid^k, \rho^k, \sigma^{k3})(i, r, \{KY, i\}_{K(r,s)}) \in M_{k3}
\end{aligned}
$$

$\Rightarrow$ { $(rid^k, \rho^k, \sigma^{k3})(r,s) \subseteq Agent_T \Rightarrow (rid^k, \rho^k, \sigma^{k3})(K(r,s)) \notin M_{k3}$, Lemma 4 }

$\exists_{j3 \in N, j3<k3, (rid^j, \rho^j, \sigma^{j3}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_l(a,b,m))$

$\wedge\ (rid^k, \rho^k, \sigma^{k3})(\{KY, i\}_{K(r,s)}) \sqsubseteq (rid^j, \rho^j, \sigma^{j3})(m)$

Op basis van de protocolbeschrijving kan de conclusie getrokken worden dat er twee matchen mogelijk zijn, namelijk die van een $send_2$ en van een $send_3$. We weten echter dat $(rid^k, \rho^k, \sigma^{k3})(i) \in Agent_T$ omdat deze meegezonden is in bericht 3, van een betrouwbare server. Indien de intruder $(rid^k, \rho^k, \sigma^{k3})(\{KY, i\}_{K(r,s)})$ uit een $send_2$ geleerd zou zijn dan zou de intruder ook een $K(i,s)$ kennen. Dit zou echter een tegenspraak opleveren omdat dezelfde instantie van i en s in de verzameling $Agent_T$ voorkomen. Dus kan het bericht door de intruder niet uit de $send_2$ geleerd zijn en moet het wel afkomstig zijn uit een $send_3$.

$\Rightarrow$ { protocolbeschrijving }

$\exists_{j3 \in N, j3<k3, (rid^j, \rho^j, \sigma^{j3}) \in Inst}$:

$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(i,r,TX))$

$\wedge\ (rid^k, \rho^k, \sigma^{k3})(\{i, KY\}_{K(r,s)}) = (rid^j, \rho^j, \sigma^{j3})(TX)$

$\Rightarrow$ { kies: $j3 \in N, (rid^j, \rho^j, \sigma^{j3}) \in Inst$, zodat: $j3 < k3 \wedge$

$(rid^k, \rho^k, \sigma^{k3})(\{i, KY\}_{K(r,s)}) = (rid^j, \rho^j, \sigma^{j3})(TX)$ } (I)

$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(i,r,TX))$

$\Rightarrow$ { Lemma 1, kies: $j2 \in N$ zodat: $j2 < j3$,

en kies: $\sigma^{j2} \in Inst$ zodat: $\sigma^{j2} \subseteq \sigma^{j3}$ }

$\alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(s, i, \{n_i, r, KX, TX\}_{K(i,s)}))$

We hebben een instantie voor de initiatorrol en bijbehorende run gevonden. Vervolgens zullen we bekijken of we uit het voorkomen van een $read_2$ in run rid^j (initiatorrun) een instantie voor de serverrol kunnen afleiden.

- { herhaling }

$\alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(\{n_i, r, KX, TX\}_{K(i,s)}))$

$\Rightarrow$ { Lemma 2 }

$(rid^j, \rho^j, \sigma^{j2})(\{n_i, r, KX, TX\}_{K(i,s)}) \in M_{j2}$

$\Rightarrow$ { $(rid^j, \rho^j, \sigma^{j3})(i,s) = (rid^k, \rho^k, \sigma^{k3})(i,s) \subseteq Agent_T$ Lemma 4 }

$\exists_{i2 \in N, i2<j2, (rid^i, \rho^i, \sigma^{i2}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, send_l(a,b,m))$

$\wedge\ (rid^j, \rho^j, \sigma^{j2})(\{n_i, r, KX, TX\}_{K(i,s)}) \sqsubseteq (rid^i, \rho^i, \sigma^{i2})(m)$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_2$ plaats hebben gevonden.

$\Rightarrow$ { protocolbeschrijving }

$\exists_{i2 \in N, i2<j2, (rid^i, \rho^i, \sigma^{i2}) \in Inst}$:

$\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, send_2(s, i, \{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)}))$

$\wedge\, (rid^i, \rho^i, \sigma^{i2})(\{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)})$

$= (rid^j, \rho^j, \sigma^{j2})(\{n_i, r, KX, TX\}_{K(i,s)})$

$\Rightarrow$ { kies: $i2 \in N, (rid^i, \rho^i, \sigma^{i2}) \in Inst$, zodat: $i2 < j2 \wedge$

$(rid^i, \rho^i, \sigma^{i2})(\{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)})$

$= (rid^j, \rho^j, \sigma^{j2})(\{n_i, r, KX, TX\}_{K(i,s)})$ } (II)

$\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, send_2(s, i, \{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)}))$

$\Rightarrow$ { Lemma 1, kies: $i1 \in N$ zodat: $i1 < i2$,

en kies: $\sigma^{i1} \in Inst$, zodat: $\sigma^{i1} \subseteq \sigma^{i2}$ }

$\alpha_{i1} = (rid^i, \rho^i, \sigma^{i1}, read_1(i, s, (i, r, P)))$

We hebben de benodigde gelijkheden voor de berichten met de labels 1, 2 en 3 afgeleid om hiervoor één label synchronisatie af te leiden. Voor de labels 3 en 4 moeten we nog gelijkheden afleiden. Om deze gelijkheden af te leiden hebben we de geheimhouding van de sessiesleutel nodig. Dat deze sleutel geheim blijft laten we zien door middel van Deelbewijs 1. We zullen nu de $read_5$ uit de run rid^k bestuderen.

$\Rightarrow$ { verzwakking }

$\alpha_{k5} = (rid^k, \rho^k, \sigma^{k5}, read_5(i, r, \{dec(n_r)\}_{KY}))$

$\Rightarrow$ { Lemma 2 }

$(rid^k, \rho^k, \sigma^{k5})(\{dec(n_r)\}_{KY}) \in M_{k5}$

$\Rightarrow$ { Deelbewijs 1, Lemma 4 }

$\exists_{h \in N, h5<k5, (rid^{jj}, \rho^{jj}, \sigma^{h5}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{h5} = (rid^{jj}, \rho^{jj}, \sigma^{h5}, send_l(a, b, m))$

$\wedge\, (rid^k, \rho^k, \sigma^{k5})(\{dec(n_r)\}_{KY}) \sqsubseteq (rid^{jj}, \rho^{jj}, \sigma^{h5})(m)$

$\Rightarrow$ { mogelijke matchen }

$\exists_{h4 \in N, h4<k5, (rid^{ii}, \rho^{ii}, \sigma^{h4}) \in Inst}$:

$\alpha_{h4} = (rid^{ii}, \rho^{ii}, \sigma^{h4}, send_4(r, i, \{n_r\}_{KY}))$

$\wedge\, (rid^{ii}, \rho^{ii}, \sigma^{i4})(\{n_r\}_{KY}) = (rid^k, \rho^k, \sigma^{k5})(\{dec(n_r)\}_{KY})$

$$
\begin{array}{ll}
& \vee \\
& \exists_{h5\in N, h5<k5, (rid^{jj},\rho^{jj},\sigma^{h5})\in Inst} : \\
& \quad \alpha_{h5} = (rid^{jj}, \rho^{jj}, \sigma^{h5}, send_5(i, r, \{dec(X)\}_{KX})) \\
& \quad \wedge\, (rid^{jj}, \rho^{jj}, \sigma^{h5})(\{dec(X)\}_{KX}) = (rid^{k}, \rho^{k}, \sigma^{k5})(\{dec(n_r)\}_{KY})
\end{array}
$$

Indien de runs gelijk zijn waarin n_r en $dec(n_r)$ gemaakt zijn, is het syntactische onmogelijk dat deze waarde gelijk aan elkaar zijn. Indien de runs verschillend zijn, is de kans dat deze waarde gelijk zijn te verwaarlozen omdat wij van nonces verwachten dat de waarde random zijn en dus onvoorspelbaar ver uit elkaar liggen.

$$
\begin{array}{ll}
\Rightarrow & \{ \text{ protocolbeschrijving } \} \\
& \exists_{h5\in N, h5<k5, (rid^{jj},\rho^{jj},\sigma^{h5})\in Inst} : \\
& \quad \alpha_{h5} = (rid^{jj}, \rho^{jj}, \sigma^{h5}, send_5(i, r, \{dec(n_r)\}_{KY})) \\
& \quad \wedge\, (rid^{jj}, \rho^{jj}, \sigma^{h5})(\{dec(X)\}_{KX}) = (rid^{k}, \rho^{k}, \sigma^{k5})(\{dec(n_r)\}_{KY}) \\
\Rightarrow & \{ \text{ kies: } h5 \in N, (rid^{jj}, \rho^{jj}, \sigma^{h5}) \in Inst, \text{ zodat: } h5 < k5 \wedge \\
& \quad (rid^{jj}, \rho^{jj}, \sigma^{h5})(\{dec(X)\}_{KX}) = (rid^{k}, \rho^{k}, \sigma^{k5})(\{dec(n_r)\}_{KY}) \} \qquad \text{(III)} \\
& \alpha_{h5} = (rid^{jj}, \rho^{jj}, \sigma^{h5}, send_5(i, r, \{i, r, s, X\}_{KX})) \\
\Rightarrow & \{ \text{ Lemma 1, kies: } h2, h4 \in N, \text{ zodat: } h2 < h4 < h5, \\
& \quad \text{en kies: } \sigma^{h2}, \sigma^{h4} \in Inst, \text{ zodat: } \sigma^{h2} \subseteq \sigma^{h4} \subseteq \sigma^{h5} \} \\
& \alpha_{h4} = (rid^{jj}, \rho^{jj}, \sigma^{h4}, read_4(r, i, \{X\}_{KX})) \wedge \\
& \alpha_{h2} = (rid^{jj}, \rho^{jj}, \sigma^{h2}, read_2(s, i, \{n_i, r, KX, TX\}_{K(i,s)}))
\end{array}
$$

We hebben een run gevonden waarin de acties behorende bij de $read_5$ en $send_4$ uit run rid^k voorkomen. We zullen proberen aan te tonen dat deze run dezelfde run is als de run rid^j die wij al eerder hebben gevonden.

$$
\begin{array}{ll}
\circ & \{ \text{ verzwakking } \} \\
& \alpha_{h2} = (rid^{jj}, \rho^{jj}, \sigma^{h2}, read_2(s, i, \{n_i, r, KX, TX\}_{K(i,s)})) \\
\Rightarrow & \{ \text{ Lemma 2 } \} \\
& (rid^{jj}, \rho^{jj}, \sigma^{h2})(\{n_i, r, KX, TX\}_{K(i,s)}) \in M_{h2} \\
\Rightarrow & \{ (rid^{jj}, \rho^{jj}, \sigma^{h5})(KX) = (rid^{k}, \rho^{k}, \sigma^{k5})(KY), \text{ Deelbewijs 1 } \} \\
& (rid^{jj}, \rho^{jj}, \sigma^{h2})(K(i, s)) \notin M_{h2} \\
\Rightarrow & \{ \text{ Lemma 4 } \} \\
& \exists_{g2\in N, g2<h2, (rid^{ii},\rho^{ii},\sigma^{g2})\in Inst, l\in Label, a,b\in Role, m\in RoleTerm} : \\
& \quad \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_l(a, b, m)) \\
& \quad \wedge\, (rid^{jj}, \rho^{jj}, \sigma^{h2})(\{n_i, r, KX, TX\}_{K(i,s)}) \sqsubseteq (rid^{ii}, \rho^{ii}, \sigma^{g2})(m)
\end{array}
$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_2$ plaats hebben gevonden.

$\Rightarrow$ { protocolbeschrijving, gelijkheden II }

$\exists_{g2 \in N, g2<h2, (rid^{ii}, \rho^{ii}, \sigma^{g2}) \in Inst}$:

$\alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(s, i, \{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)}))$

$\wedge\ (rid^{ii}, \rho^{ii}, \sigma^{g2})(\{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)})$

$= (rid^{jj}, \rho^{jj}, \sigma^{h2})(\{n_i, r, KX, TX\}_{K(i,s)})$

$\Rightarrow$ { kies: $g2 \in N, (rid^{ii}, \rho^{ii}, \sigma^{g2}) \in Inst$, zodat: $g2 < h2 \wedge$

$(rid^{ii}, \rho^{ii}, \sigma^{g2})(\{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)})$

$= (rid^{jj}, \rho^{jj}, \sigma^{h2})(\{n_i, r, KX, TX\}_{K(i,s)})$ } (IV)

$\alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(s, i, \{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)}))$

$\Rightarrow$ { gelijkheden I, II, III en IV }

$(rid^i, \rho^i, \sigma^{i2})(kir) = (rid^k, \rho^k, \sigma^{k5})(KY)$

$= (rid^j, \rho^j, \sigma^{j5})(KX) = (rid^{ii}, \rho^{ii}, \sigma^{g2})(kir)$

$\Rightarrow$ { Lemma 6 }

$rid^i = rid^{ii} \wedge \rho^i = \rho^{ii}$ (V)

$\Rightarrow$ { gelijkheden II, IV, V }

$(rid^j, \rho^j, \sigma^{j2})(n_i) = (rid^i, \rho^i, \sigma^{j2})(P)$

$= (rid^{ii}, \rho^{ii}, \sigma^{g2})(P) = (rid^{jj}, \rho^{jj}, \sigma^{h2})(n_i)$

$\Rightarrow$ { Lemma 6 }

$rid^j = rid^{jj} \wedge \rho^j = \rho^{jj}$ (VI)

We hebben laten zien dat de run rid^j en rid^{jj} aan elkaar gelijk zijn. We zullen nu laten zien dat voor alle berichten in het protocol één label synchronisatie geldt. We zullen als eerste het bericht met label 5 bekijken.

$\circ$ { herhalingen en gelijkheden I, II, III en VI }

$\alpha_{j5} = (rid^j, \rho^j, \sigma^{j5}, send_5(m_1))$

$\wedge\ \alpha_{k5} = (rid^k, \rho^k, \sigma^{k5}, read_5(m_2))$

$\wedge\ (rid^j, \rho^j, \sigma^{j5})(m_1) = (rid^k, \rho^k, \sigma^{k5})(m_2)$

$\wedge\ m_1 = (i, r, \{dec(X)\}_{KX}) \wedge m_2 = (i, r, \{dec(n_r)\}_{KY})$

$\wedge\ j5 < k5$

$\Rightarrow$ { definitie NI-SYNCH }

$1L\text{-}SYNCH(\alpha, k6, 5, rid^j, rid^k)$

We hebben eenvoudig laten zien de send en read met label 5 synchroniseren. Vervolgens bekijken we of voor label 4 synchronisatie optreedt.

