Secure Pub-Sub: Blockchain-based Fair Payment with Reputation for Reliable Cyber Physical Systems

Yanqi Zhao, Yannan Li, *Student Member, IEEE,* Qilin Mu, Bo Yang*, *Member, IEEE,* Yong Yu*, *Member, IEEE*

Abstract—The cyber physical system (CPS) has gained considerable success in large-scale distributed integration environment. In such systems, the sensor devices collect data which would be disseminated via reliable manner to all interested co-operant entities from the physical world. However, highly unreliable environment of CPS, for example, a number of limitations of existing network middle wares, makes secure and reliable data distribution services a challenge issue. In this paper, we propose a new architecture called secure pub-sub (SPS) without middle ware, i.e., blockchain-based fair payment with reputation. In SPS, publishers publish a topic on the blockchain and subscribers specify an interest message by making a deposit to subscribing the topic. Then, if the interest message matches the topic, the publisher transmits the encrypted content of the topic to the blockchain such that the subscribers can decrypt the ciphertext to obtain the content, and mark the publisher as its reputation. Finally, the publisher receives the payment from the subscriber. The new proposal provides confidentiality and reliability of data, anonymity of subscribers and payment fairness between the publishers and subscribers. Different from the traditional pub-sub services, no trusted third party is involved in our system due to employing blockchain technique. The security of the proposed SPS is analyzed as well. The implementation of the protocol on Ethereum of smart contract demonstrates the validity of SPS.

Index Terms—Blockchain, Fairness, Reputation, Anonymity, Publish/subscribe, Cyber physical systems.

1 Introduction

Cyber physical system (CPS) [1]– [4] has gained significant success in large-scale distributed integration environment in recent years. As claimed by NSF, CPS is a system *where physical and software components are deeply intertwined, each operating on different spatial and temporal scales, exhibiting multiple and distinct behavioral modalities, and interacting with each other in a myriad of ways that change with context [5]*. In such systems, large amount of entities collaborate with each other to achieve certain goals. They collect data with sensors in physical world and feed the data to computing resources. Then, cooperative entities in CPS can make a decision with sharing data and knowledge. CPS makes the communication between physical and computer components more and more scalable and flexible, which has a variety of applications in smart grid, medical monitoring, process control systems, robotics systems etc. Other applications of CPS include unmanned vehicle groups, vehicular networks [6] and autonomous transportation systems that need large amount of entities to collaborate with each other. Secure and reliable data share services [7] is essential but difficult in these applications. Due to various reasons, temporary failures and noises in communication make the reliability of participating entities problematic. In addition, in existing network middle ware cannot handle the case when providing such services in CPS.

Pub-sub (publish/subscribe) services provide loose-coupling property and inherent asynchronous communication. There are three types of pub-sub services, namely topic-based, type-based and content-based. The topic-based pub-sub service is that a publisher publishes a topic and a subscriber matches the event of interest with this topic. The type-based pub-sub service is filtering events according to their type and the subscriber matches event type directly. In content-based pub-sub services, a subscriber expresses the event content with certain predicates and matches only satisfy such predicates. Wang et al. [8] and Esposito et al. [9] investigate the security issues and requirements of pub-sub service including confidentiality, integrity, availability etc. The primary threats in pub-sub system are confidentiality and reliability of data and anonymity of subscribers. In fact, the publisher submits the event of data and inspires provided confidentiality that only the subscriber can know it. Similarly, a subscriber wants to protect his privacy and anonymity from his subscriptions. For example, on social network, user's subscription may leak his habit, hobby and even his identity. For simplicity, we denote the publisher by P and the subscriber by S. A number of solutions have been proposed to solve the issues of confidentiality and anonymity between P and S.

Related Work. Tariq et al. [10], [11] presented an approach to offering confidentiality and authentication in broker-less publish/subscribe service by using identity-

* *Corresponding Author*

- *Yanqi Zhao, Bo Yang and Yong Yu are with School of Computer Science, Shaanxi Normal University, Xi'an, 710062, China.*
  E-mail: yuyong@snnu.edu.cn
- *Yannan Li is with the Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia.*
- *Qilin Mu is with National Engineering Laboratory for Big Data Application on Improving Government Governance Capabilities, Guiyang, Guizhou, 550081, China.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2799205, IEEE Access

based signcryption [12] to sign and encrypt the data. Additionally, this approach brings fine-grained key management. The proposals in [13], [14] follow this approach to constructing secure publish/subscribe system by using fuzzy identity-based encryption or hierarchical identity-based encryption (HIBE). Anusree et al. [15] realized confidentiality in the broker-less publish/subscribe service along with forward secrecy and unforgeability by using Elliptic Curve identity based signcryption technique [16]. Malpure et al. [17] suggested to utilize attribute-based encryption (ABE) to provide secrecy and authentication in broker-less publish/subscribe services. Shitole et al. [19] used the Elliptic Curve Cryptography (ECC) algorithm to reduce the computational cost and the memory cost. Furthermore, this system supports dynamic news in which the subscriber can send updated requests and the publisher can validate the update. Ion et al. [20] combined attribute-based encryption and searchable encryption to design and implement a novel publish/subscribe system. It supports data confidentiality and access control such that only authorised parties can access it. Khoury et al. [21] proposed P3S, a novel system architecture to protect confidentiality of published data and the privacy of subscriptions. Yang et al. [22] proposed attribute-keyword based data publish/subscribe (AKPS) to protect the privacy of the published data by using a new dual-policy framework supporting multiple publishers and subscribers.

