Global Digital Identity and Public Key Infrastructure

Oleksandr Kurbatov [1][0000-0002-8237-4377], Pavel Kravchenko [2][0000-0002-0456-3295], Nikolay Poluyanenko [3][0000-0001-9386-2547], Yevhenii Demenko [3][0000-0002-5699-1400] and Tetiana Kuznetsova [3][0000-0001-6154-7139]

[1] Kharkiv National University of Radioelectronics, Kharkiv, Ukraine
olkurbatov@gmail.com
[2] Distributed Lab, Kharkiv, Ukraine
pavel@distributedlab.com
[3] V. N. Karazin Kharkiv National University, Kharkiv, Ukraine
nlfsr01@gmail.com, demenjay@gmail.com,
kuznetsova.tatiana17@gmail.com

Abstract. This paper proposes the concept of building a decentralized identification system that allows each participant to perform the role of both an identity object and an identity provider, regardless of their technical and legal capabilities. The main parameter of significance in the system is the level of trust toward the subject by the other participants of the platform and external IDs information consumers. This maintains the ability to fully manage the account and associated PII for their owner through the use of a cryptographic signature mechanism: changes to key data, PII and identifiers can be made at any time. The proposed system does not change the existing model of trust for large providers of identification services, but it allows to increase the objectivity of information about identifiers and PII of their owners by the possibility of verification of a separate identifier to each of the participants of the platform, followed by recording the results of verification in the chain of blocks. The use of blockchain technology and a consensus-reaching mechanism make it possible to synchronize the sequence of events in the system. The described global digital identity system positions itself as a source of information about global entities of specific subjects, and related personal data sets and established identifiers, while allowing the end-user of information to make independent conclusions about the level of trust of these identifiers based on related transactions. The system described is compatible with the digital asset management infrastructure and current identification tools. Thus, the system accomplishes a number of important tasks: reducing the threshold for using digital identifiers, allowing their ubiquitous use; the ability to verify the identifier and prove its compliance without the need to build a trusted infrastructure (only cryptography and other participants` votes); reducing the likelihood of certificate substitution attacks; increase the objectivity of information about the identifiers used.

Keywords: Blockchain, Digital Identity System, Public Key Infrastructure.

1 Introduction

Today, the usual point is the independent verification of user credentials by each system individually (or by an organization to which the system owner has entrusted user identification). Very often, these methods are not very different from each other: users transmit the same set of personal data, confirm registration using traditional methods (mail, phone, etc.), as a result of which they receive a "local" identifier that is valid only within the boundaries of the selected system.

The main disadvantage of this approach is that the user must go through the same identification and registration process again and again, even if an identical data set is provided. In addition to the resource costs of these processes, this entails that identifiers obtained in different systems are in no way interconnected: information about identifiers and PII confirmations is not synchronized between identity providers, which in some cases can lead to the use of different identifiers to manipulate user permissions. Moreover, at the same time, the data is not stored in a single unified format and is not signed by the user when they are transferred, which entails the difficulty of finding the custodian of certain data and the inability to confirm their integrity and authenticity.

Global services such as Google [1], Facebook [2], Twitter [3], GitHub [4] and others allow users to sign in to other services through the OAuth protocol [5-8], but this protocol does not provide the required cryptographic reliability and uses session mechanisms to gain access to user data [9]. Existing decentralized payment systems have shown that the best practice is the cryptographic signature of each request sent to the accounting system and the signature of each response that the system returns [10].

The global digital identification system involves binding all user`s PII and his public key (as an option - a set of keys) to a unique global identifier. Where it leads:

- All information about confirmations of personal data is stored in a single system. Using mechanisms of digital signature and linking transaction sets to each other will ensure the authenticity of specific PII confirmations with time-bound events;
- The integrity and authenticity of the data binded to the account is checked exclusively by cryptographic methods (control root hash value);
- The management of personal data is completely controlled by its owner, all other participants in the system can only confirm a user-defined data set. To obtain personal data about a particular participant in the system, an identity provider must contact him directly and obtain either the necessary data set or permission to receive this data from another identity provider.

