This is an accepted version of a paper published in Proceedings of the 3rd International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec 2017).

If you wish to cite this paper, please use the following reference:

Y. Kaga, M. Fujio, K. Naganuma, K. Takahashi, T. Murakami, T. Ohki, M. Nishigaki, A Secure and Practical Signature Scheme for Blockchain Based on Biometrics, Proceedings of the 3rd International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec 2017), pp.877-891, 2017.
http://dx.doi.org/10.1007/978-3-319-72359-4_55

The original publication is available at www.springerlink.com.

A Secure and Practical Signature Scheme for Blockchain Based on Biometrics

Yosuke Kaga[1], Masakazu Fujio[1], Ken Naganuma[1], Kenta Takahashi[1], Takao Murakami[2], Tetsushi Ohki[3], and Masakatsu Nishigaki[3]

[1] Hitachi, Ltd.
[2] National Institute of Advanced Industrial Science and Technology
[3] Shizuoka University

Abstract. In a blockchain system, a blockchain transaction is protected against forgery by adding a digital signature. By digital signature verification, we can confirm that a creator of a transaction has a correct private key. However, in some critical fields, we need to prove that a creator of a transaction is a proper user. In such a case, the conventional digital signature verification cannot achieve sufficient security. Furthermore, a system that combines blockchain and IoT has been proposed. However, since an IoT device in this system automatically generates a blockchain transaction, reliable creator verification is challenging issue. To achieve reliable creator verification in the IoT blockchain system, we propose a new signature scheme for blockchain. Our contributions are as follows: (1) We propose a new secure and practical signature scheme. (2) We implement our signature scheme for an IoT blockchain system and evaluate the security and the practicality of our scheme.
In our scheme, by using user's biometric information as a private key, we prove that a creator of a transaction has a correct biometric information in the transaction verification. Since biometric information such as fingerprint, face, finger vein and so on is unique, this means that a creator of a transaction is a proper user. Moreover, the proposed signature scheme generates a short-term private key and utilizes it for creating transactions. By using this scheme, IoT device can automatically generate a new transaction. Finally, we evaluate security and practicality of the proposed scheme.

Keywords: Blockchain, Biometrics, IoT, Fuzzy signature, PBI, PKI

1 INTRODUCTION

1.1 Background and Motivation

The Bitcoin [1] was proposed in 2009 and become widespread as a cryptocurrency. The core technology of the Bitcoin is called "blockchain." Blockchain can realize a decentralized database, and it is applied to cryptocurrency and smart contract systems [2]. Blockchain will be widely used to critical social infrastructure systems such as financial ones in the future and will spread widely. For

blockchain as a critical infrastructure, highly strict verification of a blockchain transaction creator is required. However, conventional blockchain systems guarantee only that a blockchain transaction creator has a correct private key. That is, conventional blockchain systems cannot confirm that a blockchain transaction creator is a proper user. For example, there is a risk that an attacker steals a user's private key by a cyber attack and creates an illegal transaction. However, conventional blockchain systems cannot detect this attack.

Moreover, many physical devices have connected each other on a network and exchanged information. This mechanism is called IoT (Internet of Things)[3]. Recently, they introduce a collaborating system between blockchain and IoT for automatic smart contract. This collaborating system is expected to spread in the future. For example, IBM's ADEPT (Autonomous Decentralized Peer-To-Peer Telemetry)[4] has a vision called Device Democracy that proposes a scalable and secure platform with non-centralized authority. By using this ADEPT, it is possible to realize automatic and non-centralized smart contract systems. For example, an IoT device like a washing machine collects information and automatically executes a smart contract for consumables order. Even when an IoT device automatically generates a blockchain transaction, it is necessary to confirm not only that a correct device has generated a blockchain transaction but also that a proper user has generated a blockchain transaction at his intention. However, to check user's own intention from automatically generated blockchain transaction is challenging issue.

1.2 Our Contributions

In this paper, we propose a secure and practical signature scheme for IoT blockchain system based on biometrics. This method is the first study to combine blockchain and biometrics at the algorithm level as far as we know. Our method uses the fuzzy signature technology [5][6] for generating a blockchain transaction and realizes strict verification of blockchain transaction creator in IoT blockchain system. Our contributions are as follows:

1. A secure and practical signature scheme for an IoT blockchain system (Sect. 3)
   We propose a new hierarchical signature scheme based on a fuzzy key and a short-term key. This scheme enables us to use biometric information as a user's private key and achieves strict verification of blockchain transaction creator.
2. Implementation and evaluation of our signature scheme (Sect. 4, 5)
   We implement our signature scheme for an IoT blockchain system and evaluate the practicality of our scheme.

