This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Feng, Wei; Li, Yafeng; Yang, Xuetao; Yan, Zheng; Chen, Liang
Blockchain Based Data Transmission Control for Tactical DataLink

Published in:
Digital Communications and Networks

DOI:
[10.1016/j.dcan.2020.05.007](https://doi.org/10.1016/j.dcan.2020.05.007)

Published: 01/08/2021

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY-NC-ND

Please cite the original version:
Feng, W., Li, Y., Yang, X., Yan, Z., & Chen, L. (2021). Blockchain Based Data Transmission Control for Tactical DataLink. *Digital Communications and Networks*, 7(3), 285-294. <https://doi.org/10.1016/j.dcan.2020.05.007>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

Blockchain-based data transmission control for Tactical Data Link

Wei Feng^a, Yafeng Li^b, Xuetao Yang^a, Zheng Yan^{a,c,*}, Liang Chen^b^a State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, 710071, China^b The 20th Research Institute of China Electronics Technology Group Corporation, Xi'an, 710061, China^c Department of Communications and Networking, Aalto University, Espoo, 02150, Finland

ARTICLE INFO

Keywords:

Blockchain

Tactical data link

Consensus

Data transmission control

ABSTRACT

Tactical Data Link (TDL) is a communication system that utilizes a particular message format and a protocol to transmit data via wireless channels in an instant, automatic, and secure way. So far, TDL has shown its excellence in military applications. Current TDL adopts a distributed architecture to enhance anti-destruction capacity. However, It still faces a problem of data inconsistency and thus cannot well support cooperation across multiple militarily domains. To tackle this problem, we propose to leverage blockchain to build an automatic and adaptive data transmission control scheme for TDL. It achieves automatic data transmission and realizes information consistency among different TDL entities. Besides, applying smart contracts based on blockchain further enables adjusting data transmission policies automatically. Security analysis and experimental results based on simulations illustrate the effectiveness and efficiency of our proposed scheme.

1. Introduction

Tactical Data Link (TDL) is a communication system that utilizes a particular message format and a common protocol to transmit data via wireless channels in an instant, automatic, and secure way [1]. It has the advantages of high throughput and low latency, and it aims to reduce the time of the Observing, Orientating, Deciding, and Acting (OODA) loop so that the reaction time of military units is greatly shortened. Nowadays, TDL usually adopts a distributed architecture rather than a point-to-point or centralized architecture to achieve and ensure high anti-corruption capability. In summary, TDL is significant in terms of military capacity improvement and thus has attracted considerable attention in the field of military communications [2,3].

Motivations and challenges: Despite the advantages of TDL, it still faces a severe problem, i.e., automatic and adaptive cooperation among multiple militarily domains without sufficient trust, especially in inter-military-domain scenarios. In modern battlefields, it is unavoidable that combat units from different military domains cooperate with each other. A battlefield event usually requires the cooperative reaction of combat units in multiple military domains. For example, in many situations, there are several combat groups on the same battlefield. They need to work together to destroy adversary targets and reach a combat goal. In order to achieve this purpose, it is necessary for them to share

information continually. Besides, they have to reach a consensus on some decisions based on shared information and data; otherwise, the ill-advised decision made based on unsynchronized information could cause cooperation failure. Therefore, it is essential to provide a data transmission control mechanism that manages the data transmission among different combat domains and supports such crucial features as data consistency, decision automation, and action adaptivity. Although TDL supports communications between different military units, it lacks an autonomous and adaptive data transmission control scheme and is unable to control the flow and sharing of data uniformly and reliably, which is a basic requirement for cross-military cooperation in multiple fields.

However, it is challenging to control data transmission in inter-domain cooperation in a reliable way. Past work about trust management cannot solve this issue [4–8]. First, current TDLs cannot guarantee data consistency among different domains. Second, the modern military requires high autonomy, and the military information and messages should be transmitted automatically rather than artificially in order to reduce transmission delay. Third, the data transmission in TDL is dynamic, and the data should be transmitted to different nodes based on the status of a battlefield. However, these issues have rarely been explored.

The emergence of the blockchain provides an effective tool to achieve automatic data transmission in TDL. The blockchain, which was firstly

* Corresponding author.

E-mail addresses: weifeng.ft@foxmail.com (W. Feng), xxddxdd@yeah.net (Y. Li), bsyangxt@foxmail.com (X. Yang), zyan@xidian.edu.cn (Z. Yan), 772029466@qq.com (L. Chen).<https://doi.org/10.1016/j.dcan.2020.05.007>

Received 1 October 2019; Received in revised form 22 May 2020; Accepted 25 May 2020

Available online 27 June 2020

2352-8648/© 2020 Chongqing University of Posts and Telecommunications. Publishing Services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an

open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

applied in a decentralized cryptocurrency system [9], has attracted considerable attention because it offers a way of data storage with immunity, trust, and traceability in a decentralized manner. In addition, Ethereum, a platform to build smart contracts, employs the blockchain to support the trustworthy execution of codes (called a smart contract) automatically in a decentralized method [10]. Owing to Ethereum, the application of the blockchain quickly expands from cryptocurrency or decentralized ledger to Internet of Things (IoT) [11,12], smart cities, energy management, crowdsourcing [13], social networking [14], Vehicular Ad-hoc Network (VANET) [15], etc.

Applying blockchain into TDL has excellent potential to overcome the above mentioned three challenges. First, blockchain can achieve data consistency among different domains with the underlying consensus mechanism. Besides, it also ensures the integrity and immunity of data, which is very important for military scenarios because false data will have a negative impact on military decision-making. Additionally, blockchain can ensure the trustworthy execution of the smart contracts deployed in it automatically, which enables automatic data transmission control with blockchain. Its decentralization property makes it suitable for the decentralized architecture of TDLs and benefits TDL by enhancing its anti-destruction ability. To summarize, blockchain can become a useful tool to achieve automatic data transmission in the TDL system. However, based on our literature survey, there is no such a proposed solution, which applies blockchain to realize automatic and autonomous data transmission control for TDL.

