Event-B Course

11. Formal Development of a Security Protocol

(the Needham-Schroeder protocol)

Jean-Raymond Abrial

September-October-November 2011

Outline

- Requirement Document

- Refinement Strategy

- Initial Model

- Refinement

Requirement Document (1)

- The Needham-Schroeder Protocol is a security protocol

- R.M. Needham and M.D. Schroeder Using encryption for authentication in large networks of computers. CACM 21 (1978)

- Its role is to allow two agents to communicate on a network

This protocol involves two agents situated on the sites of a network	ENV-1

-The Needham-Schroeder Protocol is an authentication protocol:

- At the end, the two agents must be sure to speak to each other.

Requirement Document (2)

- There is a standard attack to this protocol

- The authentication property cannot always be guaranteed

- This attack was discovered by Lowe

- G. Lowe A Breaking and fixing the Needham-Schroeder public-key protocol using FDR. TACAS 1996 LNCS vol.1055 (1996)

- More on this later

Requirement Document (3)

- One of the agents is called the initiator.

- The other agent is called the recipient.

An execution of the protocol involves two agents: the initiator and the recipient	ENV-2

- The initiator starts the communication with the recipient.

An initiator of the protocol wants to speak to a recipient	ENV-3

Requirement Document (4)

- Many executions of the protocol can occur simultaneously

- Agents use nonces to identify executions of the protocol

- Nonces are guaranteed to be unique

Agents use unique nonces to identify specific executions of the protocol	ENV-4

- An initiator and a recipient nonce are used by an execution

A protocol execution is identified by two nonces	ENV-5

Requirement Document (5)

- Agents communicate by means of messages sent on the network

- The network is supposed to be unsecure

- Bad agents are able to do the following:
  - copy messages between sites
  - remove messages
  - modify messages they can read
  - create messages
  - ...

Bad agents can corrupt the execution of a protocol	ENV-6

Requirement Document (6)

- In spite of bad agents, we want to ensure an important property

- At the end of the execution of the protocol, we want to be sure that:
  - the initiator will speak to the recipient
  - the recipient will speak to the initiator.

- This property is called mutual authentication.

The protocol must ensure *mutual authentication* between initiators and recipients	FUN-1

- This is the main property of this protocol

Requirement Document (7)

- In order to ensure mutual authentication, agents use encryption

Encrypted messages are used for the communication between agents	ENV-7

Requirement Document (8)

- Each agent A has two keys:

  - a public key K_A (known by all agents) to encrypt messages

  - a secret key K_A^{-1} (known by A only) to decrypt messages

Encryption is ensured by means of *public keys* Decryption is ensured by means of *secret keys*	ENV-8

- An agent I can send an encrypted message to another agent R

- If the message is encrypted with K_R, then only R can decrypt it.

Requirement Document (9): the Protocol

- An agent I sent a message to an agent R with public key K_R
- This message contains the name of I and a new nonce N_I
- R decrypts the previous message with secret key K_R^{-1}
- R replies to I by sending a message with public key K_I
- This message contains the nonce N_I and a new nonce N_R
- I decrypts the previous message with secret key K_I^{-1}
- I replies to R by sending a message with public key K_R
- The previous message contains the nonce N_R

1. $I \rightarrow R$: $\{I, N_I\}_{K_R}$ 2. $R \rightarrow I$: $\{N_I, N_R\}_{K_I}$ 3. $I \rightarrow R$: $\{N_R\}_{K_R}$	FUN-2

Mutual Authentication (1)

1. $I \rightarrow R : \{I, N_I\}_{K_R}$ 2. $R \rightarrow I : \{N_I, N_R\}_{K_I}$ 3. $I \rightarrow R : \{N_R\}_{K_R}$	FUN-2

- This protocol seems to guarantee authentication for I.

- At step 2, I receives the message $\{N_I, N_R\}_{K_I}$

- This message contains the nonce N_I

- Nonce N_I was sent by I to R only since encrypted with key K_R

- Nonces are guaranteed to be unique.

- Hence the message $\{N_I, N_R\}_{K_I}$ was certainly sent by R.

Mutual Authentication (2)

1. $I \rightarrow R$: $\{I, N_I\}_{K_R}$ 2. $R \rightarrow I$: $\{N_I, N_R\}_{K_I}$ 3. $I \rightarrow R$: $\{N_R\}_{K_R}$	FUN-2

- This protocol seems to guarantee authentication for R.

