ПРИЛОЖЕНИЕ Б

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

Согласовано	**Утверждаю**
Профессор департамента программной инженерии факультета компьютерных наук, канд. техн. наук	Академический руководитель образовательной программы «Программная инженерия» профессор, канд. техн. наук
_____ С.М. Авдошин	_____ В. В. Шилов
"____"_____ 2019 г	"____"_____ 2019 г

Криптографические алгоритмы и протоколы для распределенных реестров

Руководство оператора

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.01 34 01-1

Студент группы БПИ 151 НИУ ВШЭ
_____ Куприянов К. И.
"____"_____ 2019 г

2019

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

УТВЕРЖДЕНО
RU.17701729.04.01 34 01-1

Криптографические алгоритмы и протоколы для распределенных реестров

Руководство оператора

RU.17701729.04.01 34 01-1

Листов 77

2019

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Содержание

1. Назначение программы

1.1. Наименование программы

Наименование программы на русском: “Криптографические алгоритмы и протоколы для распределенных реестров”.
Наименование на английском: “Cryptographic Algorithms and Protocols for Distributed Ledgers”.

1.2. Краткая характеристика

Программа предназначена для пользователей машин на семействе ОС GNU/Linux. Цель работы — создать удобное приложение для автоматизации программирования, которое генерировало бы готовый код блокчейна с использованием алгоритмов, выбранных пользователями.

Данный продукт будет служить “инструментарием” для программиста или любого другого интересующегося криптографическими алгоритмами и протоколами, который имел бы потребность интегрировать блокчейн в своё приложение (регистрация гостей в отеле, социальную сеть, переводы, учёт документов). Так же программа будет полезна людям, которые хотят узнать как работают современные распределённые реестры с рассмотренными аспектами. Это позволит быстро получать необходимую техническую информацию, которую с трудом можно найти в общем доступе. Программа должна предоставлять не только генерацию кода, но и дружелюбный интерфейс командной строки, в которой форматирование и подсветка не будут сбивать с толку неподготовленного пользователя.

Главной чертой данного приложения является самоподдерживаемая система по работе с исходными кодами алгоритмов, расположенными удалённо. А так же лёгкая, быстрая масштабируемость и модульность программного кода.

Приложение состоит из двух компонент:

1. Позволяющей сгенерировать код блокчейна с использованием выбранных пользователем алгоритмов
2. Является выходом первой компоненты, и по своей сути обособленным приложением — блокчейном

В дальнейшем (1) будет именоваться **компоновщик**, а (2) — **реализация блокчейна**.

2. Условия использования программы

2.1. Минимальные параметры технических средств

Для оптимальной работы приложения необходимо учесть следующие системные требования:

1. Персональный компьютер со следующими минимальными требованиями:
   (a) Операционная GNU/Linux версии ядра 4.15.0-47-generic и выше
   (b) 64-разрядный (x64) процессор
   (c) 1ГБ оперативной памяти (ОЗУ)
   (d) 100 МБ свободного места на внутреннем накопителе
2. Интерпретатор Python3.6.5 и выше

2.2. Численность и квалификация персонала

Минимальное количество персонала, требуемого для работы программы: 1 оператор. Пользователь данного программного продукта должен разбираться в командной строке (shell) GNU/Linux, иметь базовые навыки в командах, уметь устанавливать и удалять программы, запускать их. Перед использованием программы пользователь должен быть заранее проинструктирован и уведомлен о составе выполняемых функций и других характеристиках приложения, а так же сопровождён необходимой технической документацией.

3. Выполнение программы

3.1. Загрузка файлов

В данном разделе будет показано, как устанавливать программу и пользоваться ей. Для описания каждого проделанного шага будут включены описание действий и скриншоты приложения.

3.1.1. Установка программы

Дистрибутив данной программы можно будет получить с прилагаемого CD диска, либо по ссылке, считав ее с прилагаемого qr-кода(рис. 1).

Рис. 1: ссылка на программу

Чтобы начать испытания выполнения требований к функциональным характеристикам, необходимо запустить установщик программы путем выполнения инструкций (Рис. 2), написанных в репозитории данного приложения (ссылка доступна по qr коду 1).