In this paper, we explore the above issues in TDL by employing a consortium blockchain to build an automatic and adaptive data transmission control in TDL. To be specific, we propose a weighted Practical Byzantine Fault Tolerance (PBFT) to adapt to TDL. Based on the weighted PBFT, we design a consortium blockchain and employ a smart contract to build an automatic data transmission control scheme. The main contributions of this paper can be summarized as follows:

1. We improved the PBFT algorithm and proposed a weighted PBFT algorithm, which considers the variety of abilities in terms of the anti-attack capability of different military entities. Compared with PBFT, the weighted PBFT can achieve better anti-destruction ability;
2. We propose a consortium blockchain based on the weighted PBFT. We build an automatic and adaptive data transmission control scheme for TDL based on this blockchain system. The proposed scheme enables military entities to obtain data transmission strategies automatically and solves the problem of the data inconsistency in TDL;
3. We conduct theoretical analysis to prove the security of our scheme and carry out several experiments to illustrate the efficiency of our scheme.

The rest of the paper is organized as follows. In Section 2, we introduce the research background as well as related works. We present the problem statement of our research in Section 3 and detail the design of the automatic and adaptive data transmission control scheme in Section 4. We conduct a security analysis in Section 5 and evaluate the performance of our scheme in Section 6. Finally, the paper is concluded in the last section.

2. Background and related work

In this section, we introduce the development and application of TDL, as well as the recent advances of the blockchain.

2.1. Tactical data link

The first TDL system was developed and deployed in America in the 1950s [1]. The original TDL system, Link-4, greatly reduces the OODA loop time and helps quick military reaction. However, Link-4 itself faces several problems. First, Link-4 adopts point-to-point communications for

data transmission, and thus it cannot effectively support data broadcast or multi-cast. Second, it transmits data in plaintext and cannot resist eavesdropping. To tackle these problems, Link-11 was developed, which introduces encryption in TDL for data confidentiality. However, it cannot well resist interference and does not offer positioning function [1].

The above TDL systems have several drawbacks. First, they are usually designed for an individual branch of armed forces and thereby cannot well support the cooperation of different multiple armed forces. Second, they suffer from low throughput, which limits the application of TDL. Third, they adopt a point-to-point or centralized architecture, thus cannot achieve excellent anti-destruction performance.

To improve the performance of TDL in terms of cooperation, anti-destruction, and throughput, Link-16 and Link-22 were developed [1]. The two TDL systems well improve throughput. Especially, they adopt a distributed architecture, which helps to enhance anti-destruction ability. Although they support data transmission between different TDLs, they still do not support the cooperation of different armed forces.

Apart from Link-16 and Link-22, TDLs for dedicated applications were also developed and explored. For example, Dinc enables the integration of intelligent Unmanned Air Vehicles (UAVs) and navy units [16]. Raber designed a command and control scheme for the coast guard shipboard [17]. Martinez-Ruiz et al. surveyed an imagery transmission issue in Link-16 [18]. In Ref. [19], the application of TDL in naval anti-air warfare was proposed. Schrecke et al. proposed an IP address translation method for tactical networks [20,21].

Although several works are aiming at performance improvement of TDL in different scenarios [22–27], they all ignore the problem of multi-domain cooperation and automation issue in TDL. In particular, they cannot achieve adaptive and automatic data transmission control in TDL. Besides, how to guarantee data consistency among different military domains is also a problem to be solved.

2.2. Blockchain and its applications

The blockchain, the fundamental technology of Bitcoin [9], is composed of several data blocks that are linked through a one-way hash function. A blockchain system is decentralized and its security relies on a consensus mechanism rather than a fully trusted centralized party. For Bitcoin, as long as more than 25% of miners are honest, the security and trust of the blockchain can be ensured [28].

The blockchain, with its advantages in decentralization and trust, has attracted extensive attention from both academia and industry. Ethereum is a decentralized intelligent contract platform based on blockchain. With the emergence of Ethereum [10], the applications of the blockchain have expanded from cryptocurrency to various fields, like data management [29], Internet of Things (IoT) [11], crowdsourcing, social networking, VANET [15], etc. To adapt to the requirements in different scenarios, people have developed various types of blockchains. Because of the consistency, trust, decentralization, and immunity of the blockchain, it could become a useful tool to build a decentralized and robust data transmission control for TDL.

Among various blockchains, the public blockchain is not suitable for TDLs due to its openness and low efficiency. Therefore, we consider the application of the consortium blockchain in TDL. Nowadays, the consortium blockchain gains great popularity in finance and insurance, energy systems, data management, digital copyright, and Digital Asset Management (DAM), etc. For example, Androulaki et al. developed Hyperledger Fabric, which is a decentralized ledger and a trusted, smart contract execution platform [30]; Li et al. proposed a consortium blockchain-based energy trading system [31]; Dib et al. proposed the significance of the consortium blockchain in digital copyright and DAM [32]; Zhou et al. leveraged the consortium blockchain to improve the trustworthiness of network software services [33]; Guo et al. explored the application of the consortium blockchain in the banking industry [34].

Despite the popularity of the consortium blockchain, its application in TDL is rarely explored. Different from the public blockchain, the

Fig. 1. System model.

consortium blockchain usually leverages a consensus mechanism like the PBFT algorithm [30], the Raft algorithm [35], etc., rather than Proof of Work (PoW) [9], Proof of Stake (PoS) [10], etc. Among the consensus mechanisms, the PBFT algorithm outperforms others because of its excellent performance in Byzantine fault tolerance. However, it also faces some problems, e.g., high communication overhead. In addition, it attaches equal importance to each entity, which cannot reveal the situation of TDL. Therefore, the algorithm seems not proper to be directly applied to TDL.

3. Problem statement

This section introduces the system model and security model of TDL and specifies the research problems and design objectives of our work.

3.1. System and security model

In this subsection, we present a system model of TDL and specify the security model of our work as well.

System model: The TDL system is composed of a Chief Command Center (CCC), multiple Operation Command Centers (OCC) in different regions, and various types of Execution Nodes (EN), like tanks, fighters, soldiers, etc., as depicted in Fig. 1. Generally, there is a unique CCC in the whole system, which is the most powerful entity among all participants in TDL. There are also several OCCs in different regions or armies that send commands to relevant ENs. ENs can be armies, combat platforms, and sensors deployed on battlefields. They are responsible for collecting data, submitting data to corresponding OCC, and responding to commands received. CCC cooperates with different OCCs to maintain a blockchain network. The blockchain stores the key information of data transmission control, i.e., code of smart contracts for data transmission control and data transmission history. For easy presentation, we suppose the total number of the CCC and the OCCs is $N + 1$, and the set of the CCC and the OCCs is denoted as CN . Without loss of generality, we suppose cn_0 is the CCC, and $cn_i (1 \leq i \leq N)$ represents an OCC.