2 Global identifier and data associated with it

A global identifier is a unique entity within the identification system that represents a specific subject and is associated with information identifying this subject. The identifier is bound to: the public key of the owner of the identifier, a set of hash values from the identifiers of other accounting systems, as well as a set of hash values from PII,

which is tied to the account owner. All of the listed data is linked into one structure - an account (see Fig. 1).

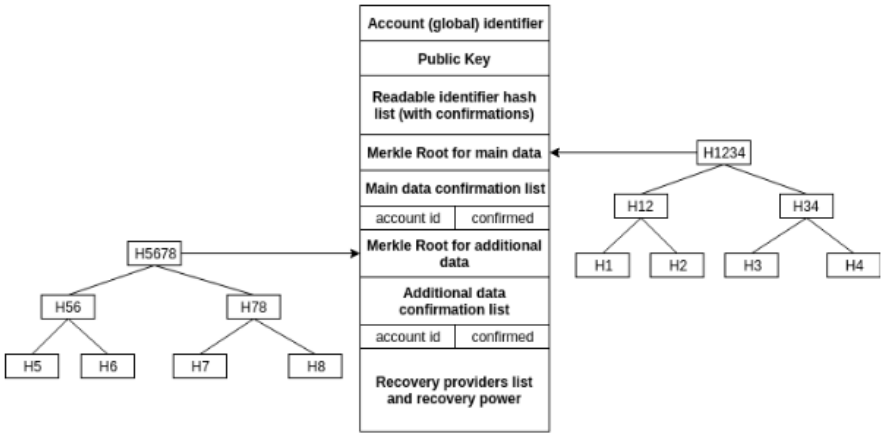

Fig. 1. Account structure in decentralized digital identity system.

"Account (global) identifier" is an immutable and unique value within the identification system, represented as a byte string. This value is generated when the identity provider creates the new user`s account. The main requirement is the uniqueness of this value within the system (this can be either randomly generated or not randomly selected).

"Public key" is a public key value, with the private key of which all transactions initiated by a particular account are signed.

"Readable identifier list" is a set of hash values from identifiers in other systems, such as email address, facebook id, etc. Also, each hash value is associated with a specific set of confirmations from other network members that they checked that the account owner owns the specified email, facebook_id, etc., which correspond to the specified hash values.

"Merkle Root for main data" is the root value from the user's main set of personal data [11]. It is assumed that the values included in this set will not be updated frequently and therefore they will remain confirmed even if other account fields are changed.

"Main data confirmation list" consists of a set of records consisting of the identifier of the provider and information about the set of main data they have verified.

"Merkle Root for additional data" is the root value of an additional set of user personal data. It is assumed that the values that are included in this set will change, and correspond to confirmations more often than the main set.

"Additional data confirmation list" consists of a set of records consisting of the identity of the provider and information about the set of additional data they have verified.

“Recovery providers list and recovery power” determine the set of identity providers (and their amount) needed to restore access to the account and change the public key.

3 Transactions and blocks

The global digital identification system provides that each participant (who has an account) can send a transaction to the network. Transactions are stored as an ordered set of blocks (see Fig. 2), each of which contains a cryptographic hash value of the previous one [12], which provides the ability to control the integrity of the entire history of events associated with global identifiers.

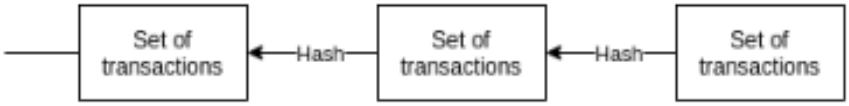

Fig. 2. Transactions arranged as a linked chain of blocks.

A transaction, in turn, consists of a set of defined operations. There are four main types of operations: the operation of creating an account; the operation of changing / adding data associated with a specific global identifier; confirmation operations of data associated with the identifier; the operation of restoring access to the account (changing the public key and the set of parties for recovery).

The account creation operation can be initiated by any existing account. As a result of the operation, a new account is created in the identification system and its identifier and public key are determined (you can additionally define fields containing hash values of personal data and their confirmations).

Data change operations include operations whose purpose is to change / add main and additional personal data of the user, as well as change the values of identifiers associated with external systems. Such transactions can only be signed by the account owner and verified using the public key value specified in the account. In fact, each user personally determines the data set that he wants to associate with the global identifier; identity providers, in turn, have the right only to confirm the data set by the user.

