www.czasopisma.pan.pl PAN www.journals.pan.pl

jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2021, VOL. 67, NO. 2, PP. 201–205
Manuscript received February 6, 2021; revised April, 2021. DOI: 10.24425/ijet.2021.135965

ID-based, Proxy, Threshold Signature Scheme

Jacek Pomykała, Henryk Kułakowski, Piotr Sapiecha, and Błażej Grela

***Abstract*—We propose the proxy threshold signature scheme with the application of elegant construction of verifiable delegating key in the ID-based infrastructure, and also with the bilinear pairings. The protocol satisfies the classical security requirements used in the proxy delegation of signing rights. The description of the system architecture and the possible application of the protocol in edge computing designs is enclosed.**

***Keywords*—ID-based, proxy, threshold, signature, scheme**

I. Introduction

INTERNET of Things (*IoT*) is a term that describes one of the newer concepts of telecommunications systems [20], [19]. It consists in connecting material objects with each other and with Internet resources by means of an extensive computer network. The *IoT* concept includes not only devices (with which you can communicate), but of course also telephones and computers, which are currently the largest group of items included in this structure. In general, *IoT* contains any element that can be linked to the World Wide Web framework [6].

These can be all modules included in smart homes (e.g. household appliances, heating installation, lighting, counters, and clocks). Let's not forget about the automotive market and cars. Sensors and their readers, such as those used in industry, trade, or transport can also be part of the telecommunication *IoT*. Cloud edge computing provides the convenience of using the cloud on edge networks. Edge clouds are hosted by microdata centers that store and process data much faster than is possible with a data center connection. Edge servers function as data processing micro-centers, providing the computing power for the edge cloud.

Blockchain is a decentralized and distributed database in an open source Internet model with a peer-to-peer architecture without central computers and no centralized place for data storage. It is used used for accounting for individual transactions, payments, or journal entries encoded with cryptographic algorithms [22], [23], [13]. The design of efficient and secure protocols, that enable the interaction of *IoT* devices with the core of blockchain networks, is a promising and rapidly evolving research area. This would have the effect of pushing the *IoT* devices out of the edge of the blockchain network, leading, again, to the centralization of operations.

In the described information processing model, the following hierarchical trust structure is possible. At the top of this hierarchy, we can distinguish cloud computing from the Authorization Center, which issues digital certificates to trusted entities. Below, in the text, we have considered the main players in the game (e.g. Companies that trade in electricity and supply it to end users). In this case, the infrastructure based on a distributed register (it would be responsible for the monitoring system) can be treated as a channel of common communication between different players in the scenario under consideration. At the same time, groups of sensors transmit information (on the level of energy consumption) to the company's servers. Such recorded events are the basis for issuing bills. As usual, the end-user pays electronic bills for energy consumption. This process can be realized with the application of smart contracts technology. Now, let us consider a group of sensors (in a large company or farm). It is possible to imagine that a trusted module (communication gate) is designated for a given group of sensors for acquisition and secure data transmission to the billing center. Such a module may receive a power of attorney to sign (sign as a proxy) the submitted measurement results. In this case, the issuer of the power of attorney is the company's headquarters. It is possible that a group of such measurement networks will be considered. Only then, their joint cooperation will allow the measurement results to be signed as a group. Hence, it seems reasonable to consider the possibility of confirming such a signature by an entity delegated to this task in the company's headquarters.

II. The problem specification

The problem of delegation of signing rights appears quite frequently in the literature, in particular within the so called group oriented society [4], [18]. In this paper, we deal with the ID-based signature scheme (see e.g. [15] for the scheme based on the bilinear pairings). Let us remark, that ID-based signatures are especially interesting when the efficient key management is required. In this setting, the bilinear pairings are the key point to obtain the elegant and computationally effective protocols with security based on the related Computational and Bilinear Diffie-Hellman problems (see [1], [7]). Let us also remind, that the first construction of the Gap Diffie-Hellman (GDH) group was proposed in [9], while in [2] and [11] the first examples of the digital signatures working in the GDH group were given.

In what follows we propose a new model of the ID-based proxy, threshold signature scheme. In contrast to the solution

This article was prepared as part of the work carried out in the following two projects: CYBER-MAZOWSZE/0057/19: *Authentication and digital identity delivery system in the PKO BP bank* and DUKAT-POIR.01.01.01-00-0756/18: *Launching services at PKO BP using tokens and smart contracts based on blockchain technology*.

