This is a formal specification for the *Swap* protocol used in Blink.

There are two participants in a swap. The participant that initiates the protocol is called the *proposer* The second participant is called the ‘partner‘.

MODULE *Swap*

internal state enum for each participant

$\text{VARIABLES } proposer\_state,\ partner\_state$

$participant\_states \triangleq \langle proposer\_state,\ partner\_state \rangle$

content of the most recent DM between parties

$\text{VARIABLES } dm$

status of onchain escrow transaction for each participant

$\text{VARIABLES } proposer\_escrow,\ partner\_escrow$

whether or not the escrow timelocks have matured

$\text{VARIABLES } proposer\_timelock\_mature,\ partner\_timelock\_mature$

collections of vars for easier UNCHANGED *assertions*

$escrows \triangleq \langle proposer\_escrow,\ partner\_escrow \rangle$
$timelocks \triangleq \langle proposer\_timelock\_mature,\ partner\_timelock\_mature \rangle$

$vars \triangleq \langle proposer\_state,\ partner\_state,\ dm,\ escrows,\ timelocks \rangle$

$$\begin{aligned} swap\_states \triangleq \{ & \text{“init”},\ \text{“proposed”},\ \text{“offered”},\ \text{“bootstrapped”},\ \text{“pendinglock”}, \\ & \text{“deposited”},\ \text{“preimagerevealed”},\ \text{“seckeyrevealed”},\ \text{“closable”},\ \text{“closed”}, \\ & \text{“closedtimelock”},\ \text{“cancelled”} \} \end{aligned}$$

$$\begin{aligned} TypeInvariant \triangleq\ & \land proposer\_state \in swap\_states \\ & \land partner\_state \in swap\_states \end{aligned}$$

$$\begin{aligned} Init \triangleq\ & \land proposer\_state = \text{“init”} \\ & \land partner\_state = \text{“init”} \\ & \land dm = \text{“”} \\ & \land proposer\_escrow = \text{“”} \\ & \land partner\_escrow = \text{“”} \\ & \land proposer\_timelock\_mature = \text{FALSE} \\ & \land partner\_timelock\_mature = \text{FALSE} \end{aligned}$$

$*$ *Convenience Definitions*

$$\begin{aligned} TimelocksOk \triangleq\ & \land proposer\_timelock\_mature = \text{FALSE} \\ & \land partner\_timelock\_mature = \text{FALSE} \end{aligned}$$

$ProposerRefund \triangleq proposer\_escrow = \text{“confirmed\_refund”}$
$PartnerRefund \triangleq partner\_escrow = \text{“confirmed\_refund”}$

note that each participant spends the OTHER *escrow in the happy cases*

$$\begin{array}{l}
\mathit{ProposerPaid} \triangleq \mathit{partner\_escrow} = \text{“confirmed\_spend”} \\
\mathit{PartnerPaid} \triangleq \mathit{proposer\_escrow} = \text{“confirmed\_spend”}
\end{array}$$

$$\mathit{TerminalStates} \triangleq \{\text{“closedsuccess”}, \text{“closedhashlock”}, \text{“closedtimelock”}\}$$