Security model: In fact, CCC is the most powerful entity among all TDL entities, and it has a strong ability to resist external attackers. For OCCs, since they may be ruined or compromised in military actions, though the probability is usually low, we assume that they may

sometimes be unavailable or compromised. Still, an adversary can only compromise a limited number of them. We also assume that the probability that an OCC is ruined or compromised is different from each other. Compared with OCCs and CCC, ENs may be ruined by the adversary easily, so we assume that they cannot be fully trusted.

In our security model, we measure the resistance capacities of the CCC and OCCs with the cost required by the attackers to compromise them. Denote the set of costs to compromise CCC or OCCs as $C = \{c_i\}_{0 \leq i \leq N}$, where c_i represents the cost to compromise cn_i . Obviously, a larger c_i means more powerful resistance capacity to attacks. Since cn_0 has a more powerful resistance capacity than the others, $\forall c_i$, we have $c_0 \geq c_i$. Without losing generality, we assume that $c_0 \geq c_1 \geq \dots \geq c_N$.

3.2. Research challenges and design goals

Herein, we specify the challenges of TDLs in automatic and trustworthy data transmission control and present our design goals. As mentioned earlier, the PBFT algorithm is not suitable for direct application to TDLs because it treats all entities equally. In TDLs, different entities have different probabilities of being ruined or compromised, which should be considered when designing the blockchain system of TDLs. Besides, the data transmission control of TDLs should be automatic, adaptive, and flexible so that it could adjust to dynamically changed battlefield situations. To summarize, the main research challenges can be summarized as below:

1. How to build a blockchain system that can consider the resistance of entities to attacks;
2. How to design an automatic data transmission control scheme for TDLs that can adapt to the dynamically changing battlefield situations and control data transmission automatically and adaptively.

Our scheme aims to overcome the above challenges and design an automatic and trustworthy data transmission control scheme for TDLs. To be specific, our design goals are summarized as follows:

1. We aim to improve the PBFT algorithm and make it suitable for TDL by considering the changing of resistance ability of different consensus nodes to attacks. The improved PBFT algorithm, i.e.,

Fig. 2. Procedures of weighted PBFT.

weighted PBFT, should achieve safety and liveness. Even if the attacker can avoid controlling some weak consistency nodes, the whole consistency mechanism can not be broken. Specifically, the cost of compromising all consensus nodes is c_T , and the weighted PBFT ensures that the cost of compromising the system is no less than $\frac{1}{3}c_T$.

2. We aim to design a flexible and efficient blockchain system for TDL based on the proposed PBFT algorithm. The proposed blockchain system should be efficient in terms of throughput and latency. It should also realize flexibility by considering the practical situation of TDLs and not introducing high latency due to block creation.
3. We aim to build an automatic and trustworthy data transmission control scheme based on smart contracts, which is built upon the proposed blockchain system with the improved weighted PBFT. The scheme can effectively control data transmission in an automatic, smart, and adaptive way. It can adaptively control the receivers and transmission priority of a data report based on the situation of the battlefield.

4. System design

In this section, we introduce the detailed design of the weighted PBFT-based on node capability, the blockchain system based on the weighted PBFT, and the blockchain-based data transmission control scheme.

4.1. Scheme overview

The whole scheme operates as follows. First, ENs sense battlefield data and send the data to corresponding OCCs. When an OCC receives data sent by an EN, it first checks whether the data is from an authorized EN. Otherwise, the OCC will ignore the data. If the data passes the verification, the OCC will take the data attributes, like content, type, location, etc., as input, invoke the smart contract in blockchain, and output the related data transmission strategy. Then, the OCC sends the data to several receivers through a pre-established secure channel according to the transmission strategy obtained by the smart contract. Besides, it will also generate a notification, including the attributes and the digest of the data, and then send the notification to the CCC. When the number of notifications received by the CCC reaches the predefined threshold, the CCC generates a block and delivers the block to all OCCs in the network. The CCC and all OCCs will employ the weighted PBFT algorithm to achieve consensus on the block.

4.2. Weighted practical byzantine fault tolerance

In this subsection, we present the design of the weighted PBFT algorithm based on node capacity, which is derived from the PBFT algorithm. Considering the diversity of TDL's ability to resist military attacks and cyber attacks, it is not reasonable to attach equal importance to all entities of TDL. For example, as the most important party in TDL, CCC

should be much more powerful and robust than OCCs in terms of anti-destruction. Therefore, CCC should play a more important role than OCCs. Inspired by this idea, we herein propose the weighted PBFT algorithm. The main difference that distinguishes the weighted PBFT from the PBFT is that it attaches a weight to each entity in the blockchain based on its ability and importance. Fig. 2 depicts the basic procedures of weighted PBFT when there exist three OCCs (including a broken OCC). The details of the weighted PBFT algorithm are described as follows:

1. **System setup:** In this process, the system establishes a series of security parameters for all system entities. First, CCC generates its public/private key pairs (PK_{cn_0}, SK_{cn_0}) .

$$\sum_{i=0}^N \omega_i = 1 \quad (1)$$

Besides, the system consists of a leader, and at the beginning of the system establishment, cn_0 acts as the leader.

2. **Request:** When a consensus node wants to record a message in the blockchain, it first generates a request with its signature to CCC. Upon receiving the request, the leader will check its validity and verify the signature. If the request is valid, the leader will broadcast a *PrePrepare* message to all consensus nodes.
3. **Preprepare:** In this process, each $cn_i \in CN$ that receives the *PrePrepare* message will check its validity by verifying its signature and the content of the message.
4. **Prepare:** If the validity check on the message holds in node cn_i , cn_i will generate a *Prepare* message and broadcast it to the other N nodes. Meanwhile, the other nodes also conduct the same process. Then, cn_i waits for *Prepare* message sent by others. Once cn_i receives sufficient *Prepare* messages, it starts the next procedure. The requirement to judge whether to start the next procedure *Commit* is presented as follows.

