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

Yang, Xuetao; Li, Yafeng; Feng, Wei; Chen, Liang; Yan, Zheng

TDL-Chain: An Intelligent Data Transmission Control System in Tactical Data Link Based on Blockchain

Published in:
Proceedings - 2020 IEEE International Conference on Blockchain, Blockchain 2020

DOI:
[10.1109/Blockchain50366.2020.00045](https://doi.org/10.1109/Blockchain50366.2020.00045)

Published: 01/11/2020

Document Version
Peer reviewed version

Please cite the original version:
Yang, X., Li, Y., Feng, W., Chen, L., & Yan, Z. (2020). TDL-Chain: An Intelligent Data Transmission Control System in Tactical Data Link Based on Blockchain. In *Proceedings - 2020 IEEE International Conference on Blockchain, Blockchain 2020* (pp. 305-312). [9284706] IEEE.
<https://doi.org/10.1109/Blockchain50366.2020.00045>

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.

TDL-Chain: An Intelligent Data Transmission Control System in Tactical Data Link Based on Blockchain

Xuetao Yang

*School of Cyber Engineering
Xidian University
Xi'an, China
bsyangxt@foxmail.com*

Yafeng Li

*The 20th Research Institute of China
Electronics Technology Group Corporation
Xi'an, China
xxddxdd@yeah.net*

Liang Chen

*The 20th Research Institute of China
Electronics Technology Group Corporation
Xi'an, China
772029466@qq.com*

Wei Feng

*State Key Laboratory of Integrated Services
Networks, School of Cyber Engineering
Xidian University
Xi'an, China
weifeng.ft@foxmail.com*

Zheng Yan✉

*State Key Laboratory of Integrated Services
Networks, School of Cyber Engineering
Xidian University
Xi'an, China
Aalto University,
Espoo, Finland
zyan@xidian.edu.cn*

Abstract—Tactical data link (TDL) is a wireless communication system that achieves instant, accurate, automatic, and secure data transmission in military scenarios. It greatly reduces the time of Observation, Orientation, Decision, and Action (OODA). Modern TDL systems utilize a distributed architecture to enhance anti-destruction capacity and support data transmission among different TDLs. However, it is difficult to ensure data consistency among TDLs, which greatly reduces joint combat capacities of TDLs. A data transmission control system that can offer data consistency among various TDLs is highly expected. Besides, data transmission in TDLs should be controlled in an automatic and intelligent way to satisfy the low latency requirement of modern wars. To tackle this problem, in this paper, we employ blockchain-based smart contracts to build up a decentralized data transmission control system for TDLs called TDL-Chain. TDL-Chain achieves automatic data transmission and guarantees information consistency among various combat entities. The evaluation based on proof-of-concept implementation demonstrates the effectiveness and efficiency of TDL-Chain.

Index Terms—Tactical Data Link, Blockchain, Smart Contract, Data Transmission Control, Data Consistency

I. INTRODUCTION

A Tactical Data Link (TDL) is a wireless communication system that transmits data with a particular format in a real-time, accurate, automatic, and confidential way [1]. It can greatly reduce the time of Observation, Orientation, Decision, and Action (OODA) loop and hence attracts considerable attention. Original TDLs adopt the point-to-point architecture, which suffers from low anti-destruction capacity and cannot well support communications among various parties. To

overcome this weakness, current TDLs employ a distributed architecture, and the break-down of a single entity will not harm the whole communication system. At present, TDLs are widely applied in different military scenarios [2]–[5].

Although TDLs support the inter-connection of various combat platforms, it cannot ensure data consistency among different TDLs of different domains [6], which reduces its cooperative combat capacity. A possible solution is to build a data transmission control system that controls cross-domain data transmission with consistency. However, it is not trivial because of the critical requirements a military scenario has specifically critical requirements. For one thing, it needs to realize data transmission automatically rather than artificially for shorter OODA loop. For another, it should intelligently adjust its transmission strategies according to the dynamic battlefield scenarios. In addition, the data transmission control system should be realized in a decentralized way to suit itself for the distributed TDL architecture and for better anti-destruction capacity as well. In summary, an automatic, intelligent, and adaptive data transmission control system with data consistency is highly expected in TDLs.