To protect subscriber's anonymity in publish/subscribe services, Daubert et al. [23] presented a new method called Probabilistic Forwarding (PF) whose core is shell game algorithm. Lee et al. [24] introduced the notion of anonymous subscription with conditional linkage, where the subscriber can anonymously access to modern web services. Yuen et al. [25] suggested a new security requirement called publisher authenticity which requires only the authenticated publisher can publish certain types of events, and identity-based signatures were used to achieve publisher authentication.

Bitcoin, a decentralized cryptocurrency, was proposed by Satoshi Nakamoto [27] in 2008. Blockchain, a hash-based data structure, is the core technology of bitcoin. Each block has a block header, a hash pointer to the previous block and a Merkle hash tree (MHT) that digests of transactions in the block in an efficient way. A trusted third party is not needed but the majority of peer nodes in this network are honest to resist 51% attack. The decentralization of blockchain can benefit many applications such as distributed storage systems [28], [29], micropayment [30], secure multiparty computation [31]– [34] etc.

Payment service is an integrant of some CPS systems such as smart grid, autonomous transportation systems, medical monitoring and automatic pilot avionics. Payment fairness without a trusted third party while keeping content's confidentiality and user's anonymity is highly needed in these applications. In the payment scenario, the publisher P receives payment from the subscriber S and S can receive the content of interested subscriptions. However, P and S do not trust each other in the payment system, which leads to a number of urgent and important issues including confidentiality of the content, anonymity of subscriber, payment fairness to be addressed.

Contributions. To solve the aforementioned problems, in this paper, we propose a new protocol called secure pub-sub (SPS), blockchain-based fair payment with reputation for reliable CPS. In SPS, publishers take advantage of hybrid encryption to guarantee the confidentiality of data, encrypt the data and transmit the ciphertext to matching subscriber. Subscribers receive the notification and decrypt the ciphertext to obtain the content. Further, weak anonymity of the subscribers is guaranteed by borrowing Bitcoin's pseudonym system. As the publisher and the subscriber do not trust each other, we take advantage of blockchain to make sure that an honest publisher can get the deposit of a malicious subscriber and an honest subscriber can withdraw deposit for malicious publisher. We use the reputation system such that a subscriber can evaluate the published event and mark the publisher based on his reputation. Finally, the smart contract validates our protocol, which can be used to the CPS for secure data sharing in general payment applications.

Paper Organization. The remainder of this paper is organized as follows. We give some preliminaries in Section 2, and present protocol model in Section 3. The detailed SPS protocol is given in Section 4. The security analysis and evaluation of the protocol are given in Section 5. Finally, we conclude the paper in Section 6.

2 Preliminaries

In this section, we review the publish/subscribe system, bitcoin system, smart contract and ethereum that will be used in this paper. Then, we describe the framework of reputation system.

2.1 Publish/subscribe system

As shown in Fig. 1, there are three participants in a traditional publish/subscribe system namely the publisher, the subscriber and the broker. The publisher publishes a message event and the subscriber submits subscription that is a message of interest. Then, the broker matches the subscription and the event between the subscriber and the publisher. With the help of the broker, the publish/subscribe network routes and forwards the packets of the event to the matching subscriber.

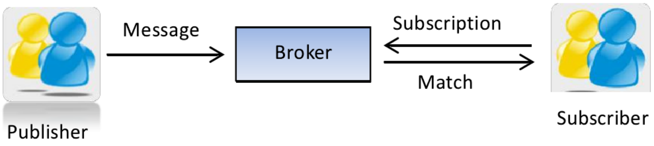

Fig. 1: Publish/subscribe system

2.2 Bitcoin system

We review two important tricks of bitcoin system called address and transaction. Bitcoin system makes use of public keys of a user rather than names as his/her pseudonyms, and the address is the hash value of the user's public key. Each user has a key pair: a private key SK and a public key PK. SK is used to sign a transaction and PK is used to validate the signature of the transaction. If a subscriber S wants to pay a publisher P d bitcoins, he makes a transaction $T_x = (T_y, PK_P, d, t, Sig_S(T_y, PK_P, d, t))$ where T_y denotes the previous transaction with the value at least d and no double-spending. (T_y, PK_P, d, t) denotes by *body*, and t is the lock time. If the signature is correct, the transaction is valid.

In the real bitcoin system, the transaction is flexibly defined by using the input-script and output-script. The scripting language of bitcoin is a stack based language [35] which is not turing-complete, with no loops. It supports the operations of hash function and basic signature algorithm. Table 1 illustrates a standard transaction.

T_x(in: T_y)
In-script: σ_S
Out-script($body, \sigma$):
$Ver_P(body, \sigma)$
Value: d
Lock time: t

TABLE 1: Transaction T_x

In transaction T_x, T_y denotes the previous transaction. In-script is the signature of subscriber S on *body* of $[T_x]$. Out-script denotes the verification statement and the value is d bitcoins. Lock time indicates the transaction is valid only after time t.

2.3 Smart contract and Ethereum

Smart contract was proposed by Nick Szabo [36], which has been expanded to blockchain. It is self-executing programming code with implementated digital contract in a secure environment. The ethereum [37] is a platform and a programming language that makes developer build next-generation distributed applications. It runs exactly as programmed without any possibility of downtime, censorship, fraud or third party interference. Ethereum can be used for programming, scattered, guarantee and trading anything such as vote, financial exchanges, the company management, contract most of the agreements and the intellectual property rights.