- { herhalingen en gelijkheden I, II, III en VI }

$$\begin{aligned}
&\alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, send_4(m_1)) \\
&\wedge\, \alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, read_4(m_2)) \\
&\wedge\, (rid^k, \rho^k, \sigma^{k4})(m_1) = (rid^j, \rho^j, \sigma^{j4})(m_2) \\
&\wedge\, m_1 = (r, i, \{n_r\}_{KY}) \wedge m_2 = (r, i, \{X\}_{KX}) \\
\Rightarrow\ & \{ \text{ Lemma 7, } (rid^k, \rho^k, \sigma^{k4})(n_r) \sqsubseteq (rid^k, \rho^k, \sigma^{k4})(m_1) \} \\
& j4 < k4 \\
\Rightarrow\ & \{ \text{ definitie NI-SYNCH } \} \\
& \mathit{1L\text{-}SYNCH}(\alpha, k6, 4, rid^k, rid^j)
\end{aligned}$$

Wederom hebben we laten zien dat de read en send voor label 4 synchroniseren. We zullen vervolgens bekijken of dit ook het geval is voor label 3.

- { herhalingen en gelijkheden I en II }

$$\begin{aligned}
&\alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, send_3(m_1)) \\
&\wedge\, \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, read_3(m_2)) \\
&\wedge\, (rid^j, \rho^j, \sigma^{j3})(m_1) = (rid^k, \rho^k, \sigma^{k3})(m_2) \\
&\wedge\, m_1 = (i, r, TX) \wedge m_2 = (i, r, \{KY, i\}_{K(r,s)}) \\
& k3 < i3 \\
\Rightarrow\ & \{ \text{ definitie NI-SYNCH } \} \\
& \mathit{1L\text{-}SYNCH}(\alpha, k6, 3, rid^j, rid^k)
\end{aligned}$$

Ook voor bericht 3 hebben we laten zien dat de read en send synchroniseren. Vervolgens bekijken we berichten 2 voor synchronisatie.

- { herhalingen en gelijkheidII }

$$\begin{aligned}
&\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, send_2(m_1)) \\
&\wedge\, \alpha_{j2} = (rid^j, \rho^j, \sigma^{j2}, read_2(m_2)) \\
&\wedge\, (rid^i, \rho^i, \sigma^{i2})(m_1) = (rid^j, \rho^j, \sigma^{j3})(m_2) \\
&\wedge\, m_1 = (s, i, \{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)}) \\
&\wedge\, m_2 = (s, i, \{n_i, r, KX, TX\}_{K(i,s)}) \\
&\wedge\, i2 < j2
\end{aligned}$$

$\Rightarrow$ { definitie NI-SYNCH }
$1L\text{-}SYNCH(\alpha, k6, 2, rid^i, rid^j)$

Nu we ook weten dat voor label 2 synchronisatie optreedt, moeten wij alleen nog maar laten zien dat de read en de send met label 1 synchroniseren.

- { herhalingen en gelijkheid II }
  $\alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(m_1))$
  $\wedge\ \alpha_{i1} = (rid^i, \rho^i, \sigma^{i1}, read_1(m_2))$
  $\wedge\ (rid^j, \rho^j, \sigma^{j1})(m_1) = (rid^i, \rho^i, \sigma^{i1})(m_2)$
  $\wedge\ m_1 = (i, s, (i, r, n_i)) \wedge m_2 = (i, s, (i, r, P))$

$\Rightarrow$ { Lemma 7, $(rid^j, \rho^j, \sigma^{j1})(n_i) \sqsubseteq (rid^j, \rho^j, \sigma^{j1})(m_1)$ }
$j1 < i1$

$\Rightarrow$ { definitie NI-SYNCH }
$1L\text{-}SYNCH(\alpha, k6, 1, rid^j, rid^i)$

We hebben tot slot aangetoond dat de send en read met label 1 synchroniseren. Voor alle labels van de acties die de claim-actie voorafgegaan zijn, hebben we laten zien dat ze synchroniseren. Het bewijs kan nu afgerond worden door het invullen van de definities. We kunnen de *cast* functie eenvoudig construeren uit de gevonden resultaten.

- { We kiezen de functie *cast* als volgt: }
  $cast(i) = rid^j \wedge cast(r) = rid^k \wedge cast(s) = rid^i$

$\Rightarrow$ { gevonden resultaten en definities }
$\forall_{\ell \in \{1,2,3,4,5\}}\ 1L\text{-}SYNCH(\alpha, k6, \ell, cast(sendrole(\ell)), cast(readrole(\ell)))$

$\equiv$ { *Definitie* }
$\exists_{cast:Inst \to RID} : cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k6, \{1,2,3,4,5\}, cast)$

$\Rightarrow$ { *Definitie* }
$NI\text{-}SYNCH(p, 6)$

4.8.4 Deelbewijs

Het volgende deelbewijs is nodig voor het bewijs voor non-injectieve synchronisatie van het NSSK protocol.

Deelbewijs 1

$$(rid^k, \rho^k, \sigma^{k5})(KY) \notin M_{k5}$$

Ad deelbewijs 1

Er zijn twee manieren waardoor deze sleutel bij een intruder bekend kan zijn. De eerste is dat de intruder de sleutel zelf gemaakt heeft. We kunnen deze mogelijkheid eenvoudig uitsluiten. Als dit het geval zou zijn, dan zou de intruder het volgende bericht gemaakt moeten hebben.

$$(rid^k, \rho^k, \sigma^{k3})(\{KY, i\}_{K(r,s)})$$

Dit is zou betekenen dat $(rid^k, \rho^k, \sigma^{k3})(K(r,s))$ bekend is bij de intruder, zie gelijkheden I en II. We weten echter dat $rng(\rho^k)) \subseteq Agent_T$ zodat we deze mogelijkheid kunnen uitsluiten.

De tweede mogelijkheid waarop deze sleutel bekend kan zijn bij een intruder is dat de intruder de sleutel geleerd heeft uit een send-actie die deze sleutel bevatte.

$$
\begin{aligned}
& (rid^k, \rho^k, \sigma^{k5})(KY) \in M_{k5} \\
\Rightarrow \quad & \{\ (rid^k, \rho^k, \sigma^{k5})(KY) \notin M_0,\ \text{Lemma 5}\ \} \\
& \exists_{g<k5,(rid^g,\rho^g,\sigma^g)\in Inst,l\in Label,a,b\in Role,m\in RoleTerm,}\ : \\
& \quad \alpha_g = (rid^g, \rho^g, \sigma^g, send_l(a,b,m)) \\
& \quad \wedge ((rid^k, \rho^k, \sigma^{k5})(KY) \sqsubseteq (rid^g, \rho^g, \sigma^g)(m) \\
& \quad \wedge (rid^k, \rho^k, \sigma^{k5})(KY) \not\triangleleft M_g \\
\Rightarrow \quad & \{\ \text{mogelijke matchen}\ \} \\
& \exists_{f2\in N, f2<k5,(rid^{iii},\rho^{iii},\sigma^{f2})\in Inst}\ : \\
& \quad \alpha_{f2} = (rid^{iii}, \rho^{iii}, \sigma^{f2}, send_2(s,i,\{P,i,kir,\{kir,i\}_{K(r,s)}\}_{K(i,s)})) \\
& \quad \wedge (rid^{iii}, \rho^{iii}, \sigma^{f2})(kir) = (rid^k, \rho^k, \sigma^{k5})(KY) \\
& \quad \wedge (rid^{iii}, \rho^{iii}, \sigma^{f2})(K(i,s)) \in M_{k5} \\
& \quad \wedge (rid^{iii}, \rho^{iii}, \sigma^{f2})(kir) \notin M_{f2} \qquad (1.1) \\
& \vee
\end{aligned}
$$

$$\exists_{f3\in N, f3<k5, (rid^{jjj},\rho^{jjj},\sigma^{f3})\in Inst}:$$
$$\alpha_{f3} = (rid^{jjj}, \rho^{jjj}, \sigma^{g}, send_3(s, r, TX))$$
$$\wedge\, (rid^{k}, \rho^{k}, \sigma^{k5})(KY) \sqsubseteq (rid^{jjj}, \rho^{jjj}, \sigma^{g})(TX)$$
$$\wedge\, (rid^{jjj}, \rho^{jjj}, \sigma^{f3})(TX) \notin M_{f3} \qquad (1.2)$$

ad 1.1

- { herhaling }

$$\exists_{f2\in N, f2<k5, (rid^{iii},\rho^{iii},\sigma^{f2})\in Inst}:$$
$$\alpha_{f2} = (rid^{iii}, \rho^{iii}, \sigma^{f2}, send_2(s, i, \{P, i, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)}))$$
$$\wedge\, (rid^{iii}, \rho^{iii}, \sigma^{f2})(kir) = (rid^{k}, \rho^{k}, \sigma^{k5})(KY)$$
$$\wedge\, (rid^{iii}, \rho^{iii}, \sigma^{f2})(K(i, s)) \in M_{k5}$$
$$\wedge\, (rid^{iii}, \rho^{iii}, \sigma^{f2})(kir) \notin M_{f2}$$

$\Rightarrow$ { gelijkheden I en II, Lemma 6 }

$$(rid^{i}, \rho^{i}, \sigma^{i2})(K(i, s)) \in M_{k5} \wedge (rid^{i}, \rho^{i}, \sigma^{i2})(i, s) \subseteq Agent_T$$

$\Rightarrow$ { logica }

Tegenspraak

ad 1.2

- { herhaling }

$$\exists_{f3\in N, f3<k5, (rid^{jjj},\rho^{jjj},\sigma^{f3})\in Inst}:$$
$$\alpha_{f3} = (rid^{jjj}, \rho^{jjj}, \sigma^{g}, send_3(s, r, TX))$$
$$\wedge\, (rid^{k}, \rho^{k}, \sigma^{k5})(KY) \sqsubseteq (rid^{jjj}, \rho^{jjj}, \sigma^{g})(TX)$$
$$\wedge\, (rid^{jjj}, \rho^{jjj}, \sigma^{f3})(TX) \notin M_{f3}$$

$\Rightarrow$ { kies: $f3 \in N, (rid^{jjj}, \rho^{jjj}, \sigma^{f3}) \in Inst$, zodat: $f3 < k3 \wedge$
$(rid^{k}, \rho^{k}, \sigma^{k3})(KY) \sqsubseteq (rid^{jjj}, \rho^{jjj}, \sigma^{f3})(TX)$ } (VII)

$$\alpha_{f3} = (rid^{jjj}, \rho^{jjj}, \sigma^{f3}, send_3(i, r, TX))$$

$\Rightarrow$ { Lemma 1, kies: $j2 \in N$ zodat: $f2 < f3$,
en kies: $\sigma^{f2} \in Inst$ zodat: $\sigma^{f2} \subseteq \sigma^{f3}$ }

$$\alpha_{f2} = (rid^{jjj}, \rho^{jjj}, \sigma^{f2}, read_2(s, i, \{n_i, r, KX, TX\}_{K(i,s)}))$$

$\Rightarrow$ { $(rid^{jjj}, \rho^{jjj}, \sigma^{f3})(TX) \notin M_{f3} \Rightarrow (rid^{jjj}, \rho^{jjj}, \sigma^{f3})(K(i, s)) \notin M_{f3}$, Lemma 4 }

$$\exists_{e2\in N, e2<f2, (rid^{iii},\rho^{iii},\sigma^{e2})\in Inst, l\in Label, a,b\in Role, m\in RoleTerm}:$$
$$\alpha_{e2} = (rid^{iii}, \rho^{iii}, \sigma^{e2}, send_l(a, b, m))$$
$$\wedge\, (rid^{jjj}, \rho^{jjj}, \sigma^{f2})(\{n_i, r, KX, TX\}_{K(i,s)}) \sqsubseteq (rid^{iii}, \rho^{iii}, \sigma^{e2})(m)$$

Volgens de protocolbeschrijving is er maar een mogelijke match en er moet dus wel een $send_2$ plaats hebben gevonden.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ protocolbeschrijving } \} \\
& \exists_{e2 \in N, e2 < f2, (rid^{iii}, \rho^{iii}, \sigma^{e2}) \in Inst} : \\
& \quad \alpha_{e2} = (rid^{iii}, \rho^{iii}, \sigma^{e2}, send_2(s, i, \{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)})) \\
& \quad \wedge (rid^{iii}, \rho^{iii}, \sigma^{e2})(\{P, r, kir, \{kir, i\}_{K(r,s)}\}_{K(i,s)}) \\
& \quad = (rid^{jjj}, \rho^{jjj}, \sigma^{f2})(\{n_i, r, KX, TX\}_{K(i,s)}) \\
\Rightarrow \quad & \{ \text{ gelijkheden I, II, VII en Lemma 6 } \} \\
& rid^i = rid^{iii} \wedge \rho^i = \rho^{iii} \\
\Rightarrow \quad & \{ \text{ gelijke instanties } \} \\
& (rid^i, \rho^i, \sigma^{i2})(\{kir, i\}_{K(r,s)}) = (rid^{jjj}, \rho^{jjj}, \sigma^{f2})(TX)
\end{aligned}
$$

