# Appendix of Modeling and Verifying NDN Access Control Using CSP

We have formalized NDN access control using CSP [1]. Due to the space limit, we only present four models in [1], and the improved models are illustrated in this appendix. In addition, auxiliary definitions are also listed here.

[1] Fei, Y., Zhu, H.: Modeling and Verifying NDN Access Control Using CSP (submitted to ICFEM2018)

## Appendix

In order to describe the improved models, we update the definition of $MSG_{dat2}$ and $MSG_{pro}$ in [1] as below.

$$\begin{aligned}
MSG_{dat2} = \; & \{msg_{dat}.a.b.n.H(k).E(k, E(k_1, c_1)), msg_{dat}.a.b.n.H(k).E(k, E(k_1, c_1)).E(k_2, c_2) \mid \\
& a, b \in \mathit{Entity}, n \in \mathit{Name}, k, k_1, k_2 \in \mathit{Key}, c_1, c_2 \in \mathit{Content}\} \\
MSG_{pro} = \; & \{msg_{pro}.E(k, E(k_1, c_1)).k_2.k_3, msg_{pro}.E(k, E(k_1, c_1)).E(k_2, c_2).k_3.k_4.k_5 \mid \\
& k, k_1, k_2, k_3, k_4, k_5 \in \mathit{Key}, c_1, c_2 \in \mathit{Content}\}
\end{aligned}$$

Table 1 represents the new involved constants and variables.

**Table 1.** New involved constants and variables

| | |
|---|---|
| **Constants** | K$_I$(public key of intruder), K$_M$(public key of ACM), K$_A$(public key of CA), K$_I^{-1}$(private key of intruder), K$_M^{-1}$(private key of ACM), K$_A^{-1}$(private key of CA) |
| **Variables** | k$_m$,k$_{m1}$,k$_{m2}$,k$_{m3}$(public key of ACM), k$_m^{-1}$,k$_{m1}^{-1}$,k$_{m2}^{-1}$(private key of ACM), k$_a$,k$_{a1}$,k$_{a2}$(public key of CA), k$_a^{-1}$,k$_{a1}^{-1}$(private key of CA), |

Here we give the improved models as below.

$$\begin{aligned}
\mathit{SystemR\_Sig} =_{df} \; & \mathit{READER\_Sig}(R, M, K, K_M, NN, ND, NDK)[|\mathit{PROCESS\_PATH}|]\mathit{PROCESS} \\
& [|\mathit{COM\_PATH}|]\mathit{ACM\_R\_Sig}(R, M, K_M, DK, DATA, NK\_e, NK\_d, HL) \\
\mathit{SystemW\_Sig} =_{df} \; & \mathit{WRITER\_Sig}(W, M, K, K_M, NN, ND, DK, DATA)[|\mathit{PROCESS\_PATH}|]\mathit{PROCESS} \\
& [|\mathit{COM\_PATH}|]\mathit{ACM\_W\_Sig}(W, M, HL, NK\_e, NK\_d, NDK) \\
\mathit{SystemR\_Sig}_I =_{df} \; & \mathit{SystemR\_Sig}[|\mathit{INTRUDER\_PATH}|]\mathit{INTUDER\_Sig} \\
\mathit{SystemW\_Sig}_I =_{df} \; & \mathit{SystemW\_Sig}[|\mathit{INTRUDER\_PATH}|]\mathit{INTUDER\_Sig} \\
\mathit{SystemR\_Sig\_C} =_{df} \; & \mathit{READER\_Sig}(R, M, K, K_I, NN, ND, NDK)[|\mathit{PROCESS\_PATH}|]\mathit{PROCESS} \\
& [|\mathit{COM\_PATH}|]\mathit{ACM\_R\_Sig}(R, M, K_M, DK, DATA, NK\_e, NK\_d, HL) \\
\mathit{SystemW\_Sig\_C} =_{df} \; & \mathit{WRITER\_Sig}(W, M, K, K_I, NN, ND, DK, DATA)[|\mathit{PROCESS\_PATH}|]\mathit{PROCESS} \\
& [|\mathit{COM\_PATH}|]\mathit{ACM\_W\_Sig}(W, M, HL, NK\_e, NK\_d, NDK) \\
\mathit{SystemR\_Sig\_C}_I =_{df} \; & \mathit{SystemR\_Sig\_C}[|\mathit{INTRUDER\_PATH}|]\mathit{INTUDER\_Sig} \\
\mathit{SystemW\_Sig\_C}_I =_{df} \; & \mathit{SystemW\_Sig\_C}[|\mathit{INTRUDER\_PATH}|]\mathit{INTUDER\_Sig}
\end{aligned}$$