2.4 Reputation system

Our protocol considers the behavior of a publisher and a subscriber for the reliability of the data. If a publisher has high reputation value, the subscriber can subscribe his topic. This is an incentive mechanism and an effective approach to monitoring the publisher. In our protocol, we set θ as the threshold value of reputation. If a publisher's reputation value R is greater than θ, the publisher is believable. We utilize the method which is used in social network [38] to instantiate a reputation system.

Reputation value assessment uses multi-dimensional way. When a publisher publishes the topic of message event, we compute the sum of score of explicit rating (from 0 to 1 score) and implicit rating as its reputation value. The two rating have different weight value α, β, respectively. The explicit rating is the direct evaluation for the message event. The implicit rating is based on the subscriber of positive activity and negative activity. The score of positive and negative activity is shown in Table 2.

Type	Activity	Score
Positive	Continuous subscription	1.0
	Recommend a subscription	0.75
	Into favorite	0.5
	Like	0.25
	Praise	0.1
Negative	Blacklist	-1.0
	Not recommend	-0.5
	Not like	-0.25

TABLE 2: Score table

The implicit rating scores are calculated separately for positive activity score and negative activity score. P represents the positive activity score. It is derived by summing the score Pe_{jk} of n_e positive activities for m_d subscribers who mark the publisher.

$$P = \sum_{j=1}^{m_d} \sum_{k=1}^{n_e} Pe_{jk}.$$

The negative activity score N is calculated by incorporating the scores of negative activity. The negative activity score N is the sum of score Ne_{jk} of n_e negative activities for m_d subscribers who mark the publisher.

$$N = \sum_{j=1}^{m_d} \sum_{k=1}^{n_e} Ne_{jk}.$$

The implicit rating E is calculated based on the positive activity score P and the negative activity score N. The log function is used to alleviate the problem that the implicit rating values dramatically change for only a few rating values.

$$E = (\frac{1}{2}e^{\frac{P}{m_d}} + \frac{1}{2}e^{\frac{N}{m_d}} - 1).$$

The explicit rating D is calculated based on the score of Dr_j for m_d explicit evaluation scores.

$$D = \frac{1}{m_d} \sum_{j=1}^{m_d} Dr_j.$$

The final reputation value R is computed by $R = \alpha D + \beta E$. For simplify, we define encapsulation function $\mathcal{F}_R$ to express marking the publisher. As shown in Table 3, $\mathcal{F}_R$ is used to derive the reputation value. First, set the reputation list to be empty. Publisher submits topic to obtain the initial reputation value and update the reputation list. The subscriber can use the reputation list and submit subscription. Finally, the subscriber returns the reputation value.

3 Components and security model of SPS

In this section, we describe the components of our protocol and its security model.

3.1 SPS model

As shown in Fig. 2, SPS has multiple publishers, logical sensors and multiple subscribers. Firstly, the publishers collect data stream with sensors and publish a topic. The subscribers submit subscription, and the publisher matches the topic and the subscription. Then, the publisher sends

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2799205, IEEE Access

Initialize: Set the reputation list to be empty.
Publish/subscribe: Publisher submits topic and obtains the initial value. According to the reputation list, the subscriber submits subscription.
Reputation: The subscriber gives the reputation rating as the feedback of subscription.

TABLE 3: Function $\mathcal{F}_R$ as the reputation function

the matching event into bitcoin system. The subscriber receives the notification to analyze the situation and provides feedback to control the physical processes. At the same time, he pays to publisher based on reputation rating. Finally, the publisher who publishes the topic can obtain bitcoins and reputation value.

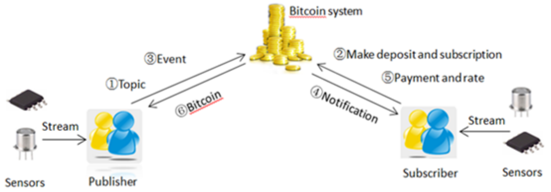

Fig. 2: SPS model

Definition 1. A SPS is a tuple of polynomial time algorithms $SPS = (Setup,\ Publish,\ Subscribe,\ Match,\ Verification\ \ and\ Payment)$ such that:

- $(params) \leftarrow Setup(1^{\kappa})$: This is a probabilistic algorithm that takes a security parameter κ as input and outputs public parameter $params$.
- $(Topic, R, T_P) \leftarrow Publish(params, \mathcal{F_R})$: The algorithm takes public parameter $params$ as input and runs reputation function $\mathcal{F_R}$. Then, it outputs the reputation value R, $Topic$ and makes transaction T_P.
- $(T_S, F) \leftarrow Subscribe(params, \mathcal{F_R}, Topic)$: The algorithm takes $Topic$ and $params$ as input and runs reputation function $\mathcal{F_R}$. Then, it outputs the transaction T_S for the subscription F.
- $(T_{P'}, C) \leftarrow Match(params, M, Enc)$: The algorithm takes $params$ and M as input and uses the symmetric encryption algorithm Enc to encrypt the message M. Then, it generates the ciphertext C and matches transaction $T_{P'}$.
- $(T_{S'}, R) \leftarrow Verification\, and\, Payment(params, M, \mathcal{F_R})$: The algorithm takes public parameter $params$ and M as input and runs reputation function $\mathcal{F_R}$. Then, it outputs reputation value R and transaction $T_{S'}$.