- At step 3, R receives the message $\{N_R\}_{K_R}$

- This message contains the nonce N_R

- Nonce N_R was sent by R to I only since encrypted with key K_I

- Nonces are guaranteed to be unique.

- Hence the message $\{N_R\}_{K_R}$ was certainly sent by I.

Attacking the Protocol (Discovered by Lowe) (1)

- An initiator I sends the message $\{I, N_I\}_{K_A}$ to a recipient A

- A happens to be an attacker.

- A decrypts this message and forward it to another recipient R.

- R is misled, it believes to have received a message from I.

- R sends back a message to I as in the normal protocol.

- The initiator I believes to have received a reply from A.

- Therefore I sends to A the acknowledgment message.

- And now A decrypts this message and forward it to R.

Attacking the Protocol (2)

1. $I \rightarrow A \ : \ \{I, N_I\}_{K_A}$
2. $A \rightarrow R \ : \ \{I, N_I\}_{K_R}$
3. $R \rightarrow I \ : \ \{N_I, N_R\}_{K_I}$
4. $I \rightarrow A \ : \ \{N_R\}_{K_A}$
5. $A \rightarrow R \ : \ \{N_R\}_{K_R}$

- At the end, A knows nonces N_I and N_R (step 1 and step 4)

- R also knows nonces N_I and N_R (step 2 and step 5).

- Further messages can then be sent to R by A.

- In such messages, the pair N_I-N_R is a justification.

- R believes such messages come from I

Attacking the Protocol (3)

- Suppose R is a bank

- A could send the following message to R:

$$\{N\_I, N\_R, \text{"Transfer some of my money into A's account"}\}_{K_R}$$

- R, the banks, believes that this message comes from I

- Then R may perform the money transfer from I to A !!!

Correcting the Protocol

- Here is, again, the faulty protocol:

1. $I \rightarrow R : \{I, N_I\}_{K_R}$ 2. $R \rightarrow I : \{N_I, N_R\}_{K_I}$ 3. $I \rightarrow R : \{N_R\}_{K_R}$	FUN-2

- Lowe proposed the following corrected protocol:

1. $I \rightarrow R : \{I, N_I\}_{K_R}$ 2. $R \rightarrow I : \{N_I, N_R, R\}_{K_I}$ 3. $I \rightarrow R : \{N_R\}_{K_R}$	FUN-2

I may check that the message comes from R

Replaying the Attack

1. $I \rightarrow A \ : \ \{I, N_I\}_{K_A}$
2. $A \rightarrow R \ : \ \{I, N_I\}_{K_R}$
3. $R \rightarrow I \ : \ \{N_I, N_R, R\}_{K_I}$

- At step 3, I may figure out that the message does not come from A

-I would expect the following message: $\{N_I, N_R, A\}_{K_I}$

- At this point, I may stop the execution of the protocol

A Proposal for the Formalization (1)

- In order to simplify the formalization, we do the following

  - We suppose that there is no attacker

  - We suppose instead that the initiator I makes a mistake

  - I does not send the msg to R, it sends it to an agent S

1. $I \rightarrow R : \{I, N_I\}_{K_S}$ 2. $S \rightarrow I : \{N_I, N_S\}_{K_I}$ 3. $I \rightarrow R : \{N_R\}_{K_S}$	FUN-3

- It is possible that S is the same as R

A Proposal for the Formalization (2)

- The corrected protocol:

1. $I \rightarrow R : \{I, N_I\}_{K_S}$ 2. $S \rightarrow I : \{N_I, N_S, S\}_{K_I}$ if $S = R$ then 3. $I \rightarrow R : \{N_R\}_{K_R}$	FUN-3

- If S is not the same as R then the protocol is stopped

- Step 3 is never executed.

Refinement Strategy: Two Separate Developments

- Protocol without mistake and no attacker

- Protocol with mistake and no attacker

- In each case:

    - Initial model without messages

    - Refinement with messages

Protocol without mistake

Initial Model: Sets, Constants, and Axioms

- Introducing a set of Agents AGT and a set of Nonces NNC

sets:	AGT
	NNC

constants:	$Initiator$
	$Recipient$

axm_1: **partition**$(AGT, Initiator, Recipient)$

These elements take account of assumptions ENV_1 and ENV_2.