$$\begin{aligned}
SystemR\_Dig =_{df}\ & READER\_Dig(R, M, K, K_M, NN, ND, NDK)[|PROCESS\_PATH|]PROCESS \\
& [|COM\_PATH|]ACM\_R\_Dig(R, M, K_M, K_A, DK, DATA, NK\_e, NK\_d, HL) \\
SystemW\_Dig =_{df}\ & WRITER\_Dig(W, M, K, K_M, K_A, NN, ND, DK, DATA)[|PROCESS\_PATH|]PROCESS \\
& [|COM\_PATH|]ACM\_W\_Dig(W, M, HL, NK\_e, NK\_d, NDK) \\
SystemR\_Dig_I =_{df}\ & SystemR\_Dig[|INTRUDER\_PATH|]INTUDER\_Dig \\
SystemW\_Dig_I =_{df}\ & SystemW\_Dig[|INTRUDER\_PATH|]INTUDER\_Dig \\
SystemR\_Dig\_C =_{df}\ & READER\_Dig(R, M, K, K_I, NN, ND, NDK)[|PROCESS\_PATH|]PROCESS \\
& [|COM\_PATH|]ACM\_R\_Dig(R, M, K_M, K_A, DK, DATA, NK\_e, NK\_d, HL) \\
SystemW\_Dig\_C =_{df}\ & WRITER\_Dig(W, M, K, K_I, K_A, NN, ND, DK, DATA)[|PROCESS\_PATH|]PROCESS \\
& [|COM\_PATH|]ACM\_W\_Dig(W, M, HL, NK\_e, NK\_d, NDK) \\
SystemR\_Dig\_C_I =_{df}\ & SystemR\_Dig\_C[|INTRUDER\_PATH|]INTUDER\_Dig \\
SystemW\_Dig\_C_I =_{df}\ & SystemW\_Dig\_C[|INTRUDER\_PATH|]INTUDER\_Dig
\end{aligned}$$

Here gives the definition of subprocesses $READER\_Sig$ and $READER\_Dig$. They simulate the read operations.

$$\begin{aligned}
& READER_1(r, m, k, k_m, nn, nd, ndk) =_{df} \\
&\quad Initialization\{n = false; d = false\} \rightarrow \\
&\quad ComRM!msg_{int}.r.m.nd \rightarrow ComRM?msg_{dat}.m.r.nd.E(dk, data) \rightarrow \\
&\quad ComRM!msg_{int}.r.m.ndk \rightarrow ComRM?msg_{dat}.m.r.ndk.E(nk\_e, dk1) \rightarrow \\
&\quad ComRM!msg_{int}.r.m.nn \rightarrow ComRM?msg_{dat}.m.r.nn.hl \rightarrow \\
&\quad ComRM!msg_{int}.r.m.nn.H(k) \rightarrow ComRM?msg_{dat}.m.r.nn.H(k1).E(k1, E((k_m^{-1}), nk\_d)) \rightarrow \\
&\quad CheckNK!msg_{pro}.E(k1, E((k_m^{-1}), nk\_d)).k^{-1}.k_m \rightarrow CheckNK?msg_{ack}.ack \rightarrow \\
&\quad \begin{pmatrix} (NKFakingSuccess\{n = true\} \rightarrow SKIP) \\ \lhd (ack == YES) \rhd (NKFakingError\{n = false\} \rightarrow SKIP) \end{pmatrix}; \\
&\quad GetData!msg_{pro}.E(dk, data).E(nk\_e, dk1).nk\_d \rightarrow GetData?msg_{ack}.ack1 \rightarrow \\
&\quad \begin{pmatrix} (DataAcquisitionSuccess\{d = true\} \rightarrow SKIP) \\ \lhd (ack1 == YES) \rhd (DataAcquisitionError\{d = false\} \rightarrow SKIP) \end{pmatrix}; \\
&\quad READER_1(r, m, k, k_m, nn, nd, ndk)
\end{aligned}$$