3.2 Security requirements

In the following, we will consider security properties of our SPS, which are defined the same as in broker-less publish/subscribe [10].

Confidentiality. In a broker-less environment, the message transmitted from publisher to subscriber is protected from illegal modification. The subscriptions of subscriber are confidential.

Authentication. In SPS, only the authorized publisher is able to publish events to system to any third parties.

Scalability. In SPS, the number of subscribers should be scalable.

Completeness. The completeness says that if an honest publisher P and an honest subscriber S perform the protocol, the honest publisher P can obtain bitcoins with reputation value and the honest subscriber S can receive required data. Moreover, the honest subscriber S can redeem his deposit.

Fairness. If a malicious subscriber S executes the SPS protocol, he cannot get his deposit back and an honest publisher can obtain the deposit of subscriber. If a malicious publisher executes the SPS protocol, his reputation value will be set to zero.

Anonymity. A subscriber executes the SPS protocol, nobody can link the pseudonym to his real identity, such as the his ID-card number, telephone number etc.

3.3 Potential attacks

The following attacks might exist in a SPS system.

Denial of service attack: An attacker can post a mass of events to network layer and make the system crash.

Unfair mark attack: Same as sybil attack, Unfair mark attack states that an attacker forges lots of subscribers to give the reputation score.

Collusion attack: Many subscribers may get together to give the reputation rating. This is an extension of unfair rating attack.

Re-entry attack: An attacker has many malicious entries to the network. When he has low reputation value, he can register as new publisher.

4 Blockchain-based fair payment with reputation

4.1 Overview

In our SPS protocol, we focus on confidentiality of the content, anonymity of subscriber, and fair payment between publisher and subscriber with the reputation value to efficient reduce the time of subscription and raise reliability of system. First, a publisher collects data stream and publishes his topic to bitcoin system. Then, a subscriber makes deposit and utilizes the ElGamal encryption scheme [39] to encrypt the topic and posts subscription to publisher. After that, the publisher matches the topic and the subscription. He encrypts the data with symmetric encryption algorithm and sends the ciphertext of data to subscriber. Finally, subscriber will pay and mark the publisher. Meanwhile, subscriber decrypts the ciphertext and analyzes the data and provides feedback to control the physical process. In SPS protocol, the subscriber must pay deposit before submitting a subscription. The deposit is denoted by b bitcoins in the transaction T_S. We suppose the deposit value always higher than the real payment. It can be redeemed when a subscriber pays for his subscription.

4.2 The detailed construction

We present our SPS protocol, which is based on ElGamal encryption scheme [39]. In SPS protocol, we assume every party has ECDSA key pair (PK, SK) and $Sig_P(m)$ is used to denote that using the secret key of P to sign on message m. The SPS consists of following algorithms, $Setup$, $Publish$, $Subscribe$, $Match$, $Verification\ and\ Payment$.

- $Setup$: It inputs security parameters κ and generates the public parameters of system as follows. Uniformly and independently choose prime p, q, where $q|p-1$ and one generator $g \in G_q$. Choose a random value $k \leftarrow \{0,1\}^\kappa$, one collision-resistent hash function $H : \{0,1\}^* \rightarrow G_q$, and function $H_1 : \{0,1\}^\kappa \rightarrow G_q$ which is efficiently invertible.
- $Publish$: A publisher uses the function $\mathcal{F_R}$ to generate his reputation value. He randomly chooses $x \in Z_q$, and computes $y = g^x$. He embeds $y, Topic$ into transaction T_P and sends it into bitcoin system

T_P(in:T_y)
In-script:$Sig_P[T_P], \mathcal{F}_R, y, Topic$
Out-script($body, \sigma_1$):
$Ver_P(body, \sigma_1)$

TABLE 4: Transaction T_P

- $Subscribe$: When a subscriber senses the posted topic, he makes deposit by generating a timed commitment with locked time t and value b and submits a subscription for his interest. At first, the subscriber picks $Topic$, and computes $w = H(Topic)$. Then, he randomly chooses $r_0 \in Z_q$, and computes the subscription $F = \{h, v_1\}$ as $h = g^{r_0}, v_1 = y^{r_0}w$. Finally, the subscriber generates a transaction T_S, and forwards it to the publisher.

T_S(in:T_x)
In-script:$Sig_S[T_S], F$
Out-script($body, \sigma_1, \sigma_2$): $Ver_S(body, \sigma_1) \bigvee$
$(Ver_S(body, \sigma_1) \wedge Ver_P(body, \sigma_2))$
Value:b

TABLE 5: Transaction T_S

- $Match$: Upon receiving the subscription, the publisher verifies $H(Topic') \overset{?}{=} \frac{v_1}{h^x}$. He randomly chooses $k_1 \in \{0,1\}^\kappa$, $r_1 \in Z_q$ and computes $k_1' = H_1(k, k_1)$, $z_1 = g^{r_1}$, $z_2 = h^{r_1} \cdot k_1'$. Then, the publisher uses the symmetric encryption algorithm to encrypt the message M, and generates the ciphertext $C = Enc(k_1, M)$. D denotes the matched information $\{z_1, z_2, C\}$, and L denotes the hash value of subscription F. Finally, the publisher sends the transaction T_M to the subscriber.