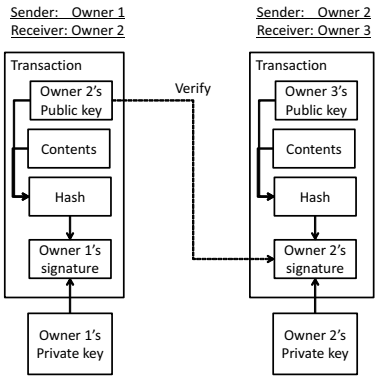

Fig. 1. An example of a Bitcoin transaction.

2 RELATED WORKS

2.1 BLOCKCHAIN

The Bitcoin [1] was proposed in 2009 and become widespread as a cryptocurrency. The core technology of the Bitcoin is blockchain which is a decentralized database. After the blockchain introduction with the Bitcoin, they applied blockchain to many types of cryptocurrencies and smart contract systems[2]. In this paper, we explain blockchain with the Bitcoin transaction as a simple example. In the other blockchain system, the model of a transaction is different from the Bitcoin's. However, the basic model of a transaction is common for the Bitcoin and the other blockchain systems. Thus we can apply our method to the other blockchain systems.

A transaction of the Bitcoin is shown in Fig. 1. In the Bitcoin system, a sender generates a transaction which includes sender's digital signature and receiver's public key. After this transaction generation, the transaction is verified whether it is valid payment or not by verifier (they are called "miner" in the Bitcoin). In this verification, the sender's digital signature is verified by the sender's public key in the previous transaction. The sender's public key in the previous transaction means that the sender has the Bitcoin, and the sender's digital signature means that the sender himself generates a payment transaction. Therefore, a verifier can confirm that the transaction is valid or not by sender's public key and a digital signature. This verification scheme is one of the core methods of blockchain.

In a typical blockchain, private keys are managed by users or membership servers to ensure security. However, private keys are at risk of leakage. When an adversary obtains a private key, it can generate arbitrary digital signatures, so the blockchain system becomes unsafe. There is a biometric authentication as a method of confirming the identity more reliably than the digital signature using the private key. For example, FIDO (Fast IDentity Online) [7] checks biometric information such as fingerprints, faces, irises and so on in secure hardware and then activates the private key. By linking such an authentication method

with blockchain, a secure blockchain system is realized. However, FIDO registers biometric information on a smart phone equipped with dedicated secure hardware and performs biometric authentication within its hardware. For this reason, when creating a signature, it is necessary to carry a smart phone with biometric information registered and to input biometric information to the smart phone. In our method, we use the fuzzy signature which can be used from any device without requiring dedicated secure hardware.

2.2 FUZZY SIGNATURE

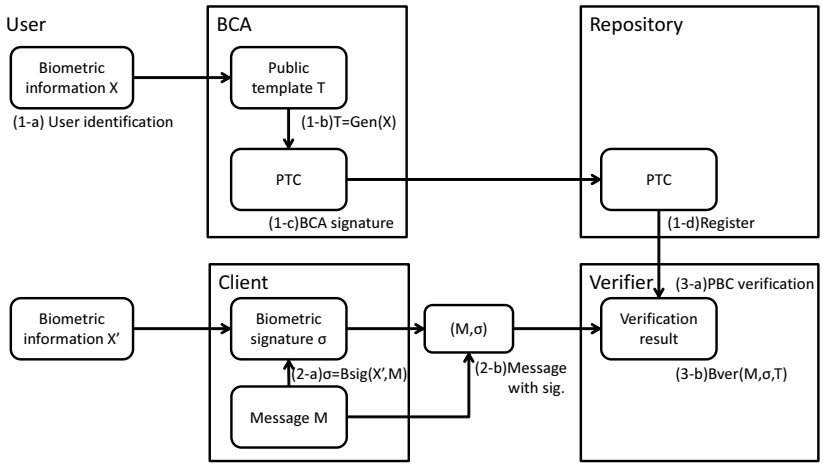

Fig. 2. The procedures of PBI.

