Decentralized Identification and Certification System

Oleksandr Kurbatov
Kharkiv National University of Radio Electronics,
Kharkiv, Ukraine,
olkurbatov@gmail.com

Oleksiy Shapoval, Nikolay Poluyanenko,
Tetiana Kuznetsova
V. N. Karazin Kharkiv National University,
Kharkiv, Ukraine,
alex.shapoval@protonmail.com, nlfsr01@gmail.com,
kuznetsova.tatiana17@gmail.com

Pavel Kravchenko
Distributed Lab,
Kharkiv, Ukraine,
pavel@distributedlab.com

***Abstract*—This article describes an approach to identification and certification in decentralized environment. The protocol proposes a way of integration for blockchain technology and web-of-trust concept to create decentralized public key infrastructure with flexible management for user identificators. Besides changing the current public key infrastructure, this system can be used in the Internet of Things (IoT). Each individual IoT sensor must correctly communicate with other components of the system it's in. To provide safe interaction, components should exchange encrypted messages with ability to check their integrity and authenticity, which is presented by this scheme.**

Keywords—blockchain technology; public key infrastructure; integrity and authenticity; decentralized system; identification and certification

I. Introduction

In a typical public-key infrastructure (PKI) scheme, the certificate issuer is a certificate authority (CA), usually a company that charges customers to issue certificates for them [1-10]. By contrast, in a web of trust scheme, individuals sign each other's keys directly, in a format that performs a similar function to a public key certificate [11-16]. Thus, there are two main approaches to building infrastructure: using hierarchical model according to the X.509 standard [17] and fully decentralized solution based on web-of-trust [18].

First approach is widely used in existing systems because it is very effective in terms of performance [1-3]: processes of recievening, updating and recalling of certificates do not need a lot of time [19-24]. In addition, such model can quickly react if keys of low-level certification authorities were compromised [25-27].

However, this public key infrastructure model has several problems [28-35]:

- an opportunity of sole censorship [28, 29];
- high-level certificate authorities need to be trustworthy [1, 2];
- the whole certificate chain up to the root certificate authority needs to be verified [28];
- problems with OCSP (Online Certificate Status Protocol) synchronization [19];
- the root certification authority is a failure point (problems with root certificate authority keys compromisation) [20];

Public key infrastructure based on web-of-trust is fully decentralized model, because each of its members provides certification and other members' certificates validation [11-14]. This approach allows to solve the problems of hierarchical model. However, it is less flexible because every action with functioning certificate (updating, recalling) has to be validated by all nodes that keep it [14-16]. One additional limitation is the difficulty of building the certificates chain with high end trust level.

In this article we describe building principles of decentralized identification and certification infrastructure, which proposes using blockchain technology together with hybrid model: two-level hierarchical structure. Its high level is comprised of local certificate authorities, which are organized into trust network and reach consensus regarding ledger state; the low level is end users. Certificate authorities act as the middle link between servers and end users and guarantee the authenticity of latter.

II. Components of Decentralized Identification and Certification System

Decentralized identification and certification system is comprised of the following components:

- CA - Certificate Authorities;
- RA - Registration Authorities;
- Personal data storages;
- Certificate storages;
- End users, systems and applications;

Schematic placement of components and their interconnection can be shown on Fig. 1.

Certification authorities are one of the main components of public key infrastructure. CA issue certificates to confirm users' rights and for systems or apps

978-1-7281-4184-8/19/$31.00 ©2019 IEEE

that need it. In the process of certificate creation CA signs it using private key thus confirming its authenticity. CA's public key is available for all functioning subjects.

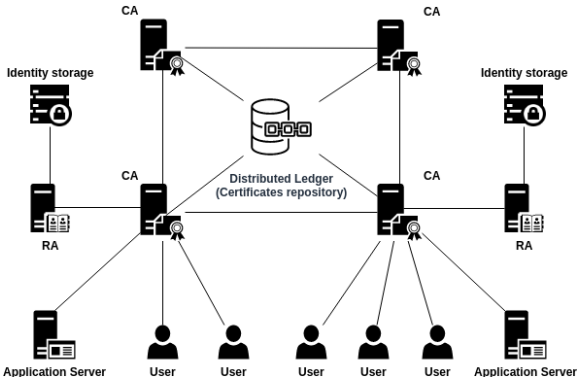

Fig. 1. Components of public key infrastructure

Certification authorities also act as platform validators i.e. they check transactions and reach agreement on the current ledger state. Ledger is an ordered chain of blocks. It guarantees the integrity of the whole transaction history (issuing, updating, recalling certificates). It is proposed to use Federated Byzantine Agreement as a consensus algorithm between two validators of the platform [36].

