Blockchain Meets Edge Computing: A Distributed and Trusted Authentication System

Shaoyong Guo, Xing Hu, Song Guo, *Senior Member, IEEE,* Xuesong Qiu, *Member, IEEE,* Feng Qi *

***Abstract*—As the great prevalence of various Internet of Things (IoT) terminals, how to solve the problem of isolated information among different IoT platforms attracts attention from both academia and industry. It is necessary to establish a trusted access system to achieve secure authentication and collaborative sharing. Therefore, this paper proposes a distributed and trusted authentication system based on Blockchain and edge computing, aiming to improve authentication efficiency. This system consists of Physical network layer, Blockchain edge layer and Blockchain network layer. Through the Blockchain network, an optimized practical Byzantine fault tolerance (PBFT) consensus algorithm is designed to construct a consortium Blockchain for storing authentication data and logs. It guarantees trusted authentication and achieves activity traceability of terminals. Furthermore, edge computing is applied in Blockchain edge nodes, to provide name resolution and edge authentication service based on smart contracts. Meanwhile, an asymmetric cryptography is designed, to prevent connection between nodes and terminals from being attacked. And a caching strategy based on edge computing is proposed to improve hit ratio. Our proposed authentication mechanism is evaluated with respect to communication and computation costs. Simulation results show that the caching strategy outperforms existing edge computing strategies by 6%-12% in terms of average delay, and 8%-14% in hit ratio.**

***Index Terms*—Blockchain, IoT, edge computing, trusted authentication, cryptography, caching strategy**

I. Introduction

As the rapid development of information and communication technology, IoT technology has been widely used in modern society [1]. Traditional IoT platforms usually adopt cloud computing to handle big data stream generated by various terminals [2]. However, these centralized platforms are isolated and incompatible from each other, facing difficulty with sharing information among different platforms. Furthermore, user privacy can be exposed easily once the centralized authority is attacked. Therefore, it is urgent to realize distributed and trusted authentication among IoT platforms.

For establishing trusted IoT platforms and supporting uniform access models, Blockchain attracts significant attentions from both academia and industry. First proposed by Satoshi Nakamoto in 2008 as a decentralized peer-to-peer network platform, Blockchain has been one of the most promising technologies to meet security requirements of IoT networks [3], [4]. It confirms integrity and validity of networks through computational-intensive tasks like Proof-of-Work puzzle.

Although many researchers are dedicated to the application of Blockchains, most of the existing Blockchains require a large amount of computing and storage resources. For example, Zhe et al. [5] proposed a reputation system for data credibility assessment, where cars with higher computation capability acted as miners. Dorri et al. [6] chose gateways as miners in Smart Home. However, few works focused on improving performance of Blockchain system when it works in the edge. Some studies adopted edge computing for supporting services in Blockchain networks [7], [8], such as blockchain based edge-as-a-service framework [9] and edge computing enabled wireless blockchain framework [10]. Therefore, this paper adopts edge computing for supporting edge authentication service in the Blockchain system. Edge computing is applied in Blockchain edge nodes to enhance computation and storage capability.

Existing works adopting Blockchain in various IoT scenarios, such as intelligent transportation [11], smart grid [9], [12], smart medical [13] and so on, are mainly designed for security or privacy. Few studies focused on efficient authentication and collaborative sharing among different platforms. Therefore, we propose a distributed and trusted authentication system that combines edge computing and Blockchain, to provide efficient authentication for smart terminals. The system consists of physical network layer, Blockchain edge layer and Blockchain network layer. Blockchain network layer works as an underlying supportive layer, storing authentication data and logs with an optimized PBFT consensus algorithm. Blockchain edge layer contains two kinds of edge nodes, in which resolution edge nodes provide name resolution service, and cache nodes achieve edge authentication service. A cryptography based on Elliptic Curve Cryptography (ECC) is designed in order to guarantee edge security. In particular, a caching strategy based on edge computing is proposed to update caching of Blockchain edge nodes and improve hit ratio.

Main contributions of this paper are summarized as follows:

- This paper proposes a distributed and trusted authentication system with Blockchain and edge computing. In the Blockchain network, an optimized PBFT consensus algorithm is designed for storing authentication data and logs. It guarantees trusted authentication and achieves activity traceability of terminals.
- A distributed authentication mechanism is designed by leveraging a dynamic name resolution strategy and ECC. With the name resolution strategy, edge nodes can syn-

This work is supported by National Key Research and Development Project (No. 2018YFB1800704), and National Natural Science Foundation of China under Grant (601702048, 61872310).

Shaoyong Guo, Xing Hu, Xuesong Qiu and Feng Qi are with the Beijing University of Posts and Communications, State Key Laboratory of Networking and Switching Technology, China (email: {syguo, xinghu, xsqiu, qifeng} @bupt.edu.cn).

