```
InitReceiver \stackrel{\Delta}{=}
                                        MODULE AuthRepSys
{\tt EXTENDS}\ Integers,\ TLC,\ Sequences,\ FiniteSets
                                                                                                                               \land \ rState = [r \in Receiver \mapsto R\_Wait]
                                                                                                                       CONSTANT Provider
CONSTANT Segment
                                                                                                                               \land \, wState = [w \in \mathit{Watchdog} \mapsto \mathit{W\_Working}]
                                                                                                                               \land wBuffer = [w \in Watchdog \mapsto \{\}]
CONSTANT Miner
CONSTANT Receiver
                                                                                                                               \land \ wProcessed = [w \in Watchdog \mapsto \{\}]
                                                                                                                       Constant Judge
CONSTANT Watchdog
                                                                                                                               \land \ txPool = \{\}
Constant PreGenPackets
                                                                                                                               \land txBlock = \{\}
CONSTANT PacketObject
                                                                                                                               \land malicious = \{\}
                                                                                                                               \land benign = Participant
Constant RToken, PToken
                                                                                                                               \wedge balance =
                                                                                                                                     [e \in Participant \mapsto [ty \in TokenType \mapsto InitialBalance]]
CONSTANT InitialBalance
                                                                                                                               \land \mathit{fPrint} = [\mathit{id} \in \mathit{PacketObject}, \ p \in \mathit{Participant} \ \mapsto \{\}]
Constant RegenTokenMin
CONSTANT TokenMin
                                                                                                                               \land malPackets = \{\}
                                                                                                                       Init \stackrel{\triangle}{=}
Constant N_-Wait, N_-NormalSend
Constant N_-PacketDrop,\ N_-Modification,\ N_-OnOff
                                                                                                                               \land InitPacketGen
                                                                                                                               \land InitSegment
Constant R_Wait, R_Eval, R_Done
CONSTANT W_-Working
                                                                                                                               \land InitReceiver
Constant RT
                                                                                                                               \land\ InitWatchdog
CONSTANT Coverage
                                                                                                                               \land\ InitBlockchain
                                                                                                                               \land channel = \{\}
CONSTANT PacketMod
                                                                                                                       PacketGen \stackrel{\Delta}{=}
                                                                                                                              \exists \ p \in pendingPackets: \exists \ n \in Segment: \exists \ pp \in pId: \\ \text{LET } msg \ \stackrel{\triangle}{=} \ [
{\tt VARIABLE}\ nState,\ rState,\ wState
Variable nBuffer, rBuffer, wBuffer
{\tt VARIABLE} \ wProcessed
                                                                                                                                            id
VARIABLE channel
                                                                                                                                             isrc \mapsto Provider,
{\tt VARIABLE}\ txPool,\ txBlock
                                                                                                                                             idst \ \mapsto n,
{\tt VARIABLE}\ pendingPackets,\ sentPackets
                                                                                                                                             dst \mapsto p.dst,
{\tt VARIABLE}\ checksum,\ checksum Id
                                                                                                                                             data \mapsto p.payload,
VARIABLE pId
                                                                                                                                             elapsed \mapsto 0,
VARIABLE balance
                                                                                                                                             cksum \mapsto checksumId]
{\tt VARIABLE} \ malicious, \ benign, \ malPackets
                                                                                                                                   IN
{\tt VARIABLE}\, f\!Print
                                                                                                                                     \land \ pendingPackets' = pendingPackets \setminus \{p\}
                                                                                                                                     \wedge \ pId' = pId \setminus \{pp\}
                                                                                                                                     \land Send(msg)
{\tt VARIABLE}\ firstSeen
                                                                                                                                     \land \ sentPackets' = sentPackets \cup \{msg\}
