Suggested LySA-encodings of the Otway-Rees protocol

Christoffer Rosenkilde Nielsen
Revised by Han Gao *
Informatics and Mathematical Modelling, Technical University of Denmark
Richard Petersens Plads, bldg. 321, DK-2800 Kongens Lyngby, Denmark
February 14, 2008

1 Encoding

The Otway-Rees protocol is defined as follows:

1. $A \to B \quad : \quad M, A, B, \{N_A, M, A, B\}_{K_A}$
2. $B \to S \quad : \quad M, A, B, \{N_A, M, A, B\}_{K_A}, \{N_B, M, A, B\}_{K_B}$
3. $S \to B \quad : \quad M, \{N_A, K_{AB}\}_{K_A}, \{N_B, K_{AB}\}_{K_B}$
4. $B \to A \quad : \quad M, \{N_A, K_{AB}\}_{K_A}$

In order to encode this in LySA, we must reorder some of the tuples, as the values that we want to bind to variables must be placed as the last values of the tuples. Furthermore we shall add a final step on the protocol where B attempts to communicate to A through the now shared key, as this will ease the validation of the output from LySATool. The resulting protocol looks as follows:

1. $A \to B \quad : \quad M, A, B, \{M, A, B, N_A\}_{K_A}$
2. $B \to S \quad : \quad M, A, B, \{M, A, B, N_A\}_{K_A}, \{M, A, B, N_B\}_{K_B}$
3. $S \to B \quad : \quad M, \{N_A, K_{AB}\}_{K_A}, \{N_B, K_{AB}\}_{K_B}$
4. $B \to A \quad : \quad M, \{N_A, K_{AB}\}_{K_A}$
5. $B \to A \quad : \quad \{MSG\}_{K_{AB}}$

The value M is used as a running serial number, and provides no actual security, thus we shall assume that it is known on advance by all participants and also by the attacker. The nonces N_A and N_B on the other hand are not known on advance by the server, and thus these must be placed at the end of each tuple.

Finally, when analysing a protocol, one must always consider which scenarios that one wishes to analyse and what possible roles an attacker may have in the protocol. We shall assume the most general scenario, namely that an arbitrary large number of initiators $(A_0, A_1, A_2, \ldots)$ attempts to establish a connection to an arbitrary large number of responders $(B_0, B_1, B_2, \ldots)$ and we shall assume that the server shares a key with each of these principals. In particular we shall assume that the attacker can also act as a legitimate principal (A_0 and B_0) and allow him to legally establish a connection with the other principals.

*Emails: crn@imm.dtu.dk, hg@imm.dtu.dk

1.1 Version A

In the first version we shall assume that $A_i \neq B_i$ and that $K_i^A \neq K_i^B$; i.e. that the principals use different addresses for initiating and responding and that they use different keys for these actions.

The resulting encoding is listed in Table 1. We have chosen to group the legitimate principals into three groups (1, 2 and 3) so that we analyse for all kinds of man-in-the-middle attacks, notice that we still analyse for an infinite number of principals, this is merely a partitioning.

We also encode each sent message analogous to the IPv4 and IPv6 standard, where we place initiator and responder as the first two elements of each tuple. These are sent in plain text and does therefore not provide any additional security to the protocol. Finally we annotate all encryptions and decryptions with unique crypto-points in order to capture any unwanted behaviour.