T_M(in:$T_{x'}$)
In-script:$Sig_P[T_M], F, D, L$
Out-script($body, \sigma_1$):
$Ver_S(body, \sigma_1) \bigwedge L = H(F) \bigwedge D$

TABLE 6: Transaction T_M

- $Verification\ and\ Payment$: When the subscriber receives the transaction T_M, he checks its correctness. First, the subscriber computes $k_s = z_2 z_1^{-r_0}$, $k_s' = H_1^{-1}(k, k_s)$. Then, he uses k_s' to decrypt $M = Dec(k_s', C)$. After the subscriber claims the message M, he will pay to the publisher by posting transaction T_{Pay} (Table 7) to bitcoin system and redeem the deposit with transaction T_{Rec} (Table 8). At the same time, he runs the function $\mathcal{F_R}$ to mark the publisher.

T_{Pay}(in:$T_{x''}$)
In-script:$Sig_S[T_{Pay}], \mathcal{F}_R, L, K$
Out-script($body, \sigma_1$):
$Ver_P(body, \sigma_1)$
Value:d'

TABLE 7: Transaction T_{Pay}

T_{Rec}(in:T_S)
In-script:$Sig_S[T_{Rec}]$
Out-script($body, \sigma_1$):
$Ver_S(body, \sigma_1)$
Value:b
Lock time:t

TABLE 8: Transaction T_{Rec}

If a publisher's reputation $R > \theta$, where θ is the threshold value of the reputation system, we assume the publisher is trusted and he can construct the transaction T_P to publish a *Topic* without making any deposit. Otherwise, we assume the publisher with reputation value less than θ is not fully trusted, then he needs to construct transaction $T_{P'}$ (Table 9) with the deposit of d' bitcoins in $Publish$.

$T_{P'}$(in:T_y)
In-script:$Sig_P[T_{P'}], \mathcal{F}_R$
Out-script($body, \sigma_1, \sigma_2$):
$Ver_S(body, \sigma_1) \bigvee Ver_P(body, \sigma_2)$
Value:d'

TABLE 9: Transaction $T_{P'}$

In our SPS protocol, we also consider the situation that when the reputation value for a deposit publisher increases over the threshold value θ, then he can redeem the deposit using transaction $Fasu_P$ (Table 10).

$Fasu_P$(in:$T_{P'}$)
In-script:$Sig_S[Fasu_P], \mathcal{F}_R, \perp$
Out-script($body, \sigma_2$):
$Ver_P(body, \sigma_2) \bigwedge \mathcal{F}_R$
Value:d'
Lock time:t_1

TABLE 10: Transaction $Fasu_P$

Moveover, we consider two special cases in our system: the first one is that the subscriber is malicious because he does not pay for the received message and the other one is that the publisher is not fully trusted, in which he does not have the claimed message.

$T_{S'}$(in:T_S)
In-script:$Sig_S[T_{S'}], Sig_P[T_{S'}], \perp$
Out-script($body, \sigma$):$Ver_P(body, \sigma)$
Value:b
Lock time:t_2

TABLE 11: Transaction $T_{S'}$

Case 1: If the subscriber is not trusted, the honest publisher can submit transaction $T_{S'}$ to receive the deposit of subscriber. In transaction $T_{S'}$ (Table 11), the lock time is t_2.

Case 2: If the publisher is malicious, the subscriber can obtain the publisher's deposit by posting transaction Get_T (Table 12). And as a punishment, the publisher's reputation value will be set zero.

Get_T(in:$T_{P'}$)
In-script:$Sig_S[Get_T], \perp$
Out-script($body, \sigma_1$):$Ver_S(body, \sigma_1)$
Value:d'
Lock time:t_3

TABLE 12: Transaction Get_T

Correctness: In our protocol, the subscriber receives the ciphertext C of data M, which can be correctly decrypted. $z_1 = g^{r_1}, z_2 = h^{r_1} \cdot k_1', C = Enc(k_1, M), Topic = Topic'$. The subscriber computes

$$\begin{aligned} k_s &= z_2 z_1^{-r_0} \\ &= (h^{r_1} \cdot k_1') \cdot (g^{r_1})^{-r_0} \\ &= h^{r_1} k_1' g^{-r_0 r_1} \\ &= g^{r_0 r_1} k_1' g^{-r_0 r_1} \\ &= k_1' \end{aligned}$$

which implies $k_s' = H_1^{-1}(k, k_1') = H_1^{-1}(k, k_s) = k_1$. Then, he decrypts $M = Dec(k_s', C)$. Thus, our protocol is correct for honest publisher and subscriber.

Remark 1. In our SPS protocol, we assume that bitcion system contains enough honest miners in which 51% attack is unavailable. The blockchain is a secure environment, it has enough bandwidth to prevent denial of service attack. For transaction $T_{S'}$ and T_{Rec}, there is a deadline $t_2 < t$. For transaction Get_T and $Fasu_P$, there is a deadline $t_3 < t_1$. We assume that the messages to be authenticated is signed by the party to avoid tampering.

Remark 2. As for unfair mark attack, we consider marking the unfair score for event. In order to handle this situation, we use the statistical method to compute standard deviation for screening. For standard deviation $SD_i(Topic_i, E)$ of each topic i, we compute

$$Fr(s) = \frac{\sum_{i=1}^{l} SD_i(Topic_i, E)}{l},$$

where l denotes the sum of score for event. If $Fr(s) > \rho_{Fr}$ that is unfair mark, where ρ_{Fr} is the threshold value.