RA conducts initial identification and registration of users. To receive a certificate user needs to reach out to RA directly. RA processes user's query and provided data. After processing, RA passes the query to CA. It is worth noting that RA processes different data including personal. Because personal data cannot be passed to third parties (and be processed by them) without user's permission, they must remain confidential. During transferring of request from RA to CA only public data and hash of all data packet should be transferred. These values must be included in open key certificate so both user (who created request) and other side (in case of disputable situations) could check that data, provided by RA was not modified and was processed correctly. Because RA is fully responsible for storing and processing users' personal data, it must have complex information protection system and safety policy regarding data storing, processing and reserving.

Identity storage keeps data which users provided to RA. Users' data are encrypted with RA's key and it only has direct access to them. This data must be stored during the whole lifespan of user's public key certificate. If user changes his data, RA also needs to update a record in Identity storage (as well as to send new hash value to CA). Third party organizations may ask RA to get user's data (for example, to check their correspondence to hash value in public key certificate). In this case referring side needs to receive user's permission to access and process his personal data (request must include what data will be handed, public key of requester for encryption and signature of this request by user's private key).

As it was mentioned earlier, certificate storage is a distributed ledger, which every CA keeps. Certificates are organized as ordered chain of blocks, with each linking to the previous one. Blocks consist of transactions, each containing operation tied to certain public key certificate. Each action (issuance, update, recalling) must be initiated using transaction. End users, systems and application servers also may have local copy of the database. It increases clarity level during validation of CAs' work and immutability of operations history.

III. System Functioning Properties

Firstly, certificate authorities have to exchange public keys among each other. Each one of them forms public key certificates for all of the others. Wherein, each of CA defines its trust level to other CAs. If trust level is maximum, CA fully trusts other CA to conduct certification for users, systems and applications [37].

After certificates are formed, each CA generates a transaction which contains a set of certificate issuing operations with corresponding details. All transactions are united into one block, which is called the genesis block. All further certificate history will be based on it.

The amount of transactions in genesis block is equal to the initial amount of CAs. The amount of issued in genesis block certificates equals $n(n-1)$, where n - initial amount of certificate authorities. It is worth mentioning that initial state of trust between certificate authorities might be heterogeneous. Depending on one CA's trust level to another, end users' certificates may have different trust level for other CAs, users, systems and applications.

CA can be added after decentralized identification and certification system has already started. To do this, new CA must ask all other CAs in the system and provide its public key. It's worth noting that every CA independently decides whether to add new CA or not.

To have the right to certify users, systems and applications, CA needs to receive certificates from all active CAs. During certificates issuance, each CA defines how much it will trust the certificates from new CA. If trust level is maximal (100), all users who received their certificates from this CA will guaranteed trust new CA's certificates. If trust level is near zero, certificates from this CA will be considered as not reliable enough and won't be accepted by verificators. In this case trust level of certificate, which was issued by other CA will be equal:

$$validity\ level = STLCAproxy\ (cert.\ CAforeign)\ SVLCAforeign(cert.\ object),$$

where *STLCAproxy (cert. CAforeign)* - Subject trust level of fiduciary certificate authority to third party CA; *SVLCAforeign(cert. object)* - Subject validity level of third party certificate authority to end user.

It is also worth mentioning that trust level between CAs may change while system is working. Each alteration of certificate trust level must be initiated by corresponding transaction. Let's note that every user independently defines necessary trust threshold, which certificate must meet to be used for interaction.

After genesis block was formed, end users, systems and applications receive certificates. To receive a public key certificate, subject refers to RA to complete initial identification and registration. RA requires necessary data from subject, processes them and put into protected storage. After this RA provides CA hash value of received data, subject's public data (which are embedded in certificate) and subject's public key. CA receives this data and forms public key certificate. Certificate receiving process is shown in Fig. 2.

