ISSN 0005-1055, Automatic Documentation and Mathematical Linguistics, 2018, Vol. 52, No. 5, pp. 265–273. © Allerton Press, Inc., 2018.
Original Russian Text © S.N. Grinyaev, R.A. Zlotin, A.I. Milushkin, D.I. Pravikov, I.A. Selionov, A.Yu. Shcherbakov, Yu.N. Shchuko, 2018, published in Nauchno-Tekhnicheskaya Informatsiya, Seriya 2: Informatsionnye Protsessy i Sistemy, 2018, No. 10, pp. 00000–00000.

INFORMATION SYSTEMS

On the Creation of a Universal Protected Trusted Digital Asset (Token)

S. N. Grinyaev[a, *], R. A. Zlotin[b, **], A. I. Milushkin[c, *], D. I. Pravikov[a, d, ****], I. A. Selionov[c, *****], A. Yu. Shcherbakov[e, ******], and Yu. N. Shchuko[e, *******]**

[a] *Gubkin Russian State University of Oil and Gas (National Research University), Moscow, Russia*
[b] *Moscow Region Chamber of Lawyers, Moscow, Russia*
[c] *Management Company of the Technopark Pushkino Project, Pushkino, Moscow Region, Russia*
[d] *Institute of Complex Security and Special Instrument Engineering, Moscow Technical University (MIREA), Moscow, Russia*
[e] *All-Russian Institute of Scientific and Technical Information (VINITI), Russian Academy of Sciences, Moscow, Russia*

**e-mail: gsn@gubkin.pro*
***e-mail: rzlotin@yandex.ru*
****e-mail: mialiv@techprom.biz*
*****e-mail: d_pravikov@mail.ru*
******e-mail: uk@techprom.biz*
*******e-mail: x509@ras.ru*
********e-mail: dir@viniti.ru*

Received July 3, 2018

Abstract—In this paper, we address the problem of developing a next-generation universal protected trusted digital asset based on a universal data structure and a new trust and security paradigm.

Keywords: trust, token, blockchain, authentication code, distributed data storage, information security, auction, encryption, integrity control, smart contract, token development tools, digital financial asset, transaction, and platform operator

DOI: 10.3103/S0005105518050059

INTRODUCTION

The problem of the "digital transformation" of the Russian economy, including the formation of a trusted and consistent digital environment for the development and application of national digital technologies, is of great importance. This field is under a great deal of "pressure" from foreign cryptocurrency technologies, including the technologies that provide circulation of tokens and digital assets, distributed data storage, and distributed ledgers (blockchains). In addition, active attempts are made to create real systems for a wide range of problems based on distributed ledgers and various cryptocurrencies. However, a number of studies have shown that despite their declared qualities these systems do not possess information-security properties, which determines their exploratory character.

The uncritical copying of the Western technologies can have detrimental consequences for the Russia's technological security and independence, especially when building a digital economy, including the effect on the continuity of Russian business both at the national and international levels [1]. Therefore, it is essential to create a technological platform for the digital transformation that meets the following requirements:

- trust (open universal structure and open source code) [2];
- national cryptographic localization (use of national standards with allowance for the corresponding certificates received from the regulators);
- universality (the platform should be applicable in the digital asset technology and management of goods and services, including government procurement).

As a result of the intensive investigation of distributed ledger technologies by Russian specialists, various scientific teams proposed their own developments. In this investigation, the representatives of academic and university science joined their efforts with the leading tech companies. This work is a result of theoretical studies and their prototyping that, given adequate support, make it possible to implement original Russian information technologies that can facilitate the development of the digital economy in the Russian Federation.

1. LEMMA ABOUT THE SECURITY OF SUBJECT–OBJECT SYSTEMS

It was found that, despite the assurances of various developers, the introduction of the blockchain technology engenders new types of threats, which were discussed in some detail in [3] and on the Forklog website. As an example, we can consider the following problem with a smart contract, which was described in various publications. At some point in the execution of a smart contract, for further computations, it becomes necessary to obtain the current US dollar exchange rate. Technically, this is done through the Oraclize, which accesses the corresponding site that is considered a trusted source. If this site was taken by malefactors, then they become capable of committing various attacks on the system, beginning with the inaccessibility of the site and ending with the manipulation of the US dollar exchange rate.