                                                                                                                                     \land checksum' =
\begin{array}{l} \textit{genVars} \ \stackrel{\triangle}{=} \ \langle \textit{pendingPackets}, \ \textit{sentPackets}, \ \textit{pId}, \ \textit{checksum}, \ \textit{checksumId} \rangle \\ \textit{commVars} \ \stackrel{\triangle}{=} \ \langle \textit{channel} \rangle \end{array}
                                                                                                                                           [checksum\_\texttt{EXCEPT}~![checksumId] = \langle p.dst,~p.payload \rangle]
\begin{array}{c} commVars \stackrel{\triangle}{=} \langle channel \rangle \\ nodeVars \stackrel{\triangle}{=} \langle nState, nBuffer \rangle \\ \\ \end{array}
                                                                                                                                     \land checksumId' = checksumId + 1
                                                                                                                                     \land \ \mathtt{UNCHANGED} \ \langle node \mathit{Vars}, \ \mathit{receiverVars}, \ \mathit{watchdogVars}, \\
blockchainVars
                                                                                                                       \begin{array}{c} SegmentRecv(n) \stackrel{\triangle}{=} \\ & \land \ \underline{n}State[n] = N\_Wait \end{array}
                                         fPrint, malPackets>
                                                                                                                               \land \exists p \in channel:
vars \stackrel{\Delta}{=} \langle genVars, commVars, nodeVars, receiverVars, watchdogVars,
                                                                                                                                       \land p.idst = n
                                                                                                                                      blockchain Vars
RECURSIVE Sum(\_,\_) Sum(f,~S) \overset{\Delta}{=} \text{ if } S = \{\} \text{ then } 0
                                                                                                                                      \land UNCHANGED \langle genVars, receiverVars, watchdogVars,
                                                 ELSE LET x \stackrel{\Delta}{=} \text{CHOOSE } x \in S : \text{TRUE}
                                                                                                                                            blockchain Vars
                                                              IN f[x] + Sum(f, S \setminus \{x\})
                                                                                                                       SegmentNormalSend(n) \stackrel{\Delta}{=}
\begin{array}{c} \mathit{Send}(p) \stackrel{\Delta}{=} \mathit{channel'} = \mathit{channel} \cup \{p\} \\ \mathit{Receive}(p) \stackrel{\Delta}{=} \mathit{channel'} = \mathit{channel} \setminus \{p\} \end{array}
                                                                                                                               \land nState[n] = N\_NormalSend
                                                                                                                                     \begin{array}{l} \mathbf{ET} & \overset{\triangle}{\rightarrow} Head(nBuf\!f\!er[n]) \\ et & \overset{\triangle}{\rightarrow} CHOOSE \ x \in NormalTxTime : \texttt{TRUE} \\ np & \overset{\triangle}{\rightarrow} [p \ \texttt{EXCEPT} \ !.isrc = n, \ !.idst = RT[n], \\ & \ !.elapsed = @ + et] \end{array} 
                                                                                                                               \wedge LET
\begin{array}{ccc} \mathit{Transfer}(\mathit{src}, \, \mathit{dst}, \, \mathit{ty}, \, \mathit{amount}, \, \mathit{f}) \, \stackrel{\Delta}{=} \\ \text{Let} \, \, \mathit{tx} \, \stackrel{\Delta}{=} \, [\mathit{from} \mapsto \mathit{src}, \, \mathit{to} \mapsto \mathit{dst}, \\ \end{array}
                       token \mapsto ty, \ value \ \mapsto amount, \ fingerprint \mapsto f]
        in txPool' = txPool \cup \{tx\}
                                                                                                                                   IN Send(np)
                                                                                                                               CheckBalance(address, ty, amount) \stackrel{\Delta}{=}
                                                                                                                               \land \ \mathtt{UNCHANGED} \ \langle \mathit{genVars}, \ \mathit{receiverVars}, \ \mathit{watchdogVars}, \\
        balance[address][ty] \geq amount
                                                                                                                                     blockchainVars
 \begin{array}{ll} Evaluator & \stackrel{\triangle}{=} Receiver \cup Watchdog \\ Participant & \stackrel{\triangle}{=} Segment \cup Evaluator \cup PacketObject \\ TokenType & \triangleq \{RToken,\ PToken\} \\ NormalTxTime & \vdash 1 \dots 10 \end{array} 
                                                                                                                       SegmentPacketDrop(n) \stackrel{\Delta}{=}
                                                                                                                                \begin{array}{l} \land nState[n] = N.PacketDrop \\ \land nBuffer' = [nBuffer\ \texttt{EXCEPT}\ !\ [n] = Tail(@)] \\ \land nState' = [nState\ \texttt{EXCEPT}\ !\ [n] = N.Wait] \end{array} 
GetFingerprint(p) \stackrel{\Delta}{=} \langle p.id, p.dst, p.data, p.idst \rangle
                                                                                                                               \land \ \mathtt{UNCHANGED} \ \langle \mathit{genVars}, \ \mathit{receiverVars}, \ \mathit{watchdogVars}, \ \mathit{commVars}, \\
                                                                                                                                     blockchainVars
InitPacketGen \stackrel{\Delta}{=}
        \land\ checksumId = 1
                                                                                                                       SegmentOnOffForwarding(n) \stackrel{\Delta}{=}
                                                                                                                               \land checksum = [i \in checksum Id .
