Zero-Knowledge Proof Trustless P2P Fiat-to-Crypto On-Ramp for Cardano: a progress report

ZkFold

March 21, 2025

Trustless P2P on-ramp is a smart contract app that connects cryptocurrency sellers and buyers. A seller deposits funds into the smart contract and ask for a fiat payment. Once the payment is delivered, the buyer can generate a cryptographic proof and unlock the funds from the smart contract.

1 Smart contract

The *onRamp* smart contract is parametrized by:

```
data OnRampParams = OnRampParams
  { feeAddress       :: Address
  , feeValue         :: Value
  , fiatPubKeyBytes  :: BuiltinByteString
  }
```

providing the fee address and value, as well as the public key of the fiat witness certifying the trade's fiat payment.

Each sell offer is sent to a UTxO containing the datum:

```
data OnRampDatum = OnRampDatum
    { paymentInfoHash   :: BuiltinByteString
    , sellPriceUsd      :: Integer
    , valueSold         :: Value
    , sellerPubKeyBytes :: BuiltinByteString
    , buyerPubKeyHash   :: Maybe PubKeyHash
    , timelock          :: Maybe POSIXTime
    }
```

providing the hash of the payment information, the sell price (in USD), the value offered, the seller's public key, a place holder for the buyer's pub-key hash, and a place holder for a time lock serving as deadline after which the sell order can be closed if no fiat payment was received. Since at the moment of posting the sell offer the buyer is unknown, initial buyer's pub-key hash field is set to `Nothing`.

During the P2P trade sequence, there are are three actions that the *onRamp* smart contract deals with:

```
data OnRampRedeemer
  = Update BuiltinByteString  -- Signed updated datum by seller
  -- ^ Update the bid with the buyer's public key hash and the timelock.
  | Claim BuiltinByteString
  -- ^ Buyer claims the value.
  | Cancel
  -- ^ Seller cancels the value selling.
```

1. The `Update` redeemer provides the seller's signature certifying the updated datum with agreed upon buyer's pub-key hash, as well as a deadline for sending out the fiat payment. The UTxO with the sell offer is updated with this datum.

2. The `Claim` redeemer provides the fiat witness' signature certifying fiat payment.

3. The `Cancel` redeemer allows the seller to cancel the operation after the deadline is reached.

The smart contract performs following checks (depending on the redeemer):

1.1 Update

- The new UTxO with the updated datum contains the original value and does not leave the *onRamp* script.

- The payment info, the sell price, and the seller's public key on the datum have not been altered.

- A buyer's pub-key-hash has been provided.

- A deadline time-lock in the future has been provided.

- The updated datum has been signed by the seller.

1.2 Claim

- The value is sent to the address corresponding to the buyer's pub-key hash registered on the datum of the UTxO being consumed.
- The value sent to buyer corresponds to the whole value of the consumed UTxO.
- Cryptographic proof of fiat payment is provided.
- The correct fee is sent ot the fee address.

In the current version, cryptographic proof of fiat payment is represented by the fiat witness having signed the payment info (hash of). In future versions this can be replaced with a ZK proof of the fiat payment.

1.3 Cancel

- The time-lock deadline for the UTxO (sell offer) being canceled is in the past.
- The value is sent to the original seller's address.

2 Off-chain code (backend)

The main logic of the off-chain code is written in Haskell and resides on the `backends` directory. Executables are named after the subdirectories within.

It is assumed that the private and public keys of the fee address wallet, the seller's wallets, and the buyer's wallet, have all been set-up. For simplicity, it is assumed that files containing private and public keys are of the form *name*`.skey` and *name*`.vkey`, respectively, where *name* is a string identifying the wallet (e.g. 'alice', 'bob', etc.)

Module `src/ZkFold/Cardano/P2P/Example.hs` contains our working example for a function mocking the payment information simply as the hash of a string (wallet's name). It also contains a working example of a "choice value" function needed to have a notion of "best" sell-offer.

2.1 `p2p-init-transaction`

Takes two arguments: a) name of wallet receiving fees, and b) fee address. For simplicity, it is assumed that the public key of the fiat witness corresponds to the fees wallet.