Remark 3. According to the reputation system, we will base on the average score of all reputation values to deal with collusion attack. A method to prevent publisher reentry attack is to link the IP address to the publisher as unique identification. In our SPS protocol, the blockchain is a secure environment that the number of subscribers are scalable. Only the authorized publisher can publish the topic and obtain the reputation value.

5 SECURITY AND IMPLEMENTATION

In this section, we firstly analyze the security of SPS, and then report its performance.

5.1 Security analysis

We give four lemmas to demonstrate the security of the proposed SPS protocol.

Lemma 1. Our SPS protocol satisfies the property of confidentiality.

Proof: In SPS, publisher P encrypts message M with the symmetric encryption algorithm and generates the ciphertext $C = Enc(k_1, M)$. Then, subscriber S decrypts the data by $M = Dec(k_s', C)$ where $k_s' = k_1$. The message is confidential if the underlying symmetric encryption algorithm is a secure algorithm. The confidentiality of subscription is protected by leveraging ElGamal encryption scheme. When the publisher receives the subscription, he computes $H(Topic') \overset{?}{=} \frac{v_1}{h^x}$ and obtains the topic. By the IND-CPA security of ElGamal Encryption [40], we can achieve the confidentiality of data and subscription.

Lemma 2. Our SPS protocol satisfies the property of completeness.

Proof: In normal case, when publisher P and subscriber S perform the protocol, the publisher P will gain the bitcoin with reputation value and the subscriber S receives data and marks the publisher. At last, the subscriber S can redeem his deposit.

Lemma 3. Our SPS protocol satisfies fairness.

Proof: Firstly, we consider that subscriber S is dishonest and publisher P is honest. In this case, the subscriber S can get the notification of data and deposit back before time t. We assume the subscriber can get the notification and pay nothing to publisher P. But he cannot redeem deposit before time t. As mentioned in case 1, the publisher P can obtain the deposit of the subscriber's. In this case, dishonest subscriber gets a contradiction. So, the probability for a cheating subscriber winning in this case is negligible.

Then, we consider the case that subscriber S is honest but publisher P is dishonest. As for the low reputation value of publisher, his reputation value will be set zero. The subscriber S may put the publisher into blacklist. At the same time, based on case 2 the subscriber S will get the deposit. We say that is contradictory with getting bitcoin and high reputation value. So the probability for a malicious publisher winning in this case is negligible. Therefore, our protocol satisfies the security of requirement of fairness.

Lemma 4. Our SPS protocol achieves anonymity.

Proof: In our protocol, we use the bitcoin pseudonym system to construct the SPS protocol. It can obtain weak anonymity by using the pseudonym mechanism. The publisher P and subscriber S can not link the pseudonym to their real identities. Thus, our blockchain-based SPS satisfies the property of anonymity.

Scheme	Confidentiality	Verifiability	Anonymous	No Turst	Fairness	Reputation
Tariq [10]	√	√	×	×	−	−
Tariq [11]	√	√	×	×	−	−
Huang [41]	√	√	√	×	√	×
Our protocol	√	√	√	√	√	√

TABLE 13: Comparing with other schemes

5.2 Performance analysis

In this section, we compare the performance of our protocol with the protocols proposed recently. Table 13 shows the comparison among the three schemes. Tariq et al. [10], [11] are broker-less publish/subscribe service and Huang et al. [41] is outsourcing computation scheme. Tariq et al [10], [11] and our protocol protect the confidentiality of data that is verifiable. However, Tariq et al [10], [11] do not consider the anonymity of subscriber and fair payment problem. Huang et al. [41] solves fair payment problem by involving semi-trusted third party.

5.3 Performance evaluation.

In this section, we provide performance evaluation of the proposed protocol. The complexity analysis of our protocol is shown in Table 14, where Exp denotes exponentiation in G_q and Mul denotes multiplications in G_q. It is executed on intel(R) Core(TM) i5-4590S CPU3.00GHz with 4.00GB of RAM and Miracl library. The Table 15 shows the gas cost of the SPS. We implement it on intel(R) Core(TM) i5-2450M CPU2.50GHz and 4.00GB of RAM with Ethereum in solidity code, a programming language for writing contracts on Ethereum [42]. We execute it on a private test network for smart contract with Ethereum wallet by Solidity on the Web3j [1]. The gas cost be used in smart contract which are provided in Table 15 that estimates for gas cost with deploy new contract and running different functions of the SPS. As of June, 2017, gas price is 0.02ether per million gas. Bitcoin performance analysis, we reference the btc-testnet [43] to simulate the bitcoin transaction.