Informally, the problem is as follows: while the smart contract operates with the data and other elements within the blockchain platform (e.g., operations with a cryptocurrency), we can discuss definite security. However, if we try to expand the application domain of distributed ledgers, then the inevitable communication with the external environment creates various security threats.

Formally, the statement about the insecurity of communication with external data sources can be formulated as the following lemma.

Suppose that we have a system B that includes n subjects and m objects:

$$B = (S,O) : |S| = n,\ |O| = m.$$

For this system, access isolation is given by the Cartesian product of the sets of subjects and objects by a set of access rights:

$$(S \times O) \to R.$$

This mapping fully describes the capabilities of the set of subjects with respect to the set of objects (in terms of the problem, this mapping closes the system). The system described above is considered secure if the mapping possesses two basic properties: completeness and consistency. Completeness means that an access right is set for each pair (subject–object), while consistency means that, when assigning access rights, there is no pair (subject–object) for which a particular access right is simultaneously prohibited and allowed.

We suppose that a subject S_i that has certain access rights R_{ik} with respect to an existing object O_k is generated. In the system B, the subject S_i can be generated from a source object O_j by another subject S_e (e.g., an executor atom or other execution environment of the smart contract).

Here, there are three possible cases.

1. The subject S_i coincides with an existing subject ($i \le n$) and its access rights coincide with the existing rights; this is a trivial tautological case.

2. The subject S_i coincides with an existing subject ($i \le n$) but its access rights do not coincide with the existing rights; in this case, the consistency property is violated.

3. The subject S_i does not coincide with existing subjects ($i > n$) and its access rights are set only for the object O_k; in this case, the completeness property is violated.

Thus, the appearance of an additional active entity in the closed system causes the violation of its security. Therefore, the majority of the blockchain systems that provide the circulation and use of cryptocoins cannot yet be considered secure.

2. OURCOIN: A TRUSTED DIGITAL ASSET

To demonstrate some approaches to the development of distributed ledger technologies with information security properties, it was proposed to create a protected cryptocurrency called OURCOIN.

OURCOIN is a digital financial asset, a digital property created by the OURCOIN platform with the use of cryptographic means, which is a measure of value recognized as such by an unlimited number of legal entities.

Below, we define the terms used to describe the operation of the system.

OURCOIN token (digital financial asset) is a digital property created by the OURCOIN platform with the use of cryptographic means that certifies the extent of the rights granted by the OURCOIN project's offer to its owner. The token is fixed in digital form (as a special file or blockchain link/atom), assigned a unique ID number, and characterized by a nominal value, creation time, lifetime (validity time), and owner's network name; it also allows its parameters to be checked for integrity.

Participant of the OURCOIN digital transaction register (for short, the participant; can also be referred to as the subscriber, user, or client) is a physical or juridical person that passed validation (identification) in the OURCOIN platform, received a unique network name, and conducts digital transactions in accordance with the rules of the digital transaction register.

Network name is a unique sequence of 32 hexadecimal digits (for a 16-byte record) that is assigned to the participant and is associated with his or her private key.

A participant's private key is a random number generated during the first or repeated registration of the participant that is used to generate (fix) and check authentication codes for the tokens/cryptocurrency.

The authentication code (AC) of the token/cryptocurrency is the information embedded into the token/cryptocurrency that allows its parameters to be checked for integrity and restored if they were damaged or intentionally modified. Below, we also refer to the AC as the signature. The AC (signature) can be private (generated by the user), transport (used to transfer tokens among the participants, as well as to check the received tokens for integrity), and root (generated by the transaction registration center).

The token generation center (TGC) is a juridical person that uses unique software of the OURCOIN platform to generate impersonal primary tokens characterized only by their ID numbers, nominal values, and lifetimes.

The transaction registration center (TRC) is a juridical or physical person, a participant of the digital transaction register, that performs activities on validating and registering digital records in the digital transactions register in accordance with the rules of the digital transaction register by assigning ACs to tokens/cryptocurrency and sending them to users.

The TRC is a prototype authorized operator of the investment platform, which is mentioned in the digital economy bills adopted by the State Duma of the Russian Federation in the first reading (in particular, the draft federal law "On alternative ways of attracting investments (crowdfunding)"). In the future, the TRC can serve as an operator of the platforms that provide government services, notarial services, Russian cryptocurrency circulation, and judicial protection (in the Russian Federation) of the transactions conducted in the distributed ledger.