$$\begin{aligned}
& READER_2(r, m, k, k_m, k_a, nn, nd, ndk) =_{df} \\
&\quad READER_1(r, m, k, k_m, nn, nd, ndk)[[ \\
&\quad ComRM?msg_{dat}.m.r.nn.H(k1).E(k1, E((k_m^{-1}), nk\_d)) \\
&\quad \leftarrow ComRM?msg_{dat}.m.r.nn.H(k1).E(k1, E((k_m^{-1}), nk\_d)).E((k_a^{-1}), k_m), \\
&\quad CheckNK!msg_{pro}.E(k1, E((k_m^{-1}), nk\_d)).k^{-1}.k_m \\
&\quad \leftarrow CheckNK!msg_{pro}.E(k1, E((k_m^{-1}), nk\_d)).E((k_a^{-1}), k_m).k^{-1}.k_m.k_a]]
\end{aligned}$$

$$\begin{aligned}
& READER\_Sig(r, m, k, k_m, nn, nd, ndk) =_{df} \\
&\quad READER_1(r, m, k, k_m, nn, nd, ndk)[[ \\
&\quad ComRM?\{|ComRM|\} \leftarrow ComRM?\{|ComRM|\}, ComRM?\{|ComRM|\} \leftarrow FakeRM2?\{|ComRM|\}, \\
&\quad ComRM!\{|ComRM|\} \leftarrow ComRM!\{|ComRM|\}, ComRM!\{|ComRM|\} \leftarrow FakeRM2!\{|ComRM|\}]]
\end{aligned}$$

$$\begin{aligned}
& READER\_Dig(r, m, k, k_m, nn, nd, ndk) =_{df} \\
&\quad READER_2(r, m, k, k_m, nn, nd, ndk)[[ \\
&\quad ComRM?\{|ComRM|\} \leftarrow ComRM?\{|ComRM|\}, ComRM?\{|ComRM|\} \leftarrow FakeRM2?\{|ComRM|\}, \\
&\quad ComRM!\{|ComRM|\} \leftarrow ComRM!\{|ComRM|\}, ComRM!\{|ComRM|\} \leftarrow FakeRM2!\{|ComRM|\}]]
\end{aligned}$$

Here lists the definition of subprocesses *WRITER_Sig* and *WRITER_Dig*. They simulate the write operations.

$$\begin{aligned}
&\mathit{WRITER}_1(w, m, k, k_m, nn, nd, dk, data) =_{df}\\
&\quad \mathit{Initialization}\{n = \mathit{false}\} \rightarrow\\
&\quad \mathit{ComWM}!msg_{int}.w.m.nn \rightarrow \mathit{ComWM}?msg_{dat}.m.w.nn.hl \rightarrow\\
&\quad \mathit{ComWM}!msg_{int}.w.m.nn.H(k) \rightarrow\\
&\quad \mathit{ComWM}?msg_{dat}.m.w.nn.H(k1).E(k1, E((k_m^{-1}), (nk\_e, nk\_d))) \rightarrow\\
&\quad \mathit{ComWM}!msg_{dat}.w.m.nd.E(dk, data) \rightarrow \mathit{ComWM}?msg_{int}.m.w.ndk \rightarrow\\
&\quad \mathit{ComWM}!msg_{dat}.w.m.ndk.E(nk\_e, dk) \rightarrow\\
&\quad \mathit{CheckNK}!msg_{pro}.E(k1, E((k_m^{-1}), (nk\_e, nk\_d))).k^{-1}.k_m \rightarrow \mathit{CheckNK}?msg_{ack}.ack \rightarrow\\
&\quad \begin{pmatrix} (\mathit{NKFakingSuccess}\{n = \mathit{true}\} \rightarrow \mathit{SKIP})\\ \lhd(ack == \mathit{YES}) \rhd (\mathit{NKFakingError}\{n = \mathit{false}\} \rightarrow \mathit{SKIP}) \end{pmatrix};\\
&\quad \mathit{WRITER}_1(w, m, k, k_m, nn, nd, dk, data)
\end{aligned}$$

