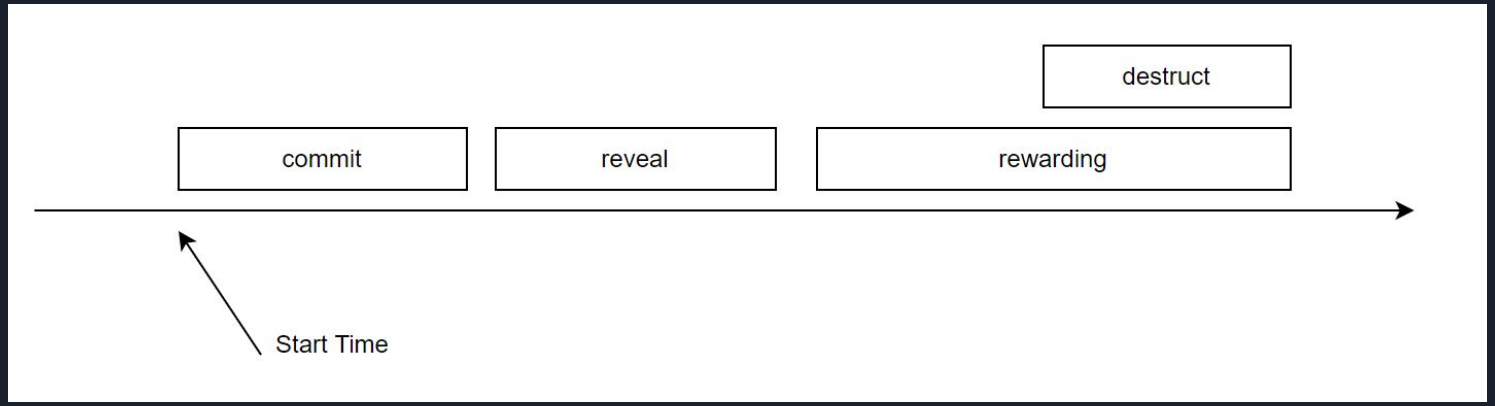
CRNL: Commit-reveal numeric lottery

Астахов Сергей

Agenda

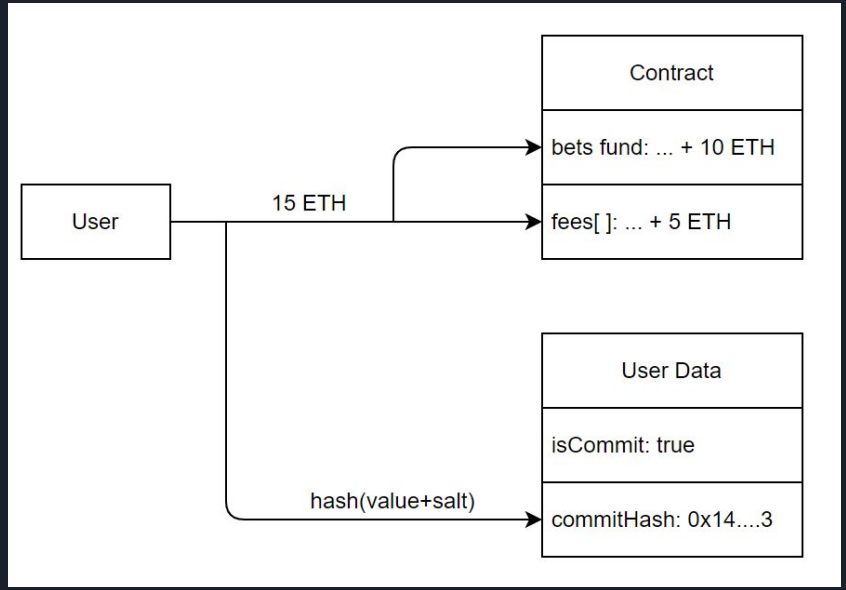
- Стадии работы контракта
  - commit
  - reveal
  - rewarding
  - destructing
- Решенные проблемы безопасности
- Дополнительные опции
- Возможные варианты наследуемых контрактов