Song Guo is with the Department of Computing and Research Institute for Sustainable Urban Development, Hong Kong Polytechnic University, Hong Kong (email: song.guo@polyu.edu.hk).

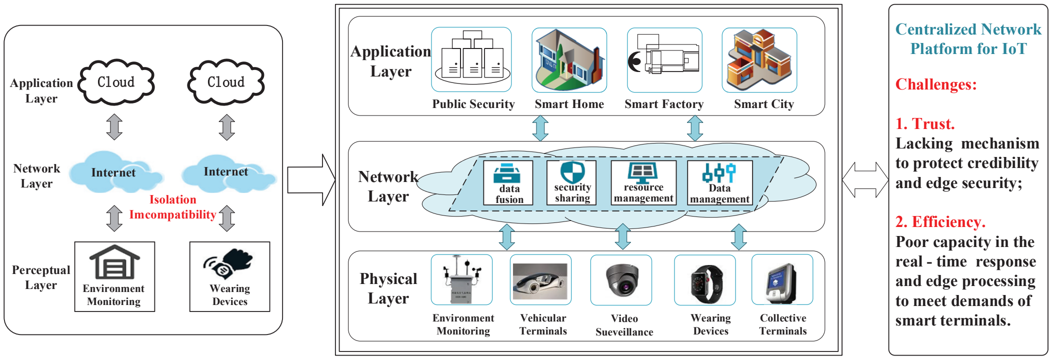

Fig. 1. IoT architecture

chronize terminal data timely. Meanwhile, the cryptography can preserve identity confidentiality and communication security among edge nodes and terminals.

- A caching strategy based on belief propagation (BP) algorithm is put forward to improve hit ratio and minimize delay. Compared with traditional caching strategies that cannot deal with mobile terminals, the strategy relying on smart contracts can optimize allocation of caching space dynamically.

The rest of this paper is organized as follows: Section II reviews the related work. Section III presents the system model. A distributed authentication mechanism introduced in Section IV. Then we propose a caching strategy according to BP algorithm in Section V. The experimental results and corresponding discussions are described in Section VI. Section VII is the conclusion and future work.

II. Related Work

A. Development of IoT

Integrating a plethora of various sensors, actuators, and smart meters across a wide spectrum of businesses [16], IoT technology goes through four stages of developments as shown in Fig. 1. Early forms of IoT platforms are highly isolated, incompatible and proprietary connected islands. After that, centralized processing architectures relying on cloud computing are adopted to handle massive data flow, making a difference to the manner of data processing. Peer et al. [17] proposed cloud-based biometric services to provide powerful storage and unprecedented processing power. Kohlwey et al. [18] presented a Hadoop-based cloud computing system, which improved matching efficiency in image recognition.

Though cloud computing improves computational and storage capability in centralized architecture, it performs poorly in promoting edge processing and protecting data security. Xu et al. [19] reviewed researches on IoT from the industrial perspective, and introduced main technologies for supporting IoT, such as social networks, radio-frequency identification (RFID), cloud computing, etc. They indicated that information security and privacy protection were two difficult issues in IoT because of devices' mobility and network complexity.

Many novel protocols and networks are applied to ensure secure authentication in IoT, e. g., cooperative message authentication protocols [20], group-oriented-range-bound authenticated key agreements [21], lightweight mutual authentication protocols [22], etc. However, they usually lack flexibility and scalability. Therefore, with the prevalent of digital cryptocurrency, Blockchain is introduced as a promising solution to provide protected privacy and trusted authentication service.

B. Trusted Authentication Based on Blockchain

Generally, the using of Blockchain can be concluded as 3 types: acting as a distributed ledger, realizing decentralized storing, or supporting distributed services relying on smart contracts. For example, Matev et al. [23] introduced an autonomous Blockchain to select the most convenient electric terminal charging station. Li et al. [24] designed a privacy-preserving incentive announcement network that achieved conditional privacy for mobile terminals. Khan et al. [25] proposed Blockchain based electrical transactions in microgrid. Then Blockchain is adopted to support device certificating and real-time monitoring in this paper.

Nevertheless, authentication efficiency is a challenge to be solved in the cloud-based Blockchain network. Tselios et al. [26] regarded Blockchain as a significant security factor for software defined network (SDN) based cloud computing infrastructure. Ali et al. [27] focused on solving trust problem in cloud-centric IoT network by smart contracts. Haipeng et al. [28] modeled the interaction between the cloud provider and miners as a Stackelberg game, to optimize resource management and pricing problem. Although dedicated to ensuring data validity, these works ignore vast costs on energy and computation resources. To promote edge processing, edge computing is considered as a new computing paradigm.