Blockchain [7] is a decentralized ledger technology, in which data are packaged into blocks that are connected with a hash function to form a data chain. The underlying consensus mechanism and the data structure of blockchain ensure data integrity, consistency, and immunity. The advantages of blockchain, i.e., data consistency, decentralization, credibility, and traceability, make it suitable for TDL scenarios. Besides, it

is feasible to realize various functions in an automatic way by employing the smart contract platform based on blockchain. Therefore, we consider blockchain an effective technology for implementing an automatic, intelligent, and adaptive data transmission control system with data consistency in TDLs. Till now, several blockchain-based platforms are developed, and Ethereum and HyperLedger Fabric are two of the most ones. Their main difference lies in that Ethereum allows every node to work as a consensus nodes freely, while HyperLedger Fabric only allows authorized nodes to work as miners. For the sake of safety and efficiency, we use HyperLedger Fabric to implement the data transmission control system for TDLs.

Contributions: In this paper, we propose to employ blockchain to implement a decentralized and intelligent data transmission control system, named TDL-Chain. We implement TDL-Chain to support inter-domain communications among various TDLs and fulfill the specific requirements of modern TDLs. Driven by the military requirements on TDL, TDL-Chain builds intelligent data transmission control on top of HyperLedger Fabric. Besides, we introduce a cloud server for storage to reduce the storage overhead of blockchain, and the blockchain only stores the metadata such as contract codes and data digests. We further apply Attribute-Based Encryption (ABE) to offer fine-grained flexible access control for data security. In summary, the main contributions of this paper can be summarized as follows:

- 1) We employ blockchain-based smart contracts [8] to design a blockchain-based intelligent data transmission control system for TDLs (i.e., TDL-Chain) to achieve data consistency, autonomy, and intelligence.
- 2) We design a cloud-based off-chain storage system for storage efficiency and employ Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [9] to ensure data confidentiality with fine-grained access control.
- 3) We implement TDL-Chain with HyperLedger Fabric and conduct a series of experiments to evaluate its performance, the experimental results demonstrate its effectiveness and efficiency.

The rest of the paper is organized as follows. Section II elaborates on our research background and related works. Section III presents the problem statement of our work including system model, security model, and preliminaries. In Section IV, we describe the design details of TDL-Chain, followed by TDL-Chain implementation and performance evaluation in Section V. Finally, we conclude our paper in the last section.

II. RELATED WORKS

In this section, we introduce the research and development of TDL and blockchain.

A. Tactical Data Link (TDL)

TDL was proposed in the 1950s to reduce the OODA loop time. It is significant for the battlefield to make wise decisions and respond to battlefield events quickly [10]. Original

TDLs (such as Link-4) used a point-to-point data transmission method, which cannot effectively support data broadcasting. Besides, data in them are transmitted in plaintext, which fails to resist eavesdropping and suffers from poor confidentiality. In response to these weaknesses, the United States has further developed the Link-11 that supports data encryption to enhance data security.

For better resistance to destruction, the United States further developed Link-16 and Link-22, both of which adopt a fully distributed architecture. In Link-22, all nodes are equivalent and realize same functions, which increases the flexibility and reliability. Compared with the centralized and point-to-point architectures, the anti-destruction ability of Link-22 is greatly improved. Moreover, Link-22 also supports inter-TDL communications, and hence, it enhances the joint combat capability [10]. However, it fails to ensure data consistency among different TDLs, which is an essential feature for cooperative combat actions. So far, the TDL has been greatly developed, but there still lacks exploration on how to realize automatic and intelligent data transmission control across multiple tactical domains with data consistency.

B. Blockchain and its Applications

Blockchain is a decentralized ledger technology, in which data blocks are connected with hash function to form a data chain [12]. It utilizes a consensus mechanism to ensure the consistency of the data stored by different entities [11]. So far, blockchain receives considerable attention owing to its immutability, decentralization, consistency, immunity, and traceability [7].

There are three types of blockchain, i.e., public blockchain, private blockchain, and consortium blockchain. Among them, the public blockchain adopts a decentralized architecture where all information in the blockchain is open to the public. In a public blockchain, any node can become a miner to generate new blocks, record transactions, and maintain the blockchain system. The private chain is a centralized architecture, and the generation and maintenance of the system are completely completed by a centralized entity. The consortium blockchain adopts a semi-decentralized architecture. This is because the maintenance of a consortium blockchain is usually conducted by some authorized nodes deployed by several centralized parties, while other nodes only have access to the data in blockchain according to their permissions.