              (checksum\dot{I}d + Cardinality(PreGenPackets)) \mapsto \langle \rangle ]
         \land pId = PacketObject
         \land pendingPackets = PreGenPackets
                                                                                                                                                                 = IF b THEN N_NormalSend ELSE N_PacketDrop
                                                                                                                               \land UNCHANGED \langle genVars, receiverVars, watchdogVars, commVars,
        \land sentPackets = \{\}
                                                                                                                                     nBuffer, blockchainVars\rangle
InitSegment \stackrel{\Delta}{=}
                                                                                                                       SegmentModification(n) \stackrel{\Delta}{=}
         \land nState = [n \in Segment \mapsto N_Wait]
         \land nBuffer = [n \in Segment \mapsto \langle \rangle]
                                                                                                                               \land nState[n] = N\_Modification
```

```
\wedge LET
                                                                                                                                                                                                                THEN balance' = [balance \ EXCEPT
                     p \triangleq
                     \begin{array}{ll} & \stackrel{\triangle}{\to} Head(nBuffer[n]) \\ et & \stackrel{\triangle}{=} CHOOSE \ x \in NormalTxTime : \text{TRUE} \\ np & \stackrel{\triangle}{=} \left[p \ \text{EXCEPT} \ ! . isrc = n, \ ! . idst = RT[p.dst], \\ & \quad ! . elapsed = @ + et, \ ! . data = @ + PacketMod] \end{array}
                                                                                                                                                                                                                ![e] = [@EXCEPT ![ty] = @+2*RegenTokenMin]]
              \land nBuffer' = [nBuffer \text{ except } ! [n] = Tail(@)] 
                                                                                                                                                                                            Confirm\,Tx\ \stackrel{\Delta}{=}
             \wedge \ nState' = [nState \ \texttt{EXCEPT} \ ! [n] = N_{-}Wait]
                                                                                                                                                                                                       \exists tx \in txPool:
                                                                                                                                                                                                                \land \ txPool' = txPool \setminus \{tx\}
             \land UNCHANGED \langle genVars, receiverVars, watchdogVars,
                                                                                                                                                                                                                 \begin{array}{l} \text{At } bot = \underbrace{\text{Lit} \ bot}_{\text{LET } e} \underbrace{\text{Let} \ bot}_{\text{LET } e} \underbrace{\text{Lit} 
                       blockchainVars
ReceiverRecv(r) \triangleq
                                                                                                                                                                                                                           \land rState[r] = R_-Wait
             \land \exists p \in channel:
                        \land \ p.idst = r
                        \land Receive(p)
                                                                                                                                                                                                                                  \begin{aligned} sp &= t. \text{Jinger print}[S] \text{IN} \\ \text{IF } fPrint[id, \ dst] &= \{\} \\ \text{THEN } fPrint' &= [fPrint \ \text{EXCEPT} \ ! \ [id, \ dst] = \{sfp\}] \\ \text{ELSE } \text{UNCHANGED } fPrint \end{aligned} 
                        \land \ \ \mathsf{UNCHANGED} \ \langle \mathit{genVars}, \ \mathit{nodeVars}, \ \mathit{rState}, \ \mathit{watchdogVars}, \ \mathit{txBlock}, \\
                                                                                                                                                                                                                           \land \text{ IF } n \in Segment \cup PacketObject \Rightarrow \land JudgeDecision(n)
                    balance,\ malicious,\ benign,\ fPrint,\ malPackets\rangle
                                                                                                                                                                                                                                                                              \land IntegrityCheck(tx.fingerprint)
Reliable Delivery(p1, p2) \stackrel{\Delta}{=}
                                                                                                                                                                                                                                      THEN DecideBenign(n) \land \mathtt{UNCHANGED} \ \langle malicious, \ malPackets \rangle
             \land \ p2.elapsed - p1.elapsed \in NormalTxTime
                                                                                                                                                                                                                                      \land \ p1.data = p2.data
                                                                                                                                                                                                                                                            \land \ \mathtt{UNCHANGED} \ benign
IsPacketSeenFirst(p) \stackrel{\Delta}{=}
                                                                                                                                                                                                                           \land e \in Evaluator
             \land \exists w \in Watchdog : p \in wBuffer[w]
                                                                                                                                                                                                                                      \land \ balance[e][tx.token] \leq \textit{TokenMin}
             \land \forall n \in Segment : p \notin \{nBuffer[n][np] : np \in DOMAIN \ nBuffer[n]\}
                                                                                                                                                                                                                                                 \Rightarrow EvaluatorReview(e, tx.token)
                                                                                                                                                                                                                        ELSE UNCHANGED \langle txBlock, \ balance \rangle