Suppose $CN_{Prepare}$ is the set of nodes (including cn_i) whose *Prepare* message has been received by cn_i , then, cn_i enters the next procedure if and only if the following equation holds:

$$\sum_{cn_j \in CN_{Prepare}} \omega_j > \frac{2}{3} \quad (2)$$

5. **Commit:** When a node cn_i starts the *Commit* procedure, it generates a *Commit* message and broadcasts it to all other nodes in the blockchain system. Similarly, there are also some other nodes starting *Commit* as well. cn_i can thus wait for the *Commit* messages generated by other nodes. The number of the *Commit* messages received by cn_i indicates whether the system has reached consensus. To be specific, suppose CN_{Commit} is the set of nodes (including cn_i) whose *Commit* message has been received by cn_i , then cn_i considers the procedure of *Commit* ends, and the system has reached the consensus, if and only if the following equation holds:

$$\sum_{cn_j \in CN_{Commit}} \omega_j > \frac{2}{3} \quad (3)$$

Weight setting: In TDL, different cn_i has a different resistance to attacks, which is measured by the cost needed by attackers to compromise it. Generally, attackers will minimize their costs to compromise the system. PBFT attaches equal importance to each consensus node, which enables attackers to compromise the whole system at a small cost by selecting consensus nodes with low resistance. Informally, given $c_T = \sum_{0 \leq j \leq N} c_i$, attackers may be able to compromise the blockchain with cost less than $\frac{1}{3}c_T$ by selecting $N/3$ nodes with the lowest cost in CN . To tackle this problem, we proposed the weighted PBFT, and the weight of each consensus node is related to the cost needed by attackers to compromise

Fig. 3. Block structure.

it. We aim to ensure that the total cost to compromise the system is at least $\frac{1}{3}c_T$ regardless of the strategies of attackers.

In order to achieve this goal, in our design, a consensus node cn_i has a higher weight if the corresponding c_i is higher. To be specific, suppose that the weight of the consensus node cn_i in TDL is denoted as ω_i , then $\omega_i = \frac{c_i}{c_T}$. We think the method is reasonable since it guarantees that the total weight of compromised consensus nodes is linear to the cost of attackers. As a result, the attackers have no way to gain more weight with a low cost. While in PBFT, it is possible for the attackers to compromise the consensus nodes with weaker abilities to reduce the total cost, which illustrates the superiority of our scheme.

Leader change: The cn_i with the highest resistance capacity acts as the leader. At the beginning of the system setup, the cn_0 acts as the leader. However, cn_0 may be compromised or ruined. In this case, the cn_i , except the cn_0 with the highest resistance capacity, will be selected as the next leader for block creation. Specifically, if the next leader detects that the current leader is ruined or compromised, it generates a signal and broadcasts the signal to all cn_i . If the system reaches a consensus on the signal, the current leader will be excluded from the system, and the next leader starts performing its duty.

The process of **leader change** can be divided into three procedures, i.e., **leader revocation**, **weight update**, and **new leader establishment**.

1. **Leader revocation:** Represent current leader as cn_{i_0} , and any node that detects misbehaviors or breakdown of cn_{i_0} can generate a *Leader Revocation* message, denoted as M_{rev,i_0} , and then broadcast the message to all consensus nodes.

All consensus nodes respond to M_{rev,i_0} via the weighted PBFT. If M_{rev,i_0} is accepted by most of the consensus nodes, they could reach a consensus on M_{rev,i_0} . Once the consensus on M_{rev,i_0} is confirmed, cn_{i_0} is revoked.

2. **Weight update:** Once the revocation of cn_{i_0} is confirmed, cn_{i_0} will be excluded from the system and can no longer participate in the consensus process. All unrevoked consensus nodes remove cn_{i_0} from their communication list, and each consensus node cn_i recalculate its weight as: $\omega_i = \frac{c_i}{c_T - c_{i_0}}$

3. **New leader establishment:** We suppose that $c_0 \geq c_1 \geq \dots \geq c_N$ and that the honest consensus node with the highest broken cost will behave as a new leader. Therefore, when cn_{i_0} is revoked, cn_{i_0+1} will take over the role of the leader. cn_{i_0+1} will generate a leader establishment message, i.e., M_{est,i_0+1} . Then, cn_{i_0+1} broadcasts M_{est,i_0+1} with its signature on it to all unrevoked consensus nodes. If most of the unrevoked consensus nodes reach consensus on M_{est,i_0+1} , cn_{i_0+1} starts to work as the next leader.

4.3. Blockchain based on weighted PBFT

Based on the weighted PBFT algorithm proposed in Section 4.2, we further build a blockchain system for TDL. The block structure is shown in Fig. 3. It contains the ID of its previous block, i.e., the hash of the previous block; the block number, which indicates the location of the block in the blockchain; smart contracts for data transmission control, with which TDL entities can obtain the data transmission strategies in a trustworthy way; smart contracts to update the contracts of data transmission control; the records of data transmission, the signature on the block content generated by the block creators.

Block creation: The weighted PBFT-based blockchain contains a leader responsible for block creation. When the leader collects a certain number of notifications from other consensus nodes, it will create a block and leverage the weighted PBFT algorithm to reach the consensus on the block content. However, in reality, especially in military scenarios, there are certainly some urgent messages that have to be sent to other consensus nodes without any delay. When there is an urgent message, and the number of messages has not yet reached a predefined threshold, it will cause serious risks and loss. To tackle this problem, we attach a priority factor to each type of message and propose three criteria for block generation as listed below:

1. The number of messages received exceeds the predefined threshold Thr_{Num} ;
2. The leader receives a message whose priority is higher than a predefined threshold Thr_{Pri} .
3. The leader's waiting time has exceeded the predefined threshold Thr_{Time} .

4.4. Blockchain-based data transmission control

Based on the proposed blockchain, we design a data transmission control scheme for TDL, as well as a self-adaptive smart contract update scheme.

4.4.1. Automatic data transmission

The TDLs request the automatic control of the data transmission. Specifically, all OCCs, when receiving data from ENs, will obtain a data transmission strategy indicating to whom the message should be sent. In this way, the collected data can be automatically transmitted to expected destination entities. Besides, the OCCs will also send a notification that indicates data transmission history to the blockchain system. The whole process is composed of a series of procedures, as described below:

- 1) The leader establishes a smart contract of data transmission control, with data type, data content, etc. as input data, and outputs a data transmission strategy that indicates the receiver of input data, receiving priority, etc. Then, the leader generates a new block by inserting the smart contract into it and broadcasts the block to all the consensus nodes. The consensus nodes verify the validity of the smart contract and load the weighted PBFT algorithm to reach the consensus on the block. Once a consensus node convinces that the block has been accepted by most network entities, it will consider the smart contract recorded in the block to be valid.