3.1.2. Использование приложения компоновщик

Далее приложение необходимо запустить. Запускается компоновщик при помощи команды

$$\text{gsl --init --name myledger --path } \tilde{}/\text{tmp/gsl}$$

где *myledger* — название будущего блокчейна, а */tmp/gsl* — путь до создания директории с блокчейном.

Installation

```
$    cd gsl
$    export PYTHONPATH=$PYTHONPATH"$(pwd)/src"
$    pip3 install . --user
```

Then, to successfully launch app, it is needed to have a config with path `/etc/gsl/config.yaml`

Example `config.yaml` :

```
# Example and debug configuration
#
init_dir: /home/coldmind/Projects/gsl/result # At the moment, ANY directory will be OK
```

Рис. 2: Инструкция по установке программы

Приложение отобразит приветственное сообщение с вариантами выбора алгоритмов (пометка 1 на Рис. 3). Варианты подсвечены в зависимости от степени поддержки алгоритмов программой (пометка 3 на Рис. 3). Так же есть возможность выбора значения по умолчанию, не вводя ничего (пометка 2 на Рис. 3)

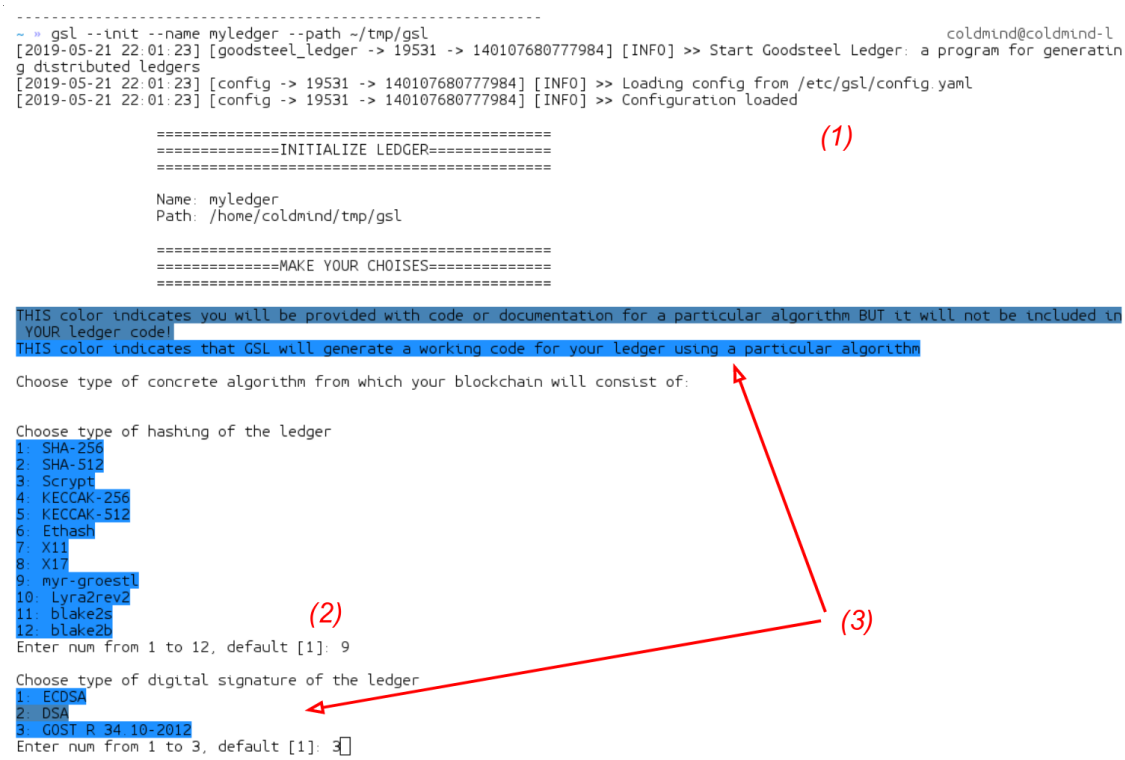

Рис. 3: Начало работы компоновщика

После выбора алгоритмов хэширования и цифровой подписи, пользователю показываются свойства/структура/другие алгоритмы распределённых реестров (Рис. 4), по которым можно получить справочную информацию (Рис. 5)

```
Now, choose related themes for which you will be provided with relevant information (links, web sites, etc.)


Option: structure of the ledger
1: Blockchain
2: DAG
3: Hashgraph
4: Holochain
5: Tempo
Enter num from 1 to 5, default [1]: 4

Option: openess of the ledger
1: Public
2: Private
Enter num from 1 to 2, default [1]: 1

Option: consensus of the ledger
1: PoW
2: PoS
3: DPoS
4: PoA
5: PoWeight
6: BFT
Enter num from 1 to 6, default [1]: 5

Option: random of the ledger
1: DRBG
2: CPRNG
Enter num from 1 to 2, default [1]: 2


The following config is to be set:
```

Рис. 4: Вывод опций по которым будет дана справочная информация

```
Holochain:
    - https://github.com/holochain/holochain-rust
Public:
    - Depends on your implementation: https://masterthecrypto.com/public-vs-private-blockchain-whats-the-difference/
PoWeight:
    - Read https://filecoin.io/filecoin.pdf
X17:
    - https://pypi.org/project/x17_hash/
CPRNG:
    - https://riptutorial.com/python/example/3857/create-cryptographically-secure-random-numbers
GOST R 34.10-2012:
    - https://pypi.org/project/pygost/
```

Рис. 5: Справочная информация в конце выполнения компоновщика

После завершения работы программы по указанной директории располагаются модули wallet.py и miner.py вместе с выбранными алгоритмами хэщирования и электронной подписи (Рис. 6).

```
~/tmp/gsl/myledger » ll                                                    coldmind@coldmind-l
total 32K
-rw-rw-r-- 1 coldmind coldmind  14K May 21 22:37 miner.py
-rw-rw-r-- 1 coldmind coldmind 2.3K May 21 22:37 mydss.py
-rw-rw-r-- 1 coldmind coldmind  291 May 21 22:37 myhashing.py
