Шардинг через механизм сообщений
(01.09.2020)

Смарт-контракты могут обмениваться сообщениями между друг другом, в том числе между разными шардами.
При создании смарт-контракта задается высота в блоках, которая определяется финальность передачи сообщения из одного щарда в другой. При достижении такого значения сообщение передается на дальнейшую обработку в смарт-контракт. При передачи сообщений внутри одного шарда - получение сообщений выполняется в следующем блоке. Минимальная высота - это фактическ величина безопасности в случае если смарт-контракт является каналом передачи ценностей или другой важной информации. Чем он больше тем более безопасным является канал, но с другой стороны - больше времени нужно ожидать пользователям для завершения операции. Процессом переноса сообщений из одного шарда в другой занимаются те майнеры, которые одновременно майнят эти два шарда. В случае если нет таких майнеров, то сообщение будет отложено и ждать их появления.

Безопасность

Безопасность каналов можно определить через стоимость атаки 51% на время которое необходимо для успешной передачи сообщения в шард приемник. Чем быстрее канал, тем дешевле атака или другими словами больше риск. Мы можем построить схему управления рисками, которая обеспечивает одновременно высокую безопасность и высокой производительности через создание двух каналов передачи ценностей:

1) Основной канал - медленный, но надежный канал. Время фиксации всегда указывается достаточно большое, например 1 млн блоков. Этот канал имеет свой токен в шарде приемнике, который и отождествляет монету шарда источника.
2) Быстрый канал используется для быстрой передачи небольшого размера ценностей (по сравнению со стоимостью атаки). Например, фиксация операции через 100 блоков. Канал тоже имеет свой токен,который будем называть **транзитным**, он нужен только для кратковременного использования.

Схема работы такова:
1. Пользователь отправляет сумму ценностей через быстрый канал (сумма ценностей всегда меньше стоимости атаки)
2. Получив их в другом шарде он их там меняет на основные токены, которые передаются через медленный канал для этого использует встроенную биржу DEX. На этом этапе работа пользователя завершается.
3. Транзитные токены отправляются в первый шард, где после получения отправляются обратно во второй шард но через медленный канал и выставляются опять на DEX (для обеспечения непрерывной ликвидности

транзитных токенов). Этим занимается другой участник отношений, назовем его посредник, он профессиональный участник. Для того чтобы его мотивировать его выставлять ордера на DEX, а также замораживать свой капитал в процессе длительной но высоконадежной операций передачи ценностей он меняет токены не в соотношении 1 к 1 а с определенным дисконтом в который закладывается стоимость заморозки капитала и его прибыль. Конкретный курс будет регулировать рынок.

Отправка монет из кошелька между шардами

Есть возможность отправки монет напрямую из интерфейса кошелька между счетами соседних шардов, для это адрес счета получателя должен содержать название шарда и номера счета, разделенные двоеточием: “SHARD:AccNumber”
При этом название шарда резолвится в номер аккаунта со смарт-контрактом, который является шлюзом (как вариант этот список соответствий можно зашивать в сам интерфейс кошелька).

Причина по которой используется промежуточный смарт-контракт - это изолирование ответственности. В случае компрометации одного шарда ценности других шардов не пострадают. При кросс-переводах монеты остаются на счете смарт-контракта внутри блокчейна, в другой блокчейн попадает только своеобразная расписка, на основании которой делаются движения токенов.

Такая модель позволяет создавать шарды очень широкого назначения, например временные но с высокой производительностью внутри.
Пример: Создается шард1, продаются токены на него, пользователи начинают работать с даппами внутри него совершая 1000 tps, через некоторое время когда размер базы превышает разумный предел, например 1000 Гбайт, создается Шард2, токены меняются на токены шарда 1, все переходят работать в шард2, майнеры отключают поддержку 1-го шарда.

Новые методы в смарт-контрактах

Новый встроенный метод для передачи сообщений (в новом обновлении):
SendMessage(*ShardAcccountStr,MethodName, Params*);
где:
ShardAcccountStr - строка в формате ИмяШарда:НомерСчета
MethodName- имя метода, который будет вызван для обработки данного сообщения, этот метод должен быть помечен признаком `message`
Params - параметры, которые передаются в указанный метод

пример:

```
//shard1 TERA
"public"
function SendTest(Params)
{
   if(context.Account.Num!==context.FromNum)
       throw "Access is allowed only from your own account.";

   //на счет смарта
   Send(context.Smart.Account,Params.Sum,Params.Description);

   //в другой шард отправляем команду на перемещение токена
   SendMessage("TEST:"+Params.Gate,"SendMyToken",Params);
}



//shard2 TEST
"message"
function SendMyToken(Params)
{
   //проверка правильности вызова - из правильного шарда и правильного смарт-контракта
   if (context.Tx.ShardFrom!=="TERA")
      throw "Error cross-shard name: "+context.Tx.ShardFrom;
   if (context.Tx.AccountFrom!==100500)
       throw "Error cross-gate: "+context.Tx.AccountFrom;

   //отправка токена
   Send(Params.To, Params.Sum,Params.Description);
}
```

Планируются методы для работы с Оракулами (нет в обновлении):

OracleRequest(UrlString, ReturnMethodName)
SomeReturnMethod(UrlString, Result)

Оракул запросы должны иметь цифровую подпись…

Запрос состояния из другого шарда (нет в обновлении):

ShardAccountState (ShardAcccountStr);

OnShardAccountState(ShardAcccountStr,РезультатВызова) - событие, которое вызвывается для возвращения результата

Строковый параметр ShardAcccountStr в формате ИмяШарда:НомерСчета

Объект РезультатВызова формате:

```
{
    AccountNum:”uint”,
    Value:{SumCOIN:”uint”,SumCENT:”uint”}
    Data:”arr80”
}
```