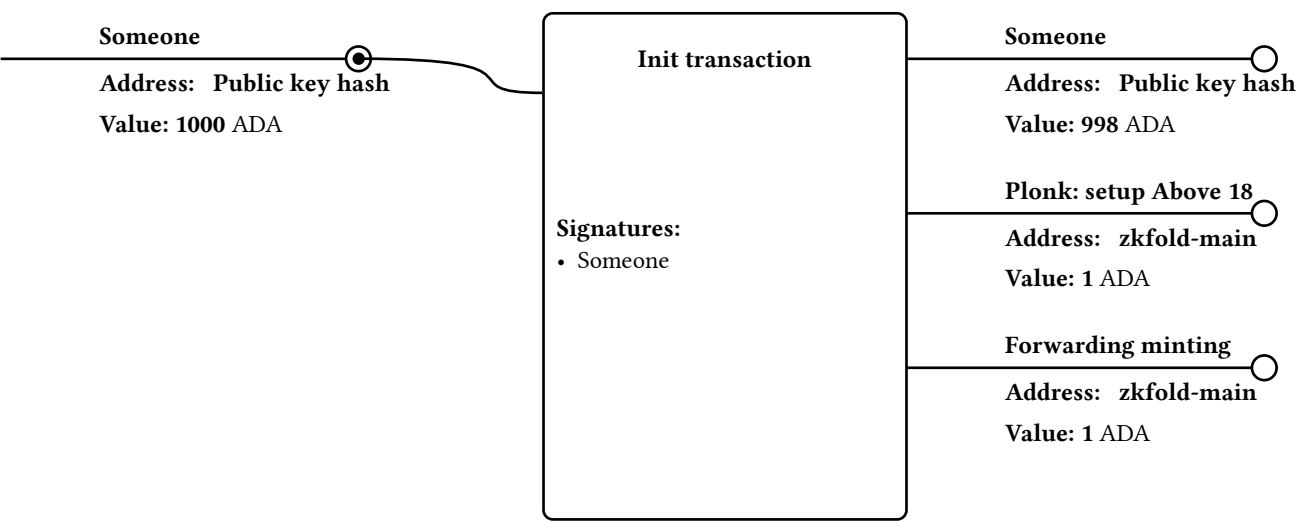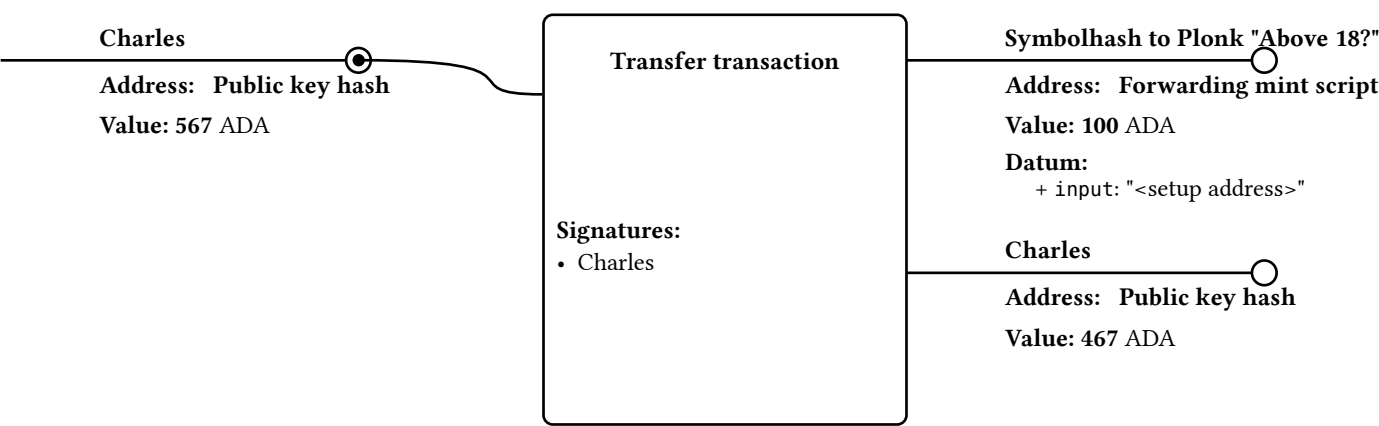# End-to-end test for the Plonk verifier