-rw-rw-r-- 1 coldmind coldmind 6.5K May 21 22:37 wallet.py
```

Рис. 6: директория со сгенерированным кодом

3.1.3. Использование приложения реализация блокчейна

Проведём сценарий использования приложения реализация блокчейна. Сгенерируем 2 кошелька, отправим с одного адреса на другой 13 единиц условной валюты

Запустить код кошелька необходимо при помощи команды python3 wallet.py и затем выбрать первую опцию. При выборе первой опции должен отображаться диалог с требованием ввести имя, и дальнейшей генерацией адреса кошелька (пары публичный-приватный ключи) (Рис. 7- -8)

```
------------------------------------------------------------
~/tmp/gsl/myledger » python3 wallet.py

            Which action would you like to take?
            1. Generate new wallet
            2. Send coins to another wallet
            3. View transactions

            1
========================================

IMPORTANT: save this credentials or you won't be able to recover your wallet

========================================

Write the name of your new address: kirill
Your new address and private key are now in the file kirill.txt
Repeat? Would you like one more action? (Y/[N])
Exiting..
------------------------------------------------------------
```

Рис. 7: Генерация адреса kirill

```
------------------------------------------------------------
~/tmp/gsl/myledger » python3 wallet.py

            Which action would you like to take?
            1. Generate new wallet
            2. Send coins to another wallet
            3. View transactions

            1
========================================

IMPORTANT: save this credentials or you won't be able to recover your wallet

========================================

Write the name of your new address: julia
Your new address and private key are now in the file julia.txt
Repeat? Would you like one more action? (Y/[N])
Exiting..
------------------------------------------------------------
```

Рис. 8: Генерация адреса julia

Таким образом, были сгенерированы 2 адреса кошельков (Рис. 9) Теперь можно

```
------------------------------------------------------------
~/tmp/gsl/myledger » cat kirill.txt                                    coldmind@coldmind-l
Private key: 28a0bc04a8c2c58df31a98779dd97de02529da91dc01c1526729525d7252ac47
Wallet address / Public key: staF5xuOTml3Bmoq+IrveFQ+Au/kD60C8TpZU/bzM2/AER6VoNOep+hrwg/DuLGwvfUkDfBNXTCkMesqDYvWLA==%
------------------------------------------------------------
~/tmp/gsl/myledger » cat julia.txt                                     coldmind@coldmind-l
Private key: bc28c8c50e56eb089d9fce56f4275671b66138ee2aec61ad22a40f7e59951a86
Wallet address / Public key: /2s/qsDSS7F0L+XN15qvG2cXHamTLMb8ipNFWycK5nOWY1t7TqcK1JU2Ayt2p/i8jPraPRd8th9N8tjeeT/jag==%
------------------------------------------------------------
~/tmp/gsl/myledger »                                                   coldmind@coldmind-l
```