Transaction is a civil deal for transferring the digital rights to the token/cryptocurrency and/or its part that is made by the participant/participants. Technically, the transaction is a system process (represented in material form) of transferring the entire nominal value of the token, or its part, from one subscriber to another. The transaction is characterized by its initiator, recipient, and (in terms of its size) by transfer, change, and commission. In this case, the nominal amount of the transfer, change, and commission is always equal to the nominal value of the original token.

The token/cryptocurrency circulation system is a system that consists of subscribers, TGC, and TRC; it provides the life cycle of the tokens in material form.

The life cycle of the token is a period of time set by the OURCOIN platform from the instant of its generation to its termination.

From a technical perspective, the life cycle of the token includes the following elementary components: generation, transfer to the user, use in transactions, and termination. As components, the life cycle also includes merging and splitting of tokens.

The termination of tokens does not pose a financial threat to the clients and does not destroy the measures of value; on the contrary, it enables the technical properties of the platform by analogy with the issue and circulation of paper currency.

3. TYPES OF TOKENS IN THE SYSTEM

Prototoken 1 is a part of the token that is generated by the TGC and is characterized only by its ID number, nominal value, and lifetime. It cannot be used in transactions because it does not have an owner.

Prototoken 2 is a part of the token that is characterized by the participant's name, TRC AC, and transport AC (the latter makes it possible to check the token for integrity when it is received by the subscriber).

A **token** has the properties of prototokens 1 and 2 and is characterized by an AC and a personal signature of its owner. Only the token is used in transactions.

Thus, prototoken 2 is obtained when generating a TRC AC and transport AC for prototoken 1, while the token is obtained when generating a private AC for prototoken 2.

A **commission token** is a token whose TRC AC and owner's code coincide; it is generated in the transactions for which the TRC charges a commission.

A **zero token** is a token generated when the change returned from the transaction is zero.

An **exchange token** is a token used for exchange operations with other digital assets; it is characterized by the integral value transferred by the subscriber owner to the subscriber broker or subscriber exchange that does not exceed the aggregate value of the tokens belonging to this owner; it can also be used to clear exchange operations in a specified period of time.

4. STRUCTURE OF THE TOKEN

The token is fixed-length (128 bytes) protected data. The structure of the data that constitute the token is shown in Table 1.

5. DESCRIPTION OF A VARIANT OF DIGITAL ASSET CIRCULATION

1. The token generation center (TGC) generates prototoken 1, which is the beginning of a transaction chain, and sends it to the transaction registration center (TRC).

2. The TRC assigns it a root signature and transport signature and then sends it (prototoken 2) to the user.

3. The user checks the transport signature of prototoken 2, signs it with his or her personal signature, and returns the token to the TRC.

4. The TRC saves the token in its database and simultaneously sends it to the TGC for archiving.

5. In the case of payment, the user creates a transaction of four candidate links: original coin, change, transfer, and commission. All these links are signed

Table 1. The structure of the token data

Index from the beginning of the token, bytes	Field length, bytes	Field	Comment
0	4	Header	
4	8	Token ID	ID number of the token generated by the TGC
12	16	Checking information E(Mi,Kt)	It is the token's ID number, creation date, and nominal value (Mi) encrypted on the token key (Kt) stored in the TGC
28	4	Type of the token	Currently, the system uses the token of type 1
32	4	Integer nominal value of the token (rubles)	
36	4	Fractional nominal value of the token (kopecks)	
40	4	Generation parameter	In this project, it is 256
44	4	Token creation date in the format (dd)(mm)(yy-2000) (3 bytes, each bracket is 1 byte)	
48	4	Token validity date in the format (dd)(mm)(yy-2000) (3 bytes, each bracket is 1 byte)	
52	4	Hierarchy level of the token	For prototokens 1 and 2, it is zero; for the token, it is one and is incremented by one with each transaction
56	8	Reserve field of the token's body	
64	16	Network name of the owner	
80	8	TGC authentication code	
88	8	Transport AC	
96	8	Owner's AC	
104	24	Reserve field for AC	

with the TRC transport signature, while the change is also signed with the personal signature of the payer.

6. The TRC checks all four links for correctness (coincidence of the sum, correctness of the transport signature and transaction number, as well as presence of the first link in the database); the TRC then sends a request to the TGC for the generation of two new prototokens with the sum of change and transfer whose ID numbers coincide with the ID number of the first link.