J. Pomykała, H. Kułakowski, B. Grela are with Warsaw University of Technology, Poland, (e-mail: J.Pomykala@mimuw.edu.pl, H.Kulakowski@tele.pw.edu.pl, blazej.grela0@gmail.com).

P. Sapiecha is with National Institute of Telecommunications in Warsaw, Poland, (email: piotrsapiecha@gmail.com).

presented in [7], here U_0 computes the delegating secret for arbitrary group UG of type $ksQ_{ID} := ksQ_0$ with the related verification key equal to $P_{ver} = P_{ver}(k) = ksP$, where k is a random element of $\mathbb{Z}_q^*$. The idea is the following: Let U_0 be a user with identity ID in the ID-based infrastructure with the trusted party PKG (Private Key Generator) having the master key s and the related public key sP. The verification key P_{ver} is approved by computing the commitment kP which serves as a proof of correctness of the related verification key $e(ksP, P) = e(kP, sP)$.

Let $I = I_k$, where:

$$I_k = (ksP, kP)$$

and

$$C = C(k, x, ID) = xQ_0 + h(I_k)ksQ_0.$$

Applying the idea of [7], we propose the following delegated signature of U_0 of the form: $\sigma(m) = (A(m), B(x), C, I)$ on the message m.

Here a random value x is verifiably distributed by U_0 among the members of UG with the aid of the interpolating polynomial of degree t with constant term equal to $x = x_{ID}$.

Let us also remark that the members U_i do not require the use of pseudo-random number generator for the derandomized Weil pairing computation [17]. It is worth saying that deterministic algorithms besides the theoretical interest also allow to reduce the computations complexity.

The above signing rights delegation is not applicable in the more narrow sense of signing rights delegation called the proxy signature, since then the related requirements (security conditions) are not satisfied (see the last two of them below). We propose to remedy this inconvenience applying the identification of the signers of the proxy group and the application of the related multisignature computed by the qualified set of users.

Therefore, we maintain the concept of computing $x = x_{ID}$ by U_0 and verifiable distribution of it among the group UG members. However, now the private inputs x_i of the signers $U_i, i \geq 1$ forming the qualifed subgroup $UG' \subset UG$ contained in part E of the signature $\sigma(m) = (A(m), B(x), C, E, I_{k,w,t})$ unable U_0 to forge the proxy signature on behalf of some U_i, nor the actual proxy signer can deny that he has signed the message m. Here $I = I_{k,w,t}$ contains additionally the information about the delegation of signing rights (warrant) and (optionally) the threshold value t. Concluding, the proxy signature (with the identification of signers belonging to the actual, qualified subgroup $UG' \subset UG$) will satisfy the required security conditions. The value $E = E(UG')$ is in fact the related multisignature [8] performed by the the actual proxy signers on the message containing the delegation data and description of UG'.

III. Related work for the proxy signatures

The proxy signatures were invented in order to keep the reliability and the continuity of services in electronic communication, when we require to delegate the signing rights to other users satisfying the security conditions pointed out below. Here a designated person or group of persons is able to sign the document on behalf of the original signer such that only the proxy signers can create the valid signature and any verifier can be convinced about the original signer's agreement on the signed message. One can distinguish two basic types of proxy signature schemes: partial delegation or delegation by warrant [10].

The warrant is used by the proxy signers to convince any user of their signing delegation power. Such a signature scheme can be used in the delegation of rights to sign messages without relying on any physical device. The delegation by a warrant may be implemented by ordinary signature schemes working in the ID-based public-key cryptosystem.

Formally, the proxy signature was introduced by: Mambo, Usuda, and Okamoto [12]. The proxy delegation by warrant approach was presented in [14] and [21]. The threshold signature schemes were first considered in [4] and [5].

IV. Communication model and signing rights delegation

We distinct the set of parties: $U_0, U_1, ..., U_n$. Let U_0 is the original signer and $UG = \{U_1, ..., U_n\}$ is called the group of delegating signers. We assume that U_0 has identity ID and the related private key $D_{ID} = sQ_{ID} := sQ_0$ and the delegation secret key equal to $D_{proxy} = ksQ_0$.