Рис. 9: Адреса кошельков

приступать к отправке условных средств с одного адреса на адрес другой. В другом окне запустить майнер командой python3 miner.py, и оставить его исполнение.

При запуске майнера, должен вестись лог о проведённых транзакциях и их валидациях (Рис. 10 - Рис. 12).

В кошельке для отправке условных средств с одного счёта на другой, необходимо выбрать вторую опцию и ввести публичный и приватный адреса отправителя, а так же публичный ключ получателя (Рис. 11). Подтвердить намерение отправить и осуществить тем самым отправку.

Теперь, можно проверить весь блокчейн на предмет совершённой транзакции. При выборе третей опции в кошельке, должен отобразиться полная цепочка транзакций (блокчейн) (Рис. 13).

На этом работу программы можно считать завершённой.

```
~/tmp/gsl/myledger » python3 miner.py                                                          coldmind@coldmind-l
 * Serving Flask app "miner" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [17/May/2019 12:50:06] "GET /txion?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
{"index": 1, "timestamp": "1558086606.8309455", "data": {"proof-of-work": 71271, "transactions": [{"from": "network", "to": "q3nf394hjg-random-miner-address-3
4nf3i4nflkn3oi", "amount": 1}]}, "hash": "820539ad1af5001742702ea099cb7e33e30e122b61c108fcc102bbc10ccc0301"}

127.0.0.1 - - [17/May/2019 12:50:06] "GET /blocks?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
127.0.0.1 - - [17/May/2019 12:50:06] "GET /txion?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
{"index": 2, "timestamp": "1558086606.8755133", "data": {"proof-of-work": 142542, "transactions": [{"from": "network", "to": "q3nf394hjg-random-miner-address-
34nf3i4nflkn3oi", "amount": 1}]}, "hash": "bf531e792069f8909969969d7f870eb9ed49bd167c6ec35c1a75f79335139161"}

127.0.0.1 - - [17/May/2019 12:50:06] "GET /blocks?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
127.0.0.1 - - [17/May/2019 12:50:06] "GET /txion?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
{"index": 3, "timestamp": "1558086606.9611478", "data": {"proof-of-work": 285084, "transactions": [{"from": "network", "to": "q3nf394hjg-random-miner-address-
34nf3i4nflkn3oi", "amount": 1}]}, "hash": "2cd997bd4563ba997e140dd38a4c4020cdc832abc1cdd5acd3a605f8a70737b8"}

127.0.0.1 - - [17/May/2019 12:50:06] "GET /blocks?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
127.0.0.1 - - [17/May/2019 12:50:07] "GET /txion?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
{"index": 4, "timestamp": "1558086607.1272025", "data": {"proof-of-work": 570168, "transactions": [{"from": "network", "to": "q3nf394hjg-random-miner-address-
34nf3i4nflkn3oi", "amount": 1}]}, "hash": "54f0d961ed8a2043d91716647e5108467cdd6e11ed502da33bdc734d1ff66c3a"}

127.0.0.1 - - [17/May/2019 12:50:07] "GET /blocks?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
127.0.0.1 - - [17/May/2019 12:50:07] "GET /txion?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
{"index": 5, "timestamp": "1558086607.4534817", "data": {"proof-of-work": 1140336, "transactions": [{"from": "network", "to": "q3nf394hjg-random-miner-address
-34nf3i4nflkn3oi", "amount": 1}]}, "hash": "1ae23f6dc5419a0579aec2ff8f2e09c92f6cfde1f49cfe2693deaab01024c731"}

127.0.0.1 - - [17/May/2019 12:50:07] "GET /blocks?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
127.0.0.1 - - [17/May/2019 12:50:08] "GET /txion?update=q3nf394hjg-random-miner-address-34nf3i4nflkn3oi HTTP/1.1" 200 -
{"index": 6, "timestamp": "1558086608.120859", "data": {"proof-of-work": 2280672, "transactions": [{"from": "network", "to": "q3nf394hjg-random-miner-address-
34nf3i4nflkn3oi", "amount": 1}]}, "hash": "bc2e393cb4ffe5d12e832ad879153266e98fe468767d6dcd964f95c045afb11c"}
```