This protocol involves two agents situated on the sites of a network	ENV-1

An execution of the protocol involves two agents: the initiator and the recipient	ENV-2

Initial Model: Variables and Invariants (1)

- Introducing the set of used nonces nnc.

- For simplification, it is partitioned: nni and nnr

variables:	nnc
	nni
	nnr

inv0_1:	$nnc \subseteq NNC$
inv0_2:	**partition**(nnc, nni, nnr)

These elements take partially account of assumption ENV_4.

Agents use unique nonces to identify specific executions of the protocol	ENV-4

Initial Model: Variables and Invariants (2)

- Introducing what is recorded with an initiator ni in the initiator site:
  - the corresponding initiator: $i1(ni)$
  - the corresponding recipient: $i2(ni)$
  - the corresponding recipient nonce: $i3(ni)$

variables: $i1$ $i2$ $i3$

inv0_3: $i1 \in nni \rightarrow Initiator$

inv0_4: $i2 \in nni \rightarrow Recipient$

inv0_5: $i3 \in nni \rightarrowtail nnr$

- $i3$ is only partial. Why?

Initial Model: Variables and Invariants (3)

- Introducing what is recorded with a recipient nr in the recipient site:
  - the corresponding recipient: $r1(nr)$
  - the corresponding initiator: $r2(nr)$
  - the corresponding recipient nonce: $r3(nr)$

variables: $r1$
$r2$
$r3$

inv0_6: $r1 \in nnr \rightarrow Recipient$

inv0_7: $r2 \in nnr \rightarrow Initiator$

inv0_8: $r3 \in nnr \rightarrowtail nni$

Initial Model: Variables and Invariants (4)

- The previous elements take account of ENV_4 and ENV_5.

Agents use unique nonces to identify specific executions of the protocol	ENV-4

A protocol execution is identified by two nonces	ENV-5

Initial Model: Events (1)

We follow the protocol (without sending messages)

1. $I \rightarrow R \ : \ \{I, N_I\}_{K_R}$ 2. $R \rightarrow I \ : \ \{N_I, N_R\}_{K_I}$ 3. $I \rightarrow R \ : \ \{N_R\}_{K_R}$	FUN-2

Event P1 corresponds to step 1 of the protocol

P1
any ni, i, r **where**
$ni \notin ncc$
$i \in Initiator$
$r \in Recipient$
then
$ncc := ncc \cup \{ni\}$
$nni := nni \cup \{ni\}$
$i1 := i1 \cup \{ni \mapsto i\}$
$i2 := i2 \cup \{ni \mapsto r\}$
end

The initiator records some data in step 1

Initial Model: Events (2)

1. $I \rightarrow R : \{I, N_I\}_{K_R}$ 2. $R \rightarrow I : \{N_I, N_R\}_{K_I}$ 3. $I \rightarrow R : \{N_R\}_{K_R}$	FUN-2

Event P2 corresponds to step 2.

P2
any ni, i, r, nr **where**
$ni \in nni \setminus \mathsf{ran}(r3)$
$i = i1(ni)$
$r = i2(ni)$
$nr \notin ncc$
then
$ncc := ncc \cup \{nr\}$
$nnr := nnr \cup \{nr\}$
$r1 := r1 \cup \{nr \mapsto r\}$
$r2 := r2 \cup \{nr \mapsto i\}$
$r3 := r3 \cup \{nr \mapsto ni\}$
end

The recipients records some data in step 2

Initial Model: Events (3)

1. $I \rightarrow R : \{I, N_I\}_{K_R}$ 2. $R \rightarrow I : \{N_I, N_R\}_{K_I}$ 3. $I \rightarrow R : \{N_R\}_{K_R}$	FUN-2

Event P3 corresponds to an initiator receiving the message sent in step 2

P3
any ni, nr **where**
$nr \mapsto ni \in r3$
$ni \notin dom(i3)$
then
$i3 := i3 \cup \{ni \mapsto nr\}$
end

The initiator finalizes his records

Cheating

P2

any ni, i, r, nr **where**
$ni \in nni \setminus \mathsf{ran}(r3)$
$i = i1(ni)$
$r = i2(ni)$
$nr \notin ncc$
then
$ncc := ncc \cup \{nr\}$
$nnr := nnr \cup \{nr\}$
$r1 := r1 \cup \{nr \mapsto r\}$
$r2 := r2 \cup \{nr \mapsto i\}$
$r3 := r3 \cup \{nr \mapsto ni\}$
end