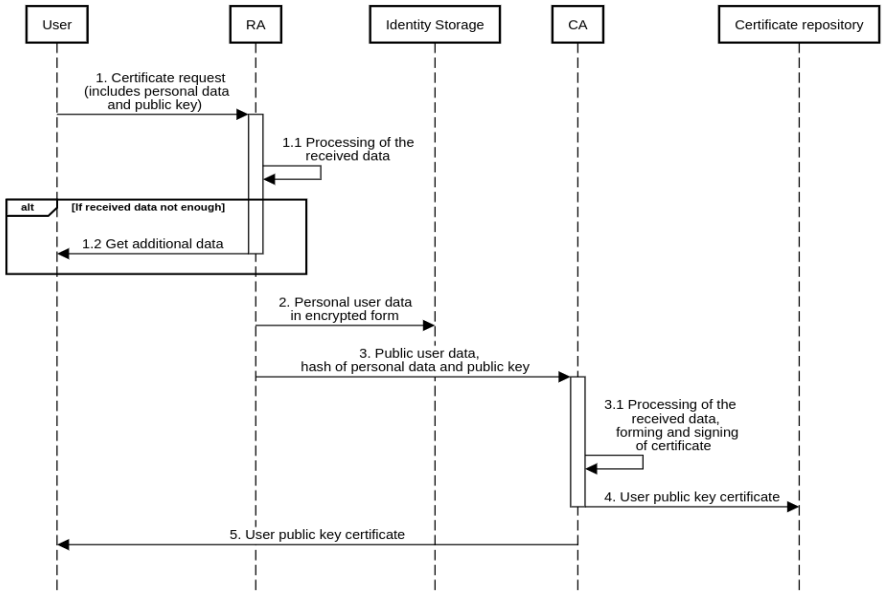

Fig. 2. Public key certificate receiving process

User sends his personal data and generated public key to RA. All data is transferred in encrypted form (directional encryption to RA). RA processes the request (identifies user). Because the goal of receiving key is stated in the request, different sets of necessary data might be needed for identification. If provided data is not enough, RA requests additional data. If user provides the data, process carries on, if not, the request gets denied.

Afterwards, RA encrypts user's personal data and put them in protected storage. Only RA can receive direct access to personal information. If third party wants to get access to personal data, it needs to receive user's permission. RA passes data set to CA, which includes: user's public data to be included in certificate, user's public key and has of his personal data. It is worth noting that all data is passed in encrypted form (directed to CA's public key) and are signed by RA.

After this CA processes data received from RA and use them to form user's public key certificate. CA forms transaction which contains certificate creation operation and distributes it among other CAs. Nodes validate transaction and reach consensus regards updating Certificates repository. When transaction is added to repository, CA returns user his public key certificate. Since then user can apply his key pair to interact with other users, systems and applications.

IV. Certificate Verification

Subject of verification needs to establish connection between the public key and and certificate's subject. It is done by checking that received certificate was issued by authorized organ with enough trust level. Certificate validation may be conducted in two ways depending on who issued the public key certificate.

Singe verification is done only if verificator fully trusts CA who issued this certificate to prover. In this case, to check public key, verificator refers to Certificate repository (which is available for all) and receive corresponding certificate. After that he checks all fields of certificate: validity time, validity level must be more than 0 (which means that certificate is not recalled) and so on. After this it checks whether this certificate was signed by fully trusted CA (for example, if verificator is a client of this exact CA).

If prover's certificate was signed by CA, which verificator doesn't trust, in this case it checks certificate chain. Unlike web of trust and X.509 standard, max amount of verified certificates in chain with this approach equals 2. To check a certificate, verifier refers to Certificate repository and receive two certificates: prover's (which is signed by its CA) and CA's (which is signed by CA and which verifier trusts). If both certificates are valid and trust level of verifier's CA to prover's CA is more or equals needed, verifier consider prover's certificate to be valid.