- 2) The blockchain-based smart contract achieves autonomic and trustworthy execution, and its result can be verified publicly. Therefore, we employ a blockchain-based smart contract to calculate data transmission strategy automatically, which also ensures the trustworthiness of the data transmission strategy. Besides, the smart contract should comprehensively consider the data content, battlefield status, the goal of the combat, etc., to generate the correct data transmission strategy. Specifically, when an OCC receives a piece of data from an EN, it will first check the validity of the data. If it passes the verification, the OCC invokes the smart contract recorded in the blockchain to obtain the transmission strategy of the data. The smart contract takes as input the data, the battlefield status, etc. Then, the OCC generates a data report, which includes the data content and the data transmission strategy. Finally, it sends the data report to the leader of the blockchain. The leader inserts the data report into the block, and all consensus nodes reach consensus on the block using the weighted PBFT.
- 3) All OCCs receiving the data report will check its validity by verifying the strategy involved in the data report that is correctly obtained through the smart contract. If the data report is valid, they transmit the data to the receivers based on the transmission strategy. Our scheme requires the receivers to send a notification to CCC, indicating that it has received the data. The notification will be inserted into the blockchain, and all consensus nodes will reach a consensus on this block. In this way, all consensus nodes are aware that the receiver has received the data. The EN that generates the data will also receive the notification from the corresponding OCC so that it can ensure that the data has been successfully transmitted.

4.4.2. Self-adaptive smart contract update

The battlefield status varies quickly all the time, which influences the data transmission strategies. For example, when the battlefield status changes, the same data collected will have different priorities and may be transmitted to different entities. Therefore, the smart contract for data transmission control will also vary based on the status. However, artificial updates on smart contracts would take time and cause long delays. Therefore, we here propose two adaptive and automatic smart contract update methods to update smart contracts adaptively.

The two proposed methods for smart contract updating are the artificial contract update method and automatic contract update method. In these two methods, the artificial method means that the leader directly sets a new contract, writes it into a new block, and enforces all consensus nodes to update the contract by reaching a consensus on the block. The difference is that the automatic method achieves contract updates by inserting a smart contract for contract updates into the blockchain ahead of time, probably when setting up the blockchain system. This smart contract automatically checks the status of the battlefield, and once it detects a change of status, it will invoke the smart contract to check whether the contract for data transmission needs to be updated. If yes, it obtains a new smart contract for data transmission control via the underlying smart contract for data transmission control contract update.

5. Security analysis

In this subsection, we analyze the safety and liveness of the proposed blockchain based on the weighted PBFT.

5.1. Safety

Security refers to the fact that there are no two valid blocks in the same location in the blockchain. We have the following proposition about safety.

Proposition 1. The weighted PBFT provides **safety** as long as the accumulated weight of compromised or ruined OCCs is less than $1/3$.

Proof. For easy presentation, we first define the following predicate, i.e., $\text{Prepared}(B, i)$. $\text{Prepared}(B, i)$ is true if and only if the Prepared message claims the block B is accepted by an honest node cn_i , and cn_i has received a number of Prepared messages from a group of nodes whose accumulated weight (including the weight of i) exceeds $2/3$.

Suppose the accumulated weight of compromised nodes is $1/3 - \delta$, where $\delta > 0$ is a quite small positive value. Then, we could infer, for an honest node cn_i , if $\text{Prepared}(B, i)$ is true, the predicate $\text{Prepared}(B', i)$ is false for all $B' \neq B$. This is because if $\text{Prepared}(B', i)$ is true, cn_i has to receive Prepare messages from a group of nodes whose accumulated weight is over $2/3$. In the worst case, all malicious nodes collude with each other and send Prepare messages on both B and B'. Since $\text{Prepared}(B, i)$ is true, the accumulated weight of honest nodes sending Prepare voting B should be over $1/3 + \delta$. Therefore, the weight of honest nodes voting B' is less than $1/3$ since honest nodes will not send Prepare messages for two different blocks. As a result, the weight of both honest and malicious nodes voting B' will be less than $2/3 - \delta$. Therefore, $\text{Prepared}(B', i)$ will never be accepted by cn_i .

With the aforementioned predicate, we further define two predicates, i.e., $\text{CommittedLocal}(B, i)$ and $\text{Committed}(B)$. Among them, $\text{CommittedLocal}(B, i)$ means that cn_i has received the Prepared messages on B from the nodes (including cn_i) whose accumulated weight has exceeded $2/3$. $\text{Committed}(B)$ means that $\text{Prepared}(B, i)$ holds true for a group of honest nodes, whose accumulated weight is more than $1/3$. With the two predicates, we have the following proposition, i.e., if $\text{CommittedLocal}(B, i)$ holds for an honest node cn_i , $\text{Committed}(B)$ is true. This is because, that $\text{CommittedLocal}(B, i)$ is true means it has received Commit messages from a group of nodes (including node i) whose accumulated weight has exceeded $2/3$. For an honest node, it will send a Commit message only when it collects enough Prepare messages, which means $\text{Prepared}(B, i)$ holds for this node. Since the total weight of all malicious nodes is $1/3 - \delta$, the honest nodes sending Commit messages on block B occupy the weight more than $1/3 + \delta$. That is, the honest nodes for whom $\text{Prepared}(B, i)$ holds occupy the weight of $1/3 + \delta$, which means that $\text{Committed}(B)$ is true.

To summarize the above analysis, we conclude that when the accumulated weight of compromised or ruined OCCs is less than $1/3$, there will be only one valid block to be accepted by a group of honest consensus nodes, of whom the accumulated weight is more than $1/3$. This means the weighted PBFT-based blockchain achieves security.

5.2. Liveness

Liveness refers to that the CCC can always generate a new block, and the block can always achieve consistency among honest nodes. The weighted PBFT-based blockchain achieves liveness as long as the accumulated weight of ruined or compromised nodes does not exceed $1/3$. The CCC generates a new block when the number of messages waiting to be recorded in the blockchain meets the threshold, or there is a message with extremely high priority, or the length of the waiting period of the CCC meets the pre-established time threshold. Therefore, the weighted PBFT-based blockchain can guarantee that a new block can be created within a limited time (i.e., the pre-established time threshold). To achieve consistency, the accumulated weight of honest nodes should be more than $2/3$. Since the sum of the weight of ruined or compromised nodes does not exceed $1/3$, the total weight of honest nodes exceeds $2/3$. As a result, the weighted PBFT can still operate smoothly. Therefore, the proposed weighted PBFT-based blockchain achieves liveness.