P3

any ni, nr **where**
$nr \mapsto ni \in r3$
$ni \notin dom(i3)$
then
$i3 := i3 \cup \{ni \mapsto nr\}$
end

In P2, the recipient is cheating (accessing Initiator's state)

In P3, the initiator is cheating (accessing recipient's state)

Formalizing Mutual Authentication (1)

The protocol must ensure *mutual authentication* between initiators and recipients	FUN-1

- $i3$ and $r3$ are converse of each other (**inv0_9**)

- Initiator with ri and recipient with nr share the same nonces.

inv0_9: $i3^{-1} \subseteq r3$

We have no equality: $i3$ is completed (in P3) after $r3$ (in P2).

Formalizing Mutual Authentication (2)

- If $nr \mapsto ni \in r3$ and $nr \mapsto r \in r1$, that is:

$$ni \mapsto r \in r3^{-1} ; r1$$

- r believes that he will speak to the initiator associated with ni

- So, we must be sure that the pair $ni \mapsto r$ belongs to $i2$

- r is then sure to speak to the initiator that wants to speak to him

inv0_10: $r3^{-1} ; r1 \subseteq i2$

- This invariant is maintained by our three events

Formalizing Mutual Authentication (3)

- If $ni \mapsto nr \in i3$ and $ni \mapsto i \in i1$, that is:

$$nr \mapsto i \in i3^{-1} ; i1$$

- i believes that he will speak to the recipient associated with nr

- So, we must be sure that the pair $nr \mapsto i$ belongs to $r2$

- i is then sure to speak to the recipient that speaks to him

thm0_1: $i3^{-1} ; i1 \subseteq r2$

- The statement is **thm0_1** in fact a *theorem*

- It is easily proved thanks to the following invariant and $i3^{-1} \subseteq r3$:

inv0_11: $r3 ; i1 = r2$

Proofs

- There are 30 proof obligations.

- All discharged automatically by the prover of the Rodin Platform

Refinement: Set

- Introducing the encrypted messages and removing the cheating.

- An agent will not be able to look at the state of other agents

sets: MSG

Refinement: Variables and Invariants (1)

variables: msg
$msg1$
$msg2$
$msg3$
$crypto$

1. $I \rightarrow R \ : \ \{I, N_I\}_{K_R}$ 2. $R \rightarrow I \ : \ \{N_I, N_R\}_{K_I}$ 3. $I \rightarrow R \ : \ \{N_R\}_{K_R}$	FUN-2

- msg is the set of messages circulating so far in the network.
- msg is partitioned into three sets $msg1$, $msg2$, and $msg3$.
- $crypto$ records the agent owning the key encrypting each message.

inv1_1: $msg \subseteq MSG$

inv1_2: partition$(msg, msg1, msg2, msg3)$

inv1_3: $crypto \in msg \rightarrow AGT$

Refinement: Variables and Invariants (2)

variables: $m1\_ini$
$m1\_nni$

1. $I \rightarrow R : \{I, N_I\}_{K_R}$
2. $R \rightarrow I : \{N_I, N_R\}_{K_I}$
3. $I \rightarrow R : \{N_R\}_{K_R}$

FUN-2

inv1_4: $m1\_ini \in msg1 \rightarrow Initiator$

inv1_5: $m1\_nni \in msg1 \rightarrowtail\!\!\!\!\!\rightarrow nni$

inv1_6: $m1\_nni^{-1} ; crypto = i2$

inv1_7: $m1\_nni^{-1} ; m1\_ini = i1$

inv0_3: $i1 \in nni \rightarrow Initiator$

inv0_4: $i2 \in nni \rightarrow Recipient$

Refinement: Variables and Invariants (3)

variables: $m2\_nni$
$m2\_nnr$

1. $I \rightarrow R : \{I, N_I\}_{K_R}$
2. $R \rightarrow I : \{N_I, N_R\}_{K_I}$
3. $I \rightarrow R : \{N_R\}_{K_R}$

