

EOSIO/Antelope Transaction

```
somecontract::someaction(...);
```

...

```
thezeostoken::begin( $\Pi$ , tx)
{
  // list of EOSIO actions which depend
  // on privacy actions of the ZEOS
  // smart contract:
  randcontract::someaction(ztx, ...);
  othercontrac::someaction(...);
  thezeostoken::exec(ztx);
  thirdcontrac::someaction(ztx, ...);

  // verification of Halo 2 proof bundle
  //  $\Pi$  which validates all zactions in
  // the above list of EOSIO actions:
  thezeostoken::verify( $\Pi$ , X);
}
```

```
thezeostoken::step()
{
  // The 'step()' action is just
  // a wrapper for the actual
  // EOSIO action to be executed.
  randcontract::someaction(ztx, ...);

  // In addition to that it auto-
  // matically executes all
  // zactions which are packed
  // as payload into the above
  // EOSIO action.
  thezeostoken::exec(ztx);
}
```

```
thezeostoken::step()
{
  // Not all actions may contain
  // zactions and are just normal
  // EOSIO actions with no privacy
  // dependencies.
  othercontrac::someaction(...);
}
```

```
thezeostoken::step()
{
  // A 'step()' might contain no
  // third party action but just
  // an 'exec' to execute a
  // sequence of zactions.
  thezeostoken::exec(ztx);
}
```

```
thezeostoken::step()
{
  // It is important that the num
  // of 'step()'s in the EOSIO tx
  // equals the number of actions
  // in the list of actions 'tx'
  // or 'begin' will fail.
  thirdcontrac::someaction(ztx, ...);

  // The number of zactions might
  // vary from action to action.
  thezeostoken::exec(ztx);
}
```

...

validated by

Time/Execution Order

