The decentralized red envelope

1. Traditional red envelopes and decentralized red envelopes

Traditional red envelopes, such as QQ, WeChat, etc., use a lottery system to issue a corresponding proportion of the amount to users at the bottom. The red envelope distribution in this mode is completely centralized.

Decentralized red envelopes are the application of decentralized random numbers to the red envelope distribution scenario， which achieved true random distribution of red envelopes, ensuring that each user can receive the corresponding amount fairly. It is different from the traditional red envelope distribution process. The random number here is generated on the blockchain, which is publicly visible and completely decentralized. The biggest difference between decentralized red envelopes and traditional red envelopes is fairness. The decentralized red envelope uses a decentralized random number, which is generated by multiple parties. At the same time, it also realizes the open and verifiable random number, which is more fair than the traditional centralized server pseudo-random number generation method of the traditional red envelope.

2. The paper foundation and existing scheme of decentralized red envelope

The core of the decentralized red envelope is the generation of decentralized and verifiable random numbers. The concept and theoretical prototype of verifiable random numbers were proposed by Micali. The main idea is to extend the pseudo-random number generation function to increase verifiability and unpredictability.In Micali's research, participants can use the pseudo-random formula v=fs(x) to generate the independent variable x and evidence Proof at any point in the domain to test the randomness and unpredictability of the output v of fs(x). This mathematical relationship is called a verifiable random function VRF.The verifiable random function of this VRF provides participants with a high sense of trust in participation, which can ensure the effective participation and self-certification of participants.

At this stage, there has been more research on the generation and application of decentralized random numbers. In the paper "A Blockchain-based Random Number Generation Algorithm and the Application in Blockchain Games" by the Mingxiao Du team, a block-based Random number generation scheme for chain games:

1.The game provider generates a random number Np and a public-private key pair, the game provider uses the public key to encrypt the random number to E(Np), and sends the message massage: {E(Np) & GameID} to the blockchain

2. The blockchain uses smart contract to detect GameID, if it is a new game, record the corresponding information record:{ txid1 & E(Np) & GameID }, and return txid1 to the Game provider

3. Game provider sends txid1 to all participants, and participants check the information by themselves based on txid1. Participant i sends his random number to the Game provider massage: {GameID & Ni }, and the Game provider sends the collected random number to the blockchain contract record: {E(Np) & GameID & (N1...Ni. ..Nn)}

4. Smart contract detects information, generates record: {txid2 & txid1 & E(Np) & GameID & (N1...Ni...Nn) }, returns txid2 to Game provider

5. Game provider broadcasts txid2 to all participants. After all participants agree, they can start to generate random numbers. The parameters include block_txid2_hash, txid2, N(p), (N1...Ni...Nn), and use f(x) to generate random numbers k

6. Use k to play the game, after the game provider uploads the record and the result of the game to the smart contract

7.Smart contract generates transaction information massage: {txid3 & txid2 & txid1 & E(Np) & GameID & (N1...Ni...Nn) & result & Operation & Private Key}

8.Participants use private keys and txid3 to check the authenticity of all information

It can be seen that there are three obvious shortcomings in this solution. The first is that the interaction cycle is too long, the second is that the transmission of random numbers is transmitted in plain text, and the third is that it still relies on the random number Np provided by the central game provider. However, This solution provides decentralized ideas to a certain extent.

The randao project based on Ethereum is a relatively simple and efficient decentralized random number solution. Its basic implementation ideas are as follows:

1. Collect the encrypted hash value of the random number provided by the participant, and collect the encrypted hash within a certain time interval for later verification. In order to ensure that the participant completes the entire random process, it is necessary to provide gas as a pledge and send it to the contract account together with the hash

2. The successfully collected hash value provider is designated as the final random number generator participant, and then the participant sends the random number to the contract account, and the contract account verifies all received random numbers as valid random numbers. Users who do not provide a random number will not affect the random number generation process, and the pledge will not be returned.

3. Use the collected random numbers as a seed to generate a final random. After the final randomization is completed, all pledge deposits can be equally distributed to participating users

In Randao, by sending the hash first and then sending the random number, the security problem of the plaintext transmission process is solved, and certain participants are prevented from cheating through gas mortgage, but the overall efficiency is preset to take up to 6 block times. , There are many interaction processes and it is not convenient for users to use.

3. The decentralized red envelope scheme we proposed

In order to adapt to the decentralized environment, we proposes a decentralized red packet scheme based on verifiable random functions(VRF).Verifiable random function is an encryption scheme that maps the input to a verifiable pseudo-random output. First, the result of VRF is a random number. Secondly, because it contains the private key signature of the generator, the verifier can determine the legitimacy of the random number through the public key.

The VRF algorithm flow of the verifiable random function is as follows:

1. The prover generates a pair of secret keys, PK and SK;
2. The prover calculates result = VRF_HASH(SK,info);
3. The prover calculates proof = VRF_Proof(SK,info);
4. The prover submits the result and proof to the verifier;

5. The verifier calculates whether result = VRF_P2H(proof) is established, if it is established, continue, otherwise abort;

6. The prover submits PK and info to the verifier;

7. The verifier calculates True/False = VRF_Verify(PK,info,proof), True means verification passed, False means verification failed.

The essence of red envelope distribution is the generation of decentralized random numbers. The solution we proposed can solve the cheating behavior of users in the process of random number generation. Each user participating in the process of red packet allocation needs to generate the corresponding zero knowledge proof through the verifiable delay function, so that anyone can verify the user's sub share generation process, ensuring the fairness and verifiability of the final random number generation.

The process of decentralization is as follows:

① Sender sends red packet on client

② The recipient can choose whether to click collect or not on the client. If not, the recipient will not participate in the distribution of red packets

③ The receiver chooses his own random number as the input, and obtains the random number sub share and zero knowledge proof through the VRF function

④ Verify the validity of the contract verification certificate. If it is invalid, it will be out. If it is valid, it will store the sub share information in the block.

⑤ Waiting for all receivers to send and verify the random number sub shares, the default generation time of 6 blocks.

⑥ The contract takes all the random number sub shares as the input of VRF, and outputs the final random number and zero knowledge proof

⑦ The final generated random number is used as seed to allocate the amount of red envelope to the receiver

⑧ The contract transfers the transaction according to the allocated red envelope amount

⑨ If the transfer is successful, the receiver will be notified at the client.