$$\begin{aligned}
&\mathit{WRITER}_2(w, m, k, k_m, k_a, nn, nd, dk, data) =_{df}\\
&\quad \mathit{WRITER}_1(w, m, k, k_m, nn, nd, dk, data)[[\\
&\quad \mathit{ComWM}?msg_{dat}.m.w.nn.H(k1).E(k1, E((k_m^{-1}), (nk\_e, nk\_d)))\\
&\quad \leftarrow \mathit{ComWM}?msg_{dat}.m.w.nn.H(k1).E(k1, E((k_m^{-1}), (nk\_e, nk\_d))).E(k_a^{-1}, k_m),\\
&\quad \mathit{CheckNK}!msg_{pro}.E(k1, E((k_m^{-1}), (nk\_e, nk\_d))).k^{-1}.k_m\\
&\quad \leftarrow \mathit{CheckNK}!msg_{pro}.E(k1, E((k_m^{-1}), (nk\_e, nk\_d))).E((k_a^{-1}), k_m).k^{-1}.k_m.k_a]]
\end{aligned}$$

$$\begin{aligned}
&\mathit{WRITER\_Sig}(w, m, k, k_m, nn, nd, dk, data) =_{df}\\
&\quad \mathit{WRITER}_1(w, m, k, k_m, nn, nd, dk, data)[[\\
&\quad \mathit{ComWM}?\{|\mathit{ComWM}|\} \leftarrow \mathit{ComWM}?\{|\mathit{ComWM}|\}, \mathit{ComWM}?\{|\mathit{ComWM}|\} \leftarrow \mathit{FakeWM2}?\{|\mathit{ComWM}|\},\\
&\quad \mathit{ComWM}!\{|\mathit{ComWM}|\} \leftarrow \mathit{ComWM}!\{|\mathit{ComWM}|\}, \mathit{ComWM}!\{|\mathit{ComWM}|\} \leftarrow \mathit{FakeWM2}!\{|\mathit{ComWM}|\}]]
\end{aligned}$$

$$\begin{aligned}
&\mathit{WRITER\_Dig}(w, m, k, k_m, k_a, nn, nd, dk, data) =_{df}\\
&\quad \mathit{WRITER}_2(w, m, k, k_m, k_a, nn, nd, dk, data)[[\\
&\quad \mathit{ComWM}?\{|\mathit{ComWM}|\} \leftarrow \mathit{ComWM}?\{|\mathit{ComWM}|\}, \mathit{ComWM}?\{|\mathit{ComWM}|\} \leftarrow \mathit{FakeWM2}?\{|\mathit{ComWM}|\},\\
&\quad \mathit{ComWM}!\{|\mathit{ComWM}|\} \leftarrow \mathit{ComWM}!\{|\mathit{ComWM}|\}, \mathit{ComWM}!\{|\mathit{ComWM}|\} \leftarrow \mathit{FakeWM2}!\{|\mathit{ComWM}|\}]]
\end{aligned}$$

Here shows the definition of subprocesses *ACM_R_Sig* and *ACM_R_Dig*. They simulate the behavior of ACM when it communicates with the readers.

$$\begin{aligned}
&\mathit{ACM\_R}_1(r, m, k_m, dk, data, nk\_e, nk\_d, hl) =_{df}\\
&\quad \mathit{ComRM}?msg_{int}.r.m.nd \rightarrow \mathit{ComRM}!msg_{dat}.m.r.nd.E(dk, data) \rightarrow\\
&\quad \mathit{ComRM}?msg_{int}.r.m.ndk \rightarrow \mathit{ComRM}!msg_{dat}.m.r.ndk.E(nk\_e, dk) \rightarrow\\
&\quad \mathit{ComRM}?msg_{int}.r.m.nn \rightarrow \mathit{ComRM}!msg_{dat}.m.r.nn.hl \rightarrow\\
&\quad \mathit{ComRM}?msg_{int}.r.m.nn.H(k) \rightarrow \mathit{ComRM}!msg_{dat}.m.r.nn.H(k).E(k, E(k_m^{-1}, nk\_d)) \rightarrow\\
&\quad \mathit{ACM\_R}_1(r, m, k_m, dk, data, nk\_e, nk\_d, hl)
\end{aligned}$$

$$\begin{aligned}
&\mathit{ACM\_R}_2(r, m, k_m, k_a, dk, data, nk\_e, nk\_d, hl) =_{df}\\
&\quad \mathit{ACM\_R}_1(r, m, k_m, dk, data, nk\_e, nk\_d, hl)[[\\
&\quad \mathit{ComRM}!msg_{dat}.m.r.nn.H(k).E(k, E(k_m^{-1}, nk\_d))\\
&\quad \leftarrow \mathit{ComRM}!msg_{dat}.m.r.nn.H(k).E(k, E(k_m^{-1}, nk\_d)).E(k_a^{-1}, k_m)]]
\end{aligned}$$

