Обмен ключами с использованием технологии **blockchain**.

engi@sumus.team

16 июля 2020 г.

1 Постановка задачи

1.1 Термины

- *Консьюмер* — сущность, запрашивающая выполнение вычислительной задачи.
- *Продюсер* — сущность, предоставляющая ресурс для выполнения вычислительной задачи.
- *blockchain* — сетевая база данных, не допускающая изменение уже внесённых записей.

1.2 Процесс

a) *Консьюмер* размещает запись RC_1 запрашивающую выполнение вычислительной задачи.

b) *Продюсер* получает запись RC_1 и размещает запись RP_{A1} о намерении предоставить ресурс для выполнения запрошенной вычислительной задачи.

c) *Консьюмер* получает запись RP_{A1} и размещает запись RC_{A1}, подтверждающую начало выполнения вычислительной задачи.

1.3 Задача

Требуется дополнить содержимое записей RC_1, RP_{A1}, RC_{A1} такими полями, чтобы выполнялись условия:

- Количество шагов обмена должно быть минимальным, желательно не более 3 – по количеству шагов *процесса*.
- В результате *процесса*, *консьюмер* и *продюсер* должны получить одинаковое число, недоступное другим обозревателям *blockchain*.
- Весь обмен при выполнении *задачи* должен происходить через *blockchain*.

2 Предлагаемое решение №1

2.1 Положения

Существует обратимое криптографическое преобразование с ключом, обладающее свойством коммутативности [1]. Обозначим пару шифрование-расшифрование таких преобразований:

$$\begin{aligned} n &\xrightarrow{\mathcal{F}(k_a)} n_a \\ n_a &\xrightarrow{\mathcal{F}^{-1}(k_a)} n \end{aligned} \tag{1}$$

Коммутативностью называется свойство преобразования для которого, при многократном шифровании, для расшифрования не важен порядок обратных преобразований.

$$\begin{aligned} n &\xrightarrow{\mathcal{F}(k_a)} n_a \xrightarrow{\mathcal{F}(k_b)} n_{ab} \\ n_{ab} &\xrightarrow{\mathcal{F}^{-1}(k_b)} n_a \xrightarrow{\mathcal{F}^{-1}(k_a)} n \\ n_{ab} &\xrightarrow{\mathcal{F}^{-1}(k_a)} n_b \xrightarrow{\mathcal{F}^{-1}(k_b)} n \end{aligned} \tag{2}$$

2.2 Алгоритм

a) *Консьюмер* создаёт:
ключ k_a,
число n

b) *Консьюмер* шифрует число n
$n \xrightarrow{\mathcal{F}(k_a)} n_a$,
заносит n_a в RC_1.

c) *Продюсер* создаёт:
ключ k_b

d) *Продюсер* получает число n_a из записи RC_1, шифрует
$n_a \xrightarrow{\mathcal{F}(k_b)} n_{ab}$,
заносит n_{ab} в RP_{A1}.

e) *Консьюмер* получает число n_{ab} из записи RP_{A1}, расшифрует
$n_{ab} \xrightarrow{\mathcal{F}^{-1}(k_a)} n_b$,
заносит n_b в RC_{A1}.

f) *Продюсер* получает число n_b из записи RC_{A1}, расшифрует
$n_b \xrightarrow{\mathcal{F}^{-1}(k_b)} n$

В результате обмена *консьюмер* и *продюсер* получают число n недоступное другим обозревателям *blockchain*.

[1] например, алгоритм RSA