In case of proxy signature, we assume that all parties: $\{U_0, U_1, ..., U_n\}$ have their identities $ID(j)$, $j = 0, 1, 2..., n$ in the ID-based infrastructure with the given trusted party PKG. The qualified set of members sign any message m on behalf of the original signer U_0. Below, we first present the ID-based delegation of signing rights based on the protocol investigated in [7] and then ID-based, threshold, proxy signature scheme (with identification of group signers).

We recall that the private key of the signer is composed of two ingredients forming the pair (x, D_{proxy}), where the second term is a variation of the private key D_{ID} of the signer U_0 multiplied by a random factor k, while the first is a secret being distributed by U_0 among the group of users UG. The variation is in fact equal to:

$$D_{k,w} = D_{k,w}(ID)$$
$$= h(I_{k,w,t})D_{proxy} = h(I_{k,w,t})ksQ_0.$$

The delegated signature has the form:

$$sigma(m) = (A(m), B(x), C(k, w, t)),$$

where:

$$A(m) = xH(ID, m),\ B(x) = xP,$$

and

$$C(k, w, t) = (xQ_0 + D_{k,w}, kP).$$

V. Proxy signing

In the proxy signing, the final signature contains additionally the extended information: $I_{k,w,t}$ together with the identities of the actual set of signers UG'. Moreover, it contains the approval E of the delegation of rights by the group members U_i, hence now, the role of $D_{k,w}$ above is played by $D_{UG'}(ID) = h(I_{k,w,t})D_{proxy}$, with $UG' \subset UG$ being the

subgroup UG taking part in the collective signing on the message $M = (m, w, t, UG')$ (attention: M is not m).

The final proxy signature under the message m has the form:

$$(A(m), B(x), C_{proxy}, I_{k,w,t}, \bar{B}, E),$$

where: $A(m)$, $B(x)$ are defined above, while $\bar{B} = (B_1, .., B_l)$, $B_i = x_i P$, for: $1 \leq i \leq n$,

$$C_{proxy} = C_{proxy}(ID, UG') = xQ_{ID} + D_{UG'}(ID),$$

where: $I_{k,w,t}$ is defined below and $l = |UG'|$. Moreover, $E = E(UG') = \sum_{i \in UG'} A_i(M)$, where:

$$A_i(M) = x_i H(\sigma(M), ID(i)),$$

and

$$\sigma(M) = (A(M), B(x), C(k, w, t))$$

and $B = x_{ID} P$ with the related terms described above.

VI. Security requirements for the proxy signature

In the paper, we assume that the proxy signatures are generated by the subgroups of UG. In fact, the proxy signature is combined from the partial proxy signatures computed by the corresponding proxy signers belonging to some subgroup $UG' \subset UG$. The computational security of the proxy signature scheme requires the following conditions to be satisfied:

Distinguishable: Proxy signature is distinguishable from the original signer signature.

Proxy-protected: No-one but the proxy signer can generate the corresponding partial proxy signature.

Secrecy: The original signer's private key cannot be derived from any information available for the proxy signers even if they collude together.

Unforgeable: No valid proxy signature should keep an honest signer as accountable for it if he did not participate in signing.

Non repudiation: Neither the original signer cannot deny having delegated the power of signing messages to the proxy signers nor the actual proxy signer can deny that he has signed the message.

VII. The protocol

The bilinear structure is given by:

$$e : G_1 \times G_2 \to G.$$

The PKG's public parameters is the tuple:

$$PT = (G_1, G_2, G, e, q, P, P_{pub}, H, h)$$

and its master secret is $s \in \mathbb{Z}_q^*$. Here: $H : \{0,1\}^* \to G_1$ and $P \in G_2$.

The basic protocol consists of the following algorithms (phases): Setup, ProxyExtract, ProxyDelegate, ProxyKeyshare, ProxySign, Verify. They are described below.

Setup: This algorithm output is the public tuple PT and the secret master key s as above.

ProxyExtract: In this phase, the original signer U_0 with identity ID_0 first selects a random $k \in \mathbb{Z}_q^*$, computes the commitment $K = kP$ and $I_k = (ksP, kP)$. Finally U_0 computes the value $D_{proxy} = ksQ_{ID_0}$, where $Q_{ID_0} = H_1(ID_0)$.