Algorithm	Computation Cost	Estimation
Publish	$1Exp$	0.000534s
Subscribe	$2Exp + 1Mul$	0.001096s
Match	$3Exp + 2Mul$	0.001658s
Verified and payment	$1Exp + 1Mul$	0.000562s

TABLE 14: The complexity analysis of SPS

Function	Gas units	Gas cost(ether)
Deploy contract	473715	0.0094743
Publish	89027	0.00178054
Subscribe	107581	0.00215162
Match	111471	0.00222942
Verification and payment	48316	0.00096632

TABLE 15: Gas cost of the SPS

6 Conclusion

In this paper, we consider data security and privacy problem for reliable CPS, and propose SPS that fairness payment with reputation based on blockchain. The publisher and subscriber can fairly exchange their items while providing confidentiality of data and anonymity of subscriber. We use hybrid encryption to guarantee the confidentiality of data. We take advantage of bitcoin system for SPS fairness, enabling malicious subscribers can not gain the deposit back and malicious publishers be punished by setting zero for his reputation value. Meanwhile the reputation value reduces the time of subscription and raises reliability of CPS.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61772326, 61572303), NSFC Research Fund for International Young Scientists (61750110528), National Cryptography Development Fund during the 13th Five-year Plan Period (MMJJ20170216), the Fundamental Research Funds for the Central Universities(GK201702004), the National Key R&D Program of China(No.2017YFB0802000), and the Project of Basic Research of Qinghai Province(2016-ZJ-776). Bo Yang and Yong Yu are the corresponding authors.

References

[1] Sha L, Gopalakrishnan S, Liu X, and Wang Q, ”Cyber-physical systems: A new frontier, in Machine Learning in Cyber Trust”. New York, NY, USA: Springer-Verlag, 2009, pp. 3-13.

[2] Horvth I and Gerritsen B., ”Cyber-physical systems: Concepts, technologies and implementation principles”, in Proc. TMCE Symp., 2012, pp. 19-36.

[3] Khaitan S K, McCalley J D., ”Design techniques and applications of cyberphysical systems: A survey”. IEEE Systems Journal, 2015, 9(2): 350-365.

[4] Ilic M. D., Xie L., Khan U. A., Moura J. M., ”Modeling of future cyberCphysical energy systems for distributed sensing and control”. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2010, 40(4), 825-838.

[5] National Science Foundation (NSF), ”Cyber Physical Systems NSF10515”, Arlington, VA, USA, 2013. [Online]. Available: http://www.nsf.gov/pubs/2010/nsf10515/nsf10515.html

[6] Reichardt D., Miglietta M., Moretti L., Morsink P., Schulz, W., ”CarTALK 2000: Safe and comfortable driving based upon inter-vehicle-communication”. In Intelligent Vehicle Symposium, 2002. IEEE Vol. 2, pp. 545-550.

[7] Kang W, Kapitanova K, Son S H. ,”RDDS: A real-time data distribution service for cyber-physical systems”. IEEE Transactions on Industrial Informatics, 2012, 8(2): 393-405.

[8] Wang C, Carzaniga A, Evans D, et al., ”Security issues and requirements for internet-scale publish-subscribe systems”. System Sciences, 2002: 3940-3947.

[9] Esposito C, Ciampi M., ”On security in publish/subscribe services: a survey”. IEEE Communications Surveys & Tutorials, 2015, 17(2): 966-997.

[10] Tariq M A, Koldehofe B, Altaweel A, et al., ”Providing basic security mechanisms in broker-less publish/subscribe systems”. In: Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems. ACM, 2010: 38-49.

[1] Solidity on the Web3j. https://ethereum.github.io/browser-solidity/ #version= soljsonv0.4.11+commit.68ef5810.js.

[11] Tariq M A, Koldehofe B, Rothermel K., ”Securing broker-less publish/subscribe systems using identity-based encryption”. IEEE transactions on parallel and distributed systems, 2014, 25(2): 518-528.

[12] Yu Y, Yang B, Sun Y, et al., ”Identity based signcryption scheme without random oracles”. Computer Standards & Interfaces, 2009, 31(1): 56-62.

[13] Maithily B, Swathi Y., ”Securing Brooker-less Publish/Subscribe System using Fuzzy Identity-Based Encryption”. International Journal of Computer Science and Information Technologies, 2015, 6(3): 2823-2826.

[14] Terkhedkar A V, Shah M A., ”Providing security mechanisms in broker-less publish/subscribe systems using hierarchical identity based encryption” Recent Trends in Electronics, Information & Communication Technology (RTEICT), International Conference on. IEEE, 2016: 641-645.

[15] Anusree P, Sreedhar S., ”A security framework for brokerless publish subscribe system using identity based signcryption”. Circuit, Power and Computing Technologies (ICCPCT), International Conference on. IEEE, 2015: 1-5.

[16] Elkamchouchi H. M., Elkheir E. A., Abouelseoud Y., ”A Pairing-Free Identity Based Tripartite Signcryption Scheme”. International Journal on Cryptography and Information Security (IJCIS), Vol.3, No. 4, 2013.

[17] Malpure V D, Deshmukh P K., ”Provide security for broker-less content based publish system using pairing based cryptography”. International Journal of Engineering Development and Research (IJEDR), 2016.vol.4 pp 1932-1938.

[18] Waters B, ”Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization”, in: PKC 2011, D. Catalano, N. Fazio, R. Gennaro and A. Nicolosi, eds, LNCS,Vol. 6571, Springer, 2011, pp. 53-70.

[19] Shitole S, Gujar A D., ”Securing broker-less publisher/subscriber systems using cryptographic technique”. Computing Communication Control and automation (ICCUBEA), International Conference on. IEEE, 2016: 1-6.

[20] Ion M, Russello G, Crispo B., ”Design and implementation of a confidentiality and access control solution for publish/subscribe systems”. Computer Networks, 2012, 56(7): 2014-2037.

[21] Khoury J, Lauer G, Pal P, et al., ”Efficient private publish-subscribe systems”. in Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), IEEE 17th International Symposium on. IEEE, 2014: 64-71.