$*$ *Happy Path*

$$\mathit{ProposeSwap} \triangleq \begin{array}[t]{l}
\land \mathit{proposer\_state} = \text{“init”} \\
\land \mathit{proposer\_state}' = \text{“proposed”} \\
\land \text{UNCHANGED } \langle \mathit{partner\_state}, \mathit{dm}, \mathit{escrows}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{OfferSwap} \triangleq \begin{array}[t]{l}
\land \mathit{proposer\_state} = \text{“proposed”} \\
\land \mathit{partner\_state} = \text{“init”} \\
\land \mathit{partner\_state}' = \text{“offered”} \\
\land \mathit{dm}' = \text{“partner\_setup”} \\
\land \text{UNCHANGED } \langle \mathit{proposer\_state}, \mathit{escrows}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{RespondToOffer} \triangleq \begin{array}[t]{l}
\land \mathit{proposer\_state} = \text{“proposed”} \\
\land \mathit{dm} = \text{“partner\_setup”} \\
\land \mathit{partner\_state} = \text{“offered”} \\
\land \begin{array}[t]{l}
\lor \begin{array}[t]{l} \land \mathit{proposer\_state}' = \text{“bootstrapped”} \\ \land \mathit{dm}' = \text{“proposer\_setup”} \end{array} \\
\lor \begin{array}[t]{l} \land \mathit{proposer\_state}' = \text{“cancelled”} \\ \land \mathit{dm}' = \text{“cancel\_swap”} \end{array}
\end{array} \\
\land \text{UNCHANGED } \langle \mathit{partner\_state}, \mathit{escrows}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{PartnerBootstrap} \triangleq \begin{array}[t]{l}
\land \mathit{dm} = \text{“proposer\_setup”} \\
\land \mathit{partner\_state} = \text{“offered”} \\
\land \mathit{partner\_state}' = \text{“bootstrapped”} \\
\land \text{UNCHANGED } \langle \mathit{proposer\_state}, \mathit{dm}, \mathit{escrows}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{PartnerConfirmAddress} \triangleq \begin{array}[t]{l}
\land \mathit{partner\_state} = \text{“bootstrapped”} \\
\land \mathit{partner\_state}' = \text{“pendinglock”} \\
\land \mathit{dm}' = \text{“partner\_address”} \\
\land \text{UNCHANGED } \langle \mathit{proposer\_state}, \mathit{escrows}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{ProposerConfirmAddress} \triangleq \begin{array}[t]{l}
\land \mathit{proposer\_state} = \text{“bootstrapped”} \\
\land \mathit{dm} = \text{“partner\_address”} \\
\land \mathit{proposer\_state}' = \text{“pendinglock”} \\
\land \mathit{dm}' = \text{“proposer\_address”} \\
\land \text{UNCHANGED } \langle \mathit{partner\_state}, \mathit{escrows}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{PartnerDeposit} \triangleq \begin{array}[t]{l}
\land \mathit{partner\_state} = \text{“pendinglock”} \\
\land \mathit{dm} = \text{“proposer\_address”} \\
\land \mathit{partner\_escrow}' = \text{“pending\_deposit”} \\
\land \mathit{partner\_state}' = \text{“deposited”} \\
\land \text{UNCHANGED } \langle \mathit{proposer\_state}, \mathit{dm}, \mathit{proposer\_escrow}, \mathit{timelocks} \rangle
\end{array}$$