In our proposed scheme, the fuzzy signature technology [5][6] is used for generating a blockchain transaction. We explain the procedures of the fuzzy signature technology in this subsection. The fuzzy signature technology is a digital signature technology which uses fuzzy data as a cryptographic key. In a conventional digital signature technology, we can use only fixed digital data as a cryptographic key. Therefore, we cannot use fuzzy biometric information such as fingerprint, face, finger-vein, and so on as a cryptographic key. By using the fuzzy signature technology, we can use fuzzy biometric information as a cryptographic key. We call a fuzzy signature generated based on biometric information as "biometric signature". For the detailed algorithm of the fuzzy signature technology, see [6].

By using the fuzzy signature technology, we can construct biometrics-based PKI (Public Key Infrastructure)[8] which uses biometric information as a user's private key. They call it the public biometrics infrastructure (PBI). The procedures of the PBI are shown in Fig. 2. The PBI requires a biometric certificate authority (BCA) and a repository in addition to the PKI components. In [5],

they propose a PBI construction method that realizes the PKI using biometric information as a user's private key. The procedures for registration, signature generation, and signature verification of the PBI using biometric signature are as follows:

1. Registration
   (a) The BCA confirms the identity of a user and then acquires user's biometric information X.
   (b) The BCA find $T = Gen(X)$. Here, T is a public template and $Gen(X)$ is a function for obtaining a public template from user's biometric information X.
   (c) The BCA issues a public template certificate (PTC) by giving a digital signature of the BCA to a set of information such as T, a user ID (UID), and an expiration date.
   (d) The BCA registers a PTC in the repository and publishes it.
2. Signature generation
   (a) A user (hereinafter referred to as "signer") generates a biometric signature $\sigma = BSig(X', M)$ from his biometric information X' and a plaintext M.
   (b) The signer transmits the pair of a plaintext and a biometric signature (M, σ) to a user who verifies a signature (hereinafter referred to as "verifier").
3. Signature verification
   (a) The verifier acquires a PTC of a signer from the repository, verifies a digital signature of the BCA attached to the PTC, and checks the expiration date of the PTC.
   (b) The verifier calculates a signature verification result $BVer(M, \sigma, T)$ from the plaintext M, the biometric signature σ, and the public template T included in the PTC. If a biometric signature is given to a plaintext M and the error between the biometric information X at registration and the biometric information X' at signature is less than a certain threshold, $BVer(M, \sigma, T) = 1$ (verification succeeded), otherwise $BVer(M, \sigma, T) = 0$ (verification failure). The successful verification means that a registered user and a signer are same persons.

In the PBI, there is no necessity to store a user's private key into a device or a cloud server. Moreover, they mathematically prove that anyone cannot estimate biometric information from a public template and a biometric signature. Thus the risk of forgery is significantly reduced in the PBI. By using the PBI, we can develop a secure signature platform.

3 A PROPOSED SCHEME

In this section, we propose a secure and practical signature scheme for an IoT blockchain system. By applying biometrics to a blockchain system, we can improve the security of a blockchain system. We propose two schemes: one is fuzzy key based signature scheme and the other is short-term key based signature scheme.

3.1 A Fuzzy Key Based Signature Scheme

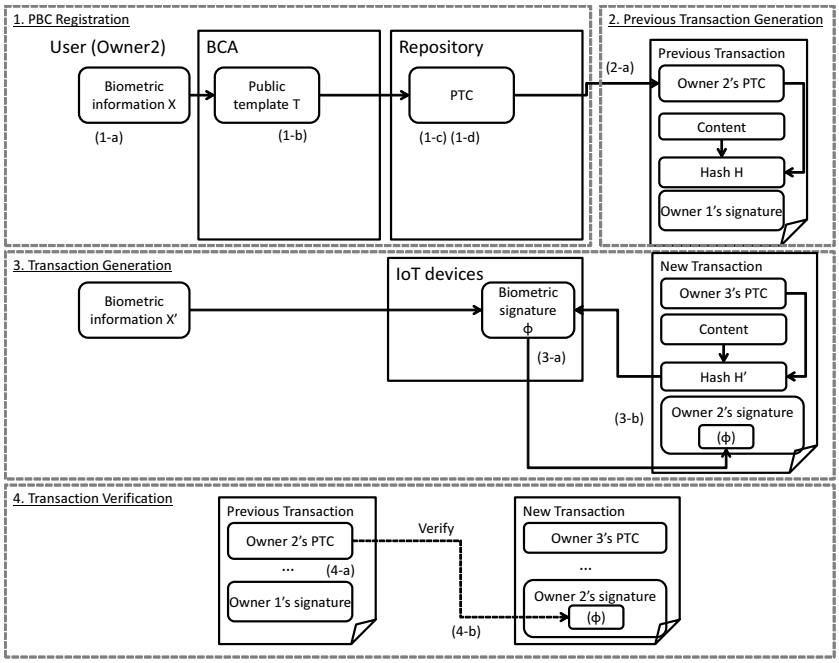

Fig. 3. The overview of the fuzzy key based signature scheme.