Nowadays, blockchain is widely applied in many fields such as cryptocurrency, smart contracts, Internet of Things (IoT), digital asset management (DSM), and intellectual property [13]. Besides, blockchain shows a promising prospect in the military field in terms of information protection, weapon equipment management, and intelligent military logistics. However, how to employ blockchain for intelligent and automatic data transmission control in TDLs is rarely explored.

1) *Attribute-based encryption (ABE)*: ABE [14] is a public key cryptographic technique in which the data owner's secret key and ciphertext depend on attributes of recipients to implement data access control. It contains two types: Key-Policy ABE(KP-ABE) and Ciphertext-Policy ABE (CP-ABE). In KP-ABE, the data access policy is embedded in the key, and the attribute set is embedded in the ciphertext. Each private key is associated with an access policy, which can control which ciphertext the key can decrypt. Differently, in CP-ABE,

TABLE I
NOTATIONS

Notations	Descriptions
CCC	The chief command center.
OCC_i	The operation command center i .
$Param$	The security parameter.
PK	The public parameter.
MSK	The master secret key for attribute-based encryption.
M	The plain message.
CT	The encrypted message of M .
T	Attribute structure.
S_i	The access policy corresponding to OCC_i .
S	The set of access policies, and $S = \{S_1, \dots, S_n\}$.
$Cert_{N_i}$	The certificate of N_i .
PK_{N_i}/SK_{N_i}	The long-term public/private key pair of N_i

the data access policy is embedded in the ciphertext, and the attribute set is embedded in the key, and it can be decrypted if and only if the attribute can meet the access policy. In the TDL scenario, the number of all entities is huge and a large amount of data is generated. However, access policies vary for different types of data. If we apply KP-ABE in the TDL-Chain, the CA needs to frequently generate attribute-based keys for different entities, which incurs great communication overhead. Therefore, our scheme adopts CP-ABE.

2) *HyperLedger Fabric*: HyperLedger Fabric is a consortium blockchain system that supports smart contracts. HyperLedger Fabric project was submitted to the Linux Foundation by IBM and DAB in 2015. It is the first open-source blockchain-based project on consortium blockchain with privacy preservation [15]. Fabric adopts logically-separated architecture for easier management of its components. Fabric provides a smart contract platform that can be used to realize various functions. Therefore, it is a suitable for TDL owing to its decentralization, scalability, efficiency, data consistency, and support to smart contracts [16]. Therefore, we choose to use HyperLedger Fabric to build the network of TDL-Chain system. For easy presentation, we summarize all the notations used in the paper in table I.

IV. DESIGN OF TDL-CHAIN

In this section, we design a blockchain-based intelligent data transmission control system for TDL.

A. Overview

As shown in Fig. 1, the TDL-Chain system consists of three parts, the blockchain network maintained by the CCC and OCC, the cloud server, and the ENs that follow the strategies and decisions issued by the blockchain to transmit data and conduct combat tasks. We have developed a management system for TDL-Chain based on the SpringBoot framework. Clients can remotely access our system to view system data. In the TDL-Chain system, CCC, OCC, and ENs cooperate to jointly maintain data consistency, secure transmission, permission control, and intelligent decision-making in military scenarios. We use smart contracts to manage data processing,

data transmission, and intelligent decision making. A large amount of data such as battlefield data collected by EN is encrypted and stored on the cloud storage by OCC using a CP-ABE, and the key of each OCC is generated by the CA.

Fig. 2. Function Design of TDL-Chain

Each type of entities implements several functions, as shown in Fig. 2. For the CCC and OCCs, each entity maintains a database that securely stores the public/private key pair, attribute-based keys, certificate, and other battlefield data, etc. The blockchain manager maintains the blockchain system with the underlying consensus mechanism. The communicator is responsible for data exchange with other entities. A contract manager that detects the blockchain status and invokes smart contracts when particular conditions are satisfied. The smart contracts in the CCC or an OCC mainly implement four types of functions: data processor, status judge, strategy maker, and data manager. Among them, the data processor processes the raw data uploaded by an EN and normalizes them in a unified data format; the status judge obtains the combat status, based on which the strategy maker further decides the data transmission strategies and combat decisions; data manager is responsible for managing the blockchain data. Finally, there is a user interface (UI) for easy interaction between human users and the system.