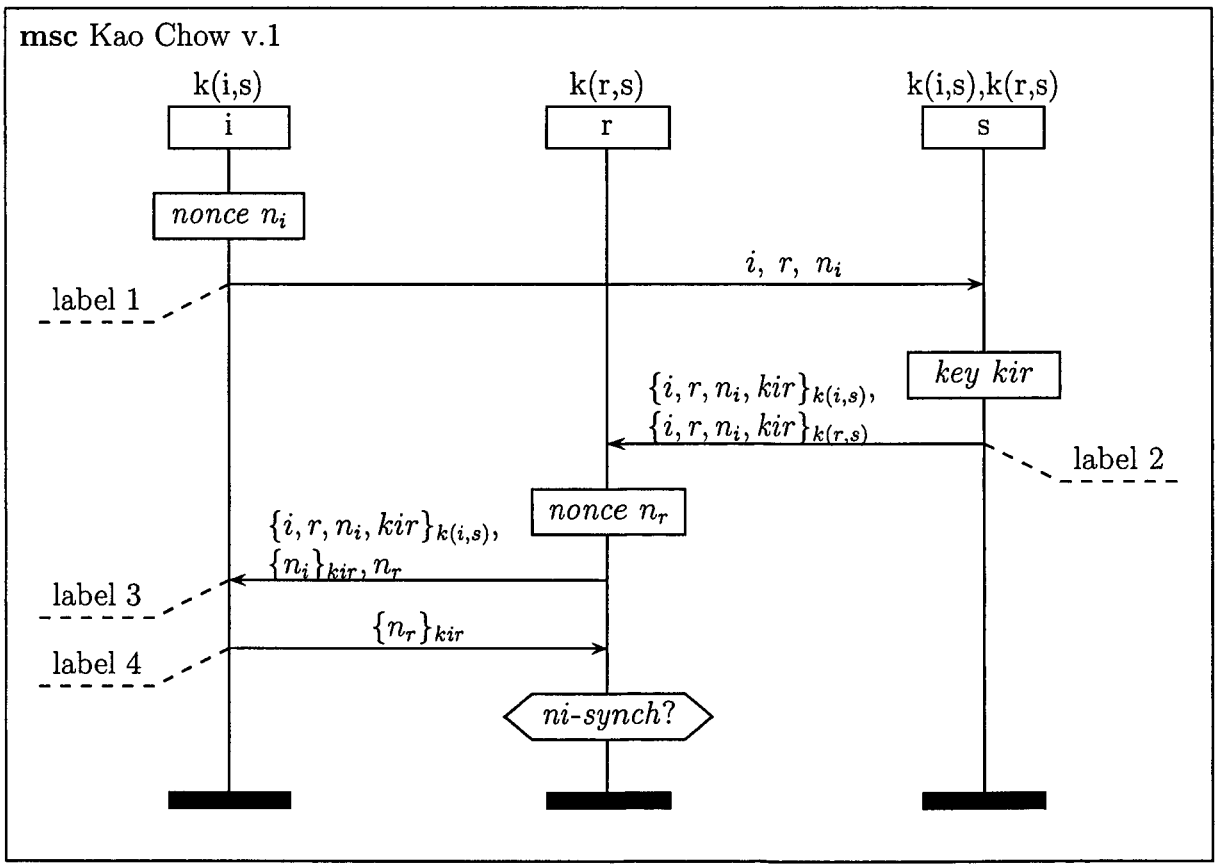
Als de intruder de sleutel $(rid^k, \rho^k, \sigma^{k5})(KY)$ uit de term $(rid^{jjj}, \rho^{jjj}, \sigma^{f2})(TX)$ geleerd heeft, dan moet de intruder ook de sleutel $(rid^i, \rho^i, \sigma^{i2})(K(r, s))$ kennen. We weten echter dat $(rid^i, \rho^i, \sigma^{i2})(r, s) \subseteq Agent_T$. We hebben een tegenspraak afgeleid en we mogen concluderen dat de sessiesleutel van de responderrol geheim blijft.

4.9 Kao Chow Authentication v.1

In deze paragraaf bestuderen we het Kao Chow Authentication v.1 protocol [4] en [8]. Dit protocol bewerkstelligt de uitwisseling van een verse symmetrische sleutel. De nieuwe sleutel wordt door een serverrol gemaakt. We zullen laten zien dat er geen non-injectieve synchronisatie geldt voor de responderrol in dit protocol. Echter door een kleine wijziging in het protocol aan te brengen treedt wel non-injectieve synchronisatie op voor de responderrol. Het bewijs hiervoor zal uitvoerig worden behandeld in paragraaf 4.9.4.

4.9.1 Analyse van het Kao Chow protocol

In Figuur 4.6 is schematisch de protocolspecificatie weergeven van "Kao Chow Authentication v.1". In eerste instantie vermoeden we dat na de laatste read-actie van de responderrol non-injectieve synchronisatie geclaimd mag worden.



Figuur 4.6: Kao Chow Authentication v.1

Bij een nadere analyse van dit protocol wordt al snel duidelijk dat non-injectieve synchronisatie voor de responderrol niet geldt. In bericht 2 wordt een ticket, $\{i, r, n_i, kir\}_{k(i,s)}$, van de server naar de responder gestuurd. Dit ticket wordt vervolgens in bericht 3 naar de initiator gestuurd. Het is niet mogelijk om te concluderen dat het ticket daad werkelijk in het bezit van de responder is geweest. We zullen deze aanval verduidelijken met een voorbeeld.

Voorbeeld

In dit voorbeeld zullen we laten zien dat het niet mogelijk is om voor het Kao Chow protocol non-injectieve synchronisatie te claimen. Een server stuurt een bericht van de vorm $(\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)})$ naar de responder. Als gevolg van deze send komt het bericht in de kennis van de intruder terecht. Een intruder weet dat de responder, voor wie dit bericht bestemd is, het eerste deel van het bericht niet kan herkennen omdat deze de sleutel $K(i, s)$ niet bezit. Een intruder kan dit bericht nu vervangen door willekeurig ander bericht. Zeg dat de intruder het bericht *Ticket* hiervoor in de plaats zet. De responder ontvangt dan een bericht van de vorm: $(Ticket, \{i, r, P, kir\}_{K(r,s)})$. De responder stuurt een bericht van de volgende vorm naar de initiator $(Ticket, \{n_i\}_{kir}, n_r)$ zoals het protocol dit voorschrijft. De intruder zal het originele bericht weer terug plaatsen op de plek waar nu *Ticket* staat. De initiator ontvangt tenslotte een bericht van de vorm: $(\{i, r, n_i, kir\}_{k(i,s)}, \{n_i\}_{kir}, n_r)$. Geen van de drie rollen in dit protocol heeft gemerkt dat het protocol niet netjes gevolgd is. Omdat we voor één label synchronisatie eisen dat agenten hetzelfde bericht ontvangen als er gestuurd is, kunnen we hier geen één label synchronisatie afleiden en dus ook geen non-injectieve synchronisatie.

We hebben laten zien dat het niet mogelijk is om voor het Kao Chow protocol non-injectieve synchronisatie af te leiden. Het is niet mogelijk om af te leiden dat hetzelfde bericht gelezen wordt als er eerder gestuurd is. Het doel van het Kao Chow protocol is de geheimhouding van de verse symmetrische sleutel verstrekt door de serverrol. Dat er geen non-injectieve synchronisatie geldt, wil niet zeggen dat een aanval mogelijk is zodat de doelstelling van dit protocol geschonden wordt.

Indien een eis zoals *logging* van belang is, dan kan dit probleem wel als zeer ernstig beschouwd worden. Als op later tijdstip bekeken wordt welke berichten de responder ontvangen heeft, kunnen dit hele andere berichten zijn dan die de server verstuurd heeft.

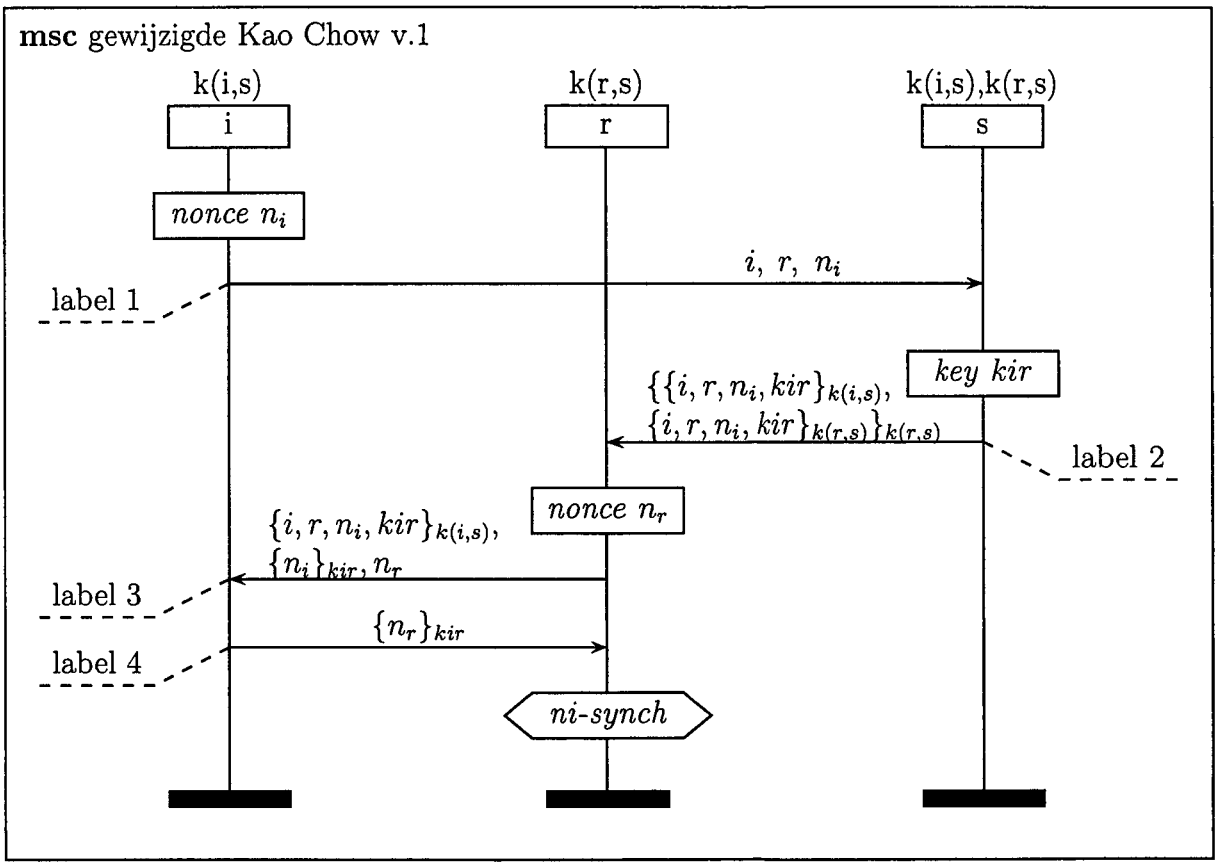
Omdat wij ons bezig houden met het bewijzen van non-injectieve synchronisatie zullen we trachten het protocol te wijzigen zodat het wel mogelijk is om non-injectieve synchronisatie af te leiden. We wijzigen het protocol door bericht 2 dat door de server naar de responder wordt gestuurd in zijn geheel te encrypten met de gemeenschappelijke sleutel van deze twee rollen. Hierdoor wordt het bericht onleesbaar voor een intruder en kan het ticket alleen door middel van een send 3 in de kennis van de intruder terecht komen, mits de intruder

de gemeenschappelijke sleutel van de server en de responder niet kent.

Het nieuwe protocol dat zo ontstaat zullen we het gewijzigde Kao Chow protocol noemen. In de volgende paragraaf geven we een volledige beschrijving van het nieuwe protocol door middel van een MSC gevolgd door de rolbeschrijving. Vervolgens zullen we non-injectieve synchronisatie voor dit protocol afleiden.

4.9.2 Protocolbeschrijving

In Figuur 4.7 is door middel van een MSC schematisch de protocolspecificatie weergegeven van de gewijzigde versie van het "Kao Chow Authentication v.1" protocol.



Figuur 4.7: Gewijzigde versie van Kao Chow Authentication v.1

4.9.3 Rolbeschrijving

De rollen in het gewijzigde Kao Chow v1 protocol kunnen als volgt beschreven worden.

Globalen:

$$\begin{aligned}&\mathit{func}\ \ K(i,s) : \mathit{key};\\ &\mathit{func}\ \ K(r,s) : \mathit{key};\end{aligned}$$

Initiatorrol (I)

$$\begin{aligned}&\mathit{Initiator}(i,r,s) =\\ &\quad \mathit{const}\ \ n_i : \mathit{nonce};\\ &\quad \mathit{var}\ \ X : \mathit{nonce};\\ &\quad \mathit{var}\ \ KX : \mathit{key};\\ &\quad \mathit{send}_1(i,s,(i,r,n_i)) \cdot\\ &\quad \mathit{read}_3(r,i,(\{i,r,n_i,KX\}_{K(i,s)},\{n_i\}_{KX},X)) \cdot\\ &\quad \mathit{send}_4(i,r,\{X\}_{KX})\end{aligned}$$

Responderrol (R)

$$\begin{aligned}&\mathit{Responder}(r,i,s) =\\ &\quad \mathit{const}\ \ n_r : \mathit{nonce};\\ &\quad \mathit{var}\ \ Y : \mathit{nonce};\\ &\quad \mathit{var}\ \ KY : \mathit{key};\\ &\quad \mathit{var}\ \ T : \mathit{term};\\ &\quad \mathit{read}_2(s,r,\{T,\{i,r,Y,KY\}_{K(r,s)}\}_{K(r,s)}) \cdot\\ &\quad \mathit{send}_3(r,i,(T,\{Y\}_{KY},n_r)) \cdot\\ &\quad \mathit{read}_4(i,r,\{n_r\}_{KY}) \cdot\\ &\quad \mathit{claim}_5(r,\mathit{ni\text{-}synch})\end{aligned}$$

Serverrol (S)

$$\begin{aligned}&\mathit{Server}(i,r,s) =\\ &\quad \mathit{const}\ \ \mathit{kir} : \mathit{SessionKey};\\ &\quad \mathit{var}\ \ P : \mathit{nonce};\\ &\quad \mathit{read}_1(i,s,(i,r,P)) \cdot\\ &\quad \mathit{send}_2(s,r,\{\{i,r,P,\mathit{kir}\}_{K(i,s)},\{i,r,P,\mathit{kir}\}_{K(r,s)}\}_{K(r,s)})\end{aligned}$$

4.9.4 Bewijs

We willen bewijzen dat de claim voor non-injectieve synchronisatie van de responderrol een correcte claim is. Voor het bewijs wordt gebruik gemaakt van diverse deelbewijzen. Deze deelbewijzen zijn terug te vinden in paragraaf 4.9.5 vanaf pagina 94. Indien de responderrol non-injectieve synchronisatie claimt betekent dit het volgende.

$$
\begin{aligned}
& NI\text{-}SYNCH(p,5) \\
\equiv \quad & \{ \text{ definitie } \} \\
& \forall_{\alpha \in T(p), k5 \in N, (rid^k, \rho^k, \sigma^{k5}) \in Inst} : \\
& \quad \alpha_{k5} = (rid^k, \rho^k, \sigma^{k5})(claim_5(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \Rightarrow \\
& \quad\quad \exists_{cast:Role \to RID} \; cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k5, prec(p,5), cast)
\end{aligned}
$$

We veronderstellen dat een instantie van de responderrol non-injectieve synchronisatie claimt en dat de agenten waarmee de responderrol meent te communiceren te vertrouwen zijn. We laten zien dat uit deze aanname non-injectieve synchronisatie is af te leiden.

$$
\begin{aligned}
& \text{Stel:} \quad \alpha \in T(p), k5 \in N, (rid^k, \rho^k, \sigma^{k5}) \in Inst \\
& \quad \text{zodat:} \quad \alpha_{k5} = (rid^k, \rho^k, \sigma^{k5}, claim_5(r, ni\text{-}synch)) \wedge rng(\rho^k) \subseteq Agent_T \\
\Rightarrow \quad & \{ \text{ Lemma 1, kies: } \sigma^{k2}, \sigma^{k3}, \sigma^{k4} \text{ zodat: } \sigma^{k2} \subseteq \sigma^{k3} \subseteq \sigma^{k4} \subseteq \sigma^{k5} \\
& \quad \text{en kies: } k2, k3, k4 \in N \text{ zodat: } k2 < k3 < k4 < k5 \ \} \\
& \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, read_2(s, r, \{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})) \wedge \\
& \alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \wedge \\
& \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, read_4(i, r, \{n_r\}_{KY}))
\end{aligned}
$$

We hebben een instantie van de responderrol geconstrueerd. Nu zullen we bekijken welke conclusies we uit deze run kunnen afleiden. We zullen eerst trachten een instantie voor de serverrol af te leiden. De sessiesleutel die de responder heeft ontvangen in bericht 2 is van groot belang voor het aantonen van gelijkheden van de diverse instanties van rollen. Om deze gelijkheden af te kunnen leiden hebben we een lokale constante nodig. De lokale constante voor de sessiesleutel vinden we in de rolspecificatie van de serverrol.