In this system, we apply the fuzzy signature technology [6] to the generation of a blockchain transaction. After generating the content of a new blockchain transaction, a user inputs his biometric information to an IoT device, and his biometric signature is attached to the blockchain transaction. A verifier of a blockchain system verifies a biometric signature of a blockchain transaction by a public template certificate (PTC). In this way, a verifier can confirm that a proper user creates a blockchain transaction. Therefore, there is no risk of successful forgery due to the theft of a user's private key.

The overview of the fuzzy key based signature scheme is shown in Fig.3. In this situation, the Owner 2 generates a new blockchain transaction. A detailed explanation of the fuzzy key based signature scheme is as follows.

1. PTC Registration
   This procedure is completely same as the PBI registration's one. See from (1-a) to (1-d) in Subsect.2.2.
2. Previous Transaction Generation
   This procedure is transaction generation from the Owner 1 to the Owner 2. The specific procedures of transaction generation are described in procedure 3.

(a) The Owner 1 sets the Owner 2's PTC to a blockchain transaction, and issues it.

3. Transaction Generation
   (a) The Owner 2 creates a new blockchain transaction which includes the Owner 3's PTC (a receiver's PTC), some contents, and their hash value H'. The Owner 2's biometric signature $\phi = BSig(X', H')$ is generated from the hash value H' using his biometric information X'.
   (b) The Owner 2 attaches the Owner 2's biometric signature ϕ to the blockchain transaction, and issues it.
4. Transaction Verification
   (a) A transaction verifier checks the expiration date of the Owner 2's PTC in the previous blockchain transaction and verifies the Owner 2's PTC by using the BCA's public key.
   (b) The transaction verifier calculates a signature verification result $BVer(H', \phi, T)$ for the hash value H', the biometric signature ϕ, and the public template T included in the PTC. If the biometric signature is given to the hash value H' and the error between the biometric information X at registration and the biometric information X' at signature is less than a certain threshold, $BVer(H', \phi, T) = 1$ (verification succeeded), otherwise it is $BVer(H', \phi, T) = 0$ (verification failure).

The fuzzy key based signature scheme need not store a user's private key in any devices or cloud servers. In this scheme, a user's biometric information acts as a user's private key. This means that a user can store his private key in his body. Therefore, we can prevent key theft and realize a highly secure blockchain system. Furthermore, the fuzzy signature generates a different PTC for each registration. Therefore, when the private key corresponding to a PTC leaks, the PTC can be updated in the same manner as the public key certificate of the PKI. However, in this method, it is necessary for a user to input biometric information every time he generates a blockchain transaction. Therefore, an IoT device cannot automatically create a blockchain transaction. Moreover, if a blockchain transaction is frequently generated, the usability of a blockchain system is reduced. To solve this problem, we propose a short-term key based signature scheme.

3.2 A Short-term Key Based Signature Scheme

In this method, a user generates a short-term key pair which consists of a short-term private key and a short-term public key in an IoT device. By attaching a user's biometric signature to a short-term public key, a user creates a short-term public key certificate (SPKC). He uses a short-term private key for generating a digital signature in a blockchain transaction. The validity of a blockchain transaction is confirmed based on three-phased hierarchical verification. The first one is PTC's verification by the BCA's public key. This phase confirms that the BCA issued a PTC. The second one is SPKC's verification by a PTC. This phase confirms that an SPKC is generated by a proper user. The third one