It is also important to consider how network behaves when one of CAs is compromised. In case of certificate redemption the X.509 standard provides an opportunity to reissue it by higher level CA. Wherein, lower level participants for some period of time lose their ability to fully function (in context of root certification authority keys compromisation this becomes a serious issue). In case with web-of-trust user's certificate redemption does not influence other users' interaction. However, it is a slow and complex process (firstly, a large number of nodes must be convinced that it is the certificate owner who wants to redeem it and that this is not an intruder trying to limit user's capabilities; after this new public key must be distributed among members while trying to avoid malefactor's attack).

V. Conclusions

Identification and certification system based on blockchain technology will be the only source of information about certificates of users, systems and applications as all history of operations on certificates will be stored and processed by different independent parties.

Architecture, which is described in this document, lets any end user or application to hold full public key certificate database (and their current statuses) with reliable synchronization and ability to check actions of all certificate authorities according to protocol rules.

Described schema lets every member of the system to independently define the level of trust to all other members. Thus, the level of objectivity and clarity of identification and certification processes rises, as every member checks other's actions and independently make a decision to trust individual subject.

Using such kind of system lets to organise infrastructure between users and individual application servers. Key feature of such structure is using single identifier (public key) to receive services from all application servers in decentralized system.

REFERENCES

[1] S. F. Mjølsnes, S. Mauw, and S. K. Katsikas, Eds., "Public Key Infrastructure," Lecture Notes in Computer Science, 2008.

[2] A. Maeda, "PKI Solutions for Trusted E-Commerce: Survey of the De Facto Standard Competition in PKI Industries," Information Technology Policy and the Digital Divide.

[3] D. Chadwick and G. Zhao, Eds., "Public Key Infrastructure," Lecture Notes in Computer Science, 2005.

[4] V. Dolgov and I. Ishchenko, "Proposals of using chameleon-signature in Ukrainian prototype of combined PKI," *2010 International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*, Lviv-Slavske, 2010, pp. 303-303.

[5] J. Lopez, P. Samarati, and J. L. Ferrer, Eds., "Public Key Infrastructure," Lecture Notes in Computer Science, 2007.

[6] J. Davies, "Implementing SSL/TLS Using Cryptography and PKI," Dec. 2010.

[7] A. S. Atzeni and A. Lioy, Eds., "Public Key Infrastructure," Lecture Notes in Computer Science, 2006.

[8] Krasnobayev V. A. Method for realization of transformations in public-key cryptography. *Telecommunications and Radio Engineering. - Volume 66, 2007 Issue 17,* pp. 1559-1572.

[9] A. Kuznetsov, I. Svatovskij, N. Kiyan and A. Pushkar'ov, "Code-based public-key cryptosystems for the post-quantum period," *2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkov, 2017, pp. 125-130. DOI: 10.1109/INFOCOMMST.2017.8246365.

[10] W. T. Polk and K. Seamons, "6th annual PKI R&D workshop 'Applications-Driven PKI' proceedings," September 2007.

[11] B. Schneier, "*Applied Cryptography, Second Edition*," John Wiley & Sons, Inc. Oct. 2015.

[12] N. Ferguson, B. Schneier, and T. Kohno, "Cryptography Engineering," Oct. 2015.

[13] G. Guo, J. Zhang, and J. Vassileva, "Improving PGP Web of Trust through the Expansion of Trusted Neighborhood," *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Aug. 2011.

[14] D. Wueppelmann, "PGP Auth: Using Public Key Encryption for Authentication on the Web." *A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of master of computer science.* Ottawa, Ontario September, 2015.

[15] K. Portz, J. M. Strong, and L. Sundby, "To Trust Or Not To Trust: The Impact Of WebTrust On The Perceived Trustworthiness Of A Web Site," *Review of Business Information Systems (RBIS)*, vol. 5, no. 3, p. 35, Jul. 2011.

[16] M. Zhu and Z. Jin, "Trust Analysis of Web Services Based on a Trust Ontology," *Lecture Notes in Computer Science*, pp. 642–648.

[17] *RFC 5280: Internet X.509 Public Key Infrastructure - Certificate and Certificate Revocation List (CRL) Profile*. [online] Available at: https://tools.ietf.org/html/rfc5280.

[18] J. Callas, "OpenPGP Message Format", *IETF RFC 4880*, Nov. 2007, [online] Available: www.ietf.org/rfc/rfc4880.txt.

