Shielded Transfers

You can construct shielded transfers similarly to transparent transfers, except that rather than providing implicit addresses as thesource andtarget , you will need to provide a spending key and payment address, respectively.

Before building your transaction, you will want to make sure theshielded context is synced .

Constructing the Transfer

This example assumes you have a spending key with sufficient shielded balance, a recipient payment address, and the public key of an implicit account with gas funds.

use namada_sdk :: args :: TxShieldedTransferData ; use namada_sdk :: masp :: PaymentAddress ; use namada_sdk :: signing :: default_sign;

let spend_key =

ExtendedSpendingKey :: from_str ("zsknam1q0medj45qqqqpq9wh90qd9c7d9f7n5xxn89h6dl54k0jfmucwn4yk7nykxwcrjmk4ylkdnlnn3wkkd9f3ul3nyw8hv5wlsfgklzr5ghzk2spzzwm05csvl2s3rn0aq7f9w4z7guul682yrw4hsmren2k2lgdp003uuj) . expect ("Invalid spending key"); let payment_addr =

PaymentAddress :: from_str ("znam1vsz9wsge4u9c0thh38vhn2z9awzfqn4540ue4haxmcxl43srptrgrtlhn6r9cd9razawuakcclc") . expect ("Invalid payment address");

sdk . shielded_mut () .await. load () .await. expect ("Could not load shielded context");

// specify the transfer arguments let data =

TxShieldedTransferData { source : spend_key, target : payment_addr, token : sdk . native_token (), amount :

InputAmount :: from_str ("5") . expect ("Invalid amount"), };

// build the tx let

mut transfer_tx_builder = sdk . new_shielded_transfer (vec! [data], vec! [spend_key]) . wrapper_fee_payer (alice_acct . public_key . clone ());

let (mut transfer_tx, signing_data) = transfer_tx_builder . build (& sdk) .await . expect ("Unable to build transfer tx");

// sign the tx sdk . sign (&mut transfer_tx, & transfer_tx_builder . tx, signing_data, default_sign, ()) .await . expect ("unable to sign transfer tx");

// submit tx to the chain match sdk . submit (transfer_tx, & transfer_tx_builder . tx) .await { Ok (res) =>

println! ("Tx result: {:?}" , res), Err (e) =>

println! ("Tx error: {:?}" , e) }

Shielded sync Proof of stake transactions