7. The TGC generates new prototokens 1 for the transfer and change and sends them to the TRC.

8. The TRC signs the transfer (prototoken 1) with the root and transport signatures and sends it to the payee; it also signs the change with the root and transport signatures and sends it to the payer.

9. The payer and payee sign the change and transfer (prototokens 2), respectively, with their personal signatures and then return the tokens to the TRC.

The transaction is complete. Operations with new tokens can then be carried out.

Each token always has three signatures (root, transport, and personal); prototokens and tokens with defect validity features (signature or AC) are not used in transactions.

6. MECHANISMS FOR THE TECHNICAL IMPLEMENTATION OF THE TRUSTED TOKEN

As noted above, there are a number of security threats to the blockchain technologies and cryptocurrencies implemented on their basis (related to their system architecture). Information security should be provided using not only properly selected and implemented cryptographic schemes but also other solutions that ensure the robustness and reliability of the system.

The problem is that a conventional coin has the property of indivisibility and its physical transfer means its withdrawal from one person in favor of the other. Unlike a physical coin, a token is a binary sequence that is repeatedly copied during its circulation. At a certain level of generalization, it is assumed that this duplication can be neglected. However, some studies have shown that duplicates can be a source of vulnerabilities for the entire system.

As an example, we consider a crypto payment to a supplier of goods with the participation of a trusted center (the necessity for this center has been substantiated in other works). This payment implies the transfer of cryptocurrency with a possible charge of a commission and the return of the change. From a perspective of distributed systems, we have a system with three independent transaction processing points. Since the system of crypto payments supports the use of standard (personal) computers without high reliability requirements, the permanent operation of the transaction processing points is not guaranteed. Using the Internet, as the most accessible medium, to organize the communication among the processing points means the use of insecure communication channels.

As a result, in the process of payment, a model of threats occurs that is associated with the (in)operability of the transaction processing points and communication channels; this can cause failures in the system due to the processing of complex transactions at independent points. In this example, the owner of the cryptocurrency wallet should wait until the token is verified and processed so as not to try to use it again in the case of a failure.

These problems can be solved by supplementing the token's structure with four additional fields (token status, token feature, status change time, and waiting limit in a new status), as well as by introducing the mechanisms for their processing. Below, these fields are described in detail.

1. The token status takes the following values:

(1) it can be used (by analogy with a coin in a pocket);

(2) in processing (the coin is conventionally withdrawn from its owner for processing);

(3) it cannot be used (the coin is available for its owner but the checking procedure showed that there are certain problems with the coin);

(4) spent.

2. The token feature contains the check result (e.g., checksum mismatch). The values of this field will be determined more precisely upon designing the checking module.

3. The status change time is the exact time when the token switches to a new status. As an example, the initial status "can be used" is assigned upon sending the token to a checkpoint, the status in the crypto wallet is changed to "in processing," the status change time is recorded, and the waiting limit in the new status is set.

4. The waiting limit in a new status is introduced based on the following considerations. Suppose that the token is sent to a checkpoint and has the current status "in processing" on the side of the crypto wallet. On the side of the checkpoint, a hardware failure occurs; the decision about the token has not been made and probably never will be made due to the loss of data. To eliminate the possibility of loosing the coin, a waiting time limit is introduced after which the token returns to its initial status. In this case, its feature field indicates that the decision about the token was not made because the waiting limit was exceeded. In this example, it is assumed that this problem is further settled with the assistance of platform operators.

Based on the field structure described above, we propose the following crypto payment algorithm.

Step 0: Initial state. A copy of a token data file is in the crypto wallet. The token status is "can be used." The token feature is "absent." The status change time corresponds to the time when the last user received the token. The waiting limit is -1 (no limit). There is also a copy of the token data file in the TRC; its field values are similar to those of the copy in the crypto wallet.

Step 1: Sending a splitting query to the TRC. The token status on the side of the crypto wallet is changed to "in processing." The status change time is set equal to the time of receiving the splitting query. The waiting limit is t_{thres}. The query with the splitting parameters is send to the TRC. The copy of the token data file in the TRC has the field values similar to those at Step 0.