[19] *RFC 4158: Internet X.509 Public Key Infrastructure - Certification Path Building*. [online] Available at: https://tools.ietf.org/html/rfc4158.

[20] *RFC 6960: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. [online] Available at: https://tools.ietf.org/html/rfc6960.

[21] K. Isirova and O. Potii, "Decentralized public key infrastructure development principles," *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kiev, 2018, pp. 305-310.

[22] A. A. Kuznetsov, Yu. I. Gorbenko, D. I. Prokopovych-Tkachenko, M. S. Lutsenko, M. V. Pastukhov. "NIST PQC: Code-Based Cryptosystems." *Telecommunications and Radio Engineering*, Volume 78, 2019, Issue 5, pp. 429-441. DOI: 10.1615/TelecomRadEng.v78.i5.50.

[23] Yu.V.Stasev, A.A.Kuznetsov, "Asymmetric Code-Theoretical Schemes Constructed with the Use of Algebraic Geometric Codes." *Cybernetics and Systems Analysis*, vol. 41, Issue 3, pp. 354-363, May 2005. DOI: 10.1007/s10559-005-0069-9.

[24] B. Rajendran, "Evolution of PKI ecosystem," *2017 International Conference on Public Key Infrastructure and its Applications (PKIA)*, Bangalore, 2017, pp. 9-10.

[25] I. Gorbenko, M. Yesina and V. Ponomar, "Anonymous electronic signature method," *2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T)*, Kharkiv, 2016, pp. 47-50.

[26] H. Nanang, A. F. Misman and Z. Zulkifli, "Trust, risk and public key infrastructure model on e-procurement adoption," *2017 5th International Conference on Cyber and IT Service Management (CITSM)*, Denpasar, 2017, pp. 1-6.

[27] A. Kuznetsov, A. Pushkar'ov, N. Kiyan and T. Kuznetsova, "Code-based electronic digital signature," *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kyiv, Ukraine, 2018, pp. 331-336. DOI: 10.1109/DESSERT.2018.8409154.

[28] S. Farrell, "Not Reinventing PKI until We Have Something Better," in *IEEE Internet Computing*, vol. 15, no. 5, pp. 95-98, Sept.-Oct. 2011.

[29] I. M. Rodiana, B. Rahardjo and W. Aciek Ida, "Design of a Public Key Infrastructure-based Single Ballot E-Voting System," *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, Bandung - Padang, Indonesia, 2018, pp. 6-9.

[30] O. Potii, Y. Gorbenko and K. Isirova, "Post quantum hash based digital signatures comparative analysis. Features of their implementation and using in public key infrastructure," *2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkov, 2017, pp. 105-109.

[31] A. Kuznetsov, M. Lutsenko, N. Kiian, T. Makushenko and T. Kuznetsova, "Code-based key encapsulation mechanisms for post-quantum standardization," *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kyiv, Ukraine, 2018, pp. 276-281. DOI: 10.1109/DESSERT.2018.8409144.

[32] Yu.V. Stasev, A.A. Kuznetsov. "Asymmetric code-theoretical schemes constructed with the use of algebraic geometric codes". *Kibernetika i Sistemnyi Analiz*, No. 3, pp. 47-57, May-June 2005.

[33] P. Landrock, "PKI, past, present and future," *2005 The IEE Seminar on Quantum Cryptography: Secure Communications for Business (Ref. No. 2005/11310)*, London, 2005, pp. 0_12-2/17.

[34] A. Kuznetsov, A. Kiian, M. Lutsenko, I. Chepurko and S. Kavun, "Code-based cryptosystems from NIST PQC," *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kyiv, Ukraine, 2018, pp. 282-287. DOI: 10.1109/DESSERT.2018.8409145.

[35] M. Pala, S. Cholia, S. A. Rea and S. W. Smith, "Extending PKI Interoperability in Computational Grids," *2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, Lyon, 2008, pp. 645-650.

[36] David Mazieres. "*The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus*". [online] Available: https://www.stellar.org/papers/stellar-consensus-protocol.pdf.

[37] *X.1254: Entity authentication assurance framework - ITU*. [online] Available: https://www.itu.int/rec/T-REC-X.1254.