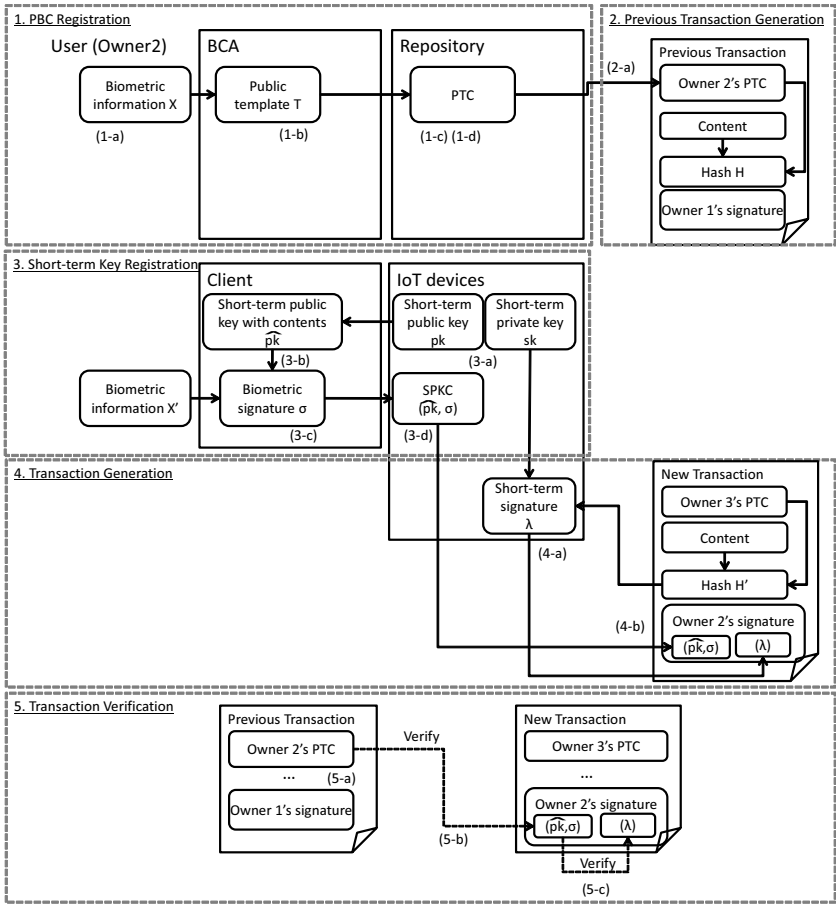

Fig. 4. The procedures of the short-term key based signature scheme.

is short-term signature's verification by an SPKC. This phase confirms that a blockchain transaction is created by using a correct short-term public key. This hierarchical verification allows a transaction verifier to verify that a proper user generated a blockchain transaction.

The overview of the short-term key based signature scheme is shown in Fig.4. A detailed explanation of the short-term key based signature scheme is as follows.

1. PTC Registration
   This procedure is completely same as the PBI registration's one. See from (1-a) to (1-d) in Subsect.2.2.
2. Previous Transaction Generation
   This procedure is entirely same as the fuzzy key based signature scheme's one. See (2-a) in Subsect.3.1.

3. Short-Term Key Registration
   In this procedure, a user generates a short-term private key and a short-term public key certificate (SPKC) and stores the keys on an IoT device.
   (a) An IoT device generates a short-term key pair which is a short-term private key sk and a short-term private key pk.
   (b) The Owner 2 creates a short-term public key with contents $\hat{pk}$ from the short-term public key pk, an expiration date, an issuer name, and so on. This information can be followed public key certificate standard X.509[9].
   (c) The Owner 2 inputs his biometric information X' and generates his biometric signature $\sigma = BSig(X', \hat{pk})$ from a short-term public key with some contents $\hat{pk}$.
   (d) An IoT device obtains an SPKC which includes a short-term public key with some contents $\hat{pk}$ and the biometric signature σ and stores it.
4. Transaction Generation
   (a) The Owner 2 creates a new blockchain transaction which includes the Owner 3's PTC (a receiver's PTC), some contents, and their hash value H' and generates a short-term signature $\lambda = Sig(H', sk)$ from the hash value H' and his short-term private key sk. Here, $Sig(A, B)$ is a function for obtaining a digital signature from a plaintext A and a private key B. Any digital signature algorithm such as RSA, DSA, ECDSA can be applied to this signature.
   (b) The Owner 2 attaches the SPKC $(\hat{pk}, \sigma)$ and the short-term signature λ to the new blockchain transaction and issues it.
5. Transaction Verification
   (a) A transaction verifier checks the expiration date of the Owner 2's PTC in the previous blockchain transaction and verifies the Owner 2's PTC by using the BCA's public key.
   (b) The transaction verifier calculates a signature verification result $BVer(pk', \sigma, T)$ for the short-term public key with some contents pk', the biometric signature σ, and the public template T included in the Owner 2's PTC. If a biometric signature is given to a short-term public key with some contents pk' and the error between the biometric information X at registration and the biometric information X' at signature is less than a certain threshold, $BVer(\hat{pk}, \sigma, T) = 1$ (verification succeeded), otherwise it is $BVer(\hat{pk}, \sigma, T) = 0$ (verification failure). The successful verification means that the SPKC is issued by a proper user.
   (c) The transaction verifier calculates a signature verification result $Ver(H', \lambda, pk)$ for the hash value H', the digital signature λ and the short-term public key pk. If a digital signature λ is valid, $Ver(H', \lambda, pk) = 1$ (verification succeeded), otherwise it is $Ver(H', \lambda, pk) = 0$ (verification failed). The successful verification means that a blockchain transaction is generated using a correct private key sk corresponding to pk.

