Atomic swaps

Lorenzo Tucci

March 17, 2024

Parties U_0 and U_1 hold assets a on blockchain $\mathbb{A}$ and assets b on chain $\mathbb{B}$ respectively.
We define with $\mathsf{swp}(a)$ and $\mathsf{swp}(b)$ the amount of the assets the parties agreed to swap before starting the protocol.

We define the following oracles to interact with the blockchains.

- $\mathsf{PubTx}(\sigma_{tx}, tx, \mathbb{A})$ publish the transaction tx with signature σ_{tx} on chain $\mathbb{A}$
- $\mathsf{InitTx}(pk_{tx}, pk_{rx}, amnt, \mathbb{A})$ create an unsigned transaction paying $amnt$ from pk_{tx} to pk_{rx} on chain $\mathbb{A}$
- $\mathsf{WatchTx}(tx, \mathbb{A})$ wait for the transaction tx to be confirmed on chain $\mathbb{A}$
- $\mathsf{GetBal}(pk, \mathbb{A})$ get the balance of assets held by pk
- $\mathsf{GetSig}(pk, \mathbb{A})$ get the signature σ_{tx} of the latest transaction in pk's record on chain $\mathbb{A}$

U_1 starts counting the timeout from the moment they send the VTD commitment to U_0, and respectively U_0 starts counting down from the moment they receive it.

If the timeout expires before the protocol is completed:

- U_1 will transfer the coins from $pk(1)$ to another wallet on $\mathbb{B}$. From this moment on, if U_0 tries to $\mathsf{PubTx}(\sigma_{\mathsf{swp}(10)}, tx_{\mathsf{swp}}, \mathbb{A})$, the transaction will get rejected.

- U_0 will wait until $\Pi_{\mathsf{VTD}}.\mathsf{ForceOp}(C)$ is completed to get the other secret key share $sk_1(01)$ of $pk(10)$ to retrieve $sk(10)$ and transfer back assets a to $pk(0)$.