Executable serializes the parameterized plutus smart contract. It also writes the cbor for the Cancel redeemer.

2.2 `p2p-add-seller`

Takes four arguments: a) name of seller's wallet, b) sell price, c) lovelace sold, and d) deadline.

Executable constructs and writes the initial datum for the sell order.

2.3 `p2p-choose-offer`

Takes as argument the string representation of the UTxO's at the *onRamp* script. This string is generated with `cardano-cli` with:

```
onRampUtxos=$(cardano-cli conway query utxo --address $onRampAddr --testnet-magic $mN --out-file /dev/stdout |
                 jq -c 'to_entries')
```

(Here shell variables `onRampAddr` and `mN` store the script's address and the chain's "magic number", respectively.)

Executable parses `onRampUtxos`, queries the UTxOs at *onRamp* script, and chooses a "best" sell-offer.

2.4 `p2p-buy-order`

Takes four arguments: a) name of buyer's wallet, b) buyer's address, c) name of seller's wallet, and d) string representation of seller's UTxO.

Executable signs the updated datum, containing the buyer's pub-key hash and time-lock, with the seller's private key. It also generates and writes the updated datum and "Update" redeemer in cbor format.

2.5 `p2p-claim-transaction`

Takes two arguments: a) name of fiat witness' wallet and b) name of seller's wallet.

Executable signs payment information (associated with the seller's name) with the private key of the fiat witness. It then generates and writes the corresponding Claim redeemer in cbor format.

3 End-to-end testing

An end-to-end trading scenario is tested with shell scripts described below. These shell scripts put together the backend Haskell executables discussed in the previous section, and build and submit transactions using the `cardano-cli` utility.

We have provided a testing scenario involving five wallets:

1. Fee wallet, also the fiat witness (*Alice*)
2. Sellers (*Barbara*, *Bob*, *Brandon*)
3. Buyer (*Charlie*)

3.1 Setup first wallet (Alice)

Alice's wallet needs to be funded with enough Ada to further fund the other participating wallets. (Working directory must be `e2e-test` .)

1. If working on local testnet, execute `./p2p/local-ini.sh` .
2. If working on preview testnet,
   (a) execute `./p2p/preview-ini.sh` ,
   (b) then fund Alice's wallet (address displayed) from the testnet-faucet[1].

Note that Alice plays the role of the fee wallet.

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/alice.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
9d9d35acf924cbfbbb7ce821b50708d06d4b5d5a23a0f6778521599d3cfbad15     0        10000000000 lovelace + TxOutDatumNone
```

3.2 Initialization

Serialize the plutus script for *onRamp* validator and fund the remaining wallets.

```
e2e-test$ ./p2p/01-p2p-init-transaction.sh

Initialization...

Creating 'onRamp' script...
Resolving dependencies...

Done serializing plutus script.

Initializing sellers...

Initializing buyer...

Funding wallets...
Estimated transaction fee: 175137 Lovelace
Transaction successfully submitted.
Waiting to see initial funding tx onchain...
```

[1]https://docs.cardano.org/cardano-testnets/tools/faucet

```
Transaction Id: b4dcb0f08f6c1866b0a21e28cd459b6116478e7c2e9c4737690bd69b38520572

Initialization completed.
```

After initialization, each seller (Barbara, Bob and Brandon) has 50 Ada, and buyer (Charlie) has 10 Ada.

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/barbara.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
b4dcb0f08f6c1866b0a21e28cd459b6116478e7c2e9c4737690bd69b38520572     0        50000000 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/bob.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
b4dcb0f08f6c1866b0a21e28cd459b6116478e7c2e9c4737690bd69b38520572     1        50000000 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/brandon.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
b4dcb0f08f6c1866b0a21e28cd459b6116478e7c2e9c4737690bd69b38520572     2        50000000 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/charlie.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
b4dcb0f08f6c1866b0a21e28cd459b6116478e7c2e9c4737690bd69b38520572     3        10000000 lovelace + TxOutDatumNone
e2e-test$
```