   If all of the signature verifications (5-a), (5-b), and (5-c) are successful, transaction verification is successful. If one or more signature verification fails, transaction verification fails.

The short-term key based signature scheme stores a short-term private key in an IoT device. Therefore, there is a risk that an attacker steals a short-term private key and successfully spoofs a digital signature in a blockchain transaction. However, this risk can be reduced compared to the conventional private key based signature scheme.

For example, suppose that we set the validity period of a short-term public key certificate to one day. An IoT device can continually generate a blockchain transaction by user's updating of a short-term public key certificate once a day. In this case, spoofing will not succeed if it takes more than one day for a cyber attack, theft of an encrypted short-term private key, decryption of a short-term private key, and attack using the decrypted short-term private key.

Furthermore, with this method, the user does not need to input his biometric information every time an IoT device generates a blockchain transaction. Therefore, it is possible to achieve high usability than the fuzzy key based signature scheme.

4 DISCUSSION ON SECURITY

Table 1. The security of each signature scheme.

Signature scheme	(T1)	(T2)	(T3)
PKSS	-	**Low**	High
FKSS	High	High	High
SKSS	High	Middle - High	High

We discuss on the security of the proposed schemes and confirm their effectiveness. In this paper, "security" is defined as resistance to spoofing or signature forgery in a signature scheme. We address the threats of the blockchain system and discuss security against three signature schemes: the conventional private key based signature scheme (PKSS), our fuzzy key based signature scheme (FKSS) and our short-term key based signature scheme (SKSS).

(T1) Issuing a short-term public key certificate corresponding to a short-term private key of imposter user
This threat is that an imposter user's short-term public key certificate is issued as a genuine user's one. By using this imposter user's short-term public key certificate, the imposter user can forge genuine user's signature. In the PKSS, this threat does not occur, because we do not use a short-term public key certificate in this signature scheme. In the FKSS and the SKSS, there are three attack patterns against this threat: (T1-a) forcing a genuine user to issue an illegal short-term public key certificate of an imposter user, (T1-b) forging the biometric signature of a short-term public key certificate,

and (T1-c) issuing a short-term public key certificate of an imposter user by collusion between a genuine user and an imposter user.

In the thread (T1-a), there is an attack that an imposter user sends his short-term public key to a genuine user and asks him to generate his biometric signature to an attacker's short-term public key. By using this signed attacker's key as a short-term public key certificate, the attacker can forge a blockchain transaction. As countermeasures against this attack, there are two kinds of methods. One is that a user separates biometric information for each purpose (for example, he use a fingerprint of an index finger for signing to a document and use a fingerprint of a middle finger for issuing a certificate). The other is that a user adds signature purpose information (for example, signature to a document or issuing a certificate) to his biometric signature. In this way, biometric signatures assigned for different purposes cannot be used to issue a short-term public key certificate. Thus transaction verification is failed.

Concerning the threat (T1-b), if it is hard to forge a biometric signature, issuing an illegal short-term public key certificate is difficult. For example, the fuzzy signature proposed in [6] is CMA - EUF (Existential Unforgeability against Adaptive Chosen Message Attacks) which means that it is hard to forge a biometric signature. By using such a secure algorithm for biometric signature, we can sufficiently reduce a risk to this threat.

In the threat (T1-c), a genuine user intentionally issues a short-term public key certificate of an imposter user. The imposter user creates a genuine user's blockchain transaction by using the short-term public key certificate and a genuine user later denies that he generated a blockchain transaction. Concerning this attack, a genuine user issues a short-term public key certificate in a correct procedure. Thus it is difficult to prevent this attack using any signature scheme. Therefore, (T1-c) is out of our scheme's scope. The FKSS and the SKSS are safe against the threads (T1-a) and (T1-b). Thus the security of these schemes is high.