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ verzwakking } \} \\
& \alpha_{k2} = (rid^k, \rho^k, \sigma^{k2}, read_2(s, r, \{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}))
\end{aligned}
$$

$\Rightarrow$ { Lemma 2 }

$(rid^k, \rho^k, \sigma^{k2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \in M_{k2}$

$\Rightarrow$ { $(rid^k, \rho^k, \sigma^{k2})(r, s) \subseteq Agent_T \Rightarrow (rid^k, \rho^k, \sigma^{k2})(K(r, s)) \notin M_{k2}$, Lemma 4 }

$\exists_{i2 \in N, i2 < k2, (rid^i, \rho^i, \sigma^{i2}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, send_l(a, b, m))$

$\wedge\, (rid^k, \rho^k, \sigma^{k2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \sqsubseteq (rid^i, \rho^i, \sigma^{i2})(m)$

$\Rightarrow$ { Deelbewijs 4, mogelijke matchen }

$\exists_{i2 \in N, i2 < k2, (rid^i, \rho^i, \sigma^{i2}) \in Inst}$:

$\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, send_2(i, r, (\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)})))$

$\wedge\, (rid^k, \rho^k, \sigma^{k2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})$

$= (rid^i, \rho^i, \sigma^{i2})(\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)})$

$\Rightarrow$ { kies: $i2 \in N, (rid^i, \rho^i, \sigma^{i2}) \in Inst$, zodat: $i2 < k2$

$\wedge\, (rid^i, \rho^i, \sigma^{i2})(\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)})$

$= (rid^k, \rho^k, \sigma^{k2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})$ } (I)

$\alpha_{i2} = (rid^i, \rho^i, \sigma^{i2}, send_2(s, r, \{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}))$

$\Rightarrow$ { Lemma 1, kies: σ^{i1} zodat: $\sigma^{i1} \subseteq \sigma^{i2}$, en kies: $i1 \in N$ zodat: $i1 < i2$ }

$\alpha_{i1} = (rid^i, \rho^i, \sigma^{i1}, read_1(i, s, (i, r, P)))$

We hebben een instantie en run van de serverrol gevonden. We weten ook dat de variabele KY van de responderrol gelijk is aan de lokale constante kir van de serverrol (zie gelijkheid I). Deze gelijkheid zal nog een grote rol spelen in het bewijs. We bestuderen nu de run rid^k verder en zullen proberen om uit de $read_4$ een instantie en run van de initiatorrol af te leiden.

$\Rightarrow$ { verzwakking }

$\alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, read_4(i, r, \{n_r\}_{KY}))$

$\Rightarrow$ { Lemma 2 }

$(rid^k, \rho^k, \sigma^{k4})(\{n_r\}_{KY}) \in M_{k4}$

$\Rightarrow$ { Deelbewijs 1, Lemma 4 }

$\exists_{j4 \in N, j4 < k4, (rid^j, \rho^j, \sigma^{j4}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}$:

$\alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_l(a, b, m)) \wedge (rid^k, \rho^k, \sigma^{k4})(\{n_r\}_{KY}) \sqsubseteq (rid^j, \rho^j, \sigma^{j4})(m)$

$\Rightarrow$ { Deelbewijs 2 }

$\exists_{j4 \in N, j4 < k4, (rid^j, \rho^j, \sigma^{j4}) \in Inst} : \alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_4(i, r, \{X\}_{KX}))$

$\wedge\, (rid^j, \rho^j, \sigma^{j4})(\{X\}_{KX}) = (rid^k, \rho^k, \sigma^{k4})(\{n_r\}_{KY})$

$\Rightarrow$ { kies: $j4 \in N, (rid^j, \rho^j, \sigma^{j4}) \in Inst$, zodat: $j4 < k4$,
$(rid^j, \rho^j, \sigma^{j4})(\{X\}_{KX}) = (rid^k, \rho^k, \sigma^{k4})(\{n_r\}_{KY})$ } (II)
$\alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_4(i, r, \{X\}_{KX}))$
$\Rightarrow$ { Lemma 1, kies: σ^{j1}, σ^{j3} zodat: $\sigma^{j1} \subseteq \sigma^{j3} \subseteq \sigma^{j4}$
en kies; $j1, j3 \in N$ zodat: $j1 < j3 < j4$ }
$\alpha_{j1} = (rid^j, \rho^j, \sigma^{j1}, send_1(i, s, (i, r, n_i))) \wedge$
$\alpha_{j3} = (rid^j, \rho^j, \sigma^{j2}, read_3(r, i, (\{i, r, n_i, KX\}_{K(i,s)}, \{n_i\}_{KX}, X)))$

We hebben ook een instantie en run van de initiatorrol gevonden. We kunnen nog niet voor alle labels in de verzameling van de labels synchronisatie aantonen. De benodigde gelijkheden zijn door middel van de sessiesleutel af te leiden. We weten dat de sessiesleutel in de run rid^k en rid^j gelijk zijn. De initiator ontvangt deze sleutel door middel van een $read_3$. Daarom zullen we nu bekijken welke conclusies we kunnen afleiden uit de $read_3$ in run rid^j.

$\Rightarrow$ { Lemma 2 }
$(rid^j, \rho^j, \sigma^{j3})(\{i, r, n_i, KX\}_{K(i,s)}, \{n_i\}_{KX}, X) \in M_{j3}$
$\Rightarrow$ { verzwakking }
$(rid^j, \rho^j, \sigma^{j3})(\{i, r, n_i, KX\}_{K(i,s)}) \in M_{j3}$
$\Rightarrow$ { Deelbewijs 3 }
$\exists_{h3 \in N, h3 < j3, (rid^{kk}, \rho^{kk}, \sigma^{h3}) \in Inst}$:
$\alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_3(r, i, (T, \{Y\}_{KY}, n_r)))$
$\wedge (rid^{kk}, \rho^{kk}, \sigma^{h3})(T) = (rid^j, \rho^j, \sigma^{j3})(\{i, r, n_i, KX\}_{K(i,s)})$
$\Rightarrow$ { kies: $h3 \in N, (rid^{kk}, \rho^{kk}, \sigma^{h3}) \in Inst$, zodat: $h3 < j3 \wedge$
$(rid^j, \rho^j, \sigma^{j3})(\{i, r, n_i, KX\}_{K(i,s)}) = (rid^{kk}, \rho^{kk}, \sigma^{h3})(T)$ } (III)
$\alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_3(r, i, (T, \{Y\}_{KY}, n_r)))$
$\Rightarrow$ { Lemma 1, kies: $h2 \in N, \sigma^{h2}$, zodat: $h2 < h3 \wedge \sigma^{h2} \subseteq \sigma^{h3}$ }
$\alpha_{h2} = (rid^{kk}, \rho^{kk}, \sigma^{h2}, read_2(s, r, \{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}))$
$\Rightarrow$ { Lemma 2 }
$(rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \in M_{h2}$

We weten dat $(rid^{kk}, \rho^{kk}, \sigma^{h2})(T)$ niet initieel in de kennis van een intruder bevat is omdat deze term andere termen bevat die niet bij een intruder bekend kunnen zijn. Dit levert twee mogelijkheden op.

De eerste mogelijkheid, $(rid^{kk}, \rho^{kk}, \sigma^{h2})(T) \in M_{h2}$, leidt rechtstreeks naar deelbewijs 3. Via dit deelbewijs komen we weer op hetzelfde punt terecht. Er ontstaat dus een recursie. We weten echter dat er een moment is geweest $(rid^{kk}, \rho^{kk}, \sigma^{h2})(T)$ niet bekend was bij een intruder. Dit levert dus de tweede mogelijkheid op. We kiezen voor dit moment $h2$ zodat $(rid^{kk}, \rho^{kk}, \sigma^{h2})(T) \notin M_{h2}$.

$$
\begin{aligned}
\Rightarrow \quad & \{\ (rid^{kk}, \rho^{kk}, \sigma^{h2})(T) \notin M_{h2}\ \} \\
& (rid^{kk}, \rho^{kk}, \sigma^{h3})(K(r,s)) \notin M_{h2} \\
\Rightarrow \quad & \{\ \text{Lemma 4}\ \} \\
& \exists_{g2 \in N, g2<h2, (rid^{ii}, \rho^{ii}, \sigma^{g2}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm} : \\
& \quad \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{i2}, send_l(a,b,m)) \\
& \quad \wedge\, (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i,r,Y,KY\}_{K(r,s)}\}_{K(r,s)}) \sqsubseteq (rid^{ii}, \rho^{ii}, \sigma^{g2})(m) \\
\Rightarrow \quad & \{\ \text{Deelbewijs 4, gelijkheid III}\ \} \\
& \exists_{g2 \in N, g2<h2, (rid^{ii}, \rho^{ii}, \sigma^{g2}) \in Inst} : \\
& \quad \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(i,r,(\{\{i,r,P,kir\}_{K(i,s)}, \{i,r,P,kir\}_{K(r,s)}\}_{K(r,s)}))) \\
& \quad \wedge\, (rid^{ii}, \rho^{ii}, \sigma^{g2})(\{\{i,r,P,kir\}_{K(i,s)}, \{i,r,P,kir\}_{K(r,s)}\}_{K(r,s)}) \\
& \quad = (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i,r,Y,KY\}_{K(r,s)}\}_{K(r,s)}) \\
& \quad \wedge\, (rid^{ii}, \rho^{ii}, \sigma^{g2})(\{i,r,P,kir\}_{K(i,s)}) \\
& \quad = (rid^{kk}, \rho^{kk}, \sigma^{h2})(T) \\
& \quad = (rid^{j}, \rho^{j}, \sigma^{j3})(\{i,r,n_i,KX\}_{K(i,s)}) \\
\Rightarrow \quad & \{\ \text{gelijkheden I en II}\ \} \\
& \exists_{g2 \in N, g2<h2, (rid^{ii}, \rho^{ii}, \sigma^{g2}) \in Inst} : \\
& \quad \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(i,r,(\{\{i,r,P,kir\}_{K(i,s)}, \{i,r,P,kir\}_{K(r,s)}\}_{K(r,s)}))) \\
& \quad \wedge\, (rid^{ii}, \rho^{ii}, \sigma^{g2})(\{\{i,r,P,kir\}_{K(i,s)}, \{i,r,P,kir\}_{K(r,s)}\}_{K(r,s)}) \\
& \quad = (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i,r,Y,KY\}_{K(r,s)}\}_{K(r,s)}) \\
& \quad \wedge\, (rid^{ii}, \rho^{ii}, \sigma^{g2})(kir) \\
& \quad = (rid^{j}, \rho^{j}, \sigma^{j3})(KX) \\
& \quad = (rid^{k}, \rho^{k}, \sigma^{k3})(KY) \\
& \quad = (rid^{i}, \rho^{i}, \sigma^{i2})(kir) \\
\Rightarrow \quad & \{\ \text{Lemma 6, gelijkheden I en II}\ \} \\
& \quad (rid^{k}, \rho^{k}, \sigma^{k2})(i,r,Y,KY) \\
& = (rid^{i}, \rho^{i}, \sigma^{i2})(i,r,P,kir) \\
& = (rid^{k}, \rho^{k}, \sigma^{k2})(T)
\end{aligned}
$$

$$= (rid^{kk}, \rho^{kk}, \sigma^{h2})(T)$$
$$= (rid^{j}, \rho^{j}, \sigma^{j3})(i, r, n_i, KX) \quad \text{(IV)}$$

We weten genoeg om voor alle labels eenvoudig één label synchronisatie af te leiden. We zullen hiervoor per label de benodigde informatie samenvatten. In de meeste gevallen zal er nog een klein afleiding voor de volgorde nodig zijn. Als eerste zullen we de communicatie van label 1 beschouwen.

$\circ$ { herhaling en gelijkheid IV }
$\alpha_{j1} = (rid^{j}, \rho^{j}, \sigma^{j1}, send_1(i, s, m_1))$
$\wedge\ \alpha_{i1} = (rid^{i}, \rho^{i}, \sigma^{i1}, read_1(i, s, m_2))$
$\wedge\ (rid^{j}, \rho^{j}, \sigma^{j1})(m_1) = (rid^{i}, \rho^{i}, \sigma^{i1})(m_2)$
$\wedge\ m_1 = (i, r, n_i) \wedge m_2 = (i, r, P)$
$\Rightarrow$ { Lemma 7, $(rid^{j}, \rho^{j}, \sigma^{j1})(n_i) \sqsubseteq (rid^{j}, \rho^{j}, \sigma^{j1})(m_1)$ }
$j1 < i1$
$\Rightarrow$ { definitie NI-SYNCH }
$1L\text{-}SYNCH(\alpha, k5, 1, rid^{j}, rid^{i})$

We hebben geconcludeerd dat voor label 1, één label synchronisatie geldt. Vervolgens zullen we bekijken of de send en read met label 2 synchroniseren.

