隐私预测市场

Jade Xie

2025 年 12 月 30 日

1 协议描述

1.1 阈值加密准备

阈值加密委员会：

- n 个成员的私钥：$sk_1, sk_2, \ldots, sk_n$
- 公钥：pk
- 阈值: t

对于一个用 pk 进行加密的秘文 C，当至少有 n 个成员中的 t 个成员的私钥进行解密时，都能解密成功。

1.2 下注阶段

对于一个用户，其地址为 $address_i$ ，当其对某一个预测方向 $side_i$ 下注 $amount_i$ 金额时，链下计算：

- $ct_i \leftarrow Enc(pk, side_i \| address_i)$: 阈值加密的下注方向的密文，其中 $side_i$ 是下注的方向，$address_i$ 为用户地址，为 160 位。
  - 这里我们采用 ElGamal 加密方案。
  - 将消息 $side_i \| address_i$ 映射到椭圆曲线上，

$$m \leftarrow g^{side_i \times 2^{160} + address_i \mod q}$$

  - 计算

$$\beta \xleftarrow{R} \mathbb{Z}_q, \quad v \leftarrow g^{\beta}, \quad e \leftarrow m + pk^{\beta}$$

  - 输出

$$ct_i = (v, e) \in \mathbb{G} \times \mathbb{G}$$

- $comm_i = Poseidon(side_i \| salt \| amount_i \| address_i)$: 计算承诺，其中 $salt$ 为全局公开的随机盐。
- $(\pi_i, ct_i) \leftarrow Prove(pk, comm_i, amount_i, address_i, salt; side_i)$: ZK Proof 证明下注合法。证明：
  1. $side_i \in \{0, 1\}$
  2. $amount_i > 0$，以及验证 $amount_i$ 的取值范围合法。
  3. 计算 $ct_i := Enc(pk, side_i \| address_i)$
  4. $comm_i == Poseidon(side_i \| salt \| amount_i \| address_i)$

接着，向链上提交 $(address_i, amount_i, ct_i, comm_i, \pi_i)$ 。

链上执行:

- $1/0 \leftarrow Verify(\pi_i, address_i, ammount_i, ct_i, comm_i, pk)$ ：验证 ZK Proof 是否有效。若输出为 1，表明证明有效，则链上记录下注数据

$$\{address_i, ammount_i, ct_i, comm_i\}$$

; 否则提示用户下注失败。

1.3 开奖阶段

在开奖阶段，阈值解密委员会从链上拉取所有的数据，若这一轮中链上总共有 M 个用户的有效数据，则从链上拉取的数据为

$$\{(address_i, amount_i, ct_i, comm_i)\}_{i=1}^{M}$$

在链下，计算

- 对于 $i = 1, \ldots, M$, 计算 $(side_i, address_i) \leftarrow Dec(ct_i, sk_1, sk_2, \ldots, sk_t)$: 解密委员会中要至少有 t 个成员才能对阈值加密的密文 $ct_1, \ldots, ct_M$ 进行解密。
- $(\pi_{batch}, sum_0, sum_1) \leftarrow BatchProve\left(pk, salt, \{(comm_i, ammount_i)\}_{i=1}^{M}; \{side_i, address_i\}_{i=1}^{M}\right)$: 生成批量证明并输出聚合金额，在电路中:
  1. 证明 $side_i \in \{0, 1\}$ $(i = 1, \ldots, M)$
  2. 证明 $comm_i == Poseidon(side_i \| salt \| amount_i \| address_i)$ $(i = 1, \ldots, M)$
  3. 计算输出 sum_0, sum_1 ,

$$sum_0 = \sum_{i=1}^{M} (1 - side_i) \times amount_i, \quad sum_1 = \sum_{i=1}^{M} side_i \times amount_i$$

向链上提交数据 $(\pi_{batch}, sum_0, sum_1)$ ，链上计算

1. $1/0 \leftarrow BatchVerify(\pi_{batch}, sum_0, sum_1, pk, salt, \{(address_i, amount_i, comm_i)\}_{i=1}^{M})$: 验证批量证明是否有效。若输出为 1，则说明批量证明有效，则继续以下流程，否则直接输出失败。
2. 更新奖金累计池，更新

$$total_0 \leftarrow total_0 + sum_0 \quad total_1 \leftarrow total_1 + sum_1$$

可以分批次得到批量证明，然后逐步更新奖金累计池，直到所有用户的数据都覆盖且不重复。

当从 Oracle 处获得获胜方向 $winning\_side$ 后，进行结算和分配奖金阶段:

1. 计算赔率 ρ:

$$\rho = \frac{total_0 + total_1}{(1 - winning\_side) \times total_0 + winning\_side \times total_1}$$

2. 对于每个用户计算奖金，若 $side_i == winning\_side$，则

$$reward_i = amount_i \times \rho,$$

用户调用合约领取收益; 否则 $reward_i = 0$ 。