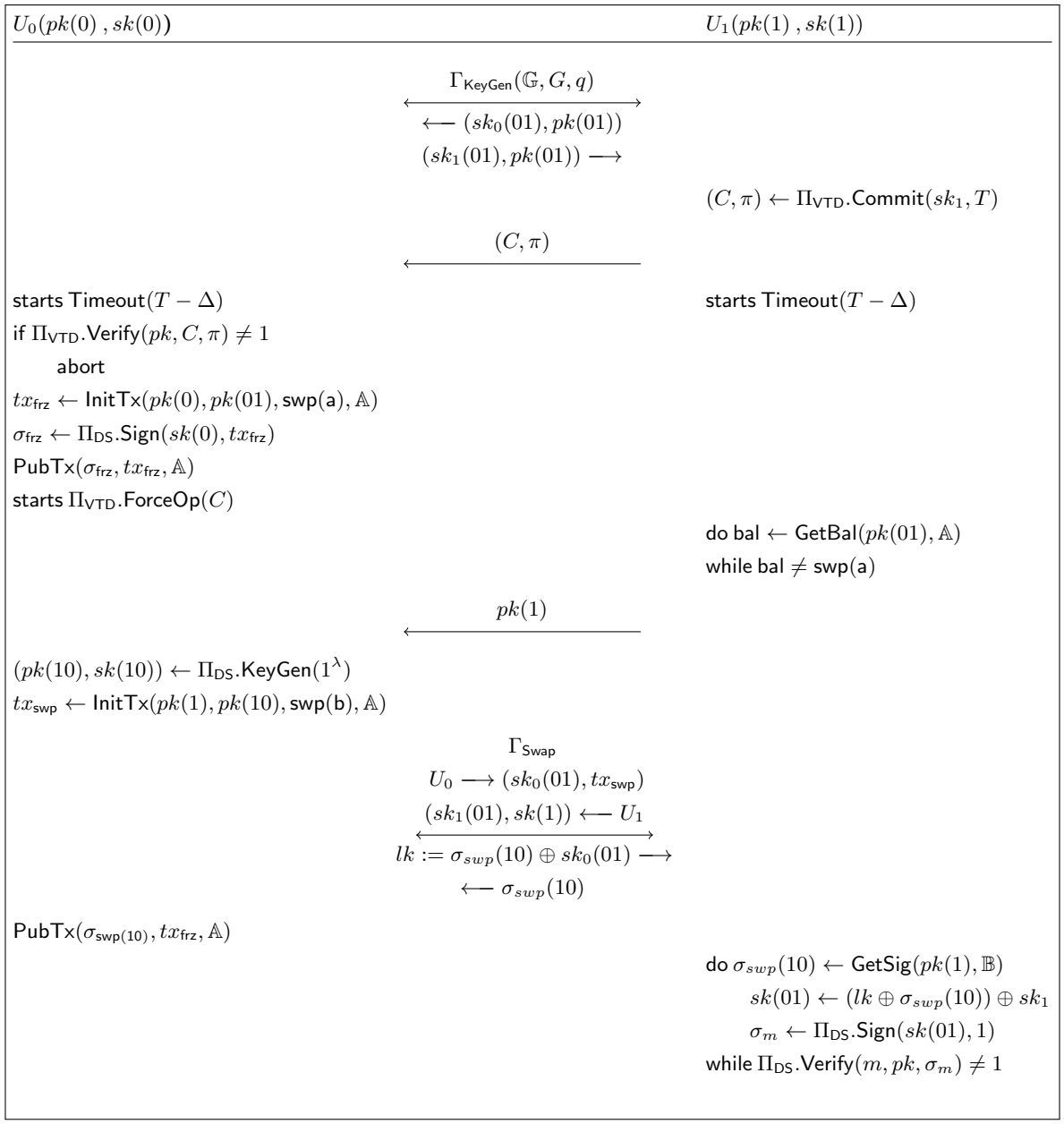

Figure 1: Protocol execution for a successful swap

Global input $(\mathbb{G}, G, q, T, \mathsf{swp}(a), \mathsf{swp}(b), \mathbb{A}, \mathbb{B})$

$(sk_0(01), pk(01)) \leftarrow$ **wait** $\Gamma_{\mathsf{KeyGen}}(\mathbb{G}, G, q)$
$(C, \pi) \leftarrow$ **wait** $\mathsf{receive}(U_1)$
if $\Pi_{\mathsf{VTD}}.\mathsf{Verify}(pk, C, \pi) \neq 1$
  return $\perp$
$\mathsf{res} \leftarrow$ **select** {
  wait {
    $\Pi_{\mathsf{VTD}}.\mathsf{ForceOp}(C)$
  }
  wait {
    $(pk(10), sk(10)) \leftarrow \Pi_{\mathsf{DS}}.\mathsf{KeyGen}(1^\lambda)$
    $tx_{\mathsf{frz}} \leftarrow$ **wait** $\mathsf{InitTx}(pk(0), pk(01), \mathsf{swp(a)}, \mathbb{A})$
    $\sigma_{\mathsf{frz}} \leftarrow \Pi_{\mathsf{DS}}.\mathsf{Sign}(sk(0), tx_{\mathsf{frz}})$
    wait $\mathsf{PubTx}(\sigma_{\mathsf{frz}}, tx_{\mathsf{frz}}, \mathbb{A})$
    $pk(1) \leftarrow$ **wait** $\mathsf{receive}(U_1)$
    $tx_{\mathsf{swp}} \leftarrow$ **wait** $\mathsf{InitTx}(pk(1), pk(10), \mathsf{swp(b)}, \mathbb{B})$
    $\sigma_{swp}(10) \leftarrow$ **wait** $\Gamma_{\mathsf{Swap}}(sk_0(01), tx_{swp})$
    wait $\mathsf{PubTx}(\sigma_{\mathsf{swp}(10)}, tx_{swp}, \mathbb{B})$
  }
}
if $\mathsf{res} \neq 1$
  $sk(01) := sk_0(01) \oplus \mathsf{res}$
  $tx_{\mathsf{rfnd}} \leftarrow$ **wait** $\mathsf{InitTx}(pk(01), pk(0), \mathsf{swp(a)}, \mathbb{A})$
  $\sigma_{\mathsf{rfnd}} \leftarrow \Pi_{\mathsf{DS}}.\mathsf{Sign}(sk(01), tx_{\mathsf{rfnd}})$
  wait $\mathsf{PubTx}(\sigma_{\mathsf{rfnd}}, tx_{rfnd}, \mathbb{A})$