Step 2: The TRC receives the query. The TRC finds its copy of the token and changes its status to "in processing." The confirmation of the status "in processing" is send to the crypto wallet in the form of a receipt. Based on the receipt, the crypto wallet sets the feature "TRC received the query" for its copy of the token.

Step 3: Sending the query to the TGC. Based on the parameters of the query, the TRC requests the generation of three (in the general case) cryptocoins whose total nominal value is equal to the nominal value of the original coin. Their field values are similar to those at Step 2 (the feature is set only on the side of the crypto wallet).

Step 4: Generating the crypto coins. The TGC generates three impersonal crypto coins with specified nominal values. The status of all three coins is "cannot be used." The copies of the files of all three crypto coins are sent to the TRC.

Step 5: Managing the crypto coins in the TRC. Having received three coins from the TGC, the TRC checks them for compliance with the query. Once the compliance is confirmed, the TRC (on its side) changes the status of the original coin to "spent," assigns three crypto coins their new owners, taking

their purpose into account (payee, change, or commission), and sets their status to "can be used." A receipt is then sent to the TGC to confirm that the crypto coins have been received and processed.

Step 6: Returning the change. The crypto wallet receives a copy of the file containing the change. The crypto wallet switches the status of the original coin from "in processing" to "spent."

7. TOKEN MANAGEMENT MODULES

7.1. A Module for Generating the User's Key, Key Container, and Network Name (InitUs)

Format: *InitUser FileUserID UserPIN RealName*<*RandomString*>, where

FileUserID is the name of a file containing a password-protected user ID (it can be associated with the user name but it must differ from *RealName*);

UserPIN is a password (PIN code or its input method, e.g., reading from a USB token) used to secure the user ID.

RealName is a text file for user data (its size does not exceed 160 bytes);

<*RandomString*> is an optional parameter used to improve the operation of the random number generator ("acceleration line").

This module creates a file "random.bin" for the random-number generator and a file with a password-protected user ID (in fact, it is a secure container for storing and transmitting the user's personal identifier/key). In addition, the module generates the user's network name that is stored in the file "RealName.net" in a human-readable format (32 hexadecimal digits). The first eight digits also define the name of a file in which the network name is stored in a binary format.

The module returns the following errors needed to integrate module calls into external applications or a smart contract:

"–1" is the security module test error;

"–2" is the call format error (wrong number of arguments);

"–3" indicates that the file *FileUserID* already exists or the file *RealName* does not exist;

"–4" is the random number generation error;

"–5" is the random number update error;

"–6" is the FileUserID writing error;

"–7" is the FileUserID reading error.

In the platform, there are two subscribers with fixed container names: 000 for the TGC container and 007 for the TRC container.

As an example, the following sequence of commands (smart contract) forms a system of three users, TGC, and TRC:

```
md 01
md 02
md 03
md CET
md CRT
copy alisa 01\alisa
initus 01\1 alisa 01\alisa 123
copy *.net 01\*.*
del *.net
copy boris 02\boris
initus 02\2 boris 02\boris 456
copy *.net 02\*.*
del *.net
copy ivan 03\ivan
initus 03\3 ivan 03\ivan 789
copy *.net 03\*.*
del *.net
initus 000 boss
copy 000 CET\000
del 000
copy gentoken.exe CET\gentoken.exe
initus 007 crt
copy 007 CRT\007
del 007
copy sendtok.exe CRT\sendtok.exe
```

It is assumed that the root directory contains the executable modules copied to the created directories. In this case, there are also text files "alisa," "boris," and "ivan" that contain the real names of the subscribers.

7.2. Token Generation Module (Gentoken)

Format: *GenToken UserPIN Token nominal*, where

UserPIN is a password (PIN code or its input method, e.g., reading from a USB token) used to restore the TGC personal identifier (key) (for the model considered, it is a method or string "boss");

Token nominal is a decimal number that does not exceed 999999.

This module also creates a file "random.bin" for the random number generator and a file "Glob.tcn" with the generated tokens and generation keys (individual for each token) that are protected by the personal identifier of the user with the name 000. It is assumed that the key container 000 is already created by the *UnitUs* procedure (module). As a result, this module yields an impersonal token of a specified nominal that has the extension *.tc0.

7.3. Module for Transferring Tokens to Users (GetCrtK)

To send tokens to users, it is required to generate transport keys. This is done either by the subscriber or

by the GetCrtK (get center registration transaction keys) module in the TRC.