ProxyDelegate: Let w be a warrant (set of identities ID_i for proxy signers $UG = \{U_1, ..., U_n\}$) delegating the signing rights of U_0 for the set of proxy signers UG). The original signer computes the value:

$$D_{k,w,t}(ID_0) = h(I_{k,w,t}) D_{proxy},$$

where: $I_{k,w,t} = (I_k, w, t)$ and t is defined below. Then w and $D_{k,w,t}(ID_0)$ is sent to the proxy group UG.

ProxyKeyshare: In this phase, U_0 selects randomly x_{ID} and the interpolating polynomial f (of degree $t-1$) such that:

$$x_{ID} = f(0) = a_0,\ f(y) = \sum_{i=0}^{t-1} a_i y^i .$$

The value $f(i) := x_{ID,i}$ is sent by a secure channel to the member U_i of proxy group UG, together with the related commitments for the coefficients of the interpolating polynomial $a_i P$, for $i = 1, 2, ..., n$. The correctness of the related shares are verified by the group UG members (see [7]). Finally, U_0 computes the values $B = B_{ID} = x_{ID} P$ and

$$C_{proxy}(ID) = x_{ID} Q_{ID} + D_{k,w,t}(ID) =$$

$$= x_{ID} Q_{ID} + h(I_{k,w,t}) D_{proxy}.$$

Proxysign: Let m be a message to be signed by the proxy group UG. Each member of UG computes the related value $A_i(m) = x_{ID,i} P_m$, where: $P_m = H(ID_0, m)$. Let UG' be the subset of actual proxy signers that takes part in the signing process. In the first part, every member of UG' selects $x_i \in \mathbb{Z}_q^*$ uniformly at random and verifies the correctness of the value $A_i(m)$ with the aids of the commitments $a_i P$ (see [7]). After its positive verification every member U_i of UG' computes the value:

$$A'_i(M) := x_i P_M = x_i H(ID_0, M),$$

where: $M = (m, w, t, UG')$ and broadcasts it in UG'. Every member of UG' verifies the correctness of the related values applying the bilinearity of e and the value of $B_i = x_i P$.

The complete signature of UG' under the message m is the tuple:

$$(A(m), B(x), C_{proxy}, I_{k,w,t}, \bar{B}, E),$$

where:

$$x = x_{ID_0}, A(m) = xP_m,$$

$$B(x) = xP$$

and

$$C_{proxy}(ID_0, UG) = xQ_0 + D_{k,w,t}(ID_0) =$$

$$xQ_{ID_0} + h(I_{k,w,t}) D_{proxy}.$$

Here: $\bar{B} = (B_1, ..., B_l)$, where: $B_i = x_i P$. Finally,

$$E = E(UG') = \sum_{i \in UG'} A'_i(\sigma(M)),$$

where the value of $A'_i(y)$ is defined above, $\sigma(M) = (A(M), B(x), C_{proxy})$ and $A(M) = H(ID, M)$.

Verify: Let $\sigma = (A, B, C_{proxy}, I_{k,w,t}, \bar{B}, E)$, be the signature on the message m with delegation approved by U_0 of identity ID_0.

An arbitrary verifier checks the equalities:

$$e(A, P) = e(P_m, B)$$

and

$$e(C_{proxy}, P) = e(xQ_{ID} + h(I_{k,w,t})ksQ_{ID_0}, P),$$

what is equal to:

$$= e(Q_{ID_0}, h(I_{k,w,t})P_{ver} + B),$$

where: $P_m = H(m, ID_0)$ and $P_{ver} = ksP$. If so, then he checks the correctness of the approvals $A'_i(M)$ verifying finally the equality:

$$e(E, P) := e\left(\sum_{i \in UG'} x_i H(\sigma(M), ID_i), P\right)$$
$$= \prod_{i \in UG'} e(H(\sigma(M), ID_i), x_i P)$$
$$= \prod_{i \in UG'} e(H(\sigma(M), ID_i), B_i).$$