Рис. 10: Лог работы майрена

```
------------------------------------------------------------
~/tmp/gsl/myledger » python3 wallet.py                                    coldmind@coldmind-l

            Which action would you like to take?
            1. Generate new wallet
            2. Send coins to another wallet
            3. View transactions

            2
From: introduce your wallet address (public key)
staF5xu0Tml3Bmoq+IrveFQ+Au/kD60C8TpZU/bzM2/AER6VoN0ep+hrwg/DuLGwvfUkDfBNXTCkMesqDYvWLA==
Introduce your private key
28a0bc04a8c2c58df31a98779dd97de02529da91dc01c1526729525d7252ac47
To: introduce destination wallet address
28a0bc04a8c2c58df31a98779dd97de02529da91dc01c1526729525d7252ac47
Amount: number stating how much do you want to send
13
=========================================


Is everything correct?

From: staF5xu0Tml3Bmoq+IrveFQ+Au/kD60C8TpZU/bzM2/AER6VoN0ep+hrwg/DuLGwvfUkDfBNXTCkMesqDYvWLA==
Private Key: 28a0bc04a8c2c58df31a98779dd97de02529da91dc01c1526729525d7252ac47
To: 28a0bc04a8c2c58df31a98779dd97de02529da91dc01c1526729525d7252ac47
Amount: 13

y/n
y
Transaction submission successful

Repeat? Would you like one more action? (Y/[N])
Exiting..
------------------------------------------------------------
```

Рис. 11: Процесс отправки средств

4. Приложение 1. Терминология

```
127.0.0.1 - - [21/May/2019 23:57:47] "GET /blocks?update=k4Odf238gn-random-dkfi3-address-k394rb
gfGKe392f HTTP/1.1" 200 -
b'\xc6&8b\x90\x1c\xcf\xb2\x0f>\xa5\x8d\x0e\xf4g\xe5\xd2\xec\x8d{\xac\x0f\x18\xbeJ\xfeo\x96,)\xe
9\xd6\x11\xe4\xe0KM\xf3\xdf\xd00\xa1d\x8cn9!\x9a\xb8\xce\xee\x8d\x19\xa3\xaa\x04\xf59\x142U\x8f
T\xef\xe3R\xcf\x08\xd0\x93]\x8cf\xe9B\x93\xb5\x7fOZ\x86\xaa\xf7\xec\xc6\xe0\xb2\xc5\xb9R\xf2R\x
04\xdf\x9d\\LDH\x93\xee\x1c\xbb!\xf6J-kH\xec9:I\xa6\xfb\xb3\xd6\xd3)34\x87\x13z2\x81I\x1f'
<class 'bytes'>
New transaction
FROM: xiY4YpAcz7IPPqWNDvRn5dLsjXusDxi+Sv5vliwp6dYR5OBLTfPf0E+hZIxuOSGauM7ujRmjqgT1ORQyVY9U7+NSz
wjQk12MZulCk7V/T1qGqvfsxuCyxblS8lIE351cTERIk+4cuyH2Si1rSOw5Okmm+7PW0ykzNIcTejKBSR8=
TO: Ld/aIVziUKTZ2rzVnuzAW14RxQDMDtYdzURYOVOif5b1dxrGZ286dxr6rmM0XdfBNeWLSJZHzeMhxnHKb4L8zk7shUX
l0jOo/D0ij0VeQq0KOUULiLS92AAWKPQCBURf3Br/X4I8M0E0lWr+ctJGBoRAiBdGcDpV6xQz4CP0cfE=
AMOUNT: 13

127.0.0.1 - - [22/May/2019 00:29:02] "POST /mycoin HTTP/1.1" 200 -
```