 WatchdogSeen(w) \stackrel{\Delta}{=}
                                                                                                                                                                                                                 \land UNCHANGED \langle genVars, nodeVars, commVars, receiverVars,
            \exists \; p \, \in \, channel :
                                                                                                                                                                                                                          watchdog Vars \rangle
                    \land \, wState[w] = \, W\_Working
                                                                                                                                                                                            SuccesfullyDelivered \stackrel{\Delta}{=}
                     \land p \notin wProcessed[w]
                     \begin{array}{l} \wedge \ (p.isrc \in Coverage[w] \lor p.idst \in Coverage[w]) \\ \wedge \ wBuffer' = [wBuffer \ \ Except \ \ ![w] = @ \cup \{p\}] \end{array} 
                                                                                                                                                                                                         \land \forall r \in Receiver :
                                                                                                                                                                                                                  \forall \ p \in \{\mathit{rBuffer}[r][x] : x \in \mathit{domain} \ \mathit{rBuffer}[r]\} :
                     \land IF IsPacketSeenFirst(p)
                                                                                                                                                                                                                          \lor checksum[p.cksum] = \langle r, p.data \rangle
                                                                                                                                                                                                         \land \ \forall \ sentPacket \in \vec{PreGenPackets}:
                             Then firstSeen' = firstSeen \cup \{p\} else unchanged firstSeen
                     \land \ \mathtt{UNCHANGED} \ \langle \mathit{genVars}, \ \mathit{nodeVars}, \ \mathit{commVars}, \ \mathit{receiverVars}, \\
                                                                                                                                                                                                                  \exists r \in Receiver :
                                                                                                                                                                                                                          sentPacket.payload \in
                               wState, wProcessed, blockchainVars \rangle
                                                                                                                                                                                                                                      \{rBuffer[r][x].data: x \in \text{domain } rBuffer[r]\}
 WatchdogCheck(w) \stackrel{\Delta}{=}
                                                                                                                                                                                            AllDelivered \stackrel{\Delta}{=}
            \exists p1 \in wBuffer[w] : \exists p2 \in wBuffer[w] :
                                                                                                                                                                                                         \land \forall r \in Receiver :
                    \wedge p1 \neq p2
                                                                                                                                                                                                                  \forall \ p \in \{\mathit{rBuffer}[r][x] : x \in \mathit{domain} \ \mathit{rBuffer}[r]\} :
                     \land \ p1.idst = p2.isrc
                     \begin{array}{l} \land p1.ust = p2.usrc \\ \land Reliable Delivery(p1,\ p2) \\ \land Transfer(w,\ p1.idst,\ RToken,\ 1,\ GetFingerprint(p2)) \\ \land wBuffer' = [wBuffer\ except \ ![w] = @ \setminus \{p1,\ p2\}] \\ \land wProcessed' = [wProcessed\ except \ ![w] = @ \cup \{p1,\ p2\}] \\ \land \text{ WNCHANGED } \langle gen\ Vars,\ node\ Vars,\ comm\ Vars,\ receiver\ Vars, \\ \hline FortScore \ wState \ trRlock\ balance.\ malicious.\ benign, \\ \end{array} 
                                                                                                                                                                                                                            \begin{array}{l} \lor \ checksum[p.cksum] = \langle r,\ p.data \rangle \\ \lor \ checksum[p.cksum] = \langle r,\ p.data - PacketMod \rangle \end{array} 
                                                                                                                                                                                                         \land \ \forall \ sentPacket \in PreGenPackets:
                                                                                                                                                                                                                  \exists r \in Receiver :
                                                                                                                                                                                                                          sentPacket.payload \in
                                                                                                                                                                                                                                       \begin{cases} rBuffer[r][x].data: x \in \text{DOMAIN } rBuffer[r]\} \cup \\ \{rBuffer[r][x].data - PacketMod: x \in \text{DOMAIN } rBuffer[r]\} \end{cases} 
                               firstSeen, wState, txBlock, balance, malicious, benign,
                              fPrint,\ malPackets\rangle
                                                                                                                                                                                            AllProcessed \stackrel{\Delta}{=}
 WatchdogReview(w) \stackrel{\Delta}{=}
                                                                                                                                                                                                         \land \ pendingPackets = \{\}
            \exists p \in wBuffer[w]:
                                                                                                                                                                                                         \land \ \forall \ n \in Segment : nBuffer[n] = \langle \rangle
                    \land IF p.idst \in Segment