3.3 Sending sell offers

Now each seller (Barbara, Bob and Brandon) posts a sell offer at the *onRamp* script. In our example, each is selling around (but not exactly) 30 Ada, for approximately (but not exactly) 30 USD; exact numbers are chosen at random.

```
e2e-test$ ./p2p/02-p2p-add-sellers.sh
Adding sellers...

Barbara sells 30031473 lovelace...

Wrote barbaraSellDatum.cbor

Sending Barbara's sell offer UTxO...
Estimated transaction fee: 174565 Lovelace
Transaction successfully submitted.

Bob sells 30016003 lovelace...

Wrote bobSellDatum.cbor

Sending Bob's sell offer UTxO...
Estimated transaction fee: 174565 Lovelace
```

```
Transaction successfully submitted.

Brandon sells 30017733 lovelace...

Wrote brandonSellDatum.cbor

Sending Brandon's sell offer UTxO...
Estimated transaction fee: 174565 Lovelace
Transaction successfully submitted.
```

After Tx is submitted, we can see the sell orders at the *onRamp* script:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/onRamp.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
0c04cc227ddbb09b796ced4d5842bcb0e236a3c9784724d6320d2db6a60e4a9d     0        30031473 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \164\149#\203\156N\"u\172\235\&9\246\ACKZx\165\165\247\
\244\144a9\169/\173;\DC3\178\228/\152\190\CAN#\161@\161@\SUB\SOH\202>qX K\249\200''\163\CAN\167\230Ww!*\154\&0\236\
\ETXJ\221\138\254\USk5P\ETB\208p\141\222\221\RS\216z\128\216z\128\255" (ScriptDataConstructor 0 [ScriptDataBytes\
 "\164\149#\203\156N\"u\172\235\&9\246\ACKZx\165\165\247\244\144a9\169/\173;\DC3\178\228/\152\190",ScriptDataNumber\
 35,ScriptDataMap [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30031473)])],ScriptData\
Bytes "K\249\200''\163\CAN\167\230Ww!*\154\&0\236\ETXJ\221\138\254\USk5P\ETB\208p\141\222\221\RS",\
ScriptDataConstructor 1 [],ScriptDataConstructor 1 []]))
39d2587446a808cf9fa7d0351b2d56e4a86b297aa622f6497a2a546dee54ac34     0        30016003 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \135h=\168\&7\DC3v\145\ETB\SO\SUB\170\&9\STX\160\DEL\225c\
\156\199\t\244'+\SO\ESCr\241\151s\244\205\CAN\RS\161@\161@\SUB\SOH\202\STX\ETXX Wc\139\182\179\169>\133<t\229H\223\
\188\242\207B\135\&3-\164\188\207\&4\ENQ\DLE\222\151\241\f\224x\216z\128\216z\128\255" (ScriptDataConstructor 0 [\
ScriptDataBytes "\135h=\168\&7\DC3v\145\ETB\SO\SUB\170\&9\STX\160\DEL\225c\156\199\t\244'+\SO\ESCr\241\151s\244\205",\
ScriptDataNumber 30,ScriptDataMap [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30016003\
)])],ScriptDataBytes "Wc\139\182\179\169>\133<t\229H\223\188\242\207B\135\&3-\164\188\207\&4\ENQ\DLE\222\151\241\f\
\224x",ScriptDataConstructor 1 [],ScriptDataConstructor 1 []]))
905577f3aae5fcc1635acb889fee41ea449570954526535907dc28dab2c61320     0        30017733 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \DLE\146\245Td\DLE\164D'\249tb\225\219\141G\138\153M$%\RS:\
\246c\169C\242\251\141I\243\CAN&\161@\161@\SUB\SOH\202\b\197X j\198\ENQ\162\148\168'\193\134\183\234\162@\199\176\
\164\SOHC\169\RS(*T\FS\233\254s7\DC14G\145\216z\128\216z\128\255" (ScriptDataConstructor 0 [ScriptDataBytes "\DLE\
\146\245Td\DLE\164D'\249tb\225\219\141G\138\153M$%\RS:\246c\169C\242\251\141I\243",ScriptDataNumber 38,ScriptDataMap\
 [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30017733)])],ScriptDataBytes "j\198\ENQ\
\162\148\168'\193\134\183\234\162@\199\176\164\SOHC\169\RS(*T\FS\233\254s7\DC14G\145",ScriptDataConstructor 1 [],\
ScriptDataConstructor 1 []]))
```

We can verify that balance on each of the seller's wallets has diminished by the amount sent to the script (around 30 Ada) plus Tx fees:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/barbara.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
0c04cc227ddbb09b796ced4d5842bcb0e236a3c9784724d6320d2db6a60e4a9d     1        19793962 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/bob.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
39d2587446a808cf9fa7d0351b2d56e4a86b297aa622f6497a2a546dee54ac34     1        19809432 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/brandon.addr) --testnet-magic 2
```

```
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
905577f3aae5fcc1635acb889fee41ea449570954526535907dc28dab2c61320     1        19807702 lovelace + TxOutDatumNone
e2e-test$
```