Data verification operations include operations whose purpose is to confirm main and additional personal data. Such transactions can be sent by identity providers (in fact, by any member of the network) in relation to a particular account. Such a transaction contains information about the identifier of the account to which the confirmation is carried out, about the value of Merkle Root, to which the data verified by it belongs, as well as about the set of data itself that has been verified.

The operation to restore access to the account can only be signed by the owners of the identifiers that were defined by the user at the time of creating or updating the account. In fact, the user personally determines the set of trusted parties (and their required number), which can confirm his identity and sign the transaction to update

the public key of the account. In fact, carrying out all types of operations described constitutes the life cycle of the account (see Fig. 3).

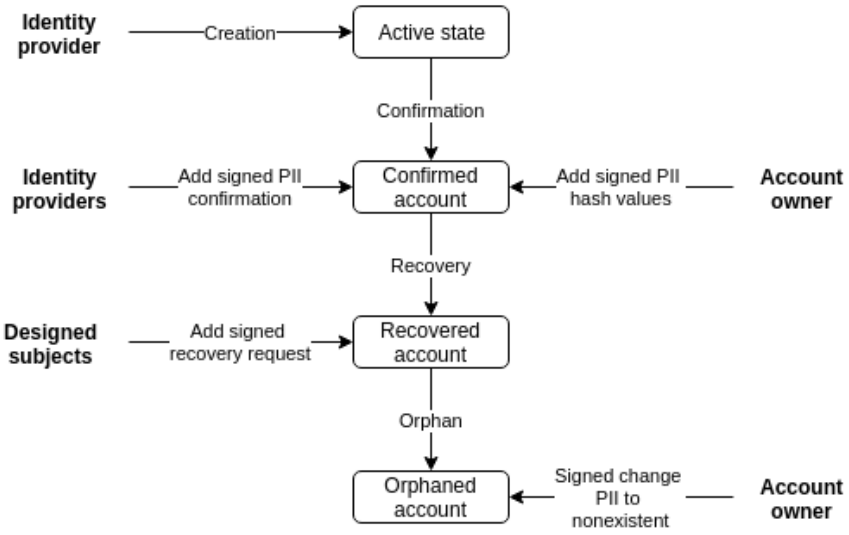

Fig. 3. Account lifecycle.

3.1 Account creation process

In order to create an account in the global digital identity system, the user needs to contact with identity provider. This can be either a separate independent entity, or tied to a specific accounting system. The registration process may differ depending on the individual provider (this can be either the personal presence of the user or remote registration), but it is worth identifying its main features [13]. In the process of creating an account, the user provides the provider with a set of personal data that must be confirmed, the Merkle Root value from the entire data set, as well as the Merkle Branch value to prove that the particular set is in the root value [11]. This allows the provider to check only specific data and agrees that they are included in the general set, and all this without providing the entire set of personal data.

Also, the user needs to generate a key pair (private and public keys) and confirm ownership of the specified email address (or other identifier, depending on those specified in the account). The account creation process is shown in the following diagram (see Fig. 4).

1. The user sends the identity provider a set of personal data that must be confirmed, the Merkle Root value for the entire personal data set, Merkle Branch as evidence that the personal data set is included in the root hash value, public key, email address and the generated identifier value. The user also defines a list of accounts that can change the value of the public key and restore access to the account.