FUN-2

inv1_8: $m2\_nnr \in msg2 \rightarrowtail\!\!\!\!\!\rightarrow nnr$

inv1_9: $m2\_nni \in msg2 \rightarrow nni$

inv1_10: $\forall\, m \cdot m \in msg2 \;\Rightarrow\; m2\_nnr(m) \mapsto m2\_nni(m) \in r3$

inv0_8: $r3 \in nnr \rightarrowtail nni$

Refinement: Variables and Invariants (4)

variables: $m3\_nnr$

inv1_11: $m3\_nnr \in msg3 \rightarrow nnr$

Refinement: Events (1)

P1

any $ni, i, r, m1$ **where**

$ni \notin ncc$

$i \in Initiator$

$r \in Recipient$

$m1 \notin msg$

then

$ncc := ncc \cup \{ni\}$

$nni := nni \cup \{ni\}$

$i1 := i1 \cup \{ni \mapsto i\}$

$i2 := i2 \cup \{ni \mapsto r\}$

$msg := msg \cup \{m1\}$

$msg1 := msg1 \cup \{m1\}$

$crypto(m1) := crypto \cup \{m1 \mapsto r\}$

$m1\_nni := m1\_nni \cup \{m1 \mapsto ni\}$

$m1\_ini := m1\_ini \cup \{m1 \mapsto i\}$

end

(abstract-)P1

any ni, i, r **where**

$ni \notin ncc$

$i \in Initiator$

$r \in Recipient$

then

$ncc := ncc \cup \{ni\}$

$nni := nni \cup \{ni\}$

$i1 := i1 \cup \{ni \mapsto i\}$

$i2 := i2 \cup \{ni \mapsto r\}$

end

Refinement: Events (2)

P2

any $m1, r, nr, m2$ **where**
- $m1 \in msg1$
- $m1\_nni(m1) \notin \mathsf{ran}(r3)$
- $nr \notin ncc$
- $r = crypto(m1)$
- $m2 \notin msg$

with
- $ni = m1\_nni(m1)$
- $i = m\_ini(m1)$

then
- $ncc := ncc \cup \{nr\}$
- $nnr := nnr \cup \{nr\}$
- $r1 := r1 \cup \{nr \mapsto r\}$
- $r2 := r2 \cup \{nr \mapsto m1\_ini(m1)\}$
- $r3 := r3 \cup \{nr \mapsto m1\_nni(m1)\}$
- $msg := msg \cup \{m2\}$
- $msg2 := msg2 \cup \{m2\}$
- $m2\_nni(m2) := m1\_nni(m1)$
- $m2\_nnr(m2) := nr$
- $crypto(m2) := m1\_ini(m1)$

end

(abstract-)P2

any ni, i, r, nr **where**
- $ni \in nni \setminus \mathsf{ran}(r3)$
- $i = i1(ni)$
- $r = i2(ni)$
- $nr \notin ncc$

then
- $ncc := ncc \cup \{nr\}$
- $nnr := nnr \cup \{nr\}$
- $r1 := r1 \cup \{nr \mapsto r\}$
- $r2 := r2 \cup \{nr \mapsto i\}$
- $r3 := r3 \cup \{nr \mapsto ni\}$

end

Refinement: Events (3)

P3
any $m2$ **where**
$m2 \in msg2$
$m2\_nni(m2) \notin \mathrm{dom}(i3)$
$i1(m2\_nni(m2)) = crypto(m2)$
with
$ni = m2\_nni(m2)$
$nr = m2\_nnr(m2)$
then
$i3 := i3 \cup \{m2\_nni(m2) \mapsto m2\_nnr(m2)\}$
end

(abstract-)P3
any ni, nr **where**
$nr \mapsto ni \in r3$
$ni \notin dom(i3)$
then
$i3 := i3 \cup \{ni \mapsto nr\}$
end

Proofs

- There are 49 proof obligations.

- All discharged automatically by the prover of the Rodin Platform

Protocol with mistake

- The initiator i records the recipient r it wants to speak to (in $i2$).

- But the initiator i send $m1$ with the public key of any agent s.

- s might be identical to r but not necessarily.

- This mistake of the initiator will break invariant **inv1_6**

$$m1\_nni^{-1} ; crypto = i2$$

- It says that the key used in $m1$ is that of the recipient (in $i2$).

- To detect the mistake, the recipient sends its name

The Protocol with Mistake

1. $I \rightarrow R \ : \ \{I, N_I\}_{K_S}$ 2. $S \rightarrow I \ : \ \{N_I, N_S, S\}_{K_I}$ if $S = R$ then 3. $I \rightarrow R \ : \ \{N_R\}_{K_R}$	FUN-3

- If S is not the same as R then the protocol is stopped

- Step 3 is never executed.