VIII. Security analysis

The security analysis refers to the security requirements defined above. The distinguishability and proxy-protected condition are clear since the private key U_0 is equal sQ_0, while his delegating key is equal to $D_{proxy} = ksQ_0$, for a random $k \in \mathbb{Z}_q^*$. The proxy protected condition follows from the approval $x_i H(\sigma(M))$ computed by the proxy signer U_i. The original signer's U_0 private key sQ_0 is not known to anyone of the users U_i, $i \geq 1$, hence the secrecy condition is satisfied. The proof of participation of the user U_i in the signature $(A, B, C, I_{k,w,t}, \bar{B}, E)$ is based on his approval B_i contained in $\bar{B}$ and $x_i H(\sigma(ID_0, M))$ contained in the signature E, hence the unforgeability condition is satisfied. Finally, the delegating user U_0 can not deny having delegated the power of signing to the proxy group since since his delegating key $D_{proxy} = ksQ_0$ can be approved by the verification key $P_{ver} = ksP$. Together with the approval of U_i equal to $x_i H(ID_0, M)$ this shows the validity of the non-repudiation security condition.

IX. Implementation

The proof of concept implementation was prepared as a script using Python programming language. The code was organized into classes describing parties taking part in our communication model which are then instantiated into objects. The communication process between these objects is then simulated accordingly to the protocol described above. There are four classes: PKG, User, ProxySigner, Verifier, which are instantiated into four sets of protocol parties (PKG, Original Signer, Proxy Signers and Verifier). The additional PublicBoard class was also designed and instantiated as an object storing signature and corresponding data that is sufficient to verify the validity of the signature. It was decided to create separate classes User, and Proxy Signer as it improves the clarity of code and is more suitable in the context of *IoT* with a hierarchical structure. It should be however noted that in the context of blockchain with decentralized peer-to-peer architecture, one user could be acting simultaneously as Original Signer and Proxy Signer depending on the process. The communication between parties is presented in Figure 1.

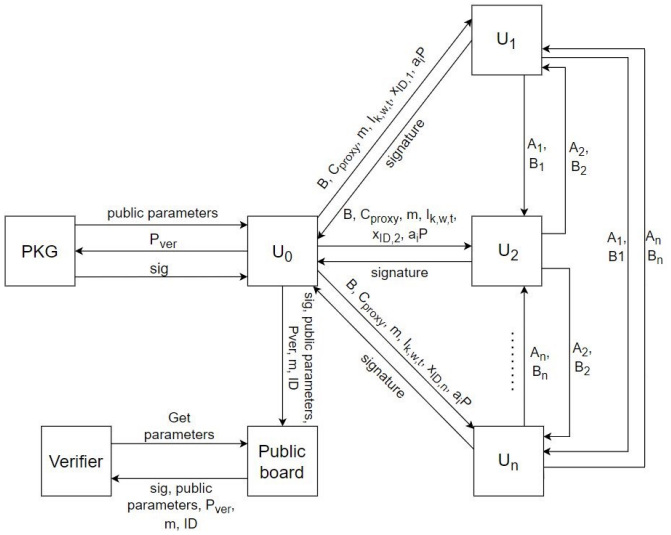

Fig. 1. System architecture

The simulation starts with PKG generating public parameters:

$$(G_1, G_2, G, e, q, P, P_{pub}, H, h),$$

which are then used by an Original Signer U_0 to calculate the verification key P_{ver}, the commitment K, delegation secret key: D_{proxy}, warrant w along with the $D_{k,w,t}$ parameter, as described in section 7. Next, U_0 creates interpolating polynomial f, calculates UG shares $x_{ID,i}$ along with $a_i P$ which are immediately distributed among proxy signers ($x_{ID,i} = f(i)$, a_i are the coefficients of the polynomial f). After computing B and C_{proxy}, the values of: $B, C_{proxy}, I_{k,w,t}, x_{ID,i}$ and $a_i P$ are sent to each U_i, for $i = 1, 2, \ldots n$. After receiving these values, each U_i calculates A_i and sends it among other members of UG' group. Then, each U_i calculates A and E before sending the complete signature $(A, B, C_{proxy}, I_{k,w,t}, E)$ to U_0. Consistently, after receiving the complete signature from every member of UG' group, U_0 sends the one of these signatures to Verifier who checks if the signature was computed correctly and can be trusted.

An open-source version of the described proof of concept protocol is available on the internet (see repository: https://github.com/blaziu1/bilinear_wallet/blob/main/bilinear_wallet.py).