$$
\begin{aligned}
\mathit{ProposerDeposit} \triangleq\ & \land \mathit{proposer\_state}\ = \text{“pendinglock”}\\
& \land \mathit{partner\_escrow} = \text{“confirmed\_deposit”}\\
& \land \mathit{proposer\_state}' = \text{“deposited”}\\
& \land \mathit{proposer\_escrow}' = \text{“pending\_deposit”}\\
& \land \textsc{unchanged}\ \langle \mathit{partner\_state},\ \mathit{dm},\ \mathit{partner\_escrow},\ \mathit{timelocks} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathit{RevealPreimage} \triangleq\ & \land \mathit{partner\_state} = \text{“deposited”}\\
& \land \mathit{proposer\_escrow} = \text{“confirmed\_deposit”}\\
& \land \mathit{partner\_escrow} = \text{“confirmed\_deposit”}\\
& \land \mathit{TimelocksOk}\\
& \land \mathit{dm}' = \text{“preimage”}\\
& \land \mathit{partner\_state}' = \text{“preimagerevealed”}\\
& \land \textsc{unchanged}\ \langle \mathit{proposer\_state},\ \mathit{escrows},\ \mathit{timelocks} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathit{ReceivePreimage} \triangleq\ & \land \mathit{dm} = \text{“preimage”}\\
& \land \mathit{proposer\_state} \notin \{\text{“closable”},\ \text{“closed”},\ \text{“closedtimelock”}\}\\
& \land \mathit{proposer\_state}' = \text{“preimagerevealed”}\\
& \land \textsc{unchanged}\ \langle \mathit{partner\_state},\ \mathit{escrows},\ \mathit{dm},\ \mathit{timelocks} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathit{SendProposerSeckey} \triangleq\ & \land \mathit{proposer\_state} = \text{“preimagerevealed”}\\
& \land \mathit{proposer\_escrow} = \text{“confirmed\_deposit”}\\
& \land \mathit{partner\_escrow} = \text{“confirmed\_deposit”}\\
& \land \mathit{TimelocksOk}\\
& \land \mathit{dm}' = \text{“proposer\_seckey”}\\
& \land \mathit{proposer\_state}' = \text{“seckeyrevealed”}\\
& \land \textsc{unchanged}\ \ \langle \mathit{partner\_state},\ \mathit{escrows},\ \mathit{timelocks} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathit{ReceiveProposerSecKey} \triangleq\ & \land \mathit{dm} = \text{“proposer\_seckey”}\\
& \land \mathit{proposer\_escrow} = \text{“confirmed\_deposit”}\\
& \land \mathit{partner\_escrow} = \text{“confirmed\_deposit”}\\
& \land \mathit{TimelocksOk}\\
& \land \mathit{partner\_state}' = \text{“seckeyrevealed”}\\
& \land \mathit{dm}' = \text{“partner\_seckey”}\\
& \land \textsc{unchanged}\ \ \langle \mathit{proposer\_state},\ \mathit{escrows},\ \mathit{timelocks} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathit{ReceivePartnerSecKey} \triangleq\ & \land \mathit{dm} = \text{“partner\_seckey”}\\
& \land \mathit{partner\_escrow} = \text{“confirmed\_deposit”}\\
& \land \mathit{TimelocksOk}\\
& \land \mathit{proposer\_state}' = \text{“closable”}\\
& \land \textsc{unchanged}\ \langle \mathit{partner\_state},\ \mathit{dm},\ \mathit{escrows},\ \mathit{timelocks} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathit{ProtocolAction} \triangleq\ & \lor \mathit{ProposeSwap}\\
& \lor \mathit{OfferSwap}\\
& \lor \mathit{RespondToOffer}\\
& \lor \mathit{PartnerBootstrap}\\
& \lor \mathit{PartnerConfirmAddress}\\
& \lor \mathit{ProposerConfirmAddress}
\end{aligned}
$$

$$\begin{array}{l}
\lor \mathit{PartnerDeposit} \\
\lor \mathit{ProposerDeposit} \\
\lor \mathit{RevealPreimage} \\
\lor \mathit{ReceivePreimage} \\
\lor \mathit{SendProposerSeckey} \\
\lor \mathit{ReceiveProposerSecKey} \\
\lor \mathit{ReceivePartnerSecKey}
\end{array}$$

∗ *Spending from escrows*

Anytime after funds have been deposited, we assume that the protocol can stall for a while and then either participant can take the refund of their escrow.

$$\mathit{ProposerSpendRefund} \triangleq \begin{array}{l}
\land \mathit{proposer\_escrow} = \text{“}\mathsf{confirmed\_deposit}\text{”} \\
\land \mathit{proposer\_timelock\_mature} = \text{TRUE} \\
\land \mathit{proposer\_escrow}' = \text{“}\mathsf{pending\_refund}\text{”} \\
\land \text{UNCHANGED } \langle \mathit{participant\_states}, \mathit{dm}, \mathit{partner\_escrow}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{ProposerReceiveRefund} \triangleq \begin{array}{l}
\land \mathit{proposer\_escrow} = \text{“}\mathsf{confirmed\_refund}\text{”} \\
\land \mathit{proposer\_state}' \quad = \text{“}\mathsf{closedtimelock}\text{”} \\
\land \text{UNCHANGED } \langle \mathit{partner\_state}, \mathit{dm}, \mathit{escrows}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{PartnerSpendRefund} \triangleq \begin{array}{l}
\land \mathit{partner\_escrow} = \text{“}\mathsf{confirmed\_deposit}\text{”} \\
\land \mathit{partner\_timelock\_mature} = \text{TRUE} \\
\land \mathit{partner\_escrow}' = \text{“}\mathsf{pending\_refund}\text{”} \\
\land \text{UNCHANGED } \langle \mathit{participant\_states}, \mathit{dm}, \mathit{proposer\_escrow}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{PartnerReceiveRefund} \triangleq \begin{array}{l}
\land \mathit{partner\_escrow} = \text{“}\mathsf{confirmed\_refund}\text{”} \\
\land \mathit{partner\_state}' \quad = \text{“}\mathsf{closedtimelock}\text{”} \\
\land \text{UNCHANGED } \langle \mathit{proposer\_state}, \mathit{dm}, \mathit{escrows}, \mathit{timelocks} \rangle
\end{array}$$