$$
\begin{array}{l}
\mathsf{let}\ X \subseteq \mathbf{N}\ \mathsf{s.t.}\ \lfloor \mathbf{N} \rfloor = \{1,2,3\}\ \mathsf{in} \\
(\nu_{i \in X}\ KA_i)(\nu_{j \in X}\ KB_j) \\
\\
|_{i \in X}\ |_{j \in X \cup \{0\}}\ !(\nu\ NA_{ij}) \\
\quad \langle A_i, B_j, M_{ij}, A_i, B_j, \{A_i, B_j, M_{ij}, NA_{ij}\}_{KA_i}[\mathsf{at}\ a1_{ij}\ \mathsf{dest}\ \{s1_{ij}\}]\rangle. \\
\quad (B_j, A_i, M_{ij};\ x1_{ij}). \\
\quad \mathsf{decrypt}\ x1_{ij}\ \mathsf{as}\ \{NA_{ij};\ xk_{ij}\}_{KA_i}[\mathsf{at}\ a2_{ij}\ \mathsf{orig}\ \{s3_{ij}\}]\ \mathsf{in} \\
\quad (B_j, A_i;\ x2_{ij}). \\
\quad \mathsf{decrypt}\ x2_{ij}\ \mathsf{as}\ \{;\ xmsg_{ij}\}_{xk_{ij}}[\mathsf{at}\ a3_{ij}\ \mathsf{orig}\ \{b3_{ij}\}]\ \mathsf{in}\ 0 \\
| \\
|_{i \in X \cup \{0\}}\ |_{j \in X}\ !(\nu\ NB_{ij}) \\
\quad (A_i, B_j, M_{ij}, A_i, B_j;\ y1_{ij}). \\
\quad \langle B_j, S, M_{ij}, A_i, B_j, y1_{ij}, \{A_i, B_j, M_{ij}, NB_{ij}\}_{KB_j}[\mathsf{at}\ b1_{ij}\ \mathsf{dest}\ \{s2_{ij}\}]\rangle. \\
\quad (S, B_j, M_{ij};\ y2_{ij}, y3_{ij}). \\
\quad \mathsf{decrypt}\ y3_{ij}\ \mathsf{as}\ \{NB_{ij};\ yk_{ij}\}_{KB_j}[\mathsf{at}\ b2_{ij}\ \mathsf{orig}\ \{s4_{ij}\}]\ \mathsf{in} \\
\quad \langle B_j, A_i, M_{ij}, y2_{ij}\rangle. \\
\quad (\nu\ MSG_{ij})\ \langle B_j, A_i, \{MSG_{ij}\}_{yk_{ij}}[\mathsf{at}\ b3_{ij}\ \mathsf{dest}\ \{a3_{ij}\}]\rangle.0 \\
| \\
|_{i \in X \cup \{0\}}\ |_{j \in X \cup \{0\}}\ ! \\
\quad (B_j, S, M_{ij}, A_i, B_j;\ z1_{ij}, z2_{ij}). \\
\quad \mathsf{decrypt}\ z1_{ij}\ \mathsf{as}\ \{A_i, B_j, M_{ij};\ zna_{ij}\}_{KA_i}[\mathsf{at}\ s1_{ij}\ \mathsf{orig}\ \{a1_{ij}\}]\ \mathsf{in} \\
\quad \mathsf{decrypt}\ z2_{ij}\ \mathsf{as}\ \{A_i, B_j, M_{ij};\ znb_{ij}\}_{KB_j}[\mathsf{at}\ s2_{ij}\ \mathsf{orig}\ \{b1_{ij}\}]\ \mathsf{in} \\
\quad (\nu\ K_{ij})\ \langle S, B_j, M_{ij}, \{zna_{ij}, K_{ij}\}_{KA_i}[\mathsf{at}\ s3_{ij}\ \mathsf{dest}\ \{a2_{ij}\}], \{znb_{ij}, K_{ij}\}_{KB_j}[\mathsf{at}\ s4_{ij}\ \mathsf{dest}\ \{b2_{ij}\}]\rangle.0
\end{array}
$$

Table 1: Otway-Rees with $A_i \neq B_i$ and $K_i^A \neq K_i^B$

1.1.1 Result

Depending on which options one runs the analysis with, the analysis will return a different number of possible violations to the annotations. This is due to the choice of allowing the attacker to behave as a legitimate principal, and violations such as $(CPDY, s_{01})$ merely means that the attacker behaved as principal 0 and initiated a protocol run with principal 1,

and at some point the server had to decrypt a message encrypted by the attacker ($CPDY$). This is not a breech of the protocol, and such violations can be sorted away by telling the analysis that the attacker is principal 0.

If this is done then the analysis on the protocol above yields the following result,

Version	$A_i \neq B_i \ \wedge K_i^A \neq K_i^B$
Result(ψ)	$\emptyset$

which means that in the case that $A_i \neq B_i$ and that $K_i^A \neq K_i^B$ no violation to the authentication property is possible.

1.2 Version B

Exercise: Please encode the Otway-Rees protocol in a scenario that the principals use the same address for initiating and responding but that they use different keys for it.

Hint: principals are referred to as I_i regardless of the role they play in the protocol.

1.3 Version C

In the third version we shall assume that $A_i \neq B_i$ and that $K_i^A = K_i^B$; i.e. that the principals use different addresses for initiating and responding but that they use the same key for it.

The resulting encoding is listed in Table 2. The change from version A is that keys are denoted KI_i regardless of whether they are used for initiating or responding.

1.3.1 Result

The analysis on the protocol above yields the following result.

Version	$A_i \neq B_i \ \wedge K_i^A = K_i^B$
Result(ψ)	$\forall i : \{(a1_{ii}, s2_{ii}), (b1_{ii}, s1_{ii}), (s4_{ii}, s2_{ii}), (s3_{ii}, b2_{ii})\}$

The analysis result tells us that it does not matter whether we distinguish between the address used as initiator and recipient in the protocol or not. However, the same does not apply to the use of keys. If we use the same key for initiating and responding then we get violations that refers to a type-flaw attack in that very special case that a principal attempts to establish a connection with itself. The attack corresponding to the violation $(a1_{ii}, s2_{ii})$ would be executed as follows:

1. $A \rightarrow I(B) : M, I_i, I_i, \{M, I_i, I_i, N_A\}_{KI_i}$
2. $I(B) \rightarrow S : M, I_i, I_i, \{M, I_i, I_i, N_A\}_{KI_i}, \{M, I_i, I_i, N_A\}_{KI_i}$
3. $S \rightarrow I(B) : M, \{N_A, K_{AB}\}_{KI_i}, \{N_A, K_{AB}\}_{KI_i}$
4. $I(B) \rightarrow A : M, \{N_A, K_{AB}\}_{KI_i}$