$$\begin{aligned}
&\mathit{ACM\_R\_Sig}(r, m, k_m, dk, data, nk\_e, nk\_d, hl) =_{df}\\
&\quad \mathit{ACM\_R}_1(r, m, k_m, dk, data, nk\_e, nk\_d, hl)[[\\
&\quad \mathit{ComRM}?\{|\mathit{ComRM}|\} \leftarrow \mathit{ComRM}?\{|\mathit{ComRM}|\}, \mathit{ComRM}?\{|\mathit{ComRM}|\} \leftarrow \mathit{FakeRM1}?\{|\mathit{ComRM}|\},\\
&\quad \mathit{ComRM}!\{|\mathit{ComRM}|\} \leftarrow \mathit{ComRM}!\{|\mathit{ComRM}|\}, \mathit{ComRM}!\{|\mathit{ComRM}|\} \leftarrow \mathit{FakeRM1}!\{|\mathit{ComRM}|\}]]
\end{aligned}$$

$$\begin{aligned}
&ACM\_R\_Dig(r, m, k_m, k_a, dk, data, nk\_e, nk\_d, hl) =_{df}\\
&\quad ACM\_R_2(r, m, k_m, k_a, dk, data, nk\_e, nk\_d, hl)[[\\
&\quad ComRM?\{|ComRM|\} \leftarrow ComRM?\{|ComRM|\}, ComRM?\{|ComRM|\} \leftarrow FakeRM1?\{|ComRM|\},\\
&\quad ComRM!\{|ComRM|\} \leftarrow ComRM!\{|ComRM|\}, ComRM!\{|ComRM|\} \leftarrow FakeRM1!\{|ComRM|\}]]
\end{aligned}$$

We illustrate the definition of subprocesses *ACM_W_Sig* and *ACM_W_Dig*. They simulate the behavior of ACM when it communicates with the writers.

$$\begin{aligned}
&ACM\_W_1(w, m, k_m, hl, nk\_e, nk\_d, ndk) =_{df}\\
&\quad Initialization\{d = false\} \rightarrow\\
&\quad ComWM?msg_{int}.w.m.nn \rightarrow ComWM!msg_{dat1}.m.w.nn.hl \rightarrow\\
&\quad ComWM?msg_{int}.w.m.nn.H(k) \rightarrow\\
&\quad ComWM!msg_{dat}.m.w.nn.H(k).E(k, E(k_m^{-1}, (nk\_e, nk\_d))) \rightarrow\\
&\quad ComWM?msg_{dat}.w.m.nd.E(dk, data) \rightarrow ComWM!msg_{int}.w.m.ndk \rightarrow\\
&\quad ComWM?msg_{dat}.m.w.ndk.E(nk\_e1, dk1) \rightarrow\\
&\quad GetData!msg_{pro}.E(dk, data).E(nk\_e1, dk1).nk\_d \rightarrow\\
&\quad GetData?msg_{ack}.ack \rightarrow\\
&\quad \left(\begin{array}{l}(DataAcquisitionSuccess\{d = true\} \rightarrow SKIP)\\ \lhd(ack == YES) \rhd (DataAcquisitionError\{d = false\} \rightarrow SKIP)\end{array}\right);\\
&\quad ACM\_W_1(w, m, hl, nk\_e, nk\_d, ndk)
\end{aligned}$$

$$\begin{aligned}
&ACM\_W_2(w, m, k_m, k_a, hl, nk\_e, nk\_d, ndk) =_{df}\\
&\quad ACM\_W_1(w, m, k_m, hl, nk\_e, nk\_d, ndk)[[\\
&\quad ComWM!msg_{dat}.m.w.nn.H(k).E(k, E(k_m^{-1}, (nk\_e, nk\_d)))\\
&\quad \leftarrow ComWM!msg_{dat}.m.w.nn.H(k).E(k, E(k_m^{-1}, (nk\_e, nk\_d)).E(K_a^{-1}, k_m)]]
\end{aligned}$$

$$\begin{aligned}
&ACM\_W\_Sig(w, m, hl, nk\_e, nk\_d, ndk) =_{df}\\
&\quad ACM\_W_1(w, m, hl, nk\_e, nk\_d, ndk)[[\\
&\quad ComWM?\{|ComWM|\} \leftarrow ComWM?\{|ComWM|\}, ComWM?\{|ComWM|\} \leftarrow FakeWM1?\{|ComWM|\},\\
&\quad ComWM!\{|ComWM|\} \leftarrow ComWM!\{|ComWM|\}, ComWM!\{|ComWM|\} \leftarrow FakeWM1!\{|ComWM|\}]]
\end{aligned}$$