Set, Constants and Axiom

sets: AGT
NNC

constants: $Initiator$
$Recipient$

axm_1: partition$(AGT, Initiator, Recipient)$

Variables and Invariants (1)

variables: nnc, nni, nnr

inv2_1: $nnc \subseteq NNC$

inv2_2: partition(nnc, nni, nnr)

Same as in previous case

inv0_1: $nnc \subseteq NNC$

inv0_2: partition(nnc, nni, nnr)

Variables and Invariants (2)

variables: $i1$
$i2$
$i3$

inv2_3: $i1 \in nni \rightarrow Initiator$

inv2_4: $i2 \in nni \rightarrow Recipient$

inv2_5: $i3 \in nni \rightarrowtail nnr$

- Same as in previous case

inv0_3: $i1 \in nni \rightarrow Initiator$

inv0_4: $i2 \in nni \rightarrow Recipient$

inv0_5: $i3 \in nni \rightarrowtail nnr$

Variables and Invariants (3)

- Variable $r3$ is now a *partial* injection only.
- Variable $r4$ is new. It records what the recipient "believes"
- Of course, it can be erroneous .
- Variable $r3$ is the corrected connection ($r3$ updated by new event P4).

variables: $r1$, $r2$, $r3$, $r4$

inv2_6: $r1 \in nnr \rightarrow Recipient$

inv2_7: $r2 \in nnr \rightarrow Initiator$

inv2_8: $r3 \in nnr \rightarrowtail\!\!\!\!\!\!\rightarrow nni$

inv2_9: $r4 \in nnr \rightarrowtail nni$

- Invariants of previous case:

inv0_6: $r1 \in nnr \rightarrow Recipient$

inv0_7: $r2 \in nnr \rightarrow Initiator$

inv0_8: $r3 \in nnr \rightarrowtail nni$

Authentication Invariants

inv2_10: $r3 \subseteq i3^{-1}$

inv2_11: $r3^{-1} ; r1 \subseteq i2$

inv2_12: $i3^{-1} ; i1 \subseteq r2$

inv2_13: $r4 ; i1 = r2$

inv2_14: $i3^{-1} \subseteq r4$

inv0_9: $i3^{-1} \subseteq r3$

inv0_10: $r3^{-1} ; r1 \subseteq i2$

thm0_1: $i3^{-1} ; i1 \subseteq r2$

inv0_11: $r3 ; i1 = r2$

- The authentication conditions are identical

Events (1)

- Q1 event of this case (with mistake)

- P1 event of previous case (without mistake)

Q1

any ni, i, r **where**

$ni \notin ncc$

$i \in Initiator$

$r \in Recipient$

then

$ncc := ncc \cup \{ni\}$

$nni := nni \cup \{ni\}$

$i1(ni) := i$

$i2(ni) := r$

end

P1

any ni, i, r **where**

$ni \notin ncc$

$i \in Initiator$

$r \in Recipient$

then

$ncc := ncc \cup \{ni\}$

$nni := nni \cup \{ni\}$

$i1(ni) := i$

$i2(ni) := r$

end

These are identical

Events (2)

Q2

any ni, i, r, nr **where**

$ni \in nni \setminus \mathsf{ran}(r4)$

$i = i1(ni)$

$r \in Recipient$

$nr \notin ncc$

then

$ncc := ncc \cup \{nr\}$

$nnr := nnr \cup \{nr\}$

$r1 := r1 \cup \{nr \mapsto r\}$

$r2 := r2 \cup \{nr \mapsto i\}$

$r4 := r4 \cup \{nr \mapsto ni\}$

end

P2

any ni, i, r, nr **where**

$ni \in nni \setminus \mathsf{ran}(r3)$

$i = i1(ni)$

$r = i2(ni)$

$nr \notin ncc$

then

$ncc := ncc \cup \{nr\}$

$nnr := nnr \cup \{nr\}$

$r1 := r1 \cup \{nr \mapsto r\}$

$r2 := r2 \cup \{nr \mapsto i\}$

$r3 := r3 \cup \{nr \mapsto ni\}$

end

We see the "mistake" in Q2. Any recipient r can accept this event.