Временные периоды



Commit

- Участник переводит ставку, honor fee (“гарантию честности”) и комиссию на адрес контракта
- Участник присылает hash(value + salt)
- Участник может поменять хеш, не внося дополнительных средств



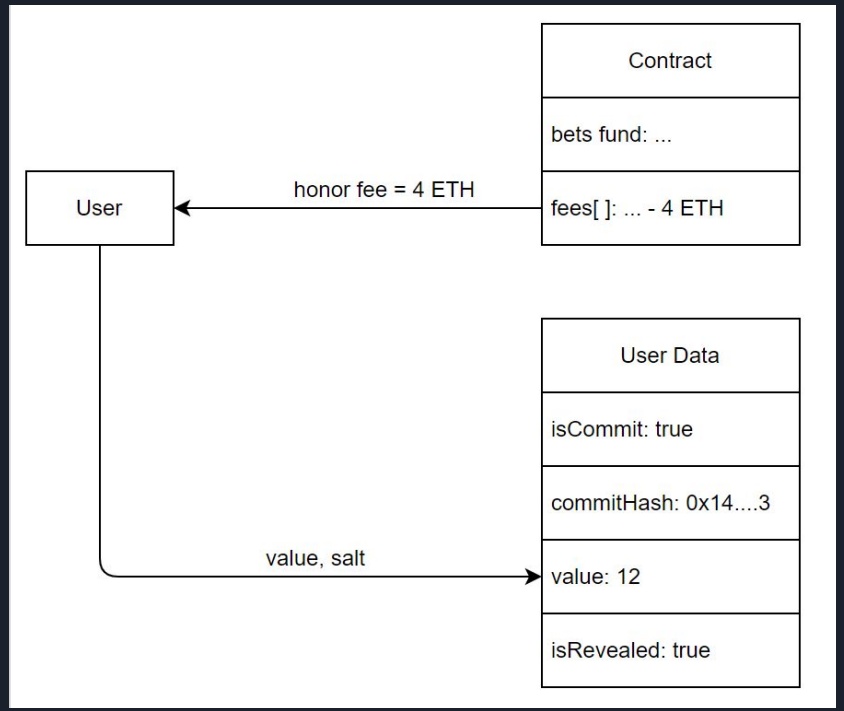
Reveal

- участник присылает value, salt

- если

  commitHash == hash(value + salt)