For the CA, it maintains a database that securely stores the information of all entities, including attribute information, permission information, certificate, long-term public/private keys, attribute-based keys. It uses a communicator to exchange information with other entities and manages these data with an entity profile manager. We also implement a key manager for the CA, which is responsible for issuing, updating, and revoking both the long-term keys and attribute-based keys for all entities.

An EN is equipped with a data collector and an executor, of which the data collector is responsible for data collection and uploading, and the executor enables the EN to execute the combat strategies received by the strategy receiver from the corresponding OCC or the CCC. It also maintains a local database that stores its personal information and key items such as long-term/attribute-based keys, combat strategies, and certificates. A UI is implemented for soldiers to use the system easier.

The cloud server is responsible for secure data storage. Its database stores all the data collected by ENs to ease the storage burden of the blockchain system. It is also responsible to distribute the data to data requestors, following the access strategies established by the CA. It mainly uses a permission verifier to check the permission of the data requestor to decide whether to share the data with it.

B. ABE Based Secure Data Storage

Before we introduce the design of the data transmission control system, we will first present the details of the data storage method of TDL-Chain. For secure storage, we use CP-ABE to achieve fine-grained data access control in the cloud server. The secure storage is composed of several procedures: *Setup*, *Encryption*, *Key Generation*, and *Decryption*, as shown in Fig. 3. The details of these procedures are described below.

- i. *Setup*: When the system sets up, the CA takes the security parameter $Param$ as input and generates the public parameter PK and a master secret key MSK for attribute-based encryption. Then, it sends PK to the CCC and all OCCs.
- ii. *Encryption*: After an OCC identified by OCC_i obtains the PK , it can encrypt the data submitted by ENs with an attribute-based access structure T . Given a data M in plaintext and the corresponding access structure T , OCC_i inputs PK , T , and M to the encryption algorithm to obtain the ciphertext CT . It then sends the CT to the cloud server for storage.
- iii. *Key Generation*: OCC_i specifies the access policies $S_1, S_2, \dots, S_n$ corresponding to $OCC_1, OCC_2 \dots OCC_n$, and sends the policy set $S = \{S_1, \dots, S_n\}$ to the CA. The CA generates corresponding decryption keys $SK_1, SK_2, \dots, SK_n$ according to the policy S and sends them to the corresponding OCC.
- iv. *Decryption*: After other OCCs obtain the cipher text CT from the cloud server, they input the CT , the public parameter PK , and the SK corresponding to the attribute set S into the decryption algorithm. If its attributes satisfy the access structure T , it can decrypt CT successfully to obtain the plaintext data M .

Fig. 3. CP-ABE based data storage of TDL-Chain

C. Intelligent Data Transmission Control

TDL-Chain achieves intelligent and automatic data transmission control and combat decision establishment with blockchain-based smart contracts. After the smart contract is compiled and deployed in the blockchain, the system can

invoke and execute the smart contracts by submitting a relevant message to the blockchain. In our proposed TDL-Chain, we also use smart contracts to manage the data in the system. The corresponding smart contracts are described in detail as below.

- 1) *Contract 1 (Data Processing)*: The contract pre-processes the raw data sensed by the ENs. It first decrypts the encrypted data and normalizes the data into a particular format for easy processing. Besides, the contract is also used to encrypt or decrypt the data with CP-ABE.
- 2) *Contract 2 (Status Judgement)*: The contract can judge the status of each node and the battlefield situation, and according to different combat environments, we can design contracts. For example, infantry operations mainly calculate infantry status and firepower values, naval operations mainly calculate warship capacity, shipboard artillery shells, etc., Air Force operations mainly decide fighter flight mileage, airborne artillery shells, etc.
- 3) *Contract 3 (Transmission Strategy and Combat Decision Establishment)*: Based on the calculation results of Contract 2, this contract analyzes the current combat status of each entity and intelligently generates transmission strategies and combat decisions. Specifically, for the EN in a disadvantaged combat state, the contract will determine the optimal support unit nearby, and strategically support this EN to achieve the needs of multi-arms cooperative operations.
- 4) *Contract 4 (Blockchain Data Management)*: The blockchain records message digests of all data and other sensitive information such as contract content. This contract is responsible for the management of the data stored on the blockchain. It can realize functions such as data appending, deletion, and querying to manage the dynamic data of blockchain at any time.

