

	MODULE <i>MCSpec</i>	
EXTENDS <i>Spec</i> , <i>Attacker</i>		
$ \begin{aligned} MCInit &\triangleq \\ &\quad \wedge \textit{Init} \\ &\quad \wedge \textit{AttackerInit} \end{aligned} $		
$ \begin{aligned} MCNext &\triangleq \\ &\quad \vee \quad \wedge \textit{Next} \\ &\quad \quad \wedge \text{UNCHANGED } \textit{attacker_vars} \\ &\quad \vee \quad \wedge \textit{AttackerAction} \\ &\quad \quad \wedge \text{UNCHANGED } \textit{agent_vars} \end{aligned} $		
$MCSpec \triangleq MCInit \wedge \Box[MCNext]_{(varsoattacker_vars)}$		
$TypeCheckSeq(seq, set) \triangleq \forall i \in 1 \dots Len(seq) : seq[i] \in set$		
$ \begin{aligned} TypeCheckSet(set, type) &\triangleq \\ &\quad \wedge IsFiniteSet(set) \\ &\quad \wedge \forall e \in set : e \in type \end{aligned} $		
$ \begin{aligned} TypeOK &\triangleq \\ &\quad \wedge \textit{used_num} \subseteq 1 \dots \textit{maxPubNum} \\ &\quad \wedge \forall s \in \textit{subscribers} \setminus \{\textit{attacker}\} : Len(store[s]) \leq \textit{maxPubNum} \\ &\quad \wedge \forall t \in \textit{topics}, q \in \{QoS0, QoS1, QoS2\} : \\ &\quad \quad \wedge TypeCheckSeq(store[broker][t][q], 1 \dots \textit{maxPubNum} * 2) \\ &\quad \quad \wedge TypeCheckSet(topic_subscribers[t][q], \textit{subscribers}) \end{aligned} $		
$ \begin{aligned} AllSubscribedBeforePublish &\triangleq \\ &\quad \forall p \in \textit{publishers} : \\ &\quad \quad (pc[p] = \text{"publishingwithqos1"} \vee pc[p] = \text{"publishingwithqos2"}) \Rightarrow \\ &\quad \quad (\forall s \in \textit{subscribers} : pc[s] = \text{"connected"} \wedge \exists t \in \textit{topics}, q \in \{QoS0, QoS1, QoS2\} : s \in \textit{topic_subscribers}[t][q]) \end{aligned} $		
$ \begin{aligned} AllMessagesSentAndPushedBeforeUnsubscribe &\triangleq \\ &\quad \forall s \in \textit{subscribers} : \\ &\quad \quad pc[s] = \text{"unsubscribing"} \Rightarrow \\ &\quad \quad (\forall p \in \textit{publishers} : store[p] = \{\}) \wedge \\ &\quad \quad (\forall t \in \textit{topics}, q \in \{QoS0, QoS1, QoS2\} : store[broker][t][q] = \langle \rangle) \end{aligned} $		
$LenConstraint \triangleq \forall a1, a2 \in \textit{agents} : Len(network[a1]) + Len(network[a2]) \leq 2$		
$ \begin{aligned} SpecConstraints &\triangleq \\ &\quad \wedge AllSubscribedBeforePublish \\ &\quad \wedge AllMessagesSentAndPushedBeforeUnsubscribe \\ &\quad \wedge LenConstraint \end{aligned} $		