2. The identity provider checks the provided personal data and checks their relation to the root value.

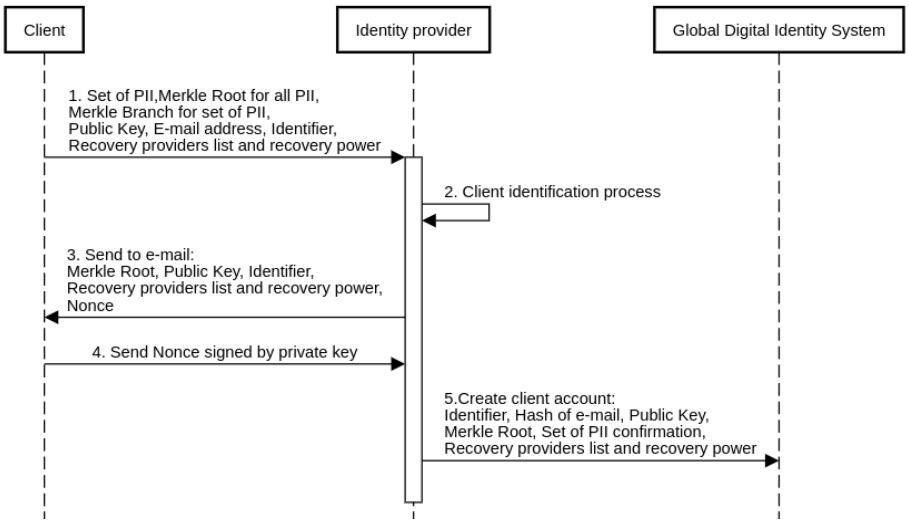

Fig. 4. Process of registration and account creation.

3. The registrar sends a e-mail confirmation to the client (not necessary e-mail, another channel can be used). The confirmation contains Merkle Root (proof that the data received by the provider was not modified), the received public key (also that it was not modified), identifier, list of trusted providers and Nonce value. All values are signed by the provider's key; accordingly, an attack on message modification during transmission is excluded.
4. The user receives a mail, checks that all the data is valid, then signs the Nonce value with his own private key (as proof that he owns the public), and then sends the signed value to the provider.
5. The registrar verifies the signature, after which it initiates the creation of the account: it fills out the necessary fields, generates a global identifier and sends the corresponding transaction.

3.2 PII confirmation process

After the user account has been created, the user determines the set of personal data that will be associated with his account (if he did not define it during creation). He collects the data in the structure of the Merkle tree and puts the root value in the account. As we noted earlier, the account contains two Merkle Root values: for a set of basic data and additional. This approach will leave the master data confirmed, even if the additional ones are changed.

After the user has determined the personal data, he needs to receive confirmation of this data from identity providers. It is important to note that the user may not want

the provider to gain access to the entire set of his personal data. This property can be provided by using Merkle trees. In this case, in the process of confirming personal data, the user must not transfer their entire set so that the provider checks their integrity and authenticity, but only the data necessary for verification and the Merkle Branch value to prove that they are in the root value (see Fig. 5).

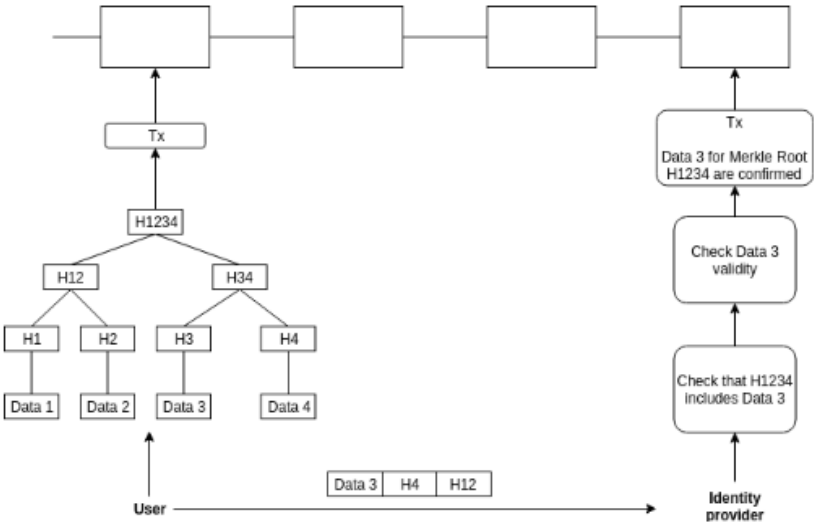

Fig. 5. The user adds his data to the Merkle tree structure, and then updates the account data. When confirming this data, they pass to the identity provider the required set and Merkle Branch for the root value.

When the identity provider receives the data set and Merkle Branch from the user, he checks that the received data matches the root hash value specified in the account, and then checks the received data. If the provider agrees with the data received, it forms a transaction, which indicates that it has confirmed a specific set of personal data that relates to a specific account identifier and a specific Merkle Root value.

If the user decides to update part of the data for a full set, in this case, the value of Merkle Root will completely change. As a result, a previously sent transaction that confirms the data for a particular Merkle Root becomes invalid. That is why the structure of the account implies dividing the data into main and additional: if the user has confirmed the basic data set (and does not change it), then regardless of whether the additional data has been updated, the basic data will remain confirmed.