  то участник получает назад honorFee, value участвует в формировании среднего

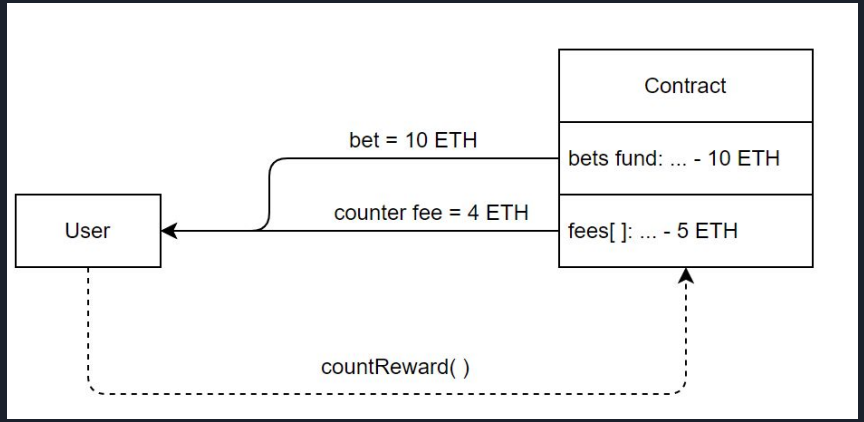


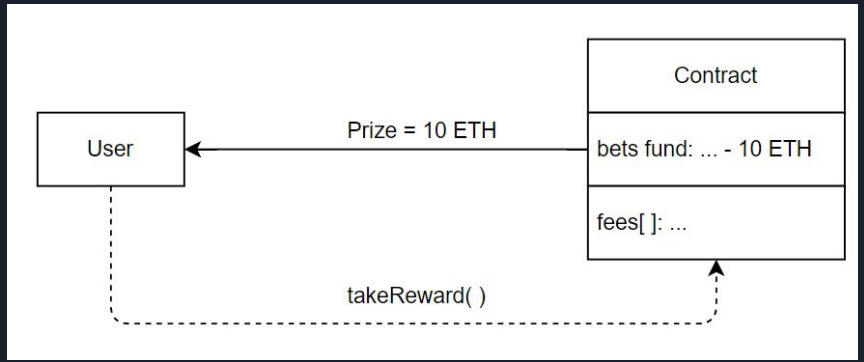
Rewarding

- Один из участников (обязательно revealed) вызывает countReward( ), находит min(|value$_i$ - avg|) и количество победителей, за что получает назад свою ставку + комиссии

- Далее участники могут вызывать takeReward( ) и получать награду, если их

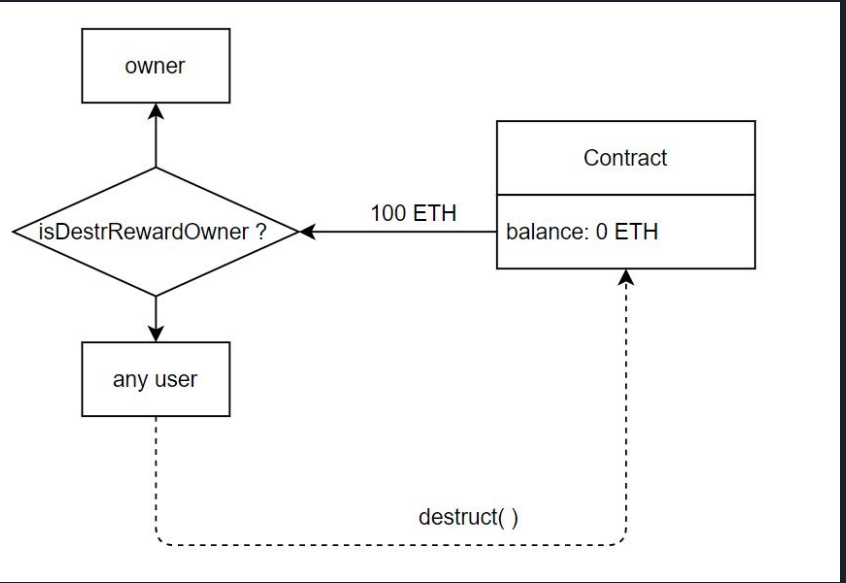
  | value - avg | == min





Destructing

- все еще можно вывести ETH
- любой пользователь сети etherium может вызвать уничтожение контракта, остаток получает sender или owner (зависит от настроек)



Проблемы безопасности

Проблема	*Решение*
Превышение gas limit	Ограничение числа участников
Множественное получение награды или комиссий	Флаги и require
Ошибки при вызове countReward(...) при 0 reveal’ов	countReward(...) может вызывать только revealed пользователь

Экономическая задача: найти оптимальное соотношение числа участников, размера ставки и комиссий

Дополнительные опции

- вывод owner fee возможен в любой момент
- при уничтожении контракта, остаток получает sender или owner (зависит от настроек)
- возможность смены owner’а
- наличие helper.py - программы для рассчета commitHash(...)

Варианты наследуемых контрактов

- TrueAvgCRNL - контракт на основе среднего значения values
- MedianCRNL - контракт на основе ⅔ медианы values
- ModCRNL - контракт на основе ∑(values) % p
- Contract factory для CRNL
- Proxy Contract для CRNL

etc.