$$\mathit{RefundAction} \triangleq \begin{array}{l}
\lor \mathit{ProposerSpendRefund} \\
\lor \mathit{ProposerReceiveRefund} \\
\lor \mathit{PartnerSpendRefund} \\
\lor \mathit{PartnerReceiveRefund}
\end{array}$$

Once a participant knows the hash preimage, they can spend via the hashlock
It s better for privacy to wait until they can do a keyspend, but its possible and makes sure that everyone gets paid if the protocol stops there.
the Partner starts off with the *preimage*, so they can spend as soon as funds are locked.

$$\mathit{PartnerSpendHashlock} \triangleq \begin{array}{l}
\land \mathit{proposer\_escrow} = \text{“}\mathsf{confirmed\_deposit}\text{”} \\
\land \mathit{TimelocksOk} \\
\quad \text{The partner spends the proposer escrow} \\
\land \mathit{proposer\_escrow}' = \text{“}\mathsf{pending\_spend}\text{”} \\
\quad \mathit{TODO}\text{: change state for partner?} \\
\land \text{UNCHANGED } \langle \mathit{partner\_escrow}, \mathit{participant\_states}, \mathit{timelocks}, \mathit{dm} \rangle
\end{array}$$

$$
\mathit{ProposerObservesPreimageOnchain} \triangleq \begin{array}{l}
\land \mathit{proposer\_escrow} = \text{“confirmed\_spend”} \\
\land \mathit{proposer\_state} \neq \text{“closable”} \\
\land \mathit{proposer\_state}' = \text{“preimagerevealed”} \\
\land \textsc{unchanged}\ \langle \mathit{partner\_state},\ \mathit{escrows},\ \mathit{timelocks},\ \mathit{dm} \rangle
\end{array}
$$

Proposer can spend from the hashlock as soon as the *preimage* is revealed, either
through the protocol or because they saw the Partner spend with it onchain.
The Proposer spends the partner escrow.

$$
\mathit{ProposerSpendHashLock} \triangleq \begin{array}{l}
\land \mathit{partner\_escrow} = \text{“confirmed\_deposit”} \\
\land \mathit{TimelocksOk} \\
\land \mathit{proposer\_state} = \text{“preimagerevealed”} \\
\land \mathit{partner\_escrow}' = \text{“pending\_spend”} \\
\land \textsc{unchanged}\ \langle \mathit{proposer\_escrow},\ \mathit{participant\_states},\ \mathit{timelocks},\ \mathit{dm} \rangle
\end{array}
$$

$$
\mathit{HashlockAction} \triangleq \mathit{PartnerSpendHashlock} \lor \mathit{ProposerObservesPreimageOnchain} \lor \mathit{ProposerSpendHashLock}
$$

The best case is where the participants spend via the keypath.