Format: *GetCRTK Network name file UserPIN*, where

Network name file is a 16-byte file yielded by the InitUs procedure that contains the binary network name of the user (subscriber);

UserPIN is a password (PIN code or its input method, e.g., reading from a USB token) used to secure the TGC transport key. This password is used to secure the container with the transport key (it has the extension *.007). The TRC must have the keys of all users. The users can generate transport keys independently and send them to the TRC; the password should be send separately (by an SMS, e-mail, or voice mail). There is another method: the users send binary files with their network names to the TRC; the TRC generates transport keys for the users and sends (using other channels) each individual user his or her password. From a security perspective, this is an equivalent scheme because the users do not know each other's passwords, and the TRC is a trusted party (a trusted component of the system).

7.4. Module for Sending the Generated Token to the User

Format: *SendTok[.exe] CrtPIN CrtKeyFile(.007) CrtKeyPIN TokenFile*, where

CrtPIN is a password (PIN code or its input method, e.g., reading from a USB token) used to secure the TRC 007 container (this password is then used to open the container with the TRC key to provide the token with the TRC AC);

CrtKeyFile(.007) is the name of a key that belongs to the user to which the token is sent (it contains the user's network name that is written in the corresponding field of the token);

CrtKeyPIN is a password that is used to open the container with the transport key (it has the extension *.007);

TokenFile is the name of an impersonal token sent to the user with the key *CrtKeyFile*(.007), which is obtained using the *GetCrtK* procedure described above.

This module yields a token with the extension *.tc7 (this token is not yet received by the user, i.e., its AC is not yet fixed).

7.5. Token Fixation Module

To fix the token, the following procedure is used: *FixToken[.exe] CrtKeyFile CrtKeyPIN UserFile UserPIN TokenFile*, where

CrtKeyFile(.007) is the name of a key that belongs to the user to which the token is sent;

CrtKeyPIN is a password (PIN code or its input method, e.g., reading from a USB token) used to open the container with the user's network key (the password specified in the method *CrtKeyPIN* is then used to open the container with the transport key; it has the extension *.007);

UserFile is the name of a file containing a password-protected user ID (container) (it can be associated with the user name but it must differ from *RealName*);

UserPIN is a password used to open the user's container;

TokenFile is the name of the token fixed by the user.

Upon fixation, the token has three authentication codes: TRC (root) AC, transport AC, and user AC, and can be used in transactions (e.g., to transfer a part of its nominal value to another user).

8. EXTENDING THE TRUSTED TOKEN CONCEPT TO A DIGITAL PROOF UNIT

The further development of the approach described above resulted in the creation of a universal digital proof unit (UDPU) that is an extension of the token technology as applied to the logistics of goods, distributed storage, transaction systems, and e-payment systems (including international ones).

A **universal digital proof unit (UDPU)** is a measure (fixed in digital form as a special file or blockchain link/atom) for storage and transfer of data, including private digital assets, e-money, non-cash money, digital rights, electronic documents, and other digital information. The UDPU is provided with a unique ID number, creation time, life time (validity time), owner's network name, and other parameters necessary for storing and exchanging information; it also allows all its parameters to be checked for integrity.

Below, we define some related terms.

A UDPU owner is a participant of the UDPU registration and circulation system (can also be referred to as subscriber, user, or client) that is assigned a unique network name with this name being necessarily included into the system's structure.

The **network name** is a unique sequence of hexadecimal digits uniquely associated with the user's private key and real name; the network name is generated using a cryptographic procedure.

The **user's private key** is a random number generated during the first or repeated user registration that serves to generate (fix) and check UDPU authentication codes.

The **authentication code (AC) of the UDPU** is the information embedded into the UDPU that allows its parameters to be checked for integrity and restored if they were damaged or intentionally modified.