Events (3)

Q3

any ni, nr **where**

$nr \mapsto ni \in r4$

$ni \notin \mathrm{dom}(i3)$

$i2(ni) = r1(nr)$

then

$i3 := i3 \cup \{ni \mapsto nr\}$

end

P3

any ni, nr **where**

$nr \mapsto ni \in r3$

$ni \notin dom(i3)$

then

$i3 := i3 \cup \{ni \mapsto nr\}$

end

The additional guard ensures that the recipient in the initiator ($i2(ni)$) is correct ($r1(nr)$)

Events (4)

Q4

any ni, nr **where**

$ni \mapsto nr \in i3$

$nr \notin \mathsf{dom}(r3)$

$i2(ni) = r1(nr)$

then

$r3 := r3 \cup \{nr \mapsto ni\}$

end

Updating the recipient information

Proofs

- There are 49 proof obligations.

- All discharged automatically by the prover of the Rodin Platform

Refinement: The state (1)

As previously

variables: msg, $msg1$, $msg2$, $msg3$, $crypto$

inv3_1: $msg \subseteq MSG$

inv3_2: partition$(msg, msg1, msg2, msg3)$

inv3_3: $crypto \in msg \rightarrow AGT$

- Invariants of previous case:

inv1_1: $msg \subseteq MSG$

inv1_2: partition$(msg, msg1, msg2, msg3)$

inv1_3: $crypto \in msg \rightarrow AGT$

Refinement: The state (2)

variables: $m1\_ini$
$m1\_nni$

inv3_4: $m1\_ini \in msg1 \rightarrow Initiator$

inv3_5: $m1\_nni \in msg1 \rightarrowtail\!\!\!\!\!\rightarrow nni$

inv3_7: $m1\_nni^{-1} ; m1\_ini = i1$

Invariants of previous case:

inv1_4: $m1\_ini \in msg1 \rightarrow Initiator$

inv1_5: $m1\_nni \in msg1 \rightarrowtail\!\!\!\!\!\rightarrow nni$

inv1_6: $m1\_nni^{-1} ; crypto = i2$

inv1_7: $m1\_nni^{-1} ; m1\_ini = i1$

- Invariant **inv1_6** has disappeared

- The message is not sent necessarily to the recorded recipient

Refinement: The state (3)

- A new "field", $m2\_rcv$, is added in the message

- $r4$ replaces $r3$

variables: $m2\_nnr$, $m2\_nni$, $m2\_rcv$

inv3_8: $m2\_nni \in msg2 \rightarrow \mathrm{ran}(r4)$

inv3_9: $m2\_nnr \in msg2 \rightarrowtail\!\!\!\!\!\rightarrow nnr$

inv3_10: $m2\_nnr = m2\_nni\ ;\ r4^{-1}$

inv3_11: $m2\_rcv \in msg2 \rightarrow Recipient$

- Invariants of previous case:

inv1_8: $m2\_nni \in msg2 \rightarrow \mathrm{ran}(r3)$

inv1_9: $m2\_nnr \in msg2 \rightarrowtail\!\!\!\!\!\rightarrow nnr$

inv1_10: $m2\_nnr = m2\_nni\ ;\ r3^{-1}$

Refinement: The state (3)

variables: $m3\_nnr$

inv3_12: $m3\_nnr \in msg3 \rightarrow nnr$

Refinement: Events (1)

Q1

any $ni, i, r, m1, s$ **where**

$ni \notin ncc$

$p \in Initiator$

$q \in Recipient$

$m1 \notin msg$

$s \in Recipient$

then

$ncc := ncc \cup \{ni\}$

$nni := nni \cup \{ni\}$

$i1 := i1 \cup \{ni \mapsto i\}$

$i2 := i2 \cup \{ni \mapsto r\}$

$msg := msg \cup \{m1\}$

$msg1 := msg1 \cup \{m1\}$

$crypto := crypto \cup \{m1 \mapsto s\}$

$m1\_nni := m1\_nni \cup \{m1 \mapsto ni\}$

$m1\_ini := m1\_ini \cup \{m1 \mapsto i\}$

end

P1

any $ni, i, r, m1$ **where**

$ni \notin ncc$

$i \in Initiator$

$r \in Recipient$

$m1 \notin msg$

then

$ncc := ncc \cup \{ni\}$

$nni := nni \cup \{ni\}$

$i1 := i1 \cup \{ni \mapsto i\}$

$i2 := i2 \cup \{ni \mapsto r\}$

$msg := msg \cup \{m1\}$

$msg1 := msg1 \cup \{m1\}$

$crypto(m1) := crypto \cup \{m1 \mapsto r\}$

$m1\_nni := m1\_nni \cup \{m1 \mapsto ni\}$

$m1\_ini := m1\_ini \cup \{m1 \mapsto i\}$

end

- In Q1, we can see the potential mistake

- The message can be sent to any recipient, not necessarily r

Refinement: Events (2)