D. Procedures

To better illustrate how TDL-Chain works, we present its procedures in Fig. 4. The whole process can be divided into seven steps, i.e., *System Initialization*, *Registration*, *Data Collection*, *Data Storage*, *Data Access*, *Consensus Establishment*, *Execution*, and *Data Access*.

When the system is initialized, each node in the system generates a unique identifier, which represents the unique identity of this entity in the system. Clients can obtain access to system data through system registration. CCC as CA will generate the corresponding public key and secret key for the required OCC based on CP-ABE. OCC can specify an access policy to enable system nodes to securely access data on the cloud. Only those who meet the policy can obtain data in plaintext. When OCC obtains the battlefield data, it will process the data through the smart contract that we designed and loaded in the blockchain. The system can analyze data through smart contracts, and generate transmission strategies and decisions. All OCCs and CCC publish related messages on the blockchain when receiving data. Nodes can check the data on the blockchain to determine whether other nodes

have received the data, thereby ensuring data consistency. The detailed procedures are described below.

- i. *System Initialization*: During this process, the CA takes the security parameters $Param$ as input to generate the public security parameters PK and the master secret key MSK . Then it broadcasts the $Param$ to all entities.
- ii. *Registration*: All the entities have to register to the CA with their own identities. We assume that for each entity N_i , it has a unique identifier denoted as ID_{N_i} . The CA verifies the validity of the identity and issues a unique certificate $Cert_{N_i}$ and a long-term key pair PK_{N_i}/SK_{N_i} to N_i . Besides, it will also decide the attribute set of N_i , and based on it the CA generates the corresponding attribute-based keys to N_i .
- iii. *Data Collection*: Some ENs are equipped with sensors to monitor the battlefield. These ENs continuously collect battlefield data and send them to the corresponding OCC, and the OCC verifies the validity of the EN to decide whether to accept the data.
- iv. *Data Storage*: After the EN node passes the verification, the OCC will assign corresponding access policies to other nodes based on their attributes. The OCC invokes *Contract 1* to normalize the data packet and encrypts it via CP-ABE. Finally, the OCC uploads the normalized and encrypted data packet CT to the cloud server and then broadcasts the data digest to the blockchain network. The cloud server stores the encrypted data and organizes the data with its corresponding hash digest, which enables others to query the data quickly and verify the integrity of the data.
- v. *Data Access*: After another OCC receives the broadcasted message digest, it can send a request to the cloud server for the encrypted data. The OCC then decrypts the ciphertext and invokes the *Contract 2* to obtain the battlefield status. Finally, the OCC sends the contract execution results to the CCC.
- vi. *Consensus Establishment*: After receiving the calculation results of other OCCs, the CCC node aggregates all the information and uses the *Contract 3* to generate the data transmission strategies and combat decisions. The transmission strategy indicates the transmission priorities and valid receivers under current battlefield status. Then the CCC sends the information transmission strategies and the combat decisions to the OCCs, which further sends the combat decisions to the corresponding ENs for execution.
- vii. *Execution*: After receiving the combat decisions, the ENs execute the commands and reply to the corresponding OCC with a confirmation message.
- viii. *Data Access*: All the entities can actively request data from the cloud server. However, because of CP-ABE, only when the attributes of a data requestor satisfy the access structure can it decrypt the ciphertext successfully. Besides, the CCC and OCCs can manage the blockchain data by using *Contract 4*.

Fig. 4. Procedures of TDL-Chain

V. IMPLEMENTATION AND PERFORMANCE EVALUATION

In this section, we conduct a series of experiments to test TDL-Chain's performance based on a proof-of-concept implementation by following established evaluation metrics.

A. Evaluation Metrics

The performance evaluation mainly focuses on the efficiency of TDL-Chain. In our evaluation, we mainly use four metrics: *Execution Time of Contract*, *Latency*, *Throughput*, and *Communication Cost*, whose definitions are presented as follows.