6. Performance evaluation

We conducted a series of experiments to verify the effectiveness and efficiency of our scheme. In this subsection, we present the experimental results on both the weighted PBFT and the data transmission control scheme.

Table 1
Experimental settings.

Experimental Settings		
Exp.1	Consensus Node Number	50
	Message Generation Rate	0 - 100 messages/s
	Block Generation Threshold	0–100
Exp.2	CCC Number	1
	OCC Number	4
	EN Number	0–350

Table 2
Data format.

Field	Description
Type	The type of the EN that generates the data
Time	The time stamp when the data is generated
ID	The unique id of the EN
LocCur	The current location of the EN
LocTAR	The next location of the EN
isMove	Whether the EN is moving
isHide	Whether the EN is well hidden
isKeep	Whether the EN is under suppression
isTired	Whether the EN is tired
TarID	The ID of the enemy with whom the EN is fighting
NumPer	The number of soldiers in the EN
NumWeapon	The number of weapons the EN is equipped with
NumStep	The moving ability of the EN

6.1. Evaluation metrics

We employed the following criteria to evaluate the performance of the weighted PBFT and the blockchain-based data transmission control for TDLs.

1. **Consensus Time:** The consensus time is measured by the time the system spends to reach a consensus on a proposed block.
2. **Block Generation Time:** The block generation time is measured as the average generation time of a single block. In our test, we configure the workload by adjusting the message frequency and the number of consensus nodes, and then we monitor the block generation time.
3. **Message Confirmation Latency:** The message confirmation latency is measured as the response time per message. We test the confirmation time that the system spends in confirming a message with regard to message generation.
4. **Throughput:** The throughput is measured as the maximum number of messages successfully recorded in the blockchain per second.
5. **Efficiency of Smart Contracts:** The efficiency of a smart contract is measured by the execution time of the smart contract. TDL usually needs lower response time and thereby it requires high smart contract efficiency.
6. **Latency:** TDLs requires low latency so that the combat units can react to the tactical issues quickly. In our analysis, the latency is measured by the length of duration from the time when an EN generates a data piece to the time when it receives the notification of data transmission.

6.2. Experimental settings

6.2.1. Experimental methodology and environments

We employed simulations to evaluate the performance of the proposed weighted PBFT and the blockchain-based data transmission control for TDLs, considering the difficulty of real-world experiments. We set up two experiments that respectively tested the performance of the weighted PBFT (Exp.1) and the blockchain-based data transmission control for TDLs (Exp.2). All the experiments were conducted on a laptop running Windows 10 with 1.7 GHz Intel Core I5 CPU and 8 GB DDR3 RAM. In Exp.1, we simulated the weighted PBFT for TDLs in the JAVA

Fig. 4. Consensus time with regard to node number.

language. In Exp.2, we implemented the blockchain-based data transmission control scheme with Hyperledger Fabric and implemented the smart contracts in the JAVA language. We tested its performance in a battlefield scenario deployed with the data obtained from a military simulation.

6.2.2. Experimental settings

In this part, we introduce the experimental settings of Exp.1 and Exp.2. For easy presentation, we summarize the settings in Table 1.

Settings of Exp.1: In Exp.1, we tested the weighted PBFT's consensus time, block generation time, and message confirmation latency under different node numbers (i.e., 50 and 100) and different message generation rates (from 0 to 100). Besides, we compared the weighted PBFT with traditional PBFT to illustrate that introducing weights does not incur high consensus latency. Based on the experimental results, we further analyzed the throughput of weighted PBFT-based blockchain.

Settings of Exp.2: In Exp.2, we implemented our blockchain-based data transmission control for TDLs based on Hyperledger Fabric, which is a popular consortium blockchain-based platform for smart contracts. The reason for employing Hyperledger Fabric rather than the weighted PBFT-based blockchain is that Exp.2 focuses on testing the efficiency of the smart contracts. Besides, building a complete blockchain-based smart contract platform is quite difficult. Therefore, we just implemented our scheme based on Hyperledger Fabric to illustrate its effectiveness.

Exp.2 is conducted using the data obtained from a military simulation, and the data format is presented in Table 2. We use these data to simulate a simple battlefield scenario. In our simulation, the whole battlefield is divided into four areas, and there is an OCC in each area. Besides, there exists a CCC on the battlefield. All the OCCs and the CCC cooperatively maintain the blockchain system. Each area contains a number of ENs, and each EN senses and uploads data to the corresponding OCC, which invokes the smart contract to determine the data receivers.

We assume that there are three types of data transmitted via TDLs, i.e., raw data sensed by the ENs, command message, and notification. The raw data is the data obtained from the military simulation, as shown in Table 2. The OCC, when receiving the data, processes the data and determines the transmission strategy for the data with the predefined smart contracts. To be specific, we designed three smart contracts for automatic data transmission control for TDLs, which are described as follows:

Fig. 5. Consensus time with regard to threshold for block generation.

1. **Contract 1 (Data Processing):** This contract processes the raw data sensed by ENs and calculates a value indicating the state that each EN is friendly or malicious.
2. **Contract 2 (Judgement Calculation):** The contract checks whether an EN is in a bad situation to judge whether it needs assistance from other ENs.
3. **Contract 3 (Transmission Strategy Calculation):** The contract determines the data transmission strategy based on the results of **Contract 1** and **Contract 2**. Specifically, for an EN requiring help, the contracts will take the status of nearby ENs as input and select a group of ENs as data receivers that do not perform any tasks, but have the ability to help the data generator.

6.3. Results of Exp.1

6.3.1. Consensus time

To demonstrate the efficiency of the weighted PBFT, we compared the consensus time of PBFT and weighted PBFT. In our experiments, we tested the time for a block to reach consistency by applying the PBFT and the weighted PBFT with regard to different thresholds and node numbers, as shown in Figs. 4 and 5. From Figs. 4 and 5, we can see that

Fig. 6. Block generation time with regard to message generation rate.