Global input $(\mathbb{G}, G, q, T, \mathsf{swp}(a), \mathsf{swp}(b), \mathbb{A}, \mathbb{B})$

$(sk_1(01), pk(01)) \leftarrow$ **wait** $\Gamma_{\mathsf{KeyGen}}(\mathbb{G}, G, q)$
$(C, \pi) \leftarrow \Pi_{\mathsf{VTD}}.\mathsf{Commit}(sk_1, T)$
$\mathsf{send}(U_1,\ (C, \pi))$
$\mathsf{res} \leftarrow$ **select** {
  wait {
    $\mathsf{timeout}(T/2)$
  }
  wait {
    do $\mathsf{bal} \leftarrow$ **wait** $\mathsf{GetBal}(pk(01), \mathbb{A})$
    while $\mathsf{bal} \neq \mathsf{swp}(a)$
    $\mathsf{send}(U_1,\ \mathsf{pk}(1))$
    $\sigma_{swp}(10) \leftarrow$ **wait** $\Gamma_{\mathsf{Swap}}(sk_1(01), sk(1))$
    do $\sigma_{\mathsf{swp}}(10) \leftarrow$ **wait** $\mathsf{GetSig}(pk(1), \mathbb{B})$
      $sk(01) \leftarrow (lk \oplus \sigma_{swp}(10)) \oplus sk_1$
      $\sigma_b \leftarrow \Pi_{\mathsf{DS}}.\mathsf{Sign}(sk(01), 1)$
    while $\Pi_{\mathsf{DS}}.\mathsf{Verify}(m, pk, \sigma_b) \neq 1$
    $tx_{\mathsf{swp}} \leftarrow$ **wait** $\mathsf{InitTx}(pk(01), pk_{\mathbb{A}}(1), \mathsf{swp(a)}, \mathbb{A})$
    $\sigma_{\mathsf{swp}} \leftarrow \Pi_{\mathsf{DS}}.\mathsf{Sign}(sk(01), tx_{\mathsf{rfnd}})$
    wait $\mathsf{PubTx}(\sigma_{\mathsf{swp}}, tx_{swp}, \mathbb{A})$
  }
}
if $\mathsf{res} \neq 1$
  $(pk(11), sk(11)) \leftarrow \Pi_{\mathsf{DS}}.\mathsf{KeyGen}(1^\lambda)$
  $tx_{\mathsf{rfnd}} \leftarrow$ **wait** $\mathsf{InitTx}(pk(1), pk(11), \mathsf{swp(b)}, \mathbb{B})$
  $\sigma_{\mathsf{rfnd}} \leftarrow \Pi_{\mathsf{DS}}.\mathsf{Sign}(sk(1), tx_{\mathsf{rfnd}})$
  wait $\mathsf{PubTx}(\sigma_{\mathsf{rfnd}}, tx_{\mathsf{rfnd}}, \mathbb{B})$

Figure 2: Full protocol execution for U_0 and U_1, respectively left and right (alternative syntax)

$U_0(pk(0)\,, sk(0))$	$U_1(pk(1)\,, sk(1))$
$sk(01) := sk_0(01) \oplus sk_1(01)$	
$\sigma_{swp}(10) \leftarrow \Pi_{\mathsf{DS}}.\mathsf{Sign}(sk(1), tx_{\mathsf{swp}})$	
$lk := \sigma_{swp}(10) \oplus sk_0(01)$	

Figure 3: Protocol definition of 2PC Γ_{Swap}