$\circ$ { herhaling en gelijkheid I }
$\alpha_{i2} = (rid^{i}, \rho^{i}, \sigma^{i2}, send_2(s, r, m_2))$
$\wedge\ \alpha_{k2} = (rid^{k}, \rho^{k}, \sigma^{k2}, read_2(s, r, m_1))$
$\wedge\ (rid^{i}, \rho^{i}, \sigma^{i2})(m_1) = (rid^{k}, \rho^{k}, \sigma^{k2})(m_2)$
$\wedge\ m_1 = \{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}$
$\wedge\ m_2 = \{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}$
$\wedge\ i2 < k2$
$\Rightarrow$ { definitie NI-SYNCH }
$1L\text{-}SYNCH(\alpha, k5, 2, rid^{i}, rid^{k})$

We hebben geconcludeerd dat voor label 2, één label synchronisatie geldt. Vervolgens zullen we bekijken of de send en read met label 3 synchroniseren.

- { herhaling en gelijkheid IV }

$$\begin{aligned}
&\alpha_{k3} = (rid^k, \rho^k, \sigma^{k3}, send_3(r, i, m_1)) \\
&\wedge\, \alpha_{j3} = (rid^j, \rho^j, \sigma^{j3}, read_3(r, i, m_2)) \\
&\wedge\, (rid^k, \rho^k, \sigma^{k3})(m_1) = (rid^j, \rho^j, \sigma^{j3})(m_2) \\
&\wedge\, (rid^k, \rho^k, \sigma^{k3})(i, r) = (rid^j, \rho^j, \sigma^{j3})(i, r) \\
&\wedge\, m_1 = (T, \{Y\}_{KY}, n_r) \\
&\wedge\, m_2 = (\{i, r, n_i, KX\}_{K(i,s)}, \{n_i\}_{KX}, X)
\end{aligned}$$

$\Rightarrow$ { Lemma 7, $(rid^k, \rho^k, \sigma^{k3})(n_r) \sqsubseteq (rid^k, \rho^k, \sigma^{k3})(m_1)$ }

$k3 < j3$

$\Rightarrow$ { definitie NI-SYNCH }

$1L\text{-}SYNCH(\alpha, k5, 3, rid^k, rid^j)$

We hebben geconcludeerd dat voor label 3, één label synchronisatie geldt. Vervolgens zullen we bekijken of de send en read met label 4 synchroniseren.

- { herhaling en gelijkheid II }

$$\begin{aligned}
&\alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_4(i, r, m_1)) \\
&\wedge\, \alpha_{k4} = (rid^k, \rho^k, \sigma^{k4}, read_4(i, r, m_2)) \\
&\wedge\, (rid^j, \rho^j, \sigma^{j4})(m_1) = (rid^k, \rho^k, \sigma^{k3})(m_2) \\
&\wedge\, (rid^j, \rho^j, \sigma^{j4})(i, r) = (rid^k, \rho^k, \sigma^{k4})(i, r) \\
&\wedge\, m_1 = \{n_r\}_{KY} \wedge m_2 = \{X\}_{KX} \\
&\wedge\, j4 < k4
\end{aligned}$$

$\Rightarrow$ { definitie NI-SYNCH }

$1L\text{-}SYNCH(\alpha, k5, 4, rid^j, rid^k)$

We hebben tot slot aangetoond dat de send en read met label 4 synchroniseren. Voor alle labels van de acties die de claim-actie voorafgegaan zijn, hebben we laten zien dat ze synchroniseren. Het bewijs kan nu afgerond worden door het invullen van de definities. We kunnen de *cast* functie eenvoudig construeren uit de gevonden resultaten.

- { We kiezen de functie *cast* als volgt: }

$cast(i) = rid^j \wedge cast(r) = rid^k \wedge cast(s) = rid^i$

$\Rightarrow$ { gevonden resultaten en definities }
$\forall_{\ell \in \{1,2,3,4\}}\ 1L\text{-}SYNCH(\alpha, k5, \ell, cast(sendrole(\ell)), cast(readrole(\ell)))$
$\equiv$ { definitie }
$\exists_{cast:Inst \to RID} : cast(r) = rid^k \wedge ML\text{-}SYNCH(\alpha, k5, \{1,2,3,4\}, cast)$
$\Rightarrow$ { definitie }
$NI\text{-}SYNCH(p, 5)$

4.9.5 Deelbewijzen

In deze paragraaf zijn de deelbewijzen te vinden die nodig zijn voor het bewijs van non-injectieve synchronisatie van het gewijzigde Kao Chow protocol. We zullen eerst de bewijzen kort als stellingen formuleren en vervolgens de bewijzen uitschrijven.

In enkele deelbewijzen worden andere deelbewijzen gebruikt. In Tabel 4.3 wordt aangegeven welke afhankelijkheden tussen de deelbewijzen bestaan.

deelbewijs	afhankelijk van deelbewijs
1	4
2	1
3	1
4	niet

Tabel 4.3: Afhankelijkheden deelbewijzen Kao Chow

Het eerste deelbewijs garandeert de geheimhouding van de sessiesleutel die door de serverrol aan de responderrol verstrekt is.

Deelbewijs 1

$(rid, \rho, \sigma^{k4})(KY) \notin M_{k4}$

Deelbewijs 2 laat zien dat als er een send is geweest waarvan bekend is dat deze send de term $(rid^k, \rho^k, \sigma^{k4})(\{n_r\}_{KY})$ bevat. Dit alleen mogelijk is als gevolg van een $send_4$.

Deelbewijs 2

$$\exists_{j4\in N, j4<k4, (rid^j,\rho^j,\sigma^{j4})\in Inst, l\in Label, a,b\in Role, m\in RoleTerm} :$$
$$\alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_l(a,b,m))$$
$$\wedge\, (rid^k, \rho^k, \sigma^{k4})(\{n_r\}_{KY}) \sqsubseteq (rid^j, \rho^j, \sigma^{j4})(m)$$
$$\Rightarrow$$
$$\exists_{j4\in N, j4<k4, (rid^j,\rho^j,\sigma^{j4})\in Inst} :$$
$$\alpha_{j4} = (rid^j, \rho^j, \sigma^{j4}, send_4(i,r,\{X\}_{KX})) \wedge$$
$$(rid^j, \rho^j, \sigma^{j4})(\{X\}_{KX}) = (rid^k, \rho^k, \sigma^{k4})(\{n_r\}_{KY})$$

Deelbewijs 3 laat zien dat als er een send geweest is waarvan bekend is, dat deze send de term $\{i, r, n_i, KX\}_{K(i,s)}$ bevat. Dit alleen mogelijk is als gevolg van een $send_3$.

Deelbewijs 3

$$\exists_{h3\in N, h3<j3, (rid^{kk},\rho^{kk},\sigma^{h3})\in Inst, l\in Label, a,b\in Role, m\in RoleTerm} :$$
$$\alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_l(a,b,m))$$
$$\wedge\, (rid^j, \rho^j, \sigma^{j4})(\{i, r, n_i, KX\}_{K(i,s)}) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{h3})(m)$$
$$\Rightarrow$$
$$\exists_{h3\in N, h3<j3, (rid^{kk},\rho^{kk},\sigma^{h3})\in Inst} :$$
$$\alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_3(r,i,(T,\{Y\}_{KY}, n_r))) \wedge$$
$$(rid^{kk}, \rho^{kk}, \sigma^{h3})(T) = (rid^j, \rho^j, \sigma^{j3})(\{i, r, n_i, KX\}_{K(i,s)})$$

Met het laatste deelbewijs laten we zien dat als we weten dat er een bericht van de vorm $\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}$ in de kennis van de intruder voorkomt deze niet geleerd kan zijn door een $send_3$.

Deelbewijs 4

$$\exists_{g3\in N, g3<k2, (rid^{kk},\rho^{kk},\sigma^{g3})\in Inst, t\in RoleTerm} :$$
$$\alpha_{g3} = (rid^{kk}, \rho^{kk}, \sigma^{g3}, send_3(r,i,(T,\{Y\}_{KY}, n_r)))$$
$$\wedge\, (rid^{kk}, \rho^{kk}, \sigma^{g3})(t) \notin M_{g3} \wedge (rid^{kk}, \rho^{kk}, \sigma^{g3})(t) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{g3})(T)$$
$$\wedge\, (rid^{kk}, \rho^{kk}, \sigma^{g3})(T)$$
$$= (rid^k, \rho^k, \sigma^{k2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})$$
$$\Rightarrow$$
$$False$$

Ad deelbewijs 1

$$
\begin{aligned}
& (rid^k, \rho^k, \sigma^{k4})(KY) \in M_{k4} \\
\Rightarrow \quad & \{ \text{ Lemma 5 } \} \\
& \exists_{g<k4,(rid,\rho,\sigma^g)\in Inst,l\in Label,a,b\in Role,m\in RoleTerm,} : \\
& \quad \alpha_g = (rid, \rho, \sigma^g, send_l(a,b,m)) \wedge ((rid^k, \rho^k, \sigma^{k4})(KY) \sqsubseteq (rid, \rho, \sigma^g)(m) \\
& \quad \wedge (rid^k, \rho^k, \sigma^{k4})(KY) \not\triangleleft M_g \\
\Rightarrow \quad & \{ \text{ protocolbeschrijving, mogelijke matchen voor m } \} \\
& \exists_{g3<k4,(rid^{kk},\rho^{kk},\sigma^{g3})\in Inst} : \\
& \quad \alpha_{g3} = (rid^{kk}, \rho^{kk}, \sigma^{g3}, send_3(r,i,(T,\{Y\}_{KY}, n_r))) \\
& \quad \wedge (rid^k, \rho^k, \sigma^{k4})(KY) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{g3})(T)) \qquad (1.1) \\
& \vee \\
& \exists_{g2<k4,(rid^{ii},\rho^{ii},\sigma^{g2})\in Inst} : \\
& \quad \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(s,r,m)) \\
& \quad m = \{\{i,r,P,kir\}_{K(i,s)}, \{i,r,P,kir\}_{K(r,s)}\}_{K(r,s)} \\
& \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^{g2})(kir) = (rid^k, \rho^k, \sigma^{k4})(KY) \\
& \quad \wedge ((rid^{ii}, \rho^{ii}, \sigma^{g2})(K(r,s)) \in M_{k4} \qquad (1.2)
\end{aligned}
$$

Het bewijs wordt opgesplitst in twee gevallen. We zullen laten zien dat beide gevallen tot een tegenspraak leiden zodat de conclusie getrokken mag worden dat de bewuste sleutel niet bij een intruder bekend kan zijn. We bekijken eerste de mogelijkheid dat de sleutel is geleerd uit een $send_3$.

ad 1.1

$$
\begin{aligned}
\Rightarrow \quad & \{ \text{ kies: } g3 < k4, (rid^{kk}, \rho^{kk}, \sigma^{g3}) \in Inst, \text{ zodat:} \\
& \quad g3 \in N \wedge (rid^k, \rho^k, \sigma^{k4})(KY) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{g3})(T) \} \qquad (v) \\
& \alpha_{g3} = (rid^{kk}, \rho^{kk}, \sigma^{g3}, send_3(r,i,(T,\{Y\}_{KY}, n_r))) \\
\Rightarrow \quad & \{ \text{ Lemma 1, kies: } \sigma^{g2}, \text{ zodat: } \sigma^{g2} \subseteq \sigma^{g3}, g2 \in N \text{ zodat } g2 < g3 \} \\
& \alpha_{g2} = (rid^{kk}, \rho^{kk}, \sigma^{g2}, read_2(s,r,\{T,\{i,r,Y,KY\}_{K(r,s)}\}_{K(r,s)})) \\
\Rightarrow \quad & \{ \text{ Lemma 2 } \} \\
& (rid^{kk}, \rho^{kk}, \sigma^{g2})(\{T,\{i,r,Y,KY\}_{K(r,s)}\}_{K(r,s)}) \in M_{k2}
\end{aligned}
$$

In het geval dat we zouden veronderstellen dat $(rid^{kk}, \rho^{kk}, \sigma^{g2})(K(r,s)) \in M_{g2}$, zou dit betekenen dat een intruder de term $(rid^{kk}, \rho^{kk}, \sigma^{g3})(T)$ al eerder kende. Deze zou dan niet

uit een $send_3$ geleerd zijn. We mogen dus veronderstellen dat $(rid^{kk}, \rho^{kk}, \sigma^{g2})(K(r,s)) \notin M_{g2}$.

$$
\begin{array}{ll}
 & (rid^{kk}, \rho^{kk}, \sigma^{g2})(K(r,s)) \notin M_{g2} \\
\Rightarrow & \{ \text{ Lemma 4, mogelijke matchen } \} \\
 & \exists_{f2 \in N, f2 < g2, (rid^{ii}, \rho^{ii}, \sigma^{f2}) \in Inst} : \\
 & \quad \alpha_{f2} = (rid^{ii}, \rho^{ii}, \sigma^{f2}, send_2(s, r, (\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}))) \\
 & \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^{f2})(\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}) \\
 & \quad = (rid^{kk}, \rho^{kk}, \sigma^{g2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \\
 & \vee \\
 & \exists_{f3 \in N, f3 < g2, (rid^{kkk}, \rho^{kkk}, \sigma^{f3}) \in Inst} : \\
 & \quad \alpha_{f3} = (rid^{kkk}, \rho^{kkk}, \sigma^{f3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \\
 & \quad \wedge (rid^{kkk}, \rho^{kkk}, \sigma^{f3})(T) \\
 & \quad = (rid^{kk}, \rho^{kk}, \sigma^{g2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \\
\Rightarrow & \{ \text{ Deelbewijs 4, kies: } f2 \in N, (rid^{ii}, \rho^{ii}, \sigma^{f2}) \in Inst, \text{ zodat: } f2 < g2 \\
 & \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^{f2})(\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}) \\
 & \quad = (rid^{kk}, \rho^{kk}, \sigma^{g2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \ \} \\
 & \alpha_{f2} = (rid^{ii}, \rho^{ii}, \sigma^{f2}, send_2(s, r, (\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}))) \\
\Rightarrow & \{ \text{ gelijkheid I, en } (rid^{k}, \rho^{k}, \sigma^{k4})(KY) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{g3})(T), \text{ zie V } \} \\
 & (rid^{k}, \rho^{k}, \sigma^{k4})(KY) = (rid^{ii}, \rho^{ii}, \sigma^{f2})(kir) \wedge (rid^{ii}, \rho^{ii}, \sigma^{f2})(K(i,s)) \in M_{f2} \\
 & \wedge (rid^{k}, \rho^{k}, \sigma^{k4})(KY) = (rid^{i}, \rho^{i}, \sigma^{i2})(kir) \\
\Rightarrow & \{ \text{ Lemma 6 } \} \\
 & rid^{i} = rid^{ii} \wedge \rho^{i} = \rho^{ii} \\
\Rightarrow & \{ \ (rid^{ii}, \rho^{ii}, \sigma^{f2})(K(i,s)) \in M_{f2} \wedge \{(rid^{i}, \rho^{i}, \sigma^{i2})(i,s)\} \subseteq Agent_T \ \} \\
 & \textit{Tegenspraak}
\end{array}
$$

