Pseudonym Pairs: A foundation for proof-of-personhood in the web 3.0 jurisdiction

Author: BipedalJoe, Year 18

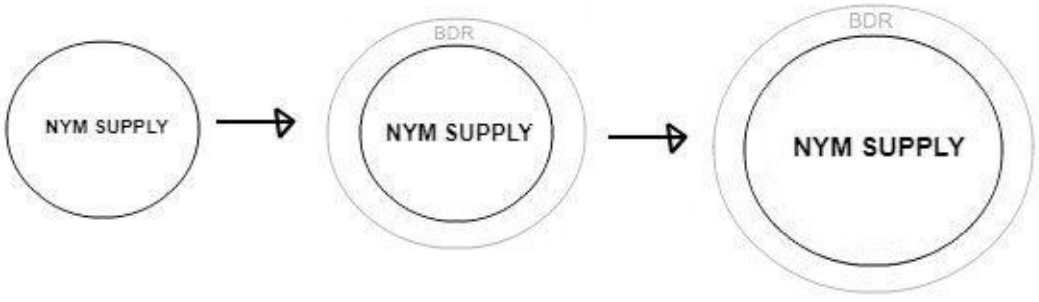
Pseudonym Pairs is a dApp for global proof-of-personhood, through monthly pseudonym events that last 20 minutes, where every single person on Earth is randomly paired together with another person, 1-on-1, to verify that the other is a person, in a pseudo-anonymous context. The events provide NYM tokens, global personhood tokens, untraceable from month to month and disposable, a sort of “temporary access tokens” similar to festival bracelets. The proof-of-personhood is that you are with the same person for the whole event.

1-on-1 verification of (pseudo-anonymous) personhood

Within the 1-on-1 pairs, people can socialize as they want, and can be seen as being employed in government positions, expected to stay within the pair for the entire duration of the pseudonym event. The 1-on-1 pairs is the standard organization, requiring mutual verification. In the case of a problem, such as a bot attacker, or, a person not showing up, people can break up their pair, to be assigned to be verified by another pair (2-on-1), similar to how people are verified at the “virtual border”. (see below)

How to opt-in to Pseudonym Pairs

The population is used to secure a "virtual border" around the network, and “border tokens” (BDR) can be bought to apply at the “virtual border” and meet a random pseudonym pair, that verify the person that opts-in. The “border tokens” are distributed through the population, each person can issue 1 BDR, and each time BDR is issued, the ability to issue one more BDR is given to a random person within the pseudonym pool, distributing the ability to invite new people onto the population as a whole, making it possible for the network to accept new people multiple times its population size, so that it can grow from 0 to potentially 5 billion people.



The population sorts themselves into pairs

The pair sorting mechanism that lets every person (or, 50% of all people, who is not mandated) sort themselves, with the function sortingHat(), that fetches a random peer from a list, forms a pair, and prunes the list by replacing the people who formed a pair with people from the beginning and end. The function has low gas-costs per person, ideal for a consensus technology like Ethereum or BitLattice.

```
function sortingHat() {
  uint ID = pseudonymID[msg.sender];
  require(ID != 0);
  // prune the list with msg.sender, so that the person does not pick themselves as a peer
  address fromBeginning = pseudonymIndex[beginning];
  pseudonymIndex[ID] = fromBeginning;
  pseudonymID[fromBeginning] = ID;
  delete pseudonymIndex[beginning];
  beginning++;
  delete pseudonymID[msg.sender];
  // select a random peer, and assign msg.sender and the peer to a pair
  uint randomNumber = uint64(sha3(sha3(block.blockhash(block.number), entropy), now)) % (end -
beginning);
  address randomPeer = pseudonymIndex[randomNumber];
  pair[pairCount][0] = msg.sender;
  pair[pairCount][1] = randomPeer;
  pairCount++;
  // prune the list with randomPeer, adding the person from the end to their position
  address fromEnd = pseudonymIndex[end];
  pseudonymIndex[randomPeer] = fromEnd;
  pseudonymID[fromEnd] = randomPeer;
  delete pseudonymIndex[end];
  delete pseudonymID[randomPeer];
  end--;
  entropy = sha3(randomNumber, msg.sender, randomPeer);
}
```

Borderless personhood tokens for a global population

The Pseudonym Pairs protocol has no way of distinguishing between people, since it treats any human being as equivalent, it cannot shut certain people out. It is borderless in that the protocol cannot know how many people it has counted unless it assumes it is everyone.

The personhood tokens are mixed, making them untraceable

When the pseudonym event is over and people have been verified, all personhood tokens are mixed, through the entire population. The mixing is simple, people continuously join mixers, incrementally increasing the number of mixers over time as people invoke mix(), and a personhood token is issued to their new public key.

```
function mix(uint _pubkey1, uint _pubkey2) {
    if(peopleCount[mixerCount] == 4) mixerCount++;
    pubkeys1[mixerCount].push(_pubkey1);
    pubkeys2[mixerCount].push(_pubkey2);
    peopleCount[mixerCount]++;
    inMixer[msg.sender] = mixerCount;
}
```

Profitability of collusion attacks

The only attack vector on the system is collusion attacks, a population of colluding people can over-number the randomization of pairs, and gain control of majority in a percentage of all pairs, two people in a pair, freeing those people to show up as a single in another pair or at the “virtual border”. That lets the colluding population sustain a population of bots and provide those bots with personhood tokens.

The profitability of the attack is quite low, the randomization of pairs from a global population, and the 1-on-1 pairs requiring 100% majority to be “hijacked”, means that a colluding population will get control of proportionally much fewer pairs than there are people in the colluding population. If 10% of the population collude, they get around 0.5% of all pairs under bot control, 0.05x a personhood token per person attacking.