- 1) *Execution Time of Smart Contracts*: The efficiency of a smart contract is measured by the execution time of the smart contract. TDLs usually require low response time and thereby require high smart contract efficiency.
- 2) *Latency*: TDLs require low latency so that the combat units can react to the tactical events quickly. In our analysis, the latency is measured by the length of duration from the time when an EN generates a data report to the time when it receives the notification of data transmission.
- 3) *Throughput*: The throughput refers to the average number of messages successfully stored in the TDL-Chain system per second.
- 4) *Communication Cost*: The communication cost generally refers to the communication volume in a system network, which is reduced by the size of the communication data and the number of data transmissions.

B. Experimental Settings

In this subsection, we introduce the experimental settings.

TDL-Chain Implementation: We implemented TDL-Chain with the Go language and Java language, and the interface of TDL-Chain is shown in Fig. 5. To be specific, we developed the TDL-Chain with Golang 1.9 and JDK 1.8 based on the components of SpringBoot 2.0.4, HyperLedger Fabric 1.1.0, and Fabric-SDK-Java 1.1.0. All the experiments are conducted on a laptop running Windows 10 host with an Intel Core i5 processor (1.7GHz) and 8G memory.

Experimental Settings: In this part, we introduce the experimental settings of the experiment, which are summarized in Table II. We simulated a real battlefield scenario with the battlefield deduction data. Specifically, we integrated the entire system under the framework of SpringBoot 2.0.4, and we built a blockchain system based on HyperLedger Fabric 1.1.0. We divided the entire battlefield into 4 regions, each of which

TABLE II
EXPERIMENTAL SETTINGS

Experimental Settings	
Development Language	Java/Go
Development Framework	SpringBoot 2.0.4
Blockchain Network Platform	HyperLedger Fabric 1.1.0
CCC Number	1
OCC Number	4
EN Number	0-350

deploys an OCC, and the four OCCs and the CCC jointly build a blockchain network. The CCC and all OCCs use docker images to simulate virtual machines running on CentOS 7.0. We used MySQL 5.6 to simulate a cloud server to provide storage services. Besides, we used Fabric-SDK-Java 1.1.0 to interact with the underlying blockchain network. With multi-threaded programming, we simulated different ENs to test system performance, and each thread simulated one EN. In the experiment, we tested the system performance with the aforementioned evaluation metrics.

Fig. 5. TDL-Chain Interface

Fig. 6. Execution Time of Smart Contracts

C. Results of Experiment

In this subsection, we present all the experimental results.

Fig. 7. Latency of TDL-Chain

1) *Execution Time of Smart Contract*: We tested the relationship between the execution time of the four smart contracts and the number of EN nodes, as shown in Fig. 6. It can be seen that the execution time of *Contract 1* and *Contract 2* is not related to the number of EN nodes. Differently, the execution time of *Contract 3* is linear with the number of EN nodes, and it increases with the increase of the EN number. The execution time of *Contract 4* first decreases as the number of ENs increases and reaches the minimum value (i.e., 17ms). Then, the curve tends to increase with the increase of the EN number. When the number of EN nodes is 100, the average execution time of *Contract 1*, *Contract 2*, *Contract 3*, and *Contract 4* are 358ms, 0.601ms, 0.855ms, and 30ms, respectively, which illustrates the efficiency of the contracts. In addition, although the execution time of *Contract 3* is proportional to the number of EN nodes, it can be seen from Fig. 6(c) that the growth rate is low. Besides, one main reason limiting the efficiency is the computing capacity of the implementation platform, and in the real world, both the CCC and OCCs are equipped with much more powerful machines, which can further reduce the execution time of contracts. In summary, our smart contracts can achieve efficient data processing, intelligent data transmission control, and decision-making operations for TDL that contains a large number of ENs

2) *Latency*: In our TDL-Chain system, using smart contracts for data transmission will cause a longer transmission delay. We tested the data transmission delay, which is defined as the length of duration from when the EN generates data packet to when the EN receives the execution command generated by the CCC with *Contract 3*. Fig. 7 shows the data transmission latency for the number of EN nodes. We observe that the data transmission latency is almost independent of the number of EN nodes. In addition, the delay is about 6.3s, which means that the data transmission is very efficient. The main reason is that, for one thing, we use HyperLedger Fabric as the underlying blockchain, which enables parallelized processing of transactions and has proven to be efficiency; for another, the efficient contracts designed by us also help reduce the latency.