Fig. 7. Block generation time with regard to node number.

the consensus time of the weighted PBFT is shorter than that of PBFT. This is because the weighted PBFT does not introduce much computation or communication burden. Fig. 4 shows the consensus time of both weighted PBFT and PBFT increases as node number increases. The reason is that the communication complexities of both algorithms are $\mathcal{O}(n^2)$, where n is the number of consensus nodes. More nodes result in a higher communication cost. Therefore, it takes more time to reach consistency when there are more nodes involved. Fig. 5 indicates that the consensus time increases with the increase of the threshold for block generation. This is because the higher threshold implies that more messages are included in block generation, which takes more time for OCCs to verify the validity.

6.3.2. Block generation time

We conducted experiments to test the block generation time with regard to the message rate. We also compared the block generation time of the weighted PBFT-based blockchain with that of the PBFT-based blockchain. The experimental results are summarized in Fig. 6. From Fig. 6, we can see that the block generation time decreases with the increase of the message generation rate at first, and then it keeps

Fig. 8. Message confirmation time with regard to message generation rate.

Fig. 9. Execution time of smart contracts with regard to the number of ENs.

unchanged. This is because a higher message generation rate makes it faster to meet the threshold for block generation. However, it takes a certain time for a block to reach consistency. In this case, the increase of the message generation rate does not make the consensus time of a block decrease because the blockchain leader does not create a new block until the previous block reaches consistency. Therefore, when the message generation rate is high enough, the block generation time almost remains unchanged. Fig. 7 plots the block generation time with regard to the number of consensus nodes, which indicates that more consensus nodes result in an increase in block generation time. This is because more nodes lead to more consensus time. Since it takes more time to reach a consensus on each block, the block generation time increases accordingly.

6.3.3. Message confirmation time

We further tested the message confirmation time. The message confirmation time refers to the time it takes when a message is submitted to the CCC and then recorded in a block, as shown in Fig. 8. Fig. 8 plots the message confirmation time with regard to the message generation rate. We can see that the confirmation time first decreases with the increase of message generation rate and then remains unchanged. The reason is that the message confirmation time is related to block generation time, and a high message generation rate makes block generation time decrease when the message generation rate is low. When the message generation rate is high enough, it does not influence the block generation time. Besides, we can also see that when the confirmation time of the weighted PBFT-based blockchain is as low as 1.1s, the proposed scheme is quite efficient.

6.3.4. Throughput

Throughput refers to the number of messages that the blockchain can process without high confirmation delay. In this part, we discuss the throughput of the weighted PBFT-based blockchain. Suppose the consensus time is T_{Con} , the threshold for block generation is Thr . When the message generation rate is high enough, the throughput TP can be calculated as $TP = \frac{Thr}{T_{Con}}$, since in this case, the block generation time converges to the consensus time because the blockchain leader quickly collects enough messages. Based on this formula, we calculate the throughput of the proposed blockchain system as 67 messages when the threshold is 100. The main reason constraining the throughput is the computational capacity of computers. Practical OCCs and the CCC will have more powerful computing power than the computer used in our experiments, which causes much-improved throughput. To achieve efficient message confirmation when the message generation rate is low, we set a time threshold for block generation, as described in Section 4. In practice, the time should be set around consensus time, so that low message confirmation delay can be ensured even there is a low message generation rate.

Fig. 10. The latency with regard to the number of EN nodes.

6.4. Results of Exp.2

6.4.1. Execution time of the smart contracts

We tested the average execution time of the three contracts with regard to the number of ENs, which are presented in Fig. 9 (a), 9(b), and 9(c), respectively. From the figures, we could see that the three contracts all achieve low execution time. When the number of ENs is 350, the average execution time is 132 ms, 0.59 ms, and 0.25 ms, respectively, which is quite efficient and means that the smart contracts will not introduce extra communication latency. Besides, we can also see that the execution time of contracts 1 and 2 is irrelevant to the EN number, while the execution time of contract 3 is linear to the number of ENs. However, considering the execution time of contract 3 is quite short, even when the EN number is high, its execution time will still keep at a low level. This means that the smart contracts can remain efficient even when there are a large number of ENs.

6.4.2. Data transmission latency

Employing the blockchain-based smart contracts for data transmission will incur a specific latency. We tested the data transmission latency, which is defined as the time duration from the time when the data is generated to the time when the data generator receives the notification message. Fig. 10 presents the data transmission latency with regard to the number of EN nodes. We observe that the data transmission latency is almost irrelevant to the number of EN nodes. Besides, the latency is around 6.5s, which implies that data transmission is very efficient.

7. Conclusions

In this paper, we investigated the automatic and adaptive data transmission control in TDL and proposed a blockchain-based data transmission scheme for TDL. We improved the traditional PBFT and proposed a weighted PBFT, which has higher anti-destruction ability than PBFT and outperforms PBFT in terms of block generation and message confirmation. We further employed the proposed weighted PBFT algorithm to build a blockchain system for TDL, based on which we designed a data transmission control scheme to achieve automatic and adaptive data transmission control. The experimental results verify the effectiveness and efficiency of our scheme.

Acknowledgments

This work is sponsored by the open grant of the Tactical Data Link Lab of the 20th Research Institute of China Electronics Technology Group Corporation, P.R. China (Grant CLDL-20182119), the National Natural Science Foundation of China under Grants 61672410 and 61802293, the Key Lab of Information Network Security, Ministry of Public Security (Grant C18614), the Academy of Finland (Grants 308087, 314203, and 335262), the Shaanxi Innovation Team project under grant 2018TD-007, and the 111 project under grant B16037.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.dcan.2020.05.007>.

