MODULE *EnCrypto*

LOCAL INSTANCE *Sequences*
LOCAL INSTANCE *Naturals*
LOCAL INSTANCE *SSWPacket*
LOCAL INSTANCE *Modbus*

LOCAL $HMACSIZE \triangleq 64$
LOCAL $MINMESSAGESIZE \triangleq 1$
LOCAL $MINMACMESSAGESIZE \triangleq 65$
LOCAL $PASSWORD \triangleq$ "lolpassword"
LOCAL $BareMessages \triangleq \{\langle$ ":", "1", "1", "0", "3", "0", "0", "6", "B", "0", "0", "0", "3", "7", "E", "C", "R",
$\langle$ "D", "9", "2", "8", "D", "9", "2", "8",
"7", "5", "3", "0", "7", "5", "3", "0",
"9", "8", "5", "C", "9", "8", "5", "C",
"E", "B", "B", "A", "E", "B", "B", "A",
"D", "9", "2", "8", "D", "9", "2", "8",
"7", "5", "3", "0", "7", "5", "3", "0",
"9", "8", "5", "C", "9", "8", "5", "C",
"E", "B", "B", "A", "E", "B", "B", "A",
"9", "8", "5", "C", "9", "8", "5", "C",
"E", "B", "B", "A", "E", "B", "B", "A",
"D", "9", "2", "8", "D", "9", "2", "8",
"7", "5", "3", "0", "7", "5", "3", "0",
"9", "8", "5", "C", "9", "8", "5", "C" $\rangle$,
$\langle\rangle\}$

$HMAC(str, pass) \triangleq \langle$ "l", "o", "l", "h", "m", "a", "c" $\rangle$ not concerned with the inner workings of *SHA*2

$SendMessage(str) \triangleq$ TRUE sending message to another cell. Assuming this works

--fair algorithm *EnCrypto*

variables $macMessage = \langle\rangle$,
$hmac = \langle\rangle$,
$bareMessage \in BareMessages$,
$flag =$ FALSE,
$generatedHMAC = \langle\rangle$,
$result =$ FALSE

begin

msgCheck: **if** $Len(bareMessage) \geq MINMESSAGESIZE$
then $hmac := HMAC(bareMessage, PASSWORD)$ **;** hash it and the password
*h*1: $macMessage := \langle$ "!" $\rangle \circ hmac \circ bareMessage$ **;**
$result := SendMessage(macMessage)$ **;**
end if ;

end algorithm

BEGIN TRANSLATION

VARIABLES $macMessage$, $hmac$, $bareMessage$, $flag$, $generatedHMAC$, $result$, pc

$vars \triangleq \langle macMessage,\ hmac,\ bareMessage,\ flag,\ generatedHMAC,\ result,\ pc \rangle$

$$
\begin{aligned}
Init \triangleq\ & \text{Global variables} \\
& \land\ macMessage = \langle\rangle \\
& \land\ hmac = \langle\rangle \\
& \land\ bareMessage \in BareMessages \\
& \land\ flag = \text{FALSE} \\
& \land\ generatedHMAC = \langle\rangle \\
& \land\ result = \text{FALSE} \\
& \land\ pc = \text{“msgCheck”}
\end{aligned}
$$

$$
\begin{aligned}
msgCheck \triangleq\ & \land\ pc = \text{“msgCheck”} \\
& \land\ \text{IF}\ Len(bareMessage) \geq MINMESSAGESIZE \\
& \qquad \text{THEN}\ \land\ hmac' = HMAC(bareMessage,\ PASSWORD) \\
& \qquad\qquad\ \ \land\ pc' = \text{“h1”} \\
& \qquad \text{ELSE}\ \land\ pc' = \text{“Done”} \\
& \qquad\qquad\ \ \land\ hmac' = hmac \\
& \land\ \text{UNCHANGED}\ \langle macMessage,\ bareMessage,\ flag,\ generatedHMAC, \\
& \qquad\qquad result \rangle
\end{aligned}
$$

$$
\begin{aligned}
h1 \triangleq\ & \land\ pc = \text{“h1”} \\
& \land\ macMessage' = \langle \text{“!”} \rangle \circ hmac \circ bareMessage \\
& \land\ result' = SendMessage(macMessage') \\
& \land\ pc' = \text{“Done”} \\
& \land\ \text{UNCHANGED}\ \langle hmac,\ bareMessage,\ flag,\ generatedHMAC \rangle
\end{aligned}
$$

$$
\begin{aligned}
Next \triangleq\ & msgCheck \lor h1 \\
& \lor\ \text{Disjunct to prevent deadlock on termination} \\
& \quad (pc = \text{“Done”} \land \text{UNCHANGED}\ vars)
\end{aligned}
$$

$$
\begin{aligned}
Spec \triangleq\ & \land\ Init \land \Box[Next]_{vars} \\
& \land\ \mathrm{WF}_{vars}(Next)
\end{aligned}
$$

$Termination \triangleq \Diamond(pc = \text{“Done”})$

END TRANSLATION

$$
\begin{aligned}
SAFETYCHECK \triangleq\ & \\
& \text{The buffer that gets forwarded to the next cell} \\
& \text{can only contain valid } SSW \text{ or be empty} \\
& \land\ IsSSW(macMessage) \lor macMessage = \langle\rangle \\
& \text{The message gets sent if and only if its valid } SSW \\
& \land\ IsSSW(macMessage) \equiv result = \text{TRUE} \\
& \text{the message in the buffer is at least the minimum } SSW \text{ size} \\
& \land\ Len(macMessage) \geq MINMACMESSAGESIZE \lor Len(macMessage) = 0
\end{aligned}
$$

the plaintext can never be sent without being processed
$\land\ bareMessage \neq macMessage$
the password is never changed
$\land\ PASSWORD =$ "lolpassword"

$LIVELINESS \triangleq$
if we get a message
then something is eventually sent sent
$\land\ Len(bareMessage) \geq MINMESSAGESIZE \rightsquigarrow result = \text{TRUE}$
if we get a message it is eventually processed
$\land\ Len(bareMessage) \geq MINMESSAGESIZE \rightsquigarrow IsSSW(macMessage)$

* Modification History
* Last modified *Tue* May 08 03:38:23 *EDT* 2018 by *SabraouM*
* Created Sun May 06 15:34:11 *EDT* 2018 by *SabraouM*