Table 2. Advantages of the universal digital proof unit

No.	Property	Provided by
1	Controlled anonymity of transactions	An original transaction protocol whereby the content of the transaction can be encrypted or opened; in the case of complete closure, no information about the transaction can be obtained
2	Controlled anonymity of names	Names can be generated in such a way that a real name cannot be derived from a network name
3	Anonymity of the UDPU	It contains only the owner's network name
4	High cryptographic security of the UDPU and transactions	Use of Russian cryptographic algorithms
5	Possibility of using any blockchain	Independence (in terms of the protocol) from the blockchain algorithm and the possibility of inserting data into a blockchain link and retrieving them from a blockchain link of any type
6	Full tracking of transactions while preserving the anonymity of the owner	The original UDPU structure in the form of a novel blockmatrix (a further development of the blockchain in 2D)
7	Possibility to restore the UDPU and transactions even after the loss of private data from the storage (wallet)	A feature of the transaction protocol with four-link confirmation and use of paired keys
8	Possibility to work with the TRC or without it (centralized and decentralized schemes)	A feature of the transaction protocol
9	Possibility to control the life cycle of the UDPU (in particular, setting a limit for the UDPU lifetime)	The original UDPU structure and the transaction protocol
10	Full protection against reuse	Storing the UDPU in the blockmatrix
11	No need for verification centers	The symmetric key system
12	Possibility to conduct the transaction with the confirmation of the transaction information by the recipient (protection against mistaken transactions)	A feature of the transaction protocol with four-link confirmation of reception
13	Technological independence from the leaders of the cryptocurrency market and regulatory agencies of the countries initiating the sanctions	The original protocol and use of cryptographic algorithms
14	Independence from the existing market of cryptocurrencies and tokens	The original protocol and use of cryptographic algorithms
15	Wide area of application	The possibility of translation into a machine-readable format (two-dimensional code)

The **generation center (GC)** is a dedicated subscriber that, at the request of other subscribers or at its own discretion, generates primary impersonal UDPUs.

The **transaction Registration Center (TRC)** is a dedicated subscriber that assigns authentication codes to the UDPUs and sends them to the users; in addition, the TRC keeps records of the UDPU circulation.

A **transaction** is a system process (represented in material form) of transferring the entire UDPU or its part from one subscriber to another; the transaction is characterized by its initiator (sender) and recipient.

The **UDPU circulation system** is a system that consists of subscribers, GC, and TRC and provides the life cycle of the UDPU in material form.

The **life cycle** consists of the following elementary components: generation, registration, transfer to the subscriber, use in transactions, and termination. As components, the life cycle also includes merging and splitting of UDPUs, as well as translating them into a machine-readable format (two-dimensional code).

The UDPU makes it possible to solve the following problems in the fields of digital assets and finance, exchange activities, and logistics of goods:

• the generation of tokens, digital coins that cannot be stolen owing to the unchangeable names of their owners;

• the complete avoidance of the threats associated with the use of foreign cryptographic algorithms;

• the creation of digital stocks and bills of arbitrary (yet unchangeable and self-restorable) values;

• safe trading by denominating currencies (including cryptocurrencies) in the UDPU;

• the solution of logistics problems by providing goods with UDPUs (as two-dimensional code markers) and storing the UDPUs in a distributed ledger;

• information protection during transactions and the protection of private and confidential data stored in a distributed ledger;

• the development of perfect systems for keeping records on works of art and their authentication.

Table 2 summarizes the advantages of the UDPU.

CONCLUSIONS

The proposed technology for creation of digital assets makes it possible to generate trusted tokens of all necessary types and conduct transactions with them. It is reasonable to implement cryptographic operations based on domestic cryptographic algorithms to provide the property of confidence and the possibility of subsequent certification and assessment of the developed solutions by state regulators. The property of confidence for the token or its extension, the UDPU, must be provided in all forms of their transformation and circulation, which is achieved only through the use of a trusted digital platform. Thus, the trusted technology is determined by the properties of the digital proof unit and trusted digital platform.

REFERENCES

1. Biktimirov, M.R. and Shcherbakov, A.Yu., Problems of the synthesis of trusted systems, *Tr. Inst. Sist. Anal. Ross. Akad. Nauk,* 2012, vol. 53, pp. 264–271.
2. Pravikov, D.I. and Shcherbakov, A.Yu., On the issue of changing the information security paradigm, *Sist. Vys. Dostupnosti,* 2018, vol. 14, no. 2, pp. 35–39.
3. Vasil'kov, G., Analysis of the main economic and technical threats for blockchain applications, Internet-Journal Forklog. https://forklog.com/razbor-osnovnyh-ekonomiko-tehnicheskih-ugroz-dlya-blokchejn-prilozhenij-prodolzhenie.

Translated by Yu. Kornienko

SPELL: OK