Q2

any $m1, r, nr, m2$ **where**
$m1 \in msg1$
$m1\_nni(m1) \notin \mathrm{ran}(r4)$
$nr \notin ncc$
$r = crypto(m1)$
$m2 \notin msg$
with
$ni = m1\_nni(m1)$
$i = m\_ini(m1)$
then
$ncc := ncc \cup \{nr\}$
$nnr := nnr \cup \{nr\}$
$r1 := r1 \cup \{nr \mapsto q\}$
$r2 := r2 \cup \{nr \mapsto m1\_ini(m1)\}$
$r4 := r4 \cup \{nr \mapsto m1\_nni(m1)\}$
$msg := msg \cup \{m2\}$
$msg2 := msg2 \cup \{m2\}$
$m2\_nni(m2) := m1\_nni(m1)$
$m2\_nnr(m2) := nr$
$crypto(m2) := m1\_ini(m1)$
$m2\_rcv(m2) := r$
end

P2

any $m1, r, nr, m2$ **where**
$m1 \in msg1$
$m1\_nni(m1) \notin \mathrm{ran}(r3)$
$nr \notin ncc$
$r = crypto(m1)$
$m2 \notin msg$
with
$ni = m1\_nni(m1)$
$i = m\_ini(m1)$
then
$ncc := ncc \cup \{nr\}$
$nnr := nnr \cup \{nr\}$
$r1 := r1 \cup \{nr \mapsto r\}$
$r2 := r2 \cup \{nr \mapsto m1\_ini(m1)\}$
$r3 := r3 \cup \{nr \mapsto m1\_nni(m1)\}$
$msg := msg \cup \{m2\}$
$msg2 := msg2 \cup \{m2\}$
$m2\_nni(m2) := m1\_nni(m1)$
$m2\_nnr(m2) := nr$
$crypto(m2) := m1\_ini(m1)$
end

- $r4$ replaces $r3$

- A new field in the message is updated

Refinement: Events (3)

Q3

any $m2, m3$ **where**

$m2 \in msg2$

$m2\_nni(m2) \notin \mathrm{dom}(i3)$

$i1(m2\_nni(m2)) = crypto(m2)$

$m2\_rcv(m2) = i2(m2\_nni(m2))$

$m3 \notin msg$

with

$ni = m2\_nni(m2)$

$nr = m2\_nnr(m2)$

then

$i3 := i3 \cup \{m2\_nni(m2) \mapsto m2\_nnr(m2)\}$

$msg := msg \cup \{m3\}$

$msg3 := msg3 \cup \{m3\}$

$m3\_nnr := m3\_nnr \cup \{m3 \mapsto m2\_nnr(m2)\}$

$crypto := crypto \cup \{m3 \mapsto m2\_rcv(m2)\}$

end

P3

any $m2$ **where**

$m2 \in msg2$

$m2\_nni(m2) \notin \mathrm{dom}(i3)$

$i1(m2\_nni(m2)) = crypto(m2)$

with

$ni = m2\_nni(m2)$

$nr = m2\_nnr(m2)$

then

$i3 := i3 \cup \{m2\_nni(m2) \mapsto m2\_nnr(m2)\}$

end

The fundamental guard for checking the name of the recipient

Refinement: Events (4)

Q4

any $m3$ **where**

$m3 \in msg3$

$m3\_nnr(m3) \notin \mathrm{dom}(r3)$

$r1(m3\_nnr(m3)) = crypto(m3)$

with

$ni = r4(m3\_nni(m3))$

$nr = m3\_nnr(m3)$

then

$r3 := r3 \cup \{m3\_nnr(m3) \mapsto r4(m3\_nnr(m3))\}$

end

- Updating $r3$

Proofs

- There are 93 proof obligations.

- Discharged automatically except 11 of them done interactively (easy)