$$\begin{aligned}
&ACM\_W\_Dig(w, m, hl, nk\_e, nk\_d, ndk) =_{df}\\
&\quad ACM\_W_2(w, m, hl, nk\_e, nk\_d, ndk)[[\\
&\quad ComWM?\{|ComWM|\} \leftarrow ComWM?\{|ComWM|\}, ComWM?\{|ComWM|\} \leftarrow FakeWM1?\{|ComWM|\},\\
&\quad ComWM!\{|ComWM|\} \leftarrow ComWM!\{|ComWM|\}, ComWM!\{|ComWM|\} \leftarrow FakeWM1!\{|ComWM|\}]]
\end{aligned}$$

We also update the definition of *PROCESS*.

$$\begin{aligned}
&PROCESS() =_{df}\\
&\quad CheckNK?msg_{pro}.E(k1, E((k_{m1}^{-1}), nk\_d)).k2^{-1}.k_{m2}\\
&\quad \left(\begin{array}{l}(checkNK!msg_{ack}.YES \rightarrow PROCESS())\\ \lhd((k1 == k2)\&\&(k_{m1} == k_{m2})\&\&(nk\_d == NK\_d\_f)) \rhd (CheckNK!msg_{ack}.NO \rightarrow PROCESS())\end{array}\right)\\
&\quad \square CheckNK?msg_{pro}.E(k1, E((k_{m1}^{-1}), nk\_d)).E((k_{a1}^{-1}), k_{m2}).k2^{-1}.k_{m3}.k_{a2} \rightarrow\\
&\quad \left(\begin{array}{l}(CheckNK!msg_{ack}.YES \rightarrow PROCESS())\\ \lhd((k1 == k2)\&\&(k_{a1} == k_{a2})\&\&(k_{m1} == k_{m3})\&\&(k_{m2} == k_{m3})\&\&(nk\_d == NK\_d\_f))\rhd\\ (CheckNK!msg_{ack}.NO \rightarrow PROCESS())\end{array}\right)\\
&\quad \square CheckNK?msg_{pro}.E(k1, E((k_{m1}^{-1}), (nk\_e, nk\_d))).k2^{-1}.k_{m2} \rightarrow\\
&\quad \left(\begin{array}{l}(CheckNK!msg_{ack}.YES \rightarrow PROCESS())\\ \lhd((k1 == k2)\&\&(k_{m1} == k_{m2})\&\&(nk\_e == NK\_e\_f)\&\&(nk\_d == NK\_d\_f))\rhd\\ (CheckNK!msg_{ack}.NO \rightarrow PROCESS())\end{array}\right)
\end{aligned}$$

$$\Box \mathit{CheckNK}?msg_{pro}.E(k1, E((k_{m1}^{-1}), (\mathit{nk\_e}, \mathit{nk\_d}))).E((k_{a1}^{-1}), k_{m2}).k2^{-1}.k_{m3}.k_{a2} \rightarrow$$

$$\begin{pmatrix} (\mathit{CheckNK}!msg_{ack}.YES \rightarrow PROCESS()) \\ \lhd((k1 == k2)\&\&(k_{a1} == k_{a2})\&\&(k_{m1} == k_{m3})\&\&(k_{m2} == k_{m3})\&\&(\mathit{nk\_e} == \mathit{NK\_e\_f})\&\&(\mathit{nk\_d} == \mathit{NK\_d\_f}))\rhd \\ (\mathit{CheckNK}!msg_{ack}.NO \rightarrow PROCESS()) \end{pmatrix}$$

$$\Box \mathit{GetData}?msg_{ack}.E(dk1, data).E(\mathit{nk\_e}, dk2).\mathit{nk\_d} \rightarrow$$

$$\begin{pmatrix} (\mathit{GetData}!msg_{ack}.YES \rightarrow PROCESS()) \\ \lhd((((\mathit{nk\_e} == \mathit{NK\_e})\&\&(\mathit{nk\_d} == \mathit{NK\_d}))||((\mathit{nk\_e} == \mathit{NK\_e\_f})\&\&(\mathit{nk\_d} == \mathit{NK\_d\_f}))) \\ \quad \&\&(dk1 == dk2)) \rhd (\mathit{GetData}!msg_{ack}.NO \rightarrow PROCESS()) \end{pmatrix}$$