3.4 Buyer selects a sell offer and posts a buy order

Buyer queries the UTxOs at the *onRamp* script for available sell offers. Then selects the best offer according to some criterion, for example the value/price ratio. Buyer communicates his intent to seller using some off-chain channel and seller signs the buy order. Datum of corresponding UTxO is updated accordingly:

```
e2e-test$ ./p2p/03-p2p-buy-order.sh

Choosing best sell offer...

Selected TxOutRef: 39d2587446a808cf9fa7d0351b2d56e4a86b297aa622f6497a2a546dee54ac34#0
Selected seller: Bob

Generating buyer's datum and redeemer...

Seller signs datum updated with buyer's pub-key-hash and time-lock...

Verifies signature:
True

Signature: 71530aa2cdd8ae2df23cc586301e9c57e7107025b1f85d6a8bc75d127f0c584eb15f17a6b1c4a729d92abe82e3ef19c1ea21f96\
5557557db59d9add6b443c301

Wrote charlieBoughtDatum.cbor
Wrote bobSoldRedeemer.cbor

Buy-order transaction...
Estimated transaction fee: 478602 Lovelace
Transaction successfully submitted.
Waiting to see buy order tx onchain...

Transaction Id: 79fc0d054d8c37e1553649c90d7d837b1c2f21f3cb407361b08cddec52f54610
```

Note that, instead of executing `./p2p/03-p2p-buy-order.sh`, we could have executed at this point `./p2p/05-p2p-cancel-transaction.sh` to cancel all sell orders.

3.5 Claim funds

After making fiat deposit, buyer wants to claim the corresponding funds in Ada. A fiat witness certifies the Tx by signing the corresponding payment info (hash of).

```
e2e-test$ ./p2p/04-p2p-claim-transaction

Fiat witness signs hash of fiat payment info...
```

```
Verifies signature:
True

Signature: 54bd2f02b1f08ddd0f295f7532feb3b7d506daf336af31e3f39cdd5fca48e46ca228f4605031cc260cc01686128112c83363d\
29c5e5a333ff5b96dce9ab35a01

Wrote bobPaymentInfoRedeemer.cbor

Claim transaction...
Estimated transaction fee: 467071 Lovelace
Transaction successfully submitted.
Waiting to see claim tx onchain...

Transaction Id: 8555e1dadf1c509a2b7025803b94a82fb5702d510211fd54eb9906b854ac6437
```