De bewuste sleutel kan door een intruder niet geleerd zijn uit een $send_3$. Vervolgens zullen we laten zien dat de aanname dat deze sleutel uit een $send_2$ is geleerd wederom tot een tegenspraak leidt.

ad 1.2

$$
\begin{array}{ll}
\Rightarrow & \{ \text{ kies: } g2 < k4, (rid^{ii}, \rho^{ii}, \sigma^{g2}) \in Inst, \text{ zodat:} \\
 & \quad g2 \in N \wedge (rid^{ii}, \rho^{ii}, \sigma^{g2})(kir) = (rid^{k}, \rho^{k}, \sigma^{k4})(KY) \ \} \qquad \text{(VI)} \\
 & \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(s, r, \{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)})) \\
 & \wedge (rid^{ii}, \rho^{ii}, \sigma^{g2})(K(r,s)) \in M_{k4}
\end{array}
$$

$\Rightarrow$ { gelijkheden I en VI }
$(rid^{ii}, \rho^{ii}, \sigma^{g2})(kir)$
$= (rid^{k}, \rho^{k}, \sigma^{k4})(KY)$
$= (rid^{i}, \rho^{i}, \sigma^{i2})(kir)$
$\Rightarrow$ { Lemma 6 }
$rid^{i} = rid^{ii} \wedge \rho^{i} = \rho^{ii}$
$\Rightarrow$ { $\{(rid^{i}, \rho^{i}, \sigma^{i2})(r,s)\} \subseteq Agent_T \wedge (rid^{ii}, \rho^{ii}, \sigma^{g2})(K(r,s)) \in M_{k4}$ }
Tegenspraak

De aanname dat de sleutel uit een $send_2$ is geleerd, leidt tot een tegenspraak. We hebben nu laten zien dat het alle mogelijkheden die voorkomen uit de aanname dat de sleutel bekend is bij een intruder tot een tegenspraak leiden. We kunnen dus concluderen dat de sleutel geheim blijft.

Ad deelbewijs 2

$$\exists_{j4 \in N, j4<k4, (rid^{j}, \rho^{j}, \sigma^{j4}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm} :$$
$$\alpha_{j4} = (rid^{j}, \rho^{j}, \sigma^{j4}, send_l(a,b,m))$$
$$\wedge (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY}) \sqsubseteq (rid^{j}, \rho^{j}, \sigma^{j4})(m)$$
$\Rightarrow$ { protocolbeschrijving }
$$\exists_{j4 \in N, j4<k4, (rid^{j}, \rho^{j}, \sigma^{j4}) \in Inst} :$$
$$\alpha_{j4} = (rid^{j}, \rho^{j}, \sigma^{j4}, send_4(i,r,\{X\}_{KX}))$$
$$\wedge (rid^{j}, \rho^{j}, \sigma^{j4})(\{X\}_{KX}) = (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY}) \quad (2.1)$$
$$\vee$$
$$\exists_{h3 \in N, h3<k4, (rid^{kk}, \rho^{kk}, \sigma^{h3}) \in Inst} :$$
$$\alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_3(r,i,(T,\{Y\}_{KY}, n_r))) \wedge$$
$$(rid^{kk}, \rho^{kk}, \sigma^{h3})(T) = (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY}) \quad (2.2)$$
$$\vee$$
$$(rid^{kk}, \rho^{kk}, \sigma^{h3})(\{Y\}_{KY}) = (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY}) \quad (2.3)$$

Er zijn drie mogelijkheden waarop de term $(rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY})$ in de kennis van een intruder terecht kan komen. Volgens het protocolbeschrijving zou dit gebeurt moeten zijn door een send zoals weergegeven wordt in geval 2.1. We zullen laten zien dat de gevallen 2.2 en 2.3 tot een tegenspraak leiden en de term $(rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY})$ alleen door de mogelijkheid zoals in geval 2.1 wordt beschreven in de kennis van de intruder terecht kan komen.

ad 2.2

$$
\begin{aligned}
\Rightarrow\ & \{\ \text{kies: } h3 \in N, (rid^{kk}, \rho^{kk}, \sigma^{h3}) \in Inst, \text{ zodat:} \\
& \quad h3 < k4, (rid^{kk}, \rho^{kk}, \sigma^{h3})(T) = (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY})\ \} \qquad \text{(VII)} \\
& \alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \\
\Rightarrow\ & \{\ \text{Lemma 1, kies: } h2 \in N, \sigma^{h2}, \text{ zodat: } h2 < h3 \wedge \sigma^{h2} \subseteq \sigma^{h3}\ \} \\
& \alpha_{h2} = (rid^{kk}, \rho^{kk}, \sigma^{h2}, read_2(s, r, \{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})) \\
\Rightarrow\ & \{\ \text{Lemma 2}\ \} \\
& (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \in M_{h2}
\end{aligned}
$$

We kunnen nu twee gevallen onderscheiden. In het eerste geval is de sleutel wel bekend bij een intruder in het tweede geval is deze onbekend.

geval 1 (sleutel bekend)

$$
\begin{aligned}
& (rid^{kk}, \rho^{kk}, \sigma^{h2})(K(r,s)) \in M_{h2} \\
\Rightarrow\ & \{\ \text{encryptie}\ \} \\
& (rid^{kk}, \rho^{kk}, \sigma^{h2})(T) \in M_{h2} \wedge (rid^{kk}, \rho^{kk}, \sigma^{h3})(T) = (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY}) \\
& \wedge (rid^{k}, \rho^{k}, \sigma^{k4})(KY) \notin M_{h2} \wedge (rid^{k}, \rho^{k}, \sigma^{k4})(n_r) \notin M_0 \\
\Rightarrow\ & \{\ \text{Lemma 4}\ \} \\
& \exists_{j4 \in N, j4 < k4, (rid^{j}, \rho^{j}, \sigma^{j4}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm}\ : \\
& \quad \alpha_{j4} = (rid^{j}, \rho^{j}, \sigma^{j4}, send_l(a, b, m)) \\
& \quad \wedge (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY}) \sqsubseteq (rid^{j}, \rho^{j}, \sigma^{j4})(m)
\end{aligned}
$$

De mogelijkheid dat deze term geleerd is uit een de send zoals beschreven in 2.2 en de sleutel bekend is bij een intruder betekent dat term $(rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY})$ al eerder in de kennis van de intruder bevat moet zijn. Dit leidt dus weer naar deelbewijs 2. Er moet dus wel een moment geweest zijn dat de sleutel niet bekend was bij de intruder.

geval 2 (sleutel onbekend)

$$
\begin{aligned}
& (rid^{kk}, \rho^{kk}, \sigma^{h2})(K(r,s)) \notin M_{h2} \\
\Rightarrow\ & \{\ \text{Lemma 4, mogelijke matchen en VII}\ \} \\
& \exists_{g2 \in N, g2 < h2, (rid^{ii}, \rho^{ii}, \sigma^{g2}) \in Inst}\ : \\
& \quad \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(s, r, (\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}))) \\
& \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^{g2})(\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}) \\
& \quad = (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \\
& \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{h3})(T) \\
& \quad = (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY})
\end{aligned}
$$

$$
\begin{array}{l}
\vee \\
\exists_{g3 \in N, g3<h3, (rid^{kkk}, \rho^{kkk}, \sigma^{g3}) \in Inst} : \\
\quad \alpha_{g3} = (rid^{kkk}, \rho^{kkk}, \sigma^{g3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \\
\quad \wedge (rid^{kkk}, \rho^{kkk}, \sigma^{g3})(T) \\
\quad = (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})
\end{array}
$$

Hierboven worden twee mogelijk matchen beschreven de eerste mogelijkheid valt af omdat de lokale sleutel KY niet gelijk kan zijn aan de globale sleutel $K(i,s)$. De tweede mogelijkheid leidt ook tot een tegenspraak. Dit kunnen we aan de hand van deelbewijs 4 concluderen.

We hebben nu laten zien dat geval 2.2 tot een tegenspraak leidt. Vervolgens zullen we laten zien dat geval 2.3 ook tot een tegenspraak leidt.

ad 2.3

$$
\begin{array}{ll}
\Rightarrow & \{ \text{ kies: } h3 \in N, (rid^{kk}, \rho^{kk}, \sigma^{h3}) \in Inst \text{ zodat: } h3 < k4 \\
& \quad (rid^{kk}, \rho^{kk}, \sigma^{h3})(\{Y\}_{KY}) = (rid^{k}, \rho^{k}, \sigma^{k4})(\{n_r\}_{KY}) \ \} \qquad \text{(VIII)} \\
& \alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \\
\Rightarrow & \{ \text{ Lemma 1, kies: } h2 \in N, \sigma^{h2}, \text{ zodat: } h2 < h3 \wedge \sigma^{h2} \subseteq \sigma^{h3} \ \} \\
& \alpha_{h2} = (rid^{kk}, \rho^{kk}, \sigma^{h2}, read_2(s, r, \{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})) \\
\Rightarrow & \{ \text{ Lemma 2 } \} \\
& (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \in M_{h2} \\
\Rightarrow & \{ \text{ gelijkheid VIII en Deelbewijs 1 } \} \\
& (rid^{kk}, \rho^{kk}, \sigma^{h2})(K(r,s)) \notin M_{h2} \\
\Rightarrow & \{ \text{ Lemma 4, 1 mogelijk match en gelijkheden I en VIII } \} \\
& \exists_{h2 \in N, h2<h2, (rid^{ii}, \rho^{ii}, \sigma^{h2}) \in Inst} : \\
& \quad \alpha_{h2} = (rid^{ii}, \rho^{ii}, \sigma^{h2}, send_2(i, r, (\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}))) \\
& \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^{h2})(\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}) \\
& \quad = (rid^{kk}, \rho^{kk}, \sigma^{h2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \\
& \quad \wedge (rid^{ii}, \rho^{ii}, \sigma^{h2})(kir) \\
& \quad = (rid^{kk}, \rho^{kk}, \sigma^{h2})(KY)
\end{array}
$$

$$
\begin{array}{ll}
& = (rid^k, \rho^k, \sigma^{k4})(KY) \\
& = (rid^i, \rho^i, \sigma^{i2})(kir) \\
\Rightarrow & \{ \text{ Lemma 6 } \} \\
& \wedge (rid^{kk}, \rho^{kk}, \sigma^{h3})(Y) \\
& = (rid^i, \rho^i, \sigma^{i2})(P) \\
& = (rid^k, \rho^k, \sigma^{k4})(n_r) \\
\Rightarrow & \{ \text{ op moment } i2 \text{ bestond } n_r \text{ nog niet } \} \\
& \textit{Tegenspraak}
\end{array}
$$

De mogelijke send-acties zoals beschreven worden in de gevallen 2.2 en 2.3 leiden beide tot een tegenspraak. We mogen dan ook concluderen dat als er een send is geweest dit alleen de send kan zijn zoals wordt beschreven in geval 2.1.

Ad deelbewijs 3

Uit Deelbewijs 1 en de gelijkheden I en II weten we dat het volgende geldt.

$$(rid^j, \rho^j, \sigma^{j3})(KX) = (rid^k, \rho^k, \sigma^{k4})(KY) = (rid^i, \rho^i, \sigma^{i2})(kir) \notin M_{k4}$$

Dit gegeven is van belang in het nu volgende bewijs.

$$
\begin{array}{ll}
& (rid^j, \rho^j, \sigma^{j3})(\{i, r, n_i, KX\}_{K(i,s)}) \in M_{j3} \\
\Rightarrow & \{ \ (rid^j, \rho^j, \sigma^{j3})(KX) \notin M_{j3} \ \} \\
& (rid^j, \rho^j, \sigma^{j3})(K(i,s)) \notin M_{j3} \\
\Rightarrow & \{ \text{ Lemma 4 } \} \\
& \exists_{h3 \in N, h3 < j3, (rid^{kk}, \rho^{kk}, \sigma^{h3}) \in Inst, l \in Label, a,b \in Role, m \in RoleTerm} : \\
& \quad \alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_l(a,b,m)) \\
& \quad \wedge (rid^j, \rho^j, \sigma^{j4})(\{i, r, n_i, KX\}_{K(i,s)}) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{h3})(m) \\
\Rightarrow & \{ \text{ mogelijke matchen } \} \\
& \exists_{h3 \in N, h3 < j3, (rid^{kk}, \rho^{kk}, \sigma^{h3}) \in Inst} : \\
& \quad \alpha_{h3} = (rid^{kk}, \rho^{kk}, \sigma^{h3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \\
& \quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{h3})(T) = (rid^j, \rho^j, \sigma^{j3})(\{i, r, n_i, KX\}_{K(i,s)}) \qquad (3.1) \\
& \vee
\end{array}
$$

$$
\begin{aligned}
&\exists_{g2<k4,(rid^{ii},\rho^{ii},\sigma^{g2})\in Inst}: \\
&\quad \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(s,r,m)) \\
&\quad m = \{\{i,r,P,kir\}_{K(i,s)}, \{i,r,P,kir\}_{K(r,s)}\}_{K(r,s)} \wedge \\
&\qquad ((rid^{ii}, \rho^{ii}, \sigma^{g2})(\{i,r,P,kir\}_{K(i,s)}) \\
&\qquad = (rid^{j}, \rho^{j}, \sigma^{j3})(\{i,r,n_i,KX\}_{K(i,s)}) \qquad (3.2a) \\
&\quad \vee \\
&\qquad (rid^{ii}, \rho^{ii}, \sigma^{g2})(\{i,r,P,kir\}_{K(r,s)}) \\
&\qquad = (rid^{j}, \rho^{j}, \sigma^{j3})(\{i,r,n_i,KX\}_{K(i,s)})) \qquad (3.2b)
\end{aligned}
$$

We zullen laten zien dat gevallen 3.2a en 3.2b tot een tegenspraak leiden zodat er wel een $send_3$ geweest moet zijn.