                            THEN Transfer(w, p.idst, PToken, 1, GetFingerprint(p))
                                                                                                                                                                                                         \land channel = \{\}
                             ELSE UNCHANGED txPool
                    IsSecurelyDelivered(p) \stackrel{\Delta}{=}
                                                                                                                                                                                                       \exists r \in Receiver :
                                                                                                                                                                                                                         \langle p.id,\ p.data \rangle \in
                    \land \ \mathtt{UNCHANGED} \ \langle \mathit{genVars}, \ \mathit{nodeVars}, \ \mathit{commVars}, \ \mathit{receiverVars}, \\
                                                                                                                                                                                                                         \{\langle rBuf\!f\!er[r][rp].id,\ rBuf\!f\!er[r][rp].data\rangle: rp\in \operatorname{domain}\ rBuf\!f\!er[r]\}
                               firstSeen, wState, txBlock, balance, malicious, benign,
                               fPrint, malPackets)
                                                                                                                                                                                            WorkingProperly \stackrel{\Delta}{=}
\begin{array}{ll} DecideMalicious(n) \stackrel{\triangle}{=} malicious' = malicious \cup \{n\} \\ DecideBenign(n) \stackrel{\triangle}{=} benign' = benign \cup \{n\} \end{array}
                                                                                                                                                                                                        \forall p \in sentPackets:
                                                                                                                                                                                                               \mathit{IsPacketSeenFirst}(p) \leadsto \ \lor \ p.id \in \mathit{malPackets}
                                                                                                                                                                                                                                                                                             \vee IsSecurelyDelivered(p)
Termination \stackrel{\Delta}{=}
           IN
                                                                                                                                                                                                         \land txPool = \{\}
                                                                                                                                                                                                         \land \ \forall \ w \in \ \widetilde{Watchdog} : wBuffer[w] = \{\}
             \land (score \ge 0)
                                                                                                                                                                                                         \land All Processed
\begin{array}{ccc} IntegrityCheck(fp) & \stackrel{\triangle}{=} \\ & \text{LET } id & \stackrel{\triangle}{=} fp[1] \\ & storedFp & \stackrel{\triangle}{=} fPrint[id, \ dst] \end{array}
                                                                                                                                                                                                         \land \ \forall \ r \in \mathit{Receiver} : \mathit{rState}[r] = \mathit{R\_Wait}
                                                                                              pfp \triangleq fp[3]
                                                                                                                                                                                                         \land \forall \ p \in \mathit{firstSeen} : p.id \in \mathit{malPackets} \lor \mathit{IsSecurelyDelivered}(p)
                                                                                                                                                                                                         ∧ UNCHANGED vars
            storedFp \neq \{\} \Rightarrow pfp \in storedFp
                                                                                                                                                                                                       LET totalBalance \stackrel{\Delta}{=} InitialBalance * Cardinality(Participant)
\begin{array}{c} \operatorname{RECURSIVE} \ Sum \ Token(\_, \ \_, \ \_) \\ Sum \ Token(f, \ S, \ t) \ \stackrel{\Delta}{=} \end{array}
                                                                                                                                                                                                       IN Sum(balance, Domain balance) = totalBalance
           \begin{array}{c} \text{If } S = \{\} \text{ Then } 0 \text{ else} \\ \text{ let } x \stackrel{\triangle}{=} \text{ choose } x \in S : \text{true} \end{array}
                       IN f[x][t] + SumToken(f, S \setminus \{x\}, t)
                                                                                                                                                                                            Delivered \triangleq
                                                                                                                                                                                                        \forall p \in sentPackets:
 \begin{array}{c} EvaluatorReview(e_{\underline{t}}\ ty)\ \stackrel{\triangle}{=}\\ \land \ \mathsf{LET}\ tTotal\ \stackrel{\triangle}{=}\ SumToken(balance,\ Evaluator,\ ty) \mathsf{IN} \end{array} 
                                                                                                                                                                                                               ENABLED IsPacketSeenFirst(p) \Rightarrow IsSecurelyDelivered(p)
                                                                                                                                                                                           Next \triangleq
                   \text{IF } (2*balance[e][ty]*Cardinality(Evaluator) < tTotal) \\
```

```
 \lor PacketGen \\ \lor \exists n \in Segment : SegmentRecv(n) \lor SegmentNormalSend(n) \\ \lor SegmentPacketDrop(n) \lor SegmentModification(n) \lor SegmentOnOffForwarding(n) \\ \lor \exists r \in Receiver : ReceiverRecv(r) \\ \lor \exists w \in Watchdog : WatchdogSeen(w) \lor WatchdogCheck(w) \\ \lor (AllProcessed \land \exists w \in Watchdog : \\ wBuffer[w] \neq \{\} \land WatchdogReview(w)) \\ \lor ConfirmTx \\ \lor Termination \\ Spec \triangleq Init \land \Box[Next]_{vars}
```