3.3 Decision making regarding identifier trust

An important feature of such a system is the independent decision-making regarding the identifier by the consumer of information. There is no information in the system that directly determines the validity of a particular identifier: there are only accounts and votes confirming the data of these accounts. Thus, the issue of trust is submitted

entirely to the client side. In this case, he can personally determine the methodology, which will be described when calculating the level of trust.

One of the methods for determining trust is the number of identifier confirmations by other network participants. However, it should be borne in mind that this method is subject to a Sybil attack - one of the account can create a large number of other accounts that confirm the identity of one of the participants in the system [14].

The second approach is to trust only a specific provider (several providers). For example, being a client of a bank, I trust its identification mechanisms and can set up my site in such a way as to trust only those accounts that have been verified by the said bank. Although this scheme is somewhat centralized (if the number of providers that the user trusts is small), it will be quite effective and expected, most used due to its simplicity.

In the third case, if the objectiveness of the result is really important for the network member, he can build complex verification algorithms that evaluate both the level of trust for providers who confirmed a specific account and providers who confirmed accounts with a specified provider (thus up to a large number of verification levels).

It is important to note that whichever way the consumer uses identification results, the system provides the ability to fully customize the verification algorithm, which lies solely on the client side.

4 Permissions to the PII receive

In order to meet the GDPR policy, it is necessary that PII users are transferred and processed only with the permission of their owners [15]. If personal data is transmitted directly by the user to the provider, everything is quite simple: when establishing a connection, the user gives permission to process his personal data. However, difficulties arise if the registrar does not request data directly from the user, but from the party that has previously authenticated it (for example, the user does not keep all the necessary data for registration on the service, however, this data was previously confirmed by the registrar and, accordingly, from him are stored).

Since the provider that stores PII users cannot distribute this data without first obtaining permission to do so, this permission must be provided by the user. Such permission must contain the date and time of its formation, the identifier of the PII owner’s account, the set of data that is allowed to be transferred to the second party, as well as the identifier of the recipient side. Note that the request must be signed with a private key that is tied to the user's account.

If the registrar receives a similar request from the user, he checks the signature value of the request using the public key that is specified in the user’s account. If the signature is valid, then he looks at what set of personal data is requested, and if he really has the required set of personal data, then he sends them and the Merkle Branch value to the requesting party (they must first establish a secure channel). The requesting party carries out a procedure similar to that described in 3: it checks that the received data is in the root hash value, and then confirms the personal data.

The timestamp in the request is necessary to ensure the confidentiality of the updated PII. If the user has updated his personal data set and does not want the party that previously received permission to access the old set to be able to receive updated data for the same permission. Therefore, when checking permission, the registrar compares the time of sending the transaction with the updated data and the time for obtaining permission, and if the time for obtaining permission is shorter, then access to personal data of users is not provided.

5 Consensus reaching and security assumption

Using a native currency for the network to motivate validators and using proof-of-work / proof of stake mechanisms for reaching consensus is not the best solution, since the quality of identification information and the reliability of the system, as practice has shown, will completely depend on the "value of this currency" [12; 16]. Moreover, a completely anonymous environment is not the most suitable for the construction of an identification system, where the level of confidence of the identification will depend on the processing power or the size of the stack of registrars, which does not exactly indicate their competence in issues of user identification.

It is not worth changing the model of trust relations to specialized providers of identification services, but you only need to expand it by combining these providers into a single network, which will contain information about issued identifiers and their confirmations. At the same time, leading identity providers will act as public sources of information about confirmed data, which will in no way affect the change in attitudes towards them. In other words, trusting the identification of a particular organization, you also continue to trust it, however, it no longer acts as an identifier provider for a specific local field, but as an identifier provider in the global system, which automatically expands the user area of activity. Thus, having confirmed the identifier with several major suppliers, the user gets rid of additional identification in local systems that trust these providers.

The most suitable model for achieving consensus for this system is the FBA algorithm [17], which is actually a projection of trust relationships between network participants at its level. Thus, the largest service providers will benefit from launching validator nodes and setting up communication between each other. Thus, each of them will be able to provide relevant information on the status of user identifiers, and relations between global suppliers will not allow one / several parties to cheat: in this case, there is a big risk of being outside the main quorums of the system, resulting in loss of trust from end users and using system identifiers.