$$
\mathit{PartnerSpendKeypath} \triangleq \begin{array}{l}
\land \mathit{proposer\_escrow} = \text{“confirmed\_deposit”} \\
\land \mathit{TimelocksOk} \\
\land \mathit{partner\_state} = \text{“seckeyrevealed”} \\
\land \mathit{proposer\_escrow}' = \text{“pending\_spend”} \\
\land \textsc{unchanged}\ \langle \mathit{partner\_escrow},\ \mathit{participant\_states},\ \mathit{timelocks},\ \mathit{dm} \rangle
\end{array}
$$

$$
\mathit{ProposerSpendKeypath} \triangleq \begin{array}{l}
\land \mathit{partner\_escrow} = \text{“confirmed\_deposit”} \\
\land \mathit{TimelocksOk} \\
\land \mathit{proposer\_state} = \text{“closable”} \\
\land \mathit{partner\_escrow}' = \text{“pending\_spend”} \\
\land \textsc{unchanged}\ \langle \mathit{proposer\_escrow},\ \mathit{participant\_states},\ \mathit{timelocks},\ \mathit{dm} \rangle
\end{array}
$$

$$
\mathit{KeypathSpendAction} \triangleq \mathit{PartnerSpendKeypath} \lor \mathit{ProposerSpendKeypath}
$$

Partner spends the proposer escrow

$$
\mathit{PartnerFinished} \triangleq \begin{array}{l}
\land \mathit{proposer\_escrow} = \text{“confirmed\_spend”} \\
\land \mathit{partner\_state}' = \text{“closed”} \\
\land \textsc{unchanged}\ \langle \mathit{proposer\_state},\ \mathit{escrows},\ \mathit{timelocks},\ \mathit{dm} \rangle
\end{array}
$$

Proposer spends the proposer escrow

$$
\mathit{ProposerFinished} \triangleq \begin{array}{l}
\land \mathit{partner\_escrow} = \text{“confirmed\_spend”} \\
\land \mathit{proposer\_state}' = \text{“closed”} \\
\land \textsc{unchanged}\ \langle \mathit{partner\_state},\ \mathit{escrows},\ \mathit{timelocks},\ \mathit{dm} \rangle
\end{array}
$$

$$
\mathit{TerminalAction} \triangleq \mathit{PartnerFinished} \lor \mathit{ProposerFinished}
$$

* Cancellation

$$
\mathit{PartnerCancel} \triangleq \begin{array}{l}
\land \mathit{dm} = \text{“cancel\_swap”} \\
\land \mathit{partner\_state} = \text{“offered”} \\
\land \mathit{partner\_state}' = \text{“cancelled”}
\end{array}
$$

$$\land \textsc{unchanged}\ \langle proposer\_state,\ dm,\ escrows,\ timelocks \rangle$$

* *Blockchain* advancing

$$
BlockConfirmation \triangleq
\begin{array}[t]{l}
\lor \land \begin{array}[t]{l}
\lor \begin{array}[t]{l}\land partner\_escrow = \text{“pending\_deposit”} \\ \land partner\_escrow' = \text{“confirmed\_deposit”}\end{array} \\
\lor \begin{array}[t]{l}\land partner\_escrow = \text{“pending\_refund”} \\ \land partner\_escrow' = \text{“confirmed\_refund”}\end{array} \\
\lor \begin{array}[t]{l}\land partner\_escrow = \text{“pending\_spend”} \\ \land partner\_escrow' = \text{“confirmed\_spend”}\end{array}
\end{array} \\
\quad\ \ \land \textsc{unchanged}\ \langle participant\_states,\ dm,\ proposer\_escrow,\ timelocks \rangle \\
\lor \land \begin{array}[t]{l}
\lor \begin{array}[t]{l}\land proposer\_escrow = \text{“pending\_deposit”} \\ \land proposer\_escrow' = \text{“confirmed\_deposit”}\end{array} \\
\lor \begin{array}[t]{l}\land proposer\_escrow = \text{“pending\_refund”} \\ \land proposer\_escrow' = \text{“confirmed\_refund”}\end{array} \\
\lor \begin{array}[t]{l}\land proposer\_escrow = \text{“pending\_spend”} \\ \land proposer\_escrow' = \text{“confirmed\_spend”}\end{array}
\end{array} \\
\quad\ \ \land \textsc{unchanged}\ \langle participant\_states,\ dm,\ partner\_escrow,\ timelocks \rangle
\end{array}
$$

Some amount of time after an escrow has been confirmed, the timelock
will mature.
It is an important safety property that the proposer timelock
matures first.

