Практическое задание 4.2

**1. Для размещения информации в блок-чейне необходимо сгенерировать ключевую пару RSA** (алгоритм подписи SHA256withRSA)

**2. Нахождение весов нейросети, на которых ошибка не превышает 0,4.**

**3. Формирование данных для размещения в блок-чейне.**

Данные упаковываются в JSON объект, имеющий следующую структуру (веса, ошибка, ваш публичный ключ):

{"w11":"","w12":"","w21":"","w22":"",

"v11":"","v12":"","v13":"","v21":"","v22":"","v23":"",

"w1":"","w2":"","w3":"","e":"","publickey":""}

Для возможности корректно проверить подпись данных они должны быть нормализованы.

Правила нормализации JSON объекта с данными:

1. порядок следования ключей закреплен приведенным выше примером;
2. ключи и значения обрамляются двойными кавычками; пробелы, и другие спецсимволы, кроме разделителей ({},:) не допускаются;
3. представление действительных чисел в 10-ном виде до 12 знаков после запятой, в числах не должно быть незначающих нулей;
4. publickey: представляется в HEX формате.

**3. Подпись данных (JSON объект с данными подписывается как строка)**

**4. Вычисление хеша предыдущего блока.**

Используется хеш-функция SHA-256. Хеш берется от совокупности:  
 **предыдущий хеш** (как массив байт) + **данные** (JSON как строка) + **подпись данных** (как массив байт) + **nonce** (целое 32 битное как массив из 4 байт, нумерация байт слева на право)

*Пример представления целого числа в виде массива байт*

*0 1 2 3*

*00000000 00000000 11001011 00111000*

Хеш предыдущего блока необходим для корректного встраивания в цепочку. Последний блок (или всю цепочку) можно запросить у арбитра.

**5. Формирование блока.**

Блок формируется как JSON объект следующей структуры:

{"prevhash":"...","data":{ … },"signature":"…","nonce":"…"}

prevhash — хеш предыдущего блока (указывается в 16 ричном представлении массива байт)

data — ваши данные

signature — подпись данных, представленных как строка (указывается в 16 ричном представлении массива байт)

nonce — целое 32 битное число, такое, что SHA256(h) ≤ N, где h — хеш формируемого блока, N — 256 битное число у которого первые 12 бит равны нулю, а остальные равны 1.

**6. Отправка блока арбитру для проверки и подписи.**

Блок отправляется POST запросом на ресурс:

[http://itislabs.ru/nbc/newblock/](http://89.108.115.118/nbc/newblock/)

Проследите наличие в заголовке

Content-Type : application/json;charset=UTF-8

При успешной проверке блок добавляется в цепочку и возвращается узлу с проставленной меткой времени и подписью арбитра. Этот блок узел может добавить в свою локальную цепочку.

**7. Подтверждение авторства.**

Блоки в цепочке анонимны. Единственный способ доказать его авторство — подтвердить , что публичный ключ, указанный в блоке, принадлежит вам. Авторство блока закрепляется за владельцем ключевой пары (идентификация по публичному ключу). Для подтверждения авторства необходимо пописать ваши данные вашим же ключем и отправить их арбитру вместе с подписью и открытым ключем.

Отправка POST запросом на ресурс [http://](http://89.108.115.118/nbc/autor)[itislabs.ru](http://89.108.115.118/nbc/newblock/)[/nbc/autor](http://89.108.115.118/nbc/autor)

JSON объекта {"autor":"...","sign":"...","publickey":"…"}

autor — ваши фамилия имя отчество, номер группы

sign — подпись строки с именем

publickey — публичная часть ключевой пары, с помощью которой подписывали блок и имя.

Проследите наличие в заголовке

Content-Type : application/json;charset=UTF-8

После успешной проверки подписи автора блоки будут деанонимизированы.

Проверка блока арбитром состоит из:

1. проверка значения ошибки нейросети по представленным параметрам (с точностью 10-6);
2. проверка валидности подписи данных.
3. на прошедший проверку блок накладывается подпись арбитра, блок помещается в цепочку, данные окончательно сформированного блока возвращаются отправителю.   
   {"prevhash":"","data":{ ... }, "signature":"", "ts":"", "arbitersignature":""}

Если одновременно на обработку попадает 2 блока, то в приоритете тот у которого ошибка меньше, этот блок фиксируется в цепочке.

Запрос узла на получение недостающих блоков (идущих после блока с указанным хешем):

[http://](http://89.108.115.118/nbc/autor)[itislabs.ru](http://89.108.115.118/nbc/newblock/)[/nbc/chain?hash=…](http://89.108.115.118/nbc/autor)

Параметр hash указывается в 16-ричном представлении.

Запрос всей цепочки:

[http://](http://89.108.115.118/nbc/autor)[itislabs.ru](http://89.108.115.118/nbc/newblock/)[/nbc/](http://89.108.115.118/nbc/autor)chain

Возвращается JSON массив из блоков, начиная со следующего, после указанного в запросе

[{"prevhash":"","data":{},"signature":"","ts":"","arbitersignature":""},...]

**Отправленные блоки:**

**http://**[**itislabs.ru**](http://89.108.115.118/nbc/newblock/)**/nbc/blockchain/autors**