The key issue at this stage is protection against spam attacks [18-22]: they cannot affect the decision-making mechanism regarding a specific identifier, since if a user trusts only a number of providers, they can (and should) be ignored by the voices of other participants in the system (this way we protect from the attack of Sybil); however, at the same time, since any user can add a transaction to the network, and in fact the number of such transactions is not limited, this can negatively affect the system's throughput, therefore, a mechanism for protecting against this from the validators

should be provided (for example, receiving rewards for adding transactions in the block - the same mempool, however, each registrar personally agrees with customers regarding the method of payment; we repeat that the internal currency in such a system should be absent).

6 Conclusions

Existing public key infrastructures were the first step in digitizing interactions between Internet users, which involved providing basic information services [23-28]. They showed that this interaction can be accomplished using cryptographic methods based on user certificates. However, such approaches are still not ubiquitous, since it is rather difficult to associate valid (in various accounting systems) user identifiers with their certificates. Integration is difficult when it comes to technical and legal compatibility (including the signature algorithms used), the need for direct trust in certification services, and problems associated with synchronizing information related to the creation / renewal / revocation of certificates.

The implementation of the identification system functioning scheme described in the document does not imply a mandatory abandonment of the existing X.509 infrastructure. Moreover, the simple integration of existing certificates is supposed by adding them to the account structure. The key idea of such a system consists precisely in transferring control of data into the hands of their owners and unifying access to various services through the use of cryptographic methods [29-33]. When user data is tied to a specific key, ownership of which proves the authenticity of the subject, only then can the system be sure of the authenticity of the requester.

References

1. Using OAuth 2.0 to Access Google APIs. [online] Available at: https://developers.google.com/identity/protocols/OAuth2
2. Facebook Login for the Web with the JavaScript SDK. [online] Available at: https://developers.facebook.com/docs/facebook-login/web/
3. OAuth with the Twitter APIs. [online] Available at: https://developer.twitter.com/en/docs/basics/authentication/overview/oauth
4. Authorizing OAuth Apps. [online] Available at: https://developer.github.com/apps/building-oauth-apps/authorizing-oauth-apps/
5. Hardt, D., ed. “The OAuth 2.0 Authorization Framework” (October 2012). doi:10.17487/rfc6749.
6. Jones, M., and D. Hardt. “The OAuth 2.0 Authorization Framework: Bearer Token Usage” (October 2012). doi:10.17487/rfc6750.
7. McGloin, M., and P. Hunt. “OAuth 2.0 Threat Model and Security Considerations.” Edited by T. Lodderstedt (January 2013). doi:10.17487/rfc6819.
8. Dronia, S., and M. Scurtescu. “OAuth 2.0 Token Revocation.” Edited by T. Lodderstedt (August 2013). doi:10.17487/rfc7009.
9. Security Issues in oauth2 and Workarounds. [online] Available at: http://blog.raremile.com/security-issues-in-oauth2-and-workarounds/