(T2) Private key leakage

This threat is that a user's private key leaks out from an IoT device, imposter user obtains it and illegally generates a blockchain transaction. This threat is caused by IoT device theft, cyber attack, and so on. In the PKSS, a long-term private key is managed in an IoT device or a cloud server. Therefore, there is a high risk that an attacker steals a private key and a forges a digital signature. In the FKSS, any private key is not managed in an IoT device. We use user's biometric information as a user's private key. Therefore, the FKSS is highly secure against the threat (T2). In the SKSS, we manage a short-term private key in an IoT device. Thus there is a risk that an attacker steals a private key and forges a digital signature. However, this risk can be significantly reduced compared to the PKSS. Since the SKSS allows a user to issue a short-term public key certificate, it is possible to shorten the expiration date of a short-term public key certificate.

For example, suppose that we set the validity period of a short-term public key certificate to one day. A user inputs his biometric information once a day

to an IoT device and issues a short-term public key certificate. As a result, the latest short-term public key certificate is always valid, so that an IoT device can generate a blockchain transaction continuously. Even if a short-term private key leaks out from an IoT device, we can sufficiently reduce the risk of illegal blockchain transaction generation by an imposter user. In other words, if an attacker takes a day or more to steal an encrypted short-term private key, decrypt it, and generate a blockchain transaction utilizing the decrypted short-term private key, the blockchain system based on the SKSS is secure. We judge this SKSS's security to be a middle to high level.

(T3) Forgery of digital signatures
This threat is to forge a digital signature for an arbitrary blockchain transaction and make the verification of a digital signature succeed. We can reduce the risk of this threat if we adopt a safe algorithm as a public key cryptography for generating a private key and a public key. In the PKSS and SKSS, if a secure signature algorithm that is difficult to be forged is used, these signature schemes are safe. In the FKSS, if we use a secure fuzzy signature algorithm [6] which has CMA - EUF (Existential Unforgeability against Adaptive Chosen Message Attacks) for generating a biometric signature, the forgery of a signature is significantly difficult.

From the above, the proposed FKSS and SKSS are safer than the conventional PKSS. Furthermore, when comparing the FKSS and the SKSS, the FKSS is more secure than the SKSS in that we do not store a short-term private key on an IoT device. Therefore, we recommend the use of the FKSS in fields where high safety is required.

5 DISCUSSION AND EXPERIMENTAL EVALUATION ON PRACTICALITY

5.1 DISCUSS ON USABILITY

Table 2. The usability of each signature scheme.

Signature scheme	Usability (Num. of user authentication)
PKSS	High(1)
FKSS	Low(mn)
SKSS	Middle(m)

In this paper, "Usability" is defined as a user's labor required to generate a blockchain transaction. Specifically, "Usability" is evaluated on the number of user authentications that is required for an IoT device to generate blockchain transactions continuously. Note that the "user authentication" includes inputting password, smart card, biometric information, and so on. The number of user

authentication is shown in Table2. Here we consider blockchain transaction generation for a specified time unit. For example, given a time unit as one day, the expected total number of blockchain transactions is expressed as mn using the number of days m and the average number of transactions per day n.

In the PKSS, a user performs authentication at an initial setting only. Thus the number of authentication is 1, and we can achieve high usability. In the FKSS, a user needs to authenticate on an IoT device each time it generates a blockchain transaction. The expected total number of user authentication is mn. Since this frequency is very high, the usability of the FKSS is low. In the SKSS, a user needs to authenticate on an IoT device each time unit, and the expected number of user authentication is m. This frequency is lower than that of the FKSS, and usability of the SKSS is the middle.

Furthermore, we compare the FKSS with the SKSS. In the FKSS, we require user's fuzzy signature generation each time an IoT device generates a blockchain transaction. For this reason, it is impossible to generate a blockchain transaction unless a user can input biometric information into an IoT device at the time. On the other hand, in the SKSS, if a user issues a short-term public key certificate once per unit time, an IoT device can continuously generate a blockchain transaction. Thus, the SKSS realizes higher usability than the FKSS.

5.2 EXPERIMENTAL EVALUATION OF IMPLEMENTABILITY

Table 3. Implementation results of each signature scheme.