$$
ProposerTimelockMature \triangleq
\begin{array}[t]{l}
\land proposer\_timelock\_mature = \text{FALSE} \\
\land partner\_timelock\_mature = \text{FALSE} \\
\land proposer\_escrow = \text{“confirmed\_deposit”} \\
\land proposer\_timelock\_mature' = \text{TRUE} \\
\land \textsc{unchanged}\ \langle participant\_states,\ dm,\ escrows,\ partner\_timelock\_mature \rangle
\end{array}
$$

$$
PartnerTimelockMature \triangleq
\begin{array}[t]{l}
\land proposer\_timelock\_mature = \text{TRUE} \\
\land partner\_escrow = \text{“confirmed\_deposit”} \\
\land partner\_timelock\_mature' = \text{TRUE} \\
\land \textsc{unchanged}\ \langle participant\_states,\ dm,\ escrows,\ proposer\_timelock\_mature \rangle
\end{array}
$$

$$TimelockMaturation \triangleq ProposerTimelockMature \lor PartnerTimelockMature$$

* Invariants and Temporal Properties

At the time when both parties have deposited to their *escrows*, the partner has
the hashlock *preimage*, but the proposer does not. The partner could wait until
right before their refund timelock matures and then take the proposer escrow via
hashlock and take their own refund via the timelock. Therefor it is critically
important that the proposer timelock matures FIRST, so that they can get their money
back if the partner is holding back the *preimage*

$$ProposerGetsRefundFirst \triangleq partner\_timelock\_mature \Rightarrow proposer\_timelock\_mature$$

If one participant gets their refund, they do not also get to spend the other escrow.
A refund of one participant always leads to a refund of the other.

A hashlock or keypath spend of one participant always leads to a spend of the other.

In other words, they can't steal money from the other participant.

$$\begin{aligned} \mathit{NobodyGetsBothEscrows} \triangleq\ & \lor\ \mathit{ProposerRefund} \leadsto \mathit{PartnerRefund} \\ & \lor\ \mathit{PartnerRefund} \leadsto \mathit{ProposerRefund} \\ & \lor\ \mathit{ProposerPaid} \leadsto \mathit{PartnerPaid} \\ & \lor\ \mathit{PartnerPaid} \leadsto \mathit{ProposerPaid} \end{aligned}$$

Once funds have been deposited, they eventually get paid out.

Once an escrow has been paid out, it doesn't get re-spent

$$\begin{aligned} \mathit{EscrowPaymentTerminal} \triangleq\ & \land\ \mathit{proposer\_state} = \text{“}\textsf{deposited}\text{”} \leadsto ((\Diamond\Box \mathit{PartnerPaid}) \lor (\Diamond\Box \mathit{ProposerRefund})) \\ & \land\ \mathit{partner\_state} = \text{“}\textsf{deposited}\text{”} \leadsto ((\Diamond\Box \mathit{ProposerPaid}) \lor (\Diamond\Box \mathit{PartnerRefund})) \end{aligned}$$

$$\begin{aligned} \mathit{Next} \triangleq\ & \lor\ \mathit{BlockConfirmation} \\ & \lor\ \mathit{ProtocolAction} \\ & \lor\ \mathit{RefundAction} \\ & \lor\ \mathit{TimelockMaturation} \\ & \lor\ \mathit{HashlockAction} \\ & \lor\ \mathit{KeypathSpendAction} \\ & \lor\ \mathit{TerminalAction} \end{aligned}$$

$$\mathit{Spec} \triangleq \mathit{Init} \land \Box[\mathit{Next}]_{\mathit{vars}} \land \mathrm{WF}_{\mathit{vars}}(\mathit{Next})$$

Assumptions that we make in this spec:

1. Participants try to make forward progress for themselves. We don't model the case where someone just leaves their money behind.

2. Transactions submitted to the *Bitcoin* are eventually mined. We assume that participant software will rebroadcast purged transactions.

3. We assume that participants can get their transactions into the next block through fee selection or fee bumping

4. We assume that both participants watch the chain and see when transactions happen.