10. Signing HTTP Messages. [online] Available at: https://tools.ietf.org/html/draft-cavage-http-signatures-10
11. Method of providing digital signatures. United States Patent US4309569A. [online] Available at: https://patents.google.com/patent/US4309569
12. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. [online] Available at: https://bitcoin.org/bitcoin.pdf
13. ISO/IEC 24760-1:2019. IT Security and Privacy – A framework for identity management – Part 1: Terminology and concepts. [online] Available at: https://www.iso.org/standard/77582.html
14. Douceur, John R. "The Sybil Attack." Lecture Notes in Computer Science (2002): 251–260. doi:10.1007/3-540-45748-8_24.
15. "GDPR ENFORCEMENT." EU General Data Protection Regulation (GDPR): An Implementation and Compliance Guide - Second Edition (n.d.): 280–292. doi:10.2307/j.ctt1trkk7x.19.
16. Kiayias, Aggelos, Alexander Russell, Bernardo David, and Roman Oliynykov. "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol." Lecture Notes in Computer Science (2017): 357–388. doi:10.1007/978-3-319-63688-7_12.
17. David Mazieres. The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus. [online] Available at: https://www.stellar.org/papers/stellar-consensus-protocol.pdf
18. John, RincyMedayil, Jacob P. Cherian, and Jubilant J Kizhakkethottam. "A Survey of Techniques to Prevent Sybil Attacks." 2015 International Conference on Soft-Computing and Networks Security (ICSNS) (February 2015). doi:10.1109/icsns.2015.7292385.
19. S. Paavolainen, T. Elo and P. Nikander, "Risks from Spam Attacks on Blockchains for Internet-of-Things Devices," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2018, pp. 314-320.doi: 10.1109/IEMCON.2018.8614837
20. Kuznetsov, A.A., A.A. Smirnov, D.A. Danilenko, and A. Berezovsky. "The statistical analysis of a network traffic for the intrusion detection and prevention systems." Telecommunications and Radio Engineering 74, no. 1 (2015): 61–78. doi:10.1615/telecomradeng.v74.i1.60.
21. Hangxia, Zhou. "Mitigating Peer-to-Peer Botnets by Sybil Attacks." 2010 International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering (2010). doi:10.1109/cicc-itoe.2010.67.
22. Kuznetsov, Alexandr, Sergii Kavun, Oleksii Smirnov, Vitalina Babenko, Oleksandr Nakisko, and Kateryna Kuznetsova. "Malware Correlation Monitoring in Computer Networks of Promising Smart Grids." 2019 IEEE 6th International Conference on Energy Smart Systems (ESS) (April 2019). doi:10.1109/ess.2019.8764228.
23. Mjølsnes, Stig F., Sjouke Mauw, and Sokratis K. Katsikas, eds. "Public Key Infrastructure." Lecture Notes in Computer Science (2008). doi:10.1007/978-3-540-69485-4.
24. Maeda, Atsuho. "PKI Solutions for Trusted E-Commerce: Survey of the De Facto Standard Competition in PKI Industries." Information Technology Policy and the Digital Divide (n.d.). doi:10.4337/9781843769781.00019.
25. Chadwick, David, and Gansen Zhao, eds. "Public Key Infrastructure." Lecture Notes in Computer Science (2005). doi:10.1007/11533733.
26. V. Dolgov and I. Ishchenko, "Proposals of using chameleon- signature in Ukrainian prototype of combined PKI," 2010 International Conference on Modern Problems of Radio En-

gineering, Telecommunications and Computer Science (TCSET), Lviv-Slavske, 2010, pp. 303-303.
27. Lopez, Javier, Pierangela Samarati, and Josep L. Ferrer, eds. “Public Key Infrastructure.” Lecture Notes in Computer Science (2007). doi:10.1007/978-3-540-73408-6.
28. Davies, Joshua. “Implementing SSL/TLS Using Cryptography and PKI” (December 27, 2010). doi:10.1002/9781118255797.
29. Gorbenko, Ivan, Olexandr Kuznetsov, Yuriy Gorbenko, Anton Alekseychuk, and Vlad Tymchenko. “Strumok Keystream Generator.” 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT) (May 2018). doi:10.1109/dessert.2018.8409147.
30. Andrushkevych, Alina, Yurii Gorbenko, Olexandr Kuznetsov, Roman Oliynykov, and Mariia Rodinko. “A Prospective Lightweight Block Cipher for Green IT Engineering.” Studies in Systems, Decision and Control (September 30, 2018): 95–112. doi:10.1007/978-3-030-00253-4_5.
31. Kuznetsov, Olexandr, Olexandr Potii, Artem Perepelitsyn, Dmytro Ivanenko, and Nikolay Poluyanenko. “Lightweight Stream Ciphers for Green IT Engineering.” Studies in Systems, Decision and Control (September 30, 2018): 113–137. doi:10.1007/978-3-030-00253-4_6.
32. Gorbenko, I., Kuznetsov, A., Gorbenko, Y., Vdovenko, S., Tymchenko, V., & Lutsenko, M. (2019). Studies on Statistical Analysis and Performance Evaluation For Some Stream Ciphers. International Journal of Computing, 18(1), 82-88.
33. Menezes, Alfred J., Paul C. van Oorschot, and Scott A. Vanstone. “Handbook of Applied Cryptography” (December 7, 2018). doi:10.1201/9780429466335.