Ad (3.2a/b)

$$
\begin{aligned}
&\circ \quad \{ \text{ herhaling en gelijkheden I en II } \} \\
&\exists_{g2<k4,(rid^{ii},\rho^{ii},\sigma^{g2})\in Inst}: \\
&\quad \alpha_{g2} = (rid^{ii}, \rho^{ii}, \sigma^{g2}, send_2(s,r,m)) \\
&\quad m = \{\{i,r,P,kir\}_{K(i,s)}, \{i,r,P,kir\}_{K(r,s)}\}_{K(r,s)} \wedge \\
&\qquad ((rid^{ii}, \rho^{ii}, \sigma^{g2})(\{i,r,P,kir\}_{K(i,s)}) \\
&\qquad = (rid^{j}, \rho^{j}, \sigma^{j3})(\{i,r,n_i,KX\}_{K(i,s)}) \\
&\quad \vee \\
&\qquad (rid^{ii}, \rho^{ii}, \sigma^{g2})(\{i,r,P,kir\}_{K(r,s)}) \\
&\qquad = (rid^{j}, \rho^{j}, \sigma^{j3})(\{i,r,n_i,KX\}_{K(i,s)})) \\
&\quad \wedge (rid^{ii}, \rho^{ii}, \sigma^{g2})(KX) = (rid^{j}, \rho^{j}, \sigma^{j3})(KX) = (rid^{i}, \rho^{i}, \sigma^{i2})(kir) \\
\Rightarrow \quad &\{ \text{ Lemma 6 } \} \\
&\quad \alpha_{i2} = (rid^{i}, \rho^{i}, \sigma^{i2}, send_2(s,r,m)) \\
&\quad m = \{\{i,r,P,kir\}_{K(i,s)}, \{i,r,P,kir\}_{K(r,s)}\}_{K(r,s)} \wedge \\
&\qquad ((rid^{i}, \rho^{i}, \sigma^{i2})(\{i,r,P,kir\}_{K(i,s)}) \\
&\qquad = (rid^{j}, \rho^{j}, \sigma^{j3})(\{i,r,n_i,KX\}_{K(i,s)}) \\
&\quad \vee \\
&\qquad (rid^{i}, \rho^{i}, \sigma^{i2})(\{i,r,P,kir\}_{K(r,s)}) \\
&\qquad = (rid^{j}, \rho^{j}, \sigma^{j3})(\{i,r,n_i,KX\}_{K(i,s)}))
\end{aligned}
$$

Een intruder kan de term $(rid^{j}, \rho^{j}, \sigma^{j3})(\{i,r,n_i,KX\}_{K(i,s)})$ alleen uit een $send_2$ geleerd hebben als deze ook in het bezit is van de sleutel $(rid^{i}, \rho^{i}, \sigma^{i2})(K(r,s))$. Maar als deze

sleutel bij de intruder bekend is, dan is ook de term $(rid^i, \rho^i, \sigma^{i2})(kir)$ bekend bij de intruder. Echter uit deelbewijs 1 weten we dat deze term niet bekend is bij de intruder. We hebben dus een tegenspraak voor de mogelijkheden 3.2a en 3.2b afgeleid.

Als de term $(rid^j, \rho^j, \sigma^{j3})(\{i, r, n_i, KX\}_{K(i,s)})$ bekend is bij een intruder dan kan dit alleen door middel van een $send_3$.

Ad deelbewijs 4

$$
\begin{aligned}
&\exists_{g3\in N, g3<k2, (rid^{kk},\rho^{kk},\sigma^{g3})\in Inst, t\in RoleTerm} : \\
&\quad \alpha_{g3} = (rid^{kk}, \rho^{kk}, \sigma^{g3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \\
&\quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{g3})(t) \notin M_{g3} \wedge (rid^{kk}, \rho^{kk}, \sigma^{g3})(t) \sqsubseteq (rid^{kk}, \rho^{kk}, \sigma^{g3})(T) \\
&\quad \wedge (rid^{kk}, \rho^{kk}, \sigma^{g3})(T) = (rid^{k}, \rho^{k}, \sigma^{k2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \\
\Rightarrow\ &\quad \{\ \text{kies: } g3 \in N, (rid^{kk}, \rho^{kk}, \sigma^{g3}) \in Inst, \text{ zodat: } g3 < k2 \\
&\qquad \wedge (rid^{kk}, \rho^{kk}, \sigma^{g3})(T) = (rid^{k}, \rho^{k}, \sigma^{k2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})\ \} \qquad \text{(IX)} \\
&\alpha_{g3} = (rid^{kk}, \rho^{kk}, \sigma^{g3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \\
\Rightarrow\ &\quad \{\ \text{Lemma 1}\ \} \\
&\alpha_{g2} = (rid^{kk}, \rho^{kk}, \sigma^{g2}, read_2(s, r, \{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})) \\
\Rightarrow\ &\quad \{\ \text{Lemma 2}\ \} \\
&(rid^{kk}, \rho^{kk}, \sigma^{g2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \in M_{g2} \\
\Rightarrow\ &\quad \{\ \text{Lemma 4, mogelijke matchen, } (rid^{kk}, \rho^{kk}, \sigma^{g3})(t) \notin M_{g3}\ \} \\
&\quad \exists_{f2\in N, f2<g2, (rid^{ii},\rho^{ii},\sigma^{f2})\in Inst} : \\
&\qquad \alpha_{f2} = (rid^{ii}, \rho^{ii}, \sigma^{f2}, send_2(s, r, (\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}))) \\
&\qquad \wedge (rid^{ii}, \rho^{ii}, \sigma^{f2})(\{\{i, r, P, kir\}_{K(i,s)}, \{i, r, P, kir\}_{K(r,s)}\}_{K(r,s)}) \\
&\qquad = (rid^{kk}, \rho^{kk}, \sigma^{g2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) \\
&\quad \vee \\
&\quad \exists_{f3\in N, f3<g2, (rid^{kkk},\rho^{kkk},\sigma^{f3})\in Inst} : \\
&\qquad \alpha_{f3} = (rid^{kkk}, \rho^{kkk}, \sigma^{f3}, send_3(r, i, (T, \{Y\}_{KY}, n_r))) \wedge \\
&\qquad \wedge (rid^{kkk}, \rho^{kkk}, \sigma^{f3})(T) \\
&\qquad = (rid^{kk}, \rho^{kk}, \sigma^{g3})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)})
\end{aligned}
$$

De eerste mogelijkheid hierboven, het geval dat er een $send_2$ heeft plaats gevonden betekent het volgende. (zie gelijk IX)

$$(rid^{k}, \rho^{k}, \sigma^{k2})(\{T, \{i, r, Y, KY\}_{K(r,s)}\}_{K(r,s)}) = (rid^{kk}, \rho^{kk}, \sigma^{g3})(T)$$

Wil er een $send_2$ plaats gevonden hebben, dan moet deze term matchen met.

$$(rid^{ii}, \rho^{ii}, \sigma^{f2})(\{i, r, P, kir\}_{K(i,s)})$$

Het zal duidelijk zijn dat er geen match optreedt. We mogen dus concluderen dat er een $send_2$ heeft plaats gevonden.

De mogelijkheid van een $send_3$ kunnen we eenvoudig uitsluiten. Als er een $send_3$ plaats gevonden zou hebben dan wijst dit recursief naar de mogelijkheid zoals we in dit deelbewijs beschouwen. We weten dat $(rid^k, \rho^k, \sigma^{k2})(T) \notin M_0$. Dus moet er ooit wel een $send_2$ geweest zijn, maar die mogelijkheid leverde nu juist een tegenspraak op. We mogen dus concluderen dat we een tegenspraak hebben afgeleid.

Hoofdstuk 5

Conclusie

In het laatste hoofdstuk evalueren we het project om in het kort aan te geven wat de grootste knelpunten geweest zijn. Vervolgens zullen we enkele punten bespreken die het meest opgevallen zijn tijdens het maken van de afleidingen. We gaan hierbij onder andere in op de structuur en complexiteit van de bewijzen voor non-injectieve synchronisatie. We besluiten het hoofdstuk met enkele aanbevelingen voor toekomstig onderzoek.

5.1 Evaluatie opdracht

Om een goed beeld te krijgen van security-protocollen en wat anderen reeds gedaan hebben op het gebied van formeel afleiden van eigenschappen aan deze protocollen, zijn we begonnen met veel te lezen over deze protocollen. Er zijn in de literatuur weinig formele definities van de veiligheidseisen te vinden en de definities die te vinden zijn verschillen allemaal licht van elkaar. Ook op het gebied van het bewijzen van correctheid van eigenschappen voor deze protocollen heeft men slechts weinig gedaan.

Aan de hand van de definitie uit [6] hebben we een begin gemaakt met het formeel verifiëren van een klein en simpel protocol. Het bleek niet eenvoudig te zijn om een bewijs te geven voor een simpel protocol. De manier waarop we de intruder modelleren en aanname over de betrouwbaarheid van agenten blijken van groot belang te zijn bij het afleiden van non-injectieve synchronisatie voor een protocol.

Bij het afleiden van het eerste protocol uit SPORE [14] bleek het niet voldoende te zijn om het rol- en runniveau met slechts een runidentifier te scheiden. We hebben functies zoals *Inst* geïntroduceerd waardoor het mogelijk geworden is om een volledig onderscheid te maken tussen deze twee niveau's. Als gevolg van dit onderscheid is het mogelijk om relatief vlot een afleiding te geven van de protocollen uit SPORE.

5.2 Structuur van de bewijzen

De bewijzen die wij in het vorige hoofdstuk voor non-injectieve synchronisatie gegeven hebben, vertonen een duidelijke structuur. Een strategie die het mogelijk maakt om dit soort bewijzen te geven, ziet er als volgt uit.

1. We veronderstellen een claim voor non-injectieve synchronisatie voor één van de rollen.
2. We veronderstellen dat er een instantie bestaat voor de rol die deze claim gedaan heeft. Teven nemen we aan dat de agenten waarmee deze instantie meent te communiceren te vertrouwen zijn.
3. Met behulp van Lemma 1 leiden we de acties af die de zojuist gevonden claim-actie voorafgegaan zijn.
4. Op de gevonden read-acties uit de run die we zojuist hebben afgeleid passen we Lemma 2 toe. Met behulp van dit lemma leiden we af dat de gelezen termen ook bekend moeten zijn bij een intruder.
5. We proberen af te leiden dat deze termen niet initieel in de intruderkennis kunnen voorkomen.
6. We proberen af te leiden dat de intruder deze termen niet gemaakt kan hebben. Stel een term van de vorm $\{t\}_k$ komt in de intruderkennis voor, dan kan de volgende strategie gebruikt worden om af te leiden dat de intruder deze term niet gemaakt kan hebben.
   (a) We leiden af dat de subterm t niet bekend is bij de intruder.
   (b) We leiden af dat de term k een geheime sleutel is, die de intruder niet bezit.
7. Indien we voor een bepaalde term kunnen afleiden dat deze niet initieel bekend is bij een intruder en ook niet gecreëerd kan zijn door een intruder, weten we dat er ooit een agent geweest is die deze term verzonden heeft. Met de lemma's 3, 4 of 5 zijn we nu in staat om een send-actie af te leiden.
8. Met behulp van de matchfunctie kunnen we bepalen welke send-acties uit de protocolbeschrijving in aanmerking komen voor het sturen van de bewuste term.
9. Indien er meer dan één send-actie in aanmerking komt, moeten we aantonen dat alleen de juiste send-actie plaats gevonden kan hebben. Voor deze send-actie kiezen we een instantie. We hebben nu een bij elkaar passende send- en read-actie gevonden.
10. Indien we meerdere instanties hebben gevonden die dezelfde rol representeren, proberen we aan te tonen dat deze instanties gelijk aan elkaar zijn. Mochten we hierin niet slagen dan passen we Lemma 1 op de gevonden send-actie toe en keren we terug naar stap 4.

11. Deze strategie herhalen we net zolang totdat we voor elke read-actie een bijbehorende send-actie gevonden hebben en dat we voor elke rol slechts één instantie afgeleid hebben die deze rol representeert.

12. Het is mogelijk dat we nog een afleiding voor de volgorde waarin acties hebben plaats gevonden moeten geven.

Met de hierboven beschreven strategie hebben we een structuur gevonden voor het bewijzen van non-injectieve synchronisatie in een security-protocol. Een volgende stap zou zijn deze structuur formeel te beschrijven. Voor deze formele beschrijving zou getracht kunnen worden bovenstaande structuur door middel van een logica te beschrijven.

5.3 Tickets

In paragraaf 4.1 zien we dat veel protocollen gebruik maken van een ticket. Voor de definitie die wij hanteren voor een ticket is de eis voor non-injectieve synchronisatie te sterk. In alle protocollen die wij hebben gemerkt als problematisch vanwege het gebruik van een ticket, leert de intruder het ticket op twee of meer manieren als het gevolg van verschillende send-acties. Deze protocollen zullen niet non-injectief synchroniseren. De meeste van deze protocollen hebben als doel de geheimhouding van een bepaalde term. Het niet optreden van non-injectieve synchronisatie als gevolg van een ticket wil niet zeggen dat de geheimhouding geschonden wordt. Het zou interessant zijn om deze protocollen binnen hetzelfde model te bestuderen voor geheimhouding van bepaalde termen.

Op het moment dat *logging* geen issue is, kunnen we ons afvragen waarom een protocol gebruik maakt van tickets. Immers de rol die het ticket doorstuurt kan niets controleren aan het ticket. Een mogelijk verklaring voor het gebruik van tickets kan de reductie van het aantal communicaties zijn. Voor deze protocollen kunnen we overwegen om het protocol te wijzigen zodat een ticket rechtstreeks wordt gestuurd en er geen sprake meer is van het gebruik van een ticket. Een wijziging aan een protocol levert echter een andere protocol zodat het lastiger wordt om een uitspraak te doen over het oorspronkelijke protocol. We prefereren daarom de tweede oplossing, een alternatieve definitie van non-injectieve synchronisatie voor deze protocollen. Deze definitie zou voor een ticket moeten garanderen dat het ticket dat verzonden wordt hetzelfde is als het ticket dat uiteindelijk ontvangen wordt. Hierbij is het niet van belang welk bericht de partij ontvangt die het ticket doorstuurt, alleen dat het ticket ongeschonden aankomt bij de uiteindelijke ontvanger.

We hebben in paragraaf 4.9 het Kao Chow protocol bestudeerd. Dit protocol synchroniseert niet non-injectief omdat het een ticket bevat. In dit protocol is een kleine wijziging aangebracht zodat het wel non-injectief synchroniseert. De wijziging die in het Kao Chow protocol toegepast wordt, is waarschijnlijk toe te passen in meer protocollen die gebruik

maken van een ticket. Dit is echter alleen interessant indien non-injectieve synchronisatie van belang is. We zien dat door het gebruik van een ticket veel meer mogelijke matchen in het bewijs ontstaan. Indien er non-injectieve synchronisatie optreedt kan er maar één match de juiste zijn. Hierdoor zullen de bewijzen van de protocollen die een ticket bevatten veel langer worden omdat aangetoond moet worden dat de juiste match plaatsgevonden heeft.

