MODULE *Contracts*

This spec captures the behaviour of commitment transactions on the two sides of a Lightning channel.

We model the various kinds of outputs a commitment transactions will have over its lifetime.

The state of the commitment transaction changes in reponse to the various actions like supercede, spend, revoke etc are taken.

We also do not deal with the communication protocol between nodes for creating and updating commitment transactions. This spec only focusses on the various commitment transaction created, revoked, spent to open, close, force close or penalise.

We ignore the details of how transactions are signed and just mark transactions as signed. This lets us focus on the specifying the behaviour of the commitment transactions without dealing with lower level complexities.

EXTENDS *Integers*, *TLC*, *Sequences*

CONSTANTS
- CSV, — The csv value to use in contracts
- $Height$ — The height up to which we run the spec

Channel contracts only ever have two parties

$Party \triangleq \{\text{“alice”}, \text{“bob”}\}$

For the first revocation we only need two keys per party

$NumKey \triangleq 2$

Set of all keys

$Key \triangleq \{\langle p, k\rangle : p \in Party, k \in 0 \mathrel{..} NumKey - 1\}$

Value to capture missing CSV in output

$NoCSV \triangleq \text{CHOOSE } c : c \notin 0 \mathrel{..} CSV$

Multisig outputs without CSV encumberance

$MultiSig \triangleq Party \times Party \times \{NoCSV\}$

Multisig outputs with CSV encumberance

$MultiSigWithCSV \triangleq Party \times Party \times \{CSV\}$

$P2PKH$ outputs, without encumbrance

$P2PKH \triangleq Key$

$AllOutput \triangleq MultiSig \cup MultiSigWithCSV \cup P2PKH$

$NoOutput \triangleq \text{CHOOSE } o : o \notin AllOutput$

Set of all signatures for all commit txs. The signature in real world is related to the commit transaction, however, leave out this complication of how the signature is generated. If there is a signature by a key on a tx, it is assumed it is correctly signed as per bitcoin's requirements

$Sig \triangleq \{\langle p, k\rangle : p \in Party, k \in 0 \,..\, NumKey - 1\}$

Value to capture unsigned transactions

$NoSig \triangleq \text{CHOOSE } s : s \notin Sig$

VARIABLES
- $outputs$, The set of all commiment transactions for both parties
- $local\_sigs$,
- $remote\_sigs$

$vars \triangleq \langle outputs, local\_sigs, remote\_sigs\rangle$

$$
\begin{aligned}
Init \triangleq\ & \land\ outputs = [p \in Party \mapsto \langle\rangle] \\
& \land\ local\_sigs = [p \in Party \mapsto NoSig] \\
& \land\ remote\_sigs = [p \in Party \mapsto NoSig]
\end{aligned}
$$

We don't define transactions using a function because using variables as functions become hard to work with in TLA+

$$
\begin{aligned}
TypeInvariant \triangleq\ & \land\ outputs \in [Party \to Seq(AllOutput)] \\
& \land\ local\_sigs \in [Party \to Sig \cup \{NoSig\}] \\
& \land\ remote\_sigs \in [Party \to Sig \cup \{NoSig\}]
\end{aligned}
$$

Helper function to get other party

$OtherParty(party) \triangleq \text{CHOOSE } p \in Party : p \neq party$

Create first commitment transactions for given parties

$$
\begin{aligned}
CreateFirstCommitmentTx(party) \triangleq\ & \land\ outputs[party] = \langle\rangle \\
& \land\ outputs' = [outputs \text{ EXCEPT } ![party] = \\
& \qquad @ \circ \langle\langle party, OtherParty(party), CSV\rangle, \\
& \qquad\quad \langle OtherParty(party), 0\rangle\rangle] \\
& \land\ local\_sigs' = [local\_sigs \text{ EXCEPT } ![party] = \langle party, 0\rangle] \\
& \land\ remote\_sigs' = [remote\_sigs \text{ EXCEPT } ![party] = \langle OtherParty(party), 0\rangle]
\end{aligned}
$$

Party p spends their commitment transaction.

If the tx is the latest commitment transaction it is succesfuly spend.

If not, ti gives the other party a chance to spend the breach remedy tx.

$SpendCommitmentTx(p,\ tx)$

$$Next \triangleq$$
$$\quad \lor \exists\, p \in Party : CreateFirstCommitmentTx(p)$$

$$Spec \triangleq Init \land \Box[Next]_{\langle vars \rangle}$$