Here the attacker I disguises itself as B and lets A believe it has a shared connection with B. The attacks corresponding to the remaining violations are variations of the attack above that all use the same type-flaw.

$$
\begin{array}{l}
\mathsf{let}\ X \subseteq \mathbf{N}\ \mathsf{s.t.}\ \lfloor \mathbf{N} \rfloor = \{1,2,3\}\ \mathsf{in} \\
(\nu_{i \in X}\ KI_i) \\
\quad \mid_{i \in X} \mid_{j \in X \cup \{0\}} !(\nu\ NA_{ij}) \\
\qquad \langle A_i, B_j, M_{ij}, A_i, B_j, \{A_i, B_j, M_{ij}, NA_{ij}\}_{KI_i} [\mathsf{at}\ a1_{ij}\ \mathsf{dest}\ \{s1_{ij}\}\,] \rangle. \\
\qquad (B_j, A_i, M_{ij};\ x1_{ij}). \\
\qquad \mathsf{decrypt}\ x1_{ij}\ \mathsf{as}\ \{NA_{ij};\ xk_{ij}\}_{KI_i} [\mathsf{at}\ a2_{ij}\ \mathsf{orig}\ \{s3_{ij}\}\,]\ \mathsf{in} \\
\qquad (B_j, A_i;\ x2_{ij}). \\
\qquad \mathsf{decrypt}\ x2_{ij}\ \mathsf{as}\ \{;\ xmsg_{ij}\}_{xk_{ij}} [\mathsf{at}\ a3_{ij}\ \mathsf{orig}\ \{b3_{ij}\}\,]\ \mathsf{in}\ 0 \\
\quad \mid \\
\quad \mid_{i \in X \cup \{0\}} \mid_{j \in X} !(\nu\ NB_{ij}) \\
\qquad (A_i, B_j, M_{ij}, A_i, B_j;\ y1_{ij}). \\
\qquad \langle B_j, S, M_{ij}, A_i, B_j, y1_{ij}, \{A_i, B_j, M_{ij}, NB_{ij}\}_{KI_j} [\mathsf{at}\ b1_{ij}\ \mathsf{dest}\ \{s2_{ij}\}\,] \rangle. \\
\qquad (S, B_j, M_{ij};\ y2_{ij}, y3_{ij}). \\
\qquad \mathsf{decrypt}\ y3_{ij}\ \mathsf{as}\ \{NB_{ij};\ yk_{ij}\}_{KI_j} [\mathsf{at}\ b2_{ij}\ \mathsf{orig}\ \{s4_{ij}\}\,]\ \mathsf{in} \\
\qquad \langle B_j, A_i, M_{ij}, y2_{ij} \rangle. \\
\qquad (\nu\ MSG_{ij})\ \langle B_j, A_i, \{MSG_{ij}\}_{yk_{ij}} [\mathsf{at}\ b3_{ij}\ \mathsf{dest}\ \{a3_{ij}\}\,] \rangle.0 \\
\quad \mid \\
\quad \mid_{i \in X \cup \{0\}} \mid_{j \in X \cup \{0\}} ! \\
\qquad (B_j, S, M_{ij}, A_i, B_j;\ z1_{ij}, z2_{ij}). \\
\qquad \mathsf{decrypt}\ z1_{ij}\ \mathsf{as}\ \{A_i, B_j, M_{ij};\ zna_{ij}\}_{KI_i} [\mathsf{at}\ s1_{ij}\ \mathsf{orig}\ \{a1_{ij}\}\,]\ \mathsf{in} \\
\qquad \mathsf{decrypt}\ z2_{ij}\ \mathsf{as}\ \{A_i, B_j, M_{ij};\ znb_{ij}\}_{KI_j} [\mathsf{at}\ s2_{ij}\ \mathsf{orig}\ \{b1_{ij}\}\,]\ \mathsf{in} \\
\qquad (\nu\ K_{ij})\ \langle S, B_j, M_{ij}, \{zna_{ij}, K_{ij}\}_{KI_i} [\mathsf{at}\ s3_{ij}\ \mathsf{dest}\ \{a2_{ij}\}\,], \{znb_{ij}, K_{ij}\}_{KI_j} [\mathsf{at}\ s4_{ij}\ \mathsf{dest}\ \{b2_{ij}\}\,] \rangle.0
\end{array}
$$

Table 2: Otway-Rees with $A_i \neq B_i$ and $K_i^A = K_i^B$

1.4 Version D

Exercise: Please encode the Otway-Rees protocol in a scenario that the principals use the same address and key for initiating and responding. Feed your encoding into the LySa tool and try to explain the result.

Hint: principals are referred to as I_i regardless of the role they play in the protocol, and keys as KI_i. This is a combination of the changes from version A to version B and from version A to version C.