5.4 Lemma's

De construeerde lemma's zijn van groot nut geweest bij het maken van de afleidingen. Deze lemma's zullen ook goed bruikbaar zijn als men aan de slag gaat om andere veiligheidseisen dan non-injectieve synchronisatie te bewijzen binnen hetzelfde model. De Lemma's 2, 3, 4 en 5 zijn gebaseerd op het Dolev-Yao intrudermodel. Indien een ander intrudermodel gekozen wordt zullen deze lemma's aangepast moeten worden.

5.5 Complexiteit van de bewijzen

Het is ons opgevallen dat de complexiteit van de bewijzen die we in het vorige hoofdstuk gegeven hebben sterk varieert. We willen graag iets meer proberen te zegen over deze complexiteit. Daarom bestuderen we enkele eigenschappen van deze protocollen en of bewijzen waarvan we vermoeden dat ze ons meer kunnen vertellen over de complexiteit van het bewijs.

In Tabel 5.1 hebben we een overzicht gegeven van de protocollen waarvoor we een afleiding van non-injectieve synchronisatie gegeven hebben. We meten de lengte van een bewijs met het aantal pagina's dat we nodig hebben gehad voor dit bewijs. Op het moment dat het noodzakelijk is om geheimhouding van een bepaalde term aan te tonen of het uitsluiten van diverse send-acties hebben wij dit bewijs ondergebracht in een deelbewijs. Om deze reden achten wij het interessant om het aantal pagina's dat het deelbewijs telt, apart weer te geven.

We kunnen niet garanderen dat de bewijzen die wij gegeven de kortste zijn. Daarom moeten we voorzichtig zijn met het afleiden van conclusies over de complexiteit op basis van de lengte van een bewijs.

5.5.1 Rollen en berichten

Als we in de Tabel 5.1 het aantal pagina's dat het bewijs telt bekijken zien we dat hier grote onderlinge verschillen tussen bestaan. Echter als we dit aantal pagina's verminderen

protocol	par	#rol	#ber	encr	geheim	#pag
BAN CCITT X.509	4.7	2	3	a	nee	5
Andrew Lowe	4.5	2	3	s	ja	5 (1)
Andrew BAN	4.6	2	4	s	nee	5
NSL	4.3	2	3	a	ja	8 (4)
Yahalom Lowe	4.4	3	5	s	ja	8 (2)
NSSK	4.8	3	5	s	ja	10 (3)
Kao Chow (gewijzigd)	4.9	3	4	s	ja	16 (10)

par = *paragraaf waar het bewijs terug te vinden is*
#rol = *aantal rollen dat voorkomt in het protocol*
#ber = *aantal berichten dat voorkomt in het protocol*
encr = *a voor asymmetrische encryptie en een s voor symmetrische encryptie*
geheim = *of er een bewijs voor geheimhouding van een term noodzakelijk is*
#pag = *totaal aantal pagina's dat het bewijs telt*
tussen haakjes het aantal pagina's dat het deelbewijs telt.

Tabel 5.1: Complexiteitstabel

met het aantal pagina's dat het deelbewijs telt, zien we dat de verschillen niet meer zo groot zijn. Voor protocollen waarin meer rollen en berichten voorkomen zien we dat de lengte van het bewijs licht toeneemt.

Bij een toename van het aantal berichten, moet er voor meer send- en read-acties gelijkheid afgeleid worden. De toename van de lengte van het bewijs is hier een logisch gevolg van. In protocollen waarin meer rollen voorkomen valt het op, dat veel werk gedaan moet worden om gelijkheid van verschillende instanties van dezelfde rollen aan te tonen. Een duidelijk voorbeeld hiervan is het NSSK protocol in paragraaf 4.8.

We vermoeden dat de bewijzen voor protocollen met meer berichten ongeveer lineair in lengte stijgen. Bij een toename van het aantal rollen zal dit volledig afhankelijk zijn van de hoeveelheid werk die nodig is, om gelijkheid van de verschillende instanties van dezelfde rollen aan te tonen.

5.5.2 Encryptie

De encryptiemethode lijkt op het eerste gezicht nauwelijks van invloed te zijn op de lengte van een bewijs. Op het moment dat alleen gebruik wordt gemaakt van globale symmetrische sleutel die geheim is, zien we dat het bewijs erg kort is. We zien dit bijvoorbeeld bij het bewijs van het Andrew BAN protocol.

5.5.3 Geheimhouding

Zoals in Tabel 5.1 te zien is, steunen veel bewijzen op de geheimhouding van één of meer termen. Op basis van onder andere het geheim zijn van deze termen is het mogelijk om een afleiding voor non-injectieve synchronisatie te geven. Met uitzondering van het deelbewijs van het Kao Chow protocol betreffen alle andere deelbewijzen alleen bewijzen voor de geheimhouding van een term.

Voor het aantonen van geheimhouding van een term, bestuderen we de mogelijkheden waarop deze term geleerd kan zijn door een intruder. We zien dat als het aantal matchen voor deze termen toeneemt, de lengte van het bewijs ook stijgt. Vanzelfsprekend neemt de lengte van het bewijs toe als voor meer termen in een protocol geheimhouding aangetoond moet worden.

In het algemeen is het lastig iets te zeggen over de complexiteit van een bewijs. We zien dat een bewijs complex wordt als de berichten weinig informatie bevatten zodat het veel werk is om gelijkheid van instanties aan te tonen. Het afleiden dat termen geheim blijven voor een intruder wordt ingewikkelder naarmate er meer matchen mogelijk zijn. Uiteraard is ook een combinatie van deze facturen mogelijk.

5.6 Aanbevelingen

We hebben enkele zaken besproken die opgevallen zijn bij het maken van afleidingen voor non-injectieve synchronisatie. Het is duidelijk dat er diverse punten zijn, waarvoor het interessant is om in de toekomst meer aandacht aan te besteden.

5.6.1 Automatisch bewijzen

Het bekijken van de mogelijkheden van automatische bewijzen en het implementeren hiervan was oorspronkelijk een nevendoel van het project. In een vrij vroeg stadium is het ons al duidelijk geworden dat we dit doel niet zouden halen.

We hebben gezien dat de bewijzen vrij snel complex worden. Voor protocollen die veel groter zijn dan de protocollen die wij bestudeerd hebben, zal het ondoenlijk zijn om handmatig een afleiding te geven. Tevens is een fout in een handmatige afleiding snel gemaakt. De mogelijkheid om automatisch een afleiding te geven zou zeer wenselijk zijn.

We hebben laten zien dat er een structuur bestaat voor de bewijzen van non-injectieve synchronisatie. Om in de bewijzen te automatiseren zal deze structuur eerst geformaliseerd moeten worden. Vervolgens kan bekeken worden of het mogelijk is deze formalisatie te implementeren. Dit zal niet eenvoudig zijn omdat in de bewijzen vaak keuzes gemaakt worden op basis van inzicht dat wij in een protocol hebben.

5.6.2 Tijd modelleren

In het model dat gebruikt is voor het afleiden van non-injectieve synchronisatie is geen notie van tijd opgenomen. Als we de lijst van protocollen in paragraaf 4.1 bekijken, zien we dat veel protocollen wel gebruik maken van tijd. Het is dus zeker interessant te bekijken of het mogelijk is om het bestaande model uit te breiden met tijd.

5.6.3 Type flaw attacks

In het model dat hier gebruikt wordt is aangenomen dat agenten in staat zijn de typen en patronen van berichten te herkennen. Het gevolg hiervan is dat een "type flaw attack" niet relevant is in dit model. Het zou zeker reëel zijn om de mogelijkheden van een type aanval mee te nemen in het model.

Deze aanpassingen kunnen eenvoudig in het model gedaan worden, het is voldoende om slechts de matchfunctie aan te passen. Op het moment dat agenten niet meer in staat zijn om te herkennen of termen van het juiste type zijn, ontstaan er veel meer mogelijkheden voor een match bij het afleiden van een send. Er zal aangetoond moeten worden dat alleen de juiste send plaats gevonden heeft. Zoals we al gezien hebben bij het bewijs voor de gewijzigde versie van het Kao Chow protocol levert dit veel extra werk op. De bewijzen voor non-injectieve synchronisatie worden dus een stuk complexer indien agenten niet meer in staat zijn type te herkennen.

Bijlage A

Protocolbeschrijvingen uit SPORE

In deze appendix worden de protocolbeschrijvingen uit SPORE [14] gegeven van de protocollen, waarvan we in hoofdstuk 4 het bewijs voor non-injectieve synchronisatie gegeven hebben.

A.1 Lowe's fixed version of Needham-Schroder Public Key

```
A,B,S :                          Principal
Na,Nb :                          Nonce
KPa,KPb,KPs,KSa,KSb,KSs :        Key
KPa,KSa :                        is a key pair
KPb,KSb :                        is a key pair
KPs,KSs :                        is a key pair
```

```
1.   A -> S  :   A,B
2.   S -> A  :   {KPb, B}KSs
3.   A -> B  :   {Na, A}KPb
4.   B -> S  :   B,A
5.   S -> B  :   {KPa, A}KSs
6.   B -> A  :   {Na, Nb, B}KPa
7.   A -> B  :   {Nb}KPb
```

A.2 Lowe's modified version of Yahalom

```
A, B, S :          principal
Na, Nb :           number fresh
Kas, Kbs, Kab :    key

A knows :   A, B, S, Kas
B knows :   B, S, Kbs
S knows :   S, A, B, Kas, Kbs


1.   A -> B  :   A, Na
2.   B -> S  :   {A, Na, Nb}Kbs
3.   S -> A  :   {B, Kab, Na, Nb}Kas
4.   S -> B  :   {A, Kab}Kbs
5.   A -> B  :   {A, B, S, Nb}Kab
```

A.3 Lowe modified BAN concrete Andrew Secure RPC

```
A, B :          principal
Kab, K'ab :     symkey
Na, Nb, N'b :   nonce
succ :          nonce -> nonce


1.   A -> B  :   A, Na
2.   B -> A  :   {Na, K'ab, B}Kab
3.   A -> B  :   {Na}K'ab
4.   B -> A  :   Nb
```

A.4 BAN modified Andrew Secure RPC

```
A, B :          principal
Kab, K'ab :     symkey
Na, Nb, N'b :   nonce
succ :          nonce -> nonce


1.   A -> B  :   A, {Na}Kab
2.   B -> A  :   {succNa, Nb}Kab
3.   A -> B  :   {succNb}Kab
4.   B -> A  :   {K'ab, N'b, Na}Kab
```

A.5 BAN modified version of CCITT X.509 (3)

```
A, B :     principal
Na, Nb :   nonce
Ya, Yb :   userdata
Xa, Xb :   userdata
PK, SK :   principal -> key (keypair)


1.   A -> B  :   A, {Na, B, Xa, {Ya}PK(B)}SK(A)
2.   B -> A  :   B, {Nb, A, Na, Xb, {Yb}PK(A)}SK(B)
3.   A -> B  :   A, {B, Nb}SK(A)
```

A.6 Needham Schroeder Symmetric Key

```
A, B, S :          principal
Na, Nb :           nonce
Kas, Kbs, Kab :    key
dec :              nonce -> nonce
```

```
1.   A -> S  :   A, B, Na
2.   S -> A  :   {Na, B, Kab, {Kab, A}Kbs}Kas
3.   A -> B  :   {Kab,A}Kbs
4.   B -> A  :   {Nb}Kab
5.   A -> B  :   {dec(Nb)}Kab
```

A.7 Kao Chow Authentication v.1

```
A, B, S :          principal
Na, Nb :           number
Kab, Kbs, Kas :    key
```

```
1.   A -> S  :   A, B, Na
2.   S -> B  :   {A, B, Na, Kab}Kas, {A, B, Na, Kab}Kbs
3.   B -> A  :   {A, B, Na, Kab}Kas, {Na}Kab, Nb
4.   A -> B  :   {Nb}Kab
```

Bijlage B

Lijst van definities

De onderstaande lijst van definities wordt gebruikt om het model in hoofdstuk 3 te beschrijven. Dit model is volledig gebaseerd op het model dat in [5] wordt beschreven. Om deze reden zijn diverse definities letterlijk uit de artikel overgenomen. Deze definities markeren we met een ster (*). Sommige definities zijn gewijzigd omdat wij het model uit het artikel ([5]) aangepast hebben. Deze definities zijn gemarkeerd met twee sterren (**).

Referenties

[1] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8:18–36, 1990.

[2] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. Technical Report 39, Digital Systems Research Center, february 1989.

[3] J.A. Clark and J.L. Jacob. A survey of authentication protocol literature. Technical Report 1.0, 1997.

[4] E. Clarke, S. Jha, and W. Marrero. Partial order reductions for security protocol verification. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 1785 of *Lecture Notes in Computer Science*, pages 503–518. Springer, 2000.

[5] C.J.F. Cremers and S. Mauw. Operational semantics of security protocols. Submitted for publication.

[6] C.J.F. Cremers, S. Mauw, and E.P. de Vink. Defining authentication in a trace model. In Theo Dimitrakos and Fabio Martinelli, editors, *FAST 2003*, Proceedings of the first international Workshop on Formal Aspects in Security and Trust, pages 131–145, Pisa, September 2003. IITT-CNR technical report.

[7] D. Dolev and A.C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, IT-29(12):198–208, March 1983.

[8] I.-Lung Kao and Randy Chow. An efficient and secure authentication protocol using uncertified keys. *SIGOPS Oper. Syst. Rev.*, 29(3):14–21, 1995.

[9] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proceedings of TACAS*, volume 1055, pages 147–166. Springer Verlag, 1996.

[10] Gavin Lowe. Some new attacks upon security protocols. In IEEE Computer Society Press, editor, *In Proceedings of the Computer Security Foundations Workshop VIII*, 1996.

[11] Gavin Lowe. Towards a completeness result for model checking of security protocols. Technical Report 1998/6, Dept. of Mathematics and Computer Science, University of Leicester, 1998.

[12] S. Mauw and V. Bos. Drawing Message Sequence Charts with LaTeX. *TUGBoat*, 22(1-2):87–92, March/June 2001.

[13] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12):993–999, 1978.

[14] SPORE. *Security Protocols Open Repository.* http://www.lsv.ens-cachan.fr/spore/.

[15] Thomson. Smartright technical white paper v1.0. Technical report, Thomson, october 2001. http://www.smartright.org.