Рис. 12: Лог регистрации новой транзакции

```
------------------------------------------------------------
~/tmp/gsl/myledger » python3 wallet.py                                    coldmind@coldmind-l

            Which action would you like to take?
            1. Generate new wallet
            2. Send coins to another wallet
            3. View transactions

            3
[{"index": "0", "timestamp": "1558468861.6067605", "data": "{'proof-of-work': 9, 'transactions': None}", "hash": "0b9cde7957ad
fce769df95dabb02a2ba60fe08c772c2ce49a7c65c77f5c9e161"}, {"index": "1", "timestamp": "1558468861.6604264", "data": "{'proof-of-
work': 71271, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "
hash": "640841869b994c0b01f077bdedba4a05ba17829219bf220b0fa21b7ff4122eff"}, {"index": "2", "timestamp": "1558468861.7079172",
"data": "{'proof-of-work': 142542, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f
', 'amount': 1}]}", "hash": "e3f92d89609ba95a67a6cf08fe21941acb4f6e92b36ad2c458d29a472a53f41f"}, {"index": "3", "timestamp": "
1558468861.7935686", "data": "{'proof-of-work': 285084, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-ad
dress-k394rbgfGKe392f', 'amount': 1}]}", "hash": "5cb2451bb5a9c095e762a7bdf8c410f412cb90bba08110adee88f75d4b8b29b8"}, {"index"
: "4", "timestamp": "1558468861.9601207", "data": "{'proof-of-work': 570168, 'transactions': [{'from': 'network', 'to': 'k4Odf
238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "015f83ca2fcf2502f5b4dc9eb36f0e9acc05c2c1d51daa3cdc2df54
f9da4b4f0"}, {"index": "5", "timestamp": "1558468862.2882986", "data": "{'proof-of-work': 1140336, 'transactions': [{'from': '
network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "3ac6afffcaab7f049c7865720f3445376
59508afb3582c642243154dc929eb09"}, {"index": "6", "timestamp": "1558468862.9600003", "data": "{'proof-of-work': 2280672, 'tran
sactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "2fd58ac1580
0945a0695f47be4485d54f0e1db3c4d0d71a3d89f47930195311c"}, {"index": "7", "timestamp": "1558468864.1199691", "data": "{'proof-of
-work': 4561344, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}"
, "hash": "5c257e176bb399c57a644b5cd38118d8c48fdb7046b9ccd95bedb61f84518a47"}, {"index": "8", "timestamp": "1558468865.952813"
, "data": "{'proof-of-work': 9122688, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe3
92f', 'amount': 1}]}", "hash": "d5303b53bcbd7ec433dae6101569bba83097a31740f720a96df7f220466d155b"}, {"index": "9", "timestamp"
: "1558468869.1114795", "data": "{'proof-of-work': 18245376, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkf
i3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "ff27888d7e4df6a553ca1c5099c4fcbbc8d33696c0c68af192f73b511872a988"}, {"i
ndex": "10", "timestamp": "1558468875.0238767", "data": "{'proof-of-work': 36490752, 'transactions': [{'from': 'network', 'to'
: 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "6358863a0819dd6a6bb2e6cc6e24ce18e188c033491ccc9
920d5c9cc4574c574"}, {"index": "11", "timestamp": "1558468886.5762112", "data": "{'proof-of-work': 72981504, 'transactions': [
{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "74bab54e0b8682f81c1c2cc
428534ce6edb4e0427de1af7115404a97960de363"}, {"index": "12", "timestamp": "1558468909.7597685", "data": "{'proof-of-work': 145
963008, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash":
 "67d61bb8ea2dd49c91724affb0ea56c99bb2f32764efa5bcffe8068bf12ee8b5"}, {"index": "13", "timestamp": "1558468955.0765815", "data
": "{'proof-of-work': 291926016, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f',
 'amount': 1}]}", "hash": "49f6b3c4ec5b9679c245da0fb092e9277896a5cc3d77c001031ee3439ad8f629"}, {"index": "14", "timestamp": "1
558469049.0494485", "data": "{'proof-of-work': 583852032, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-
address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "34a97447dd030dd42f945d6a11435f0bbbaeaec45e344796113aeee0b14bee66"}, {"inde
x": "15", "timestamp": "1558469245.1669443", "data": "{'proof-of-work': 1167704064, 'transactions': [{'from': 'network', 'to':
 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "a78903fc2ab59c634ea1a8418436f5d1529702182adee71c
18d9602053c9b271"}, {"index": "16", "timestamp": "1558469669.0417874", "data": "{'proof-of-work': 2335408128, 'transactions':
[{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash": "696cfb1d98910b6ad6cd1c
5a1a0eabf248485960 76c8ab57605cdd093bcf8abd"}, {"index": "17", "timestamp": "1558470528.4890237", "data": "{'proof-of-work': 46
70816256, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392f', 'amount': 1}]}", "hash
": "f4f46c9bc8b4771acf94d68c66c9aed212e487eca4115e69c79f0cf871fe9b58"}, {"index": "18", "timestamp": "1558472267.4413888", "da
ta": "{'proof-of-work': 9341632512, 'transactions': [{'from': 'network', 'to': 'k4Odf238gn-random-dkfi3-address-k394rbgfGKe392
f', 'amount': 1}]}", "hash": "7cee2cb4fd3580dd201426201ba93b7a951a72385fe912f6bcff6cf38a433657"}]
Repeat? Would you like one more action? (Y/[N])
Exiting..
------------------------------------------------------------
```