Fig. 8. Throughput of TDL-Chain

3) *Throughput*: We tested the relationship between the throughput of the system and the number of ENs, and the experimental results are shown in Fig. 8. It can be seen from the figure that when the number of EN is less than 250, the TPS curve rises with the increase in the number of EN. When the EN reaches 250, the throughput reaches the maximum value (58). After that, the curve gradually decreases. This is mainly due to the limited processing capacity of the server. In practice, we can improve the system's throughput by improving the computing ability of the server.

4) *Communication Cost*: We mainly evaluate the communication resulted from a data report generated by an EN. During this process, the main communication overhead is generated by the transmission of the encrypted data and the consensus process. Since we employ CP-ABE, the size of the encrypted data report is linear to the attribute number A , which is decided ahead of time and is a constant. In the worst case where every OCC needs to download the encrypted data report from the cloud server, the total communication overhead in this process should be $\mathcal{O}(AN)$, where N is the number of all consensus nodes including the CCC. For the consensus process, since HyperLedger Fabric leverages Kafka consensus mechanism, whose communication complexity is $\mathcal{O}(N)$. As a result, in the worst case, the total communication complexity should be $\mathcal{O}(NA + N)$. Since A is a constant, the communication overhead can be considered as $\mathcal{O}(N)$.

VI. CONCLUSION

In this paper, we designed and implemented TDL-Chain, a blockchain-based intelligent data transmission control system for TDL, aiming to solve the problem of information inconsistency across multiple TDL domains. TDL-Chain guarantees information consistency among combat entities in a battlefield and meets the needs of modern warfare by supporting intelligent and automatic cooperations across multiple TDL domains. Secure data sharing based on ABE can protect the security of battlefield data. System implementation based on HyperLedger Fabric and related performance tests further show the effectiveness of TDL-Chain.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61672410 and Grant 62072351; in part by the Academy of Finland under Grant 308087, Grant 314203, and Grant 335262; in part by the open grant of the Tactical Data Link Lab of the 20th Research Institute of China Electronics Technology Group Corporation, China, under Grant CLDL-20182119; in part by the Shaanxi Innovation Team Project under Grant 2018TD-007; and in part by the 111 Project under Grant B16037.

REFERENCES

- [1] J. T. Sturdy, "Military data link integration application," Honeywell Inc Albuquerque Nm Defense Avionics Systems Div, Tech. Rep., 2004.
- [2] A. E. Yilmaz, R. M. Ozel, and H. G. Ilk, "Analysis and modeling of multiple tactical data links," in *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, 2004., April 2004, pp. 763–766.
- [3] K. Dästner, B. Köhler, and F. Opitz, "Information fusion: protection of military forces and homeland security," *Infomatic 2010. Service Science—Neue Perspektiven für die Informatik. Band 2*, 2010.
- [4] W. Wen-Zheng, L. Peng-Cheng, R. Pei, and 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, April 2009, pp. 709–714.
- [5] R. Galindez and T. Davis, "Tactical satellite 3 cdl communications, a communications link for mission utility," in *MILCOM 2007 - IEEE Military Communications Conference*, Oct 2007, pp. 1–6.
- [6] J. Sorroche, "Tactical digital information link-technical advice and lexicon for enabling simulation (tadil-tales)," ASRC Communication Ltd Kirtland Afb Nm Distributed Mission Operations Center, Tech. Rep., 2005.
- [7] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)*. IEEE, 2017, pp. 557–564.
- [8] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.
- [9] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy (SP'07)*. IEEE, 2007, pp. 321–334.
- [10] A. Stoica, D. Militaru, D. Moldoveanu, and A. Popa, "Tactical data link-from link-1 to link-22," *Sci. Bull. Mircea cel Batran Naval Acad.*, vol. 19, no. 2, p. 346, 2016.
- [11] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, "A review on consensus algorithm of blockchain," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 2567–2572.
- [12] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography Mailing list at https://metzdowd.com*, 03 2009.
- [13] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.
- [14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 89–98.
- [15] O. Dib, K.-L. Brousmiche, A. Durand, E. Thea, and E. B. Hamida, "Consortium blockchains: Overview, applications and challenges," *International Journal On Advances in Telecommunications*, vol. 11, no. 1&2, 2018.
- [16] 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*, 2018, pp. 1–15.