References

- [1] A. Stoica, D. Militaru, D. Moldoveanu, A. Popa, Tactical data link - from link-1 to link-22, *Scientific Bulletin "Mircea cel Batran" Naval Academy* 19 (2) (2016) 346.
- [2] J.T. Sturdy, Military Data Link Integration Application, Tech. Rep, Honeywell Inc Albuquerque Nm Defense Avionics Systems Div, 2004.
- [3] J. Sorroche, R. Byers, N. Barrett, Tactical Digital Information Link-Technical Advice and Lexicon for Enabling Simulation (Tadil-ales) II: Link 11/11b, Tech. Rep., Asrc Communication Ltd Kirtland Afb Nm Distributed Mission Operations Center, 2006.
- [4] Z. Yan, Y. Chen, Y. Shen, Percontrep: a practical reputation system for pervasive content services, *J. Supercomput.* 70 (3) (2014) 1051–1074.
- [5] Z. Yan, P. Cofta, A mechanism for trust sustainability among trusted computing platforms, in: *International Conference on Trust, Privacy and Security in Digital Business*, Springer, 2004, pp. 11–19.
- [6] Z. Yan, Trust Modeling and Management in Digital Environments: from Social Concept to System Development, *Information Science Reference*, 2010.
- [7] Z. Yan, Trust Management in Mobile Environments: Autonomic and Useable Models, first ed., IGI Global, Hershey, PA, USA, 2013.
- [8] Z. Yan, R. MacLavery, Autonomic trust management in a component based software system, in: L.T. Yang, H. Jin, J. Ma, T. Ungerer (Eds.), *Autonomic and Trusted Computing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 279–292.
- [9] S. Nakamoto, Bitcoin: a peer-to-peer electronic cash system. <http://bitcoin.org/bitcoin.pdf>.
- [10] G. Wood, Ethereum: a secure decentralised generalised transaction ledger, *Ethereum project yellow paper* 151 (2014) 1–32.
- [11] S. Huh, S. Cho, S. Kim, Managing iot devices using blockchain platform, in: *2017 19th International Conference on Advanced Communication Technology (ICACT)*, IEEE, 2017, pp. 464–467.
- [12] K. Fan, S. Sun, Z. Yan, Q. Pan, H. Li, Y. Yang, A blockchain-based clock synchronization scheme in iot, *Future Generat. Comput. Syst.* 101 (2019) 524–533, <https://doi.org/10.1016/j.future.2019.06.007>. <http://www.sciencedirect.com/science/article/pii/S0167739X18326657>.
- [13] W. Feng, Z. Yan, Mcs-chain: decentralized and trustworthy mobile crowdsourcing based on blockchain, *Future Generat. Comput. Syst.* 95 (2019) 649–666, <https://doi.org/10.1016/j.future.2019.01.036>. URL, <http://www.sciencedirect.com/science/article/pii/S0167739X18326967>.
- [14] Z. Yan, L. Peng, Trust Evaluation Based on Blockchain in Pervasive Social Networking, *IEEE Blockchain Technical Briefs*.
- [15] M. Singh, S. Kim, Blockchain Based Intelligent Vehicle Data Sharing Framework, *arXiv preprint arXiv:1708.09721*.
- [16] M. Dinc, Design Considerations for Military Data Link Architecture in Enabling Integration of Intelligent Unmanned Air Vehicles (Uavs) with Navy Units.
- [17] M.E. Raber, The Coast Guard Shipboard Command and Control System and its Role in Future Joint Military operations., Tech. Rep, NAVAL WAR COLL NEWPORT RI, 1996.
- [18] M. Martinez-Ruiz, A. Arts-Rodrguez, J.A. Diaz-Rico, J.B. Fuentes, New initiatives for imagery transmission over a tactical data link. a case study: jpeg2000 compressed images transmitted in a link-16 network. method and results, in: *2010 - MILCOM 2010 Military Communications Conference*, 2010, pp. 1163–1168, <https://doi.org/10.1109/MILCOM.2010.5680102>.
- [19] W.D. O'Neil, The Cooperative Engagement Capability Ccc Transforming Naval Anti-air Warfare, Tech. Rep, National Defense Univ Washington DC Center for Technology and National Security Policy, 2007.
- [20] G.S. Schrecke, S. Davidson, M.A. Kahn, M.-C. Wang, M.W. Henry, Ip address translation for tactical networks, *US Patent App* 10/063 (2018) 516.
- [21] G.S. Schrecke, S. Davidson, M.A. Kahn, M.-C. Wang, M.W. Henry, Ip address translation for tactical networks, *US Patent App* 14/499 (2016) 835.
- [22] J.F. Bull, H.Z. Zebrowitz, M.A. Arnao, A versatile adaptive array for link-11 communications, in: *MILCOM 91 - Conference Record vol. 1*, 1991, pp. 241–245, <https://doi.org/10.1109/MILCOM.1991.258245>.
- [23] A.E. Yilmaz, R.M. Ozel, H.G. Ilk, Analysis and modeling of multiple tactical data links, in: *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, 2004, 2004, pp. 763–766, <https://doi.org/10.1109/SIU.2004.1338643>.
- [24] K. Dstner, B. Khler, F. Opitz, Information Fusion: Protection of Military Forces and Homeland Security, *INFORMATIK, Service Science-Neue Perspektiven fr die Informatik. Band 2*, 2010.
- [25] W. Wen-Zheng, L. Peng-Cheng, R. Pei, Z. Jing-Lun, Adaptive-frame-based dynamic slot assignment protocol for tactical data link system, in: *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing vol. 1*, 2009, pp. 709–714, <https://doi.org/10.1109/NSWCTC.2009.226>.
- [26] R. Galindez, T. Davis, Tactical satellite 3 cdl communications, a communications link for mission utility, in: *MILCOM 2007 - IEEE Military Communications Conference*, 2007, pp. 1–6, <https://doi.org/10.1109/MILCOM.2007.4455179>.
- [27] W.E. Hoekstra, Tactical Data Links and Interoperability, the Glue between Systems, Tech. Rep, Nato Consultation Command and Control Agency the, Hague (Netherlands), 2001.
- [28] I. Eyal, E.G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, *Commun. ACM* 61 (7) (2018) 95–102.
- [29] G. Zyskind, O. Nathan, et al., Decentralizing privacy: using blockchain to protect personal data. *2015 IEEE Security and Privacy Workshops*, IEEE, 2015, pp. 180–184.
- [30] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: *Proceedings of the Thirteenth EuroSys Conference*, ACM, 2018, p. 30.
- [31] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, Y. Zhang, Consortium blockchain for secure energy trading in industrial internet of things, *IEEE transactions on industrial informatics* 14 (8) (2017) 3690–3700.
- [32] O. Dib, K.-L. Brousmiche, A. Durand, E. Thea, E. B. Hamida, Consortium blockchains: overview, applications and challenges, *International Journal On Advances in Telecommunications* 11 (1&2).
- [33] L. Zhou, G. Wang, T. Cui, X. Xing, Cssp: the consortium blockchain model for improving the trustworthiness of network software services, in: *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, IEEE, 2017, pp. 101–107.
- [34] Y. Guo, C. Liang, Blockchain application and outlook in the banking industry, *Financial Innovation* 2 (1) (2016) 24.
- [35] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: *2014 USENIX Annual Technical Conference (USENIX ATC'14)*, 2014, pp. 305–319.