Рис. 13: Отображение полной цепочки транзакций

4.1. Терминология

Распределённый реестр (Distributed Ledger) — В примитивной своей реализации это распределённая база данных между сетевыми узлами или вычислительными устрой-

ствами. Каждый из узлов может получать данные других, при этом храня полную копию реестра. Обновления этих узлов происходят независимо друг от друга.

Блокчейн — Постоянно растущий список записей, называемых блоками, которые связаны и защищены с помощью криптографии. Он копируется его пользователями и устойчив к модификации. Машина с рабочей копией называется узлом.

DAG — Направленный ациклический граф. Это ориентированный граф с данными, использующий топологическую сортировку (от ранних узлов к более поздним).

Биткоин (Bitcoin) — Электронная пиринговая платёжная система, используемая в качестве финансовой единицы (криптовалюты) одноимённую сущность. Создателем биткоина выступает некто Satoshi Nakomoto [1].

Эфириум (Ethereum) — Открытая, общедоступная, вторая по популярности, распределенная вычислительная платформа на основе технологии блокчейн и операционная система с функциональностью смарт-контрактов [2]

Алгоритм консенсуса — Набор математических операций, которые необходимо выполнять для поддержания консистентности всей сети.

5. Приложение 2. Список используемой литературы

Список литературы

1. *Nakamoto S.* Bitcoin: A Peer-to-Peer Electronic Cash SyNakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Consulted, 1–9. // Journal for General Philosophy of Science. — 2008. — № 1. — С. 1—9. — ISSN 09254560. — DOI: 10.1007/s10838-008-9062-0. — arXiv: 43543534534v343453.
2. *Vitalik Buterin*. On Public and Private Blockchains. — 2015. — URL: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/ (дата обр. 22.04.2019).
3. *Документации Е. С. П.* ГОСТ 19.101-77 Виды программ и программных документов. — ИПК Издательство стандартов, 2001.
4. *Документации Е. С. П.* ГОСТ 19.102-77 Стадии разработки. — ИПК Издательство стандартов, 2001.
5. *Документации Е. С. П.* ГОСТ 19.103-77 Обозначения программ и программных документов. — ИПК Издательство стандартов, 2001.
6. *Документации Е. С. П.* ГОСТ 19.104-78 Основные надписи. — ИПК Издательство стандартов, 2001.
7. *Документации Е. С. П.* ГОСТ 19.106-78 Требования к программным документам. — ИПК Издательство стандартов, 2001.
8. *Документации Е. С. П.* ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. — ИПК Издательство стандартов, 2001.
9. *Документации Е. С. П.* ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. — ИПК Издательство стандартов, 2001.
10. *Документации Е. С. П.* ГОСТ 19.603-78 Общие правила внесения изменений. — ИПК Издательство стандартов, 2001.