We will add new element to *Fact* in [1], which is the set of facts which intruders might learn.

$$\mathit{Fact}_1 =_{df} \quad \mathit{Fact} \cup \{E(K, E(K_M^{-1}, content)) \mid content \in \{\mathit{NK\_d}, (\mathit{NK\_e}, \mathit{NK\_d}), \mathit{NK\_d\_f}, (\mathit{NK\_e\_f}, \mathit{NK\_d\_f})\}\}$$

And we also need to define a new deducing rule for the new element.

$$\{K^{-1}, E(K, E(K_M^{-1}, content)\} \mapsto E(K_M^{-1}, content)$$

We also add new definitions of how the intruders get new facts form messages:

$$\mathit{Info}(msg_{dat}.a.b.n.H(k).E(k, E(k_1, c_1))) =_{df} \{a, b, n, H(k), E(k, E(k_1, c_1))\}$$
$$\mathit{Info}(msg_{dat}.a.b.n.H(k).E(k, E(k_1, c_1)).E(k_2, c_2)) =_{df} \{a, b, n, H(k), E(k, E(k_1, c_1)), E(k_2, c_2)\}$$

where $a$,$b$ $\in$ *Entity*, $n \in$ *Name*, $k$,$k1$,$k2$ $\in$ *Key*, $c1$,$c2$ $\in$ *Content*.

Finally, we give the definition of *INTUDER_Sig* and *INTUDER_Dig*. They simulate the behavior of the intruders.

$$\begin{aligned}
& INTUDER_1(F) =_{df} \\
& \quad \Box\Box_{m \in MSG_{out}} \mathit{FakeRM1}?m \rightarrow \mathit{FakeRM2}!m \rightarrow INTUDER_1(F \cup \mathit{Info}(m)) \\
& \quad \Box\Box_{m \in (MSG_{out} \setminus MSG_{dat2})} \mathit{FakeRM2}?m \rightarrow \mathit{FakeRM1}!m \rightarrow INTUDER_1(F \cup \mathit{Info}(m)) \\
& \quad \Box\Box_{m \in MSG_{dat2}} \mathit{FakeRM2}?m \rightarrow \mathit{FakeRM1}!m[[k_m^{-1} \leftarrow K_I^{-1}, \mathit{nk\_d} \leftarrow \mathit{NK\_d\_f}]] \rightarrow \\
& \qquad INTUDER_1(F \cup \mathit{Info}(m)) \\
& \quad \Box\Box_{m \in MSG_{out}} \mathit{FakeWM1}?m \rightarrow \mathit{FakeWM2}!m \rightarrow INTUDER_1(F \cup \mathit{Info}(m)) \\
& \quad \Box\Box_{m \in (MSG_{out} \setminus MSG_{dat2})} \mathit{FakeWM2}?m \rightarrow \mathit{FakeWM1}!m \rightarrow INTUDER_1(F \cup \mathit{Info}(m)) \\
& \quad \Box\Box_{m \in MSG_{dat2}} \mathit{FakeWM2}?m \rightarrow \mathit{FakeWM1}!m[[k_m^{-1} \leftarrow K_I^{-1}, (\mathit{nk\_e}, \mathit{nk\_d}) \leftarrow (\mathit{NK\_e\_f}, \mathit{NK\_d\_f})]] \rightarrow \\
& \qquad INTUDER_1(F \cup \mathit{Info}(m)) \\
& \quad \Box\Box_{f \in Fact, f \notin F, F \mapsto f} \mathit{Initialization}\{l = false\} \rightarrow \mathit{Deduce}.f.F \rightarrow \\
& \qquad \begin{pmatrix} (\mathit{DataLeakageSuccess}\{l = true\} \rightarrow INTRUDER_1(F \cup \{f\})) \\ \lhd(f == Data) \rhd (\mathit{DataLeakageError}\{l = false\} \rightarrow INTRUDER_1(F \cup \{f\})) \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
& INTUDER\_Sig =_{df} \\
& \quad INTUDER_1(IK \cup \{K_I, K_I^{-1}, K_M\}) \\
& INTUDER\_Dig =_{df} \\
& \quad INTUDER_1(IK \cup \{K_I, K_I^{-1}, K_M, K_A\})
\end{aligned}$$