So now the bought assets are in Charlie's wallet:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/charlie.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
8555e1dadf1c509a2b7025803b94a82fb5702d510211fd54eb9906b854ac6437     0        30016003 lovelace + TxOutDatumNone
8555e1dadf1c509a2b7025803b94a82fb5702d510211fd54eb9906b854ac6437     2        6054327 lovelace + TxOutDatumNone
e2e-test$
```

and there remain only two sell offers at the *onRamp* script:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/onRamp.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
0c04cc227ddbb09b796ced4d5842bcb0e236a3c9784724d6320d2db6a60e4a9d     0        30031473 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \164\149#\203\156N\"u\172\235\&9\246\ACKZx\165\165\247\244\
\144a9\169/\173;\DC3\178\228/\152\190\CAN#\161@\161@\SUB\SOH\202>qX K\249\200``\163\CAN\167\230Ww!*\154\&0\236\ETXJ\
\221\138\254\USk5P\ETB\208p\141\222\221\RS\216z\128\216z\128\255" (ScriptDataConstructor 0 [ScriptDataBytes "\164\
\149#\203\156N\"u\172\235\&9\246\ACKZx\165\165\247\244\144a9\169/\173;\DC3\178\228/\152\190",ScriptDataNumber 35,\
ScriptDataMap [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30031473)])],ScriptDataBytes\
 "K\249\200``\163\CAN\167\230Ww!*\154\&0\236\ETXJ\221\138\254\USk5P\ETB\208p\141\222\221\RS",ScriptDataConstructor 1\
 [],ScriptDataConstructor 1 []]))
905577f3aae5fcc1635acb889fee41ea449570954526535907dc28dab2c61320     0        30017733 lovelace + TxOutDatumInline\
 BabbageEraOnwardsConway (HashableScriptData "\216y\159X \DLE\146\245Td\DLE\164D'\249tb\225\219\141G\138\153M$%\RS:\
\246c\169C\242\251\141I\243\CAN&\161@\161@\SUB\SOH\202\b\197X j\198\ENQ\162\148\168`\193\134\183\234\162@\199\176\
\164\SOHC\169\RS(*T\FS\233\254s7\DC14G\145\216z\128\216z\128\255" (ScriptDataConstructor 0 [ScriptDataBytes "\DLE\
\146\245Td\DLE\164D'\249tb\225\219\141G\138\153M$%\RS:\246c\169C\242\251\141I\243",ScriptDataNumber 38,ScriptDataMap\
 [(ScriptDataBytes "",ScriptDataMap [(ScriptDataBytes "",ScriptDataNumber 30017733)])],ScriptDataBytes "j\198\ENQ\
\162\148\168`\193\134\183\234\162@\199\176\164\SOHC\169\RS(*T\FS\233\254s7\DC14G\145",ScriptDataConstructor 1 [],\
ScriptDataConstructor 1 []]))
```

3.6 Cancel unexecuted sell orders

After the set deadline expiration (in our example, one minute after posting buy-order), seller's can reclaim the unexecuted orders:

```
e2e-test$ ./p2p/05-p2p-cancel-transaction.sh
Cancel sell orders...

Canceling Barbara's order...
Estimated transaction fee: 454643 Lovelace
Transaction successfully submitted.
Barbara's order has ben canceled.

Bob has sold or closed sell offer; nothing to cancel.

Canceling Brandon's order...
Estimated transaction fee: 454643 Lovelace
Transaction successfully submitted.
Brandon's order has ben canceled.

Waiting to see all sell orders canceled onchain...

Done.
```

So now there remain no UTxOs at the onRamp script,

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/onRamp.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
e2e-test$
```

and the balances at the seller's wallets have been updated accordingly:

```
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/barbara.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
fb604189b5d750484452f5ba81d2fb1360b2e64acd02bb37ad2ed3197dbdb318     0        30031473 lovelace + TxOutDatumNone
fb604189b5d750484452f5ba81d2fb1360b2e64acd02bb37ad2ed3197dbdb318     1        19339319 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/bob.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
39d2587446a808cf9fa7d0351b2d56e4a86b297aa622f6497a2a546dee54ac34     1        19809432 lovelace + TxOutDatumNone
e2e-test$
e2e-test$ cardano-cli conway query utxo --address $(cat ./p2p/keys/brandon.addr) --testnet-magic 2
                           TxHash                                 TxIx        Amount
--------------------------------------------------------------------------------------
813a52159491da610e5a5e5c11d4f963281e8021eb7f69f59c322e0e1645bbaf     0        30017733 lovelace + TxOutDatumNone
813a52159491da610e5a5e5c11d4f963281e8021eb7f69f59c322e0e1645bbaf     1        19353059 lovelace + TxOutDatumNone
e2e-test$
```

As expected, one of the seller's has sold around 30 Ada and the other two have recovered almost their original 50 Ada (the difference due to spent fees).