A ZK-KYC example: Alice wants to mint and send Bob a token that represents a proof of some statement about Bob.
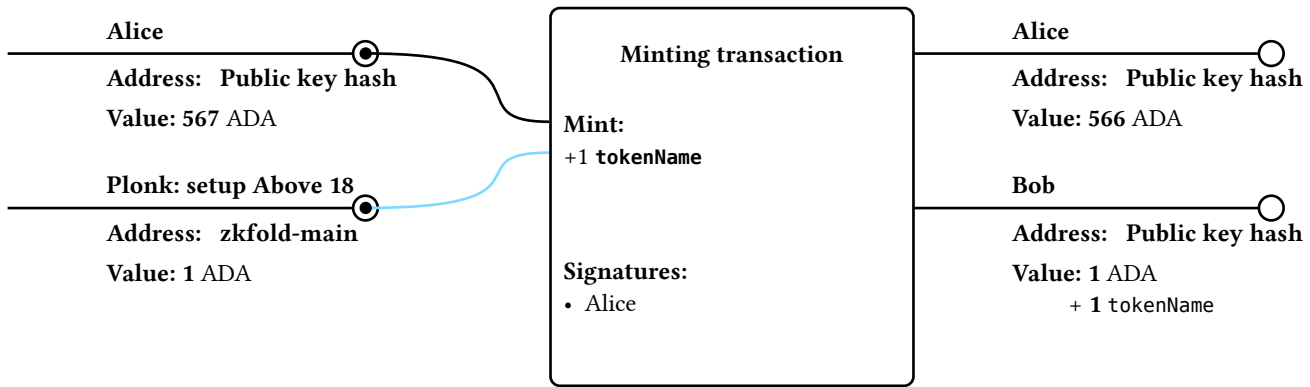
- The first transaction is a setup that needs to be performed only once for this application;
- In the second transaction, Alice minting tokens and sends ada + tokens to Bob;
- In the third transaction, Bob burning tokens.

Someone

Address: Public key hash

Value: 1000 ADA

Init transaction

Signatures:
- Someone

Someone

Address: Public key hash

Value: 998 ADA

Plonk: setup Above 18

Address: zkfold-main

Value: 1 ADA

Forwarding minting

Address: zkfold-main

Value: 1 ADA

Note: Someone posts the Plonk verifier script on-chain.

Charles

Address: Public key hash

Value: 567 ADA

Transfer transaction

Signatures:
- Charles

Symbolhash to Plonk "Above 18?"

Address: Forwarding mint script

Value: 100 ADA

Datum:
   + input: "<setup address>"

Charles

Address: Public key hash

Value: 467 ADA

Note: Charles sents ada to a smart contract address.

Alice

Address: **Public key hash**

Value: 567 ADA

Plonk: setup Above 18

Address: **zkfold-main**

Value: 1 ADA

Minting transaction

Mint:

+1 `tokenName`

Signatures:
- Alice

Alice

Address: **Public key hash**

Value: 566 ADA

Bob

Address: **Public key hash**

Value: 1 ADA

  + **1** `tokenName`

Note: Alice sents ada and plonk tokens to Bob.

Bob

Address: Public key hash

Value: 100 ADA
  + **1** tokenName

Symbolhash to Plonk "Above 18?"

Address: Forwarding mint script

Value: 100 ADA

Datum:
  + input: "<setup address>"

Plonk: setup Above 18

Address: zkfold-main

Value: 1 ADA

Forwarding minting

Address: zkfold-main

Value: 1 ADA

Burning transaction

Mint:
−1 **tokenName**

Signatures:
• Bob

Bob

Address: Public key hash

Value: 200 ADA

Note: Bob burn plonk tokens.