[22] Yang K, Zhang K, Jia X, et al., ”Privacy-preserving attribute-keyword based data publish-subscribe service on cloud platforms”. Information Sciences, 2017, 387: 116-131.

[23] Daubert J, Fischer M, Grube T, et al., ”AnonPubSub: Anonymous publish-subscribe overlays”. Computer Communications, 2016, 76: 42-53.

[24] Lee M Z, Dunn A M, Waters B, et al., ”Anon-pass: Practical anonymous subscriptions”. Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013: 319-333.

[25] Yuen T H, Susilo W, Mu Y., ”Towards a cryptographic treatment of publish/subscribe systems”. Journal of Computer Security, 2014, 22(1): 33-67.

[26] Suzuki M, Isshiki T, Tanaka K., ”Sanitizable signature with secret information”. in: Symposium on Cryptography and Information Security, 2006, 4A1-4A2.

[27] Nakamoto S., ”Bitcoin: A peer-to-peer electronic cash system”. 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf.

[28] Cai C, Yuan X, Wang C., ”Towards trustworthy and private keyword search in encrypted decentralized storage”. Communications (ICC), 2017 IEEE International Conference on. IEEE, 2017: 1-7.

[29] Wilkinson S, Lowry J, Boshevski T., ”Metadisk a blockchain-based decentralized file storage application”. Technical Report, Available: http://metadisk. org/metadisk. pdf, 2014.

[30] Chiesa A, Green M, Liu J, et al., ”Decentralized Anonymous Micropayments”. in: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2017: 609-642.

[31] Andrychowicz M, Dziembowski S, Malinowski D, et al., ”Secure multiparty computations on bitcoin”.Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, 2014: 443-458.

[32] Andrychowicz M, Dziembowski S, Malinowski D, et al., ”Fair two-party computations via bitcoin deposits”. in:International Conference on Financial Cryptography and Data Security. Springer, 2014: 105-121.

[33] Andrychowicz M, Dziembowski S, Malinowski D, et al., ”On the malleability of bitcoin transactions”. in: International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2015: 1-18.

[34] Bentov I, Kumaresan R., ”How to use bitcoin to design fair protocols”. in:International Cryptology Conference. Springer, Berlin, Heidelberg, 2014: 421-439.

[35] Bitcoin script language. https://bitcoin.org/en/developer-reference#raw-transaction-format.

[36] Szabo N., ”Formalizing and securing relationships on public networks”. First Monday 2.9 (1997).

[37] Wood G., ”Ethereum: A secure decentralised generalised transaction ledger”. Ethereum Project Yellow Paper, 2014, 151.

[38] Bok K, Yun J, Kim Y, et al., ”User Reputation computation Method+ Based on Implicit Ratings on Social Media”. KSII Transactions on Internet & Information Systems, 2017, 11(3).

[39] ElGamal T., ”A public key cryptosystem and a signature scheme based on discrete logarithms”. In: CRYPTO 1984. LNCS,vol.196,pp.10-18.

[40] Tsiounis Y, Yung M. ”On the security of ElGamal based encryption”. in: Public Key Cryptography. Springer, 1998: 117-134.

[41] Huang H, Chen X, Wu Q, et al., ”Bitcoin-based fair payments for outsourcing computations of fog devices”. Future Generation Computer Systems, https://doi.org/10.1016/j.future.2016.12.016.

[42] Solidity.http://solidity.readthedocs.io/en/latest.

[43] Btc-testnet. https://live.blockcypher.com/btc-testnet/

Yanqi Zhao is currently a master candidate of School of Computer Science, Shaanxi Normal University, Xi’an, China. His research interests are digital signatures and blockchain.

Yannan Li is currently a PhD candidate of School of Computing and Information Technology, University of Wollongong, Australia. She received her master and bachelor degree from University of Electronic Science and Technology of China in 2017 and 2014 respectively. Her research interests are digital signatures and secure cloud storage.

Qilin Mu is the deputy general manager and associate director of National Engineering Laboratory for big data application on improving government governance capabilities. He is the deputy general manager of CETC Big Data Research Institute. He worked in No.30 Research Institute of CETC as the supervisor of System Department. He obtained the second class prize of the science and technology progress award.

Bo Yang received the B.S. degree from Peking University, Beijing, China, in 1986, and the M.S.and Ph.D. degrees from Xidian University, Xi'an, China, in 1993 and 1999, respectively. From 1986 to 2005, he was with Xidian University, where he was a Professor of National Key Laboratory of ISN and the Ph.D. Supervisor from 2002 to 2005. He has served as a Program Chair of the Fourth China Conference on Information and Communications Security in 2005, a Vice Chair of the Conference of the Chinese Association for Cryptologic Research in 2009, and a General Chair of the Fifth Joint Workshop on Information Security in 2010. He is currently a Professor and Ph.D. Supervisor with the School of Computer Science, Shaanxi Normal University, Xi'an, and a Special Term Professor of Shaanxi Province. His research interests include information theory and cryptography.

Yong Yu is currently a Professor of Shaanxi Normal University, China. He holds the prestigious one hundred talent Professorship of Shaanxi Province as well. He received his Ph.D. degree in cryptography from Xidian University in 2008. He has authored over 50 referred journal and conference papers. His research interests are cryptography and its applications, especially public encryption, digital signature, and secure cloud computing. He is an Associate Editor of Soft Computing.