X. Conclusion

In this paper, we have proposed the proxy threshold signature scheme with the application of elegant construction of verifiable delegating key in the ID-based infrastructure. The protocol satisfies the classical security requirements used in the proxy delegation of signing rights. In our construction, we used the bilinear pairings concept. We enclose in the article, the description of the system architecture and the possible application of the protocol in edge computing designs. The presented protocol can be extended for the family of proxy groups generated by a given user for the threshold proxy signatures schemes [16].

References

[1] A. Boldyreva, "Threshold signatures, multi-signatures and blind signatures based on the Gap Diffie-Hellman Group signature scheme", *LNCS*, vol. 2567, pp. 31-44, 2003.

[2] D. Boneh, C. Gentry, H. Shacham, B. Lynn, "Short signatures from the Weil pairing", *Journal of Cryptology*, vol. 17, no. 4, pp. 297-319, 2004.

[3] D. Boneh, C. Gentry, H. Shacham, B. Lynn, "Aggregate and verifiably encrypted signatures from bilinear pairing", *Proceedings of Eurocrypt 2003, LNCS 2656*, pp. 416-432, 2003.

[4] Y. Desmedt, "Society and group oriented cryptography", *Crypto 87*, pp. 120-127, 1987.

[5] Y. Desmedt, Y. Frankel, "Threshold cryptosystems", *LNCS*, vol. 718, pp. 1-14, 1993.

[6] D. E. Kouicem, A. Bouabdallah, H. Lakhlef, "Internet of things security: A top-down survey", 2018

[7] W. Gao, G. Wang, X. Wang, Z. Yang, "One-Round ID-Based Threshold Signature Scheme from Bilinear Pairings", *Informatica*, Vol. 20, No. 4, pp. 461-476, 2009.

[8] K. Itakura, K. Nakamura, "A public key cryptosystem suitable for digital multisignatures", *NEC Research and Development*, pp. 1-8, vol. 71, 1983.

[9] A. Joux, "A one-round protocol for tripartite Diffie-Hellman", *Journal of Cryptology*, vol. 17, no. 4, pp. 263-276, 2004.

[10] S. Kim, S. Park, D. Won, "Proxy signatures, revisited", *LNCS*, vol. 1334, pp. 223-232, 1997.

[11] A. Lysyanskaya, "Unique signatures and verifiable random functions from the DH-DDH separation", *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, pp. 597-612, 2002.

[12] M. Mambo, K. Usuda, E. Okamoto, "Proxy signatures for delegating signing operation", *ACM Conference on Computer and Communications Security*, pp. 48-57, 1996.

[13] A. Manzoor, M. Liyanagey, A. Braekenz, S. Kanherex, M. Ylianttila, "Blockchain based Proxy Re-Encryption Scheme for Secure IoT Data Sharing", 2018.

[14] B.C. Neuman, "Proxy-based authorization and accounting for distributed systems", *Proceedings of the 13th International Conference on Distributed Computing Systems*, pp. 283-291, 1993.

[15] K.G. Paterson, "ID-based signatures from pairings on elliptic curves", *Journals and Magazines, Electronics Letters*, Volume: 38 Issue: 18, pp. 1025 - 1026, 2002.

[16] J. Pomykała and T Warchoł, "Threshold signatures in dynamic groups", *Proceedings of Future Generation Communication and Networking, IEEE Computer Science*, pp. 32-37, 2007.

[17] J. Pomykała, B. Źrałek, "A model of ID-based proxy signature scheme", *Proc. of 6-th Collecter Iberoamerica*, pp. 1-8, 2008.

[18] A. Shamir, "How to share a secret", *Communications of the ACM*, 22:612-613, 1979.

[19] N. Suryadevara and S. Mukhopadhyay, "Internet of things: A review and future perspective" *Reliance*, 2018.

[20] R. Taylor, D. Baron, and D. Schmidt, "The world in 2025-predictions for the next ten years", *Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, 2015 10th International. IEEE, pp. 192–195, 2015.

[21] V. Varadharajan, P. Allen, S. Black, "An analysis of the proxy problem in distributed systems", *Proc. IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 255-275, 1991.

[22] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts", *Consumer Electronics (ICCE), 2016 IEEE International Conference on. IEEE*, pp. 467–468, 2016.

[23] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends", *Big Data (BigData Congress), 2017 IEEE International Congress IEEE*, pp. 557–564, 2017.