Results		PKSS	FKSS	SKSS
File size	PTC	-	10 Kbyte	10 Kbyte
	Public key certificate	1 Kbyte	-	1 Kbyte
	Signature in a blockchain transaction	71 byte	71 byte	71 byte
Process time	PTC generation	-	499 msec	499 msec
	Short-term public key certificate generation	-	-	1306 msec
	Signature generation	78 msec	1306 msec	78 msec
	Signature verification	70 msec	70 msec	140 msec

EXPERIMENTAL SET-UP We implement the proposed methods and evaluate the size of files and processing time. We develop the fuzzy signature algorithm for finger-vein authentication [10]. Moreover, we use the ECDSA 256bit [11] as a digital signature algorithm in this evaluation. The ECDSA 256bit is utilized in the open source blockchain platform the Hyperledger Fabric [2].

EXPERIMENTAL RESULTS First, we evaluate the file size of a public template certificate (PTC), public key certificate and signature in a blockchain transaction. A PTC includes a public template for a finger-vein pattern, and the

file size of a PTC is 10Kbyte. This file size is larger than a traditional public key certificate's one (1 Kbyte). However, 10Kbyte is small enough for practical use. The file sizes of a public key certificate and signature are same in all methods, and they are 1Kbyte and 71byte, respectively.

Second, we evaluate the processing time for each signature scheme. The CPU and memory where we perform the evaluation are Intel Celeron N3050 1.6GHz and 4GB, respectively. This spec is too rich as an IoT device. However, we think that if sufficiently high-speed processing can be performed with this specification, practical processing time can be achieved even if IoT device processing is several times slower. PTC generation is executed one time in an initial user registration. Thus the processing time 499 msec is fast enough. Short-term public key certificate generation is performed every time unit (for example once a day). Thus, the processing time 1306 msec is also fast enough. We perform signature generation every blockchain transaction generation. In the PKSS and the SKSS, the processing time of signature generation is 78 msec, and this is significantly fast. In the FKSS, the processing time of signature generation is 1306 msec, and this is slower than the PKSS and the SKSS. However, the processing time is fast enough for practical use. We perform signature verification every blockchain transaction verification. In the SKSS, signature verification takes twice the time of the other schemes. However, 140 msec is fast enough comparing to the other blockchain procedures. In this way, you can see that the proposed schemes achieve practical file size and processing time. Therefore, we can use these schemes for a practical IoT blockchain system.

6 CONCLUSIONS

In this paper, we propose a secure and practical signature scheme for an IoT blockchain system based on biometrics. In the proposed scheme, the fuzzy signature is applied to generate a blockchain transaction. The fuzzy signature can use a user's biometric information as a user's private key. Since the proposed scheme requires biometric information at blockchain transaction generation, it is possible to achieve high security against spoofing and signature forgery. Therefore, our scheme can integrate blockchain and biometrics and achieve highly secure blockchain system. Moreover, we newly propose a short-term key based signature scheme. This method can achieve both blockchain security and usability. In the discussion and the experimental evaluation, we evaluate the security and the practicality of the proposed scheme, and the effectiveness of the proposed scheme is confirmed.

References

1. Satoshi Nakamoto.: Bitcoin: A Peer-to-Peer Electronic Cash System. Bitcoin.org (2009)
2. Cachin, Christian.: Architecture of the Hyperledger blockchain fabric. Workshop on Distributed Cryptocurrencies and Consensus Ledgers. (2016).

3. Da Xu, Li, Wu He, and Shancang Li.: Internet of things in industries: A survey. IEEE Transactions on industrial informatics 10.4 : 2233-2243 (2014).
4. Samaniego, Mayra, and Ralph Deters.: Blockchain as a Service for IoT. Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on. IEEE, (2016).
5. Takahashi, Kenta, et al.: A signature scheme with a fuzzy private key. International Conference on Applied Cryptography and Network Security. Springer, Cham (2015).
6. Matsuda, Takahiro, et al.: Fuzzy Signatures: Relaxing Requirements and a New Construction. International Conference on Applied Cryptography and Network Security. Springer International Publishing (2016).
7. FIDO Aliance, https://fidoalliance.org/
8. Nash, Andrew, William Duane, and Celia Joseph.: PKI: Implementing and Managing E-security. McGraw-Hill, Inc., (2001).
9. Myers, Michael, et al.: X. 509 Internet public key infrastructure online certificate status protocol-OCSP. No. RFC 2560. (1999).
10. Miura, Naoto, Akio Nagasaka, and Takafumi Miyatake.: Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. Machine Vision and Applications 15.4 : 194-203 (2004).
11. Johnson, Don, Alfred Menezes, and Scott Vanstone.: The elliptic curve digital signature algorithm (ECDSA). International Journal of Information Security 1.1 : 36-63 (2001).