Utilizing computation and storage capacity of edge computing, fixed and mobile terminals can be operated in distributed manners. Zehui et al. [29] analyzed the advantages of facilitating blockchain applications in future mobile IoT system. Mengting et al. [10] proposed a novel mobile edge computing enabled wireless blockchain framework, where the computation-intensive mining tasks can be offloaded to nearby edge nodes. However, these works did not consider authentication efficiency. Zhonglin et al. [30] proposed a security authentication scheme of 5G Ultra-Dense network based on Blockchain and designed a APG-PBFT algorithm. Yongxu et al. [31] constructed a decentralized platform for storing and trading information in the air-to-ground IoT network. Focused on data security, collaborative sharing among different IoT platforms were not discussed in these papers. Anish et al. [12] proposed a blockchain based edge-as-a-service framework for secure energy trading in vehicle-to-grid networks, aiming at improving performance of SDN networks. Wentong et al. [32] designed a Blockchain-based cross-domain authentication model in a distributed environment. But efficiency and limited resources of the Blockchain were ignored in the process of cross-domain authentication in [12], [32].

In summary, existing literature for Blockchain systems have achieved a variety of properties such as anonymity, decentralization and system transparency. However, less attention has been paid to achieve efficient authentication among different IoT platforms. Hence, this paper proposes a distributed and trusted authentication system based on Blockchain and edge computing. With edge computing technology, Blockchain edge nodes can offer name resolution and edge authentication service. Meanwhile, a caching strategy is designed to improve authentication efficiency further.

C. Other Preliminaries

1) Elliptic Curve Cryptography (ECC): As a public key cryptography, ECC ensures security depending on the ability to compute a point multiplication with a random point, as well as the inability to figure out a multiplicand given the original curve and product points [14].

An elliptic curve E is a plane curve over a prime finite field E_p, which is defined by the equation: $y = x^3 + ax + b$. All points on E and infinity point O form a cyclic group G. Consider two cyclic groups G_1 and G_2 with the same prime order q. G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group. Define $e : G_1 \times G_1 \to G_2$ with basic properties of bilinear map, i. e.,

- Non-degeneracy: There exists $P, Q \in G_1$ so that $e\left(P, Q\right) \neq 1$.
- Bilinearity: $e\left(P + R, Q\right) = e\left(P, Q\right) \cdot e\left(P, R\right)$ and $e\left(aP, bQ\right) = e(P, Q)^{ab}, \forall a, b \in Z_q^*, \forall P, Q, R \in G_1$.
- Computability: It is efficient to compute $e\left(P + R, Q\right), \forall P, Q, \in G_1$.

2) Consensus Algorithm: The consortium blockchain is adopted to establish a trusted authentication system in this paper. Based on the consortium blockchain, a variety of consensus algorithms have been designed as shown in Table I, such like proof of work (PoW), proof of stake (PoS), delegated PoS (dPoS), casper, proof of elapsed time (PoET) and PBFT [15]. Usually, each can be divided into three parts: verifying identity, selecting primary peers and synchronizing data in the Blockchain. To meet the high real-time requirement, the PBFT algorithm with no token required is applied.

Most alliance members forming the consortium blockchain are trusted and valid, such as governments, service operators and big enterprises. Under this circumstance, we propose an optimized PBFT algorithm to improve authentication efficiency, where the primary peer is selected through round robin rather than computing complex puzzles. Furthermore, to alleviate storage and computation burden of Blockchain, tasks of resolving and recording data generated by numerous terminals are conducted by edge nodes. In this way, the consensus algorithm is executed only for verifying identity and storing authentication logs in the Blockchain, achieving data traceability and preventing data tampering.

TABLE I
COMPARISON OF CONSENSUS ALGORITHMS

Algorithm	PoS	DPoS	Casper	PoET	PBFT
Decentralized	complete	complete	complete	semi	semi
Tokens	yes	yes	yes	no	no
Evil number	51%	51%	51%	51%	33%
Performance	relatively high	high	relatively high	high	high
Technical maturity	mature	mature	not applied	not applied	mature

III. SYSTEM MODEL

A. Blockchain Hierarchical Architecture

In this paper, a three-layer architecture is proposed to handle trust and efficiency problems. As shown in Fig. 2, the architecture consists of physical network layer, Blockchain edge layer, and Blockchain network layer. The set of terminals is denoted by $V = \{V_1, V_2, \cdots, V_M\}$. Sets of cache nodes and resolution nodes are denoted as $B_c = \{B_{c,1}, B_{c,2}, \cdots, B_{c,N}\}$, $B_r = \{B_{r,1}, B_{r,2}, \cdots, B_{r,O}\}$, respectively. Functions of each layer are described as follows:

- **Physical Network Layer:** A large number of fixed and mobile terminals are used in IoT to realize monitoring and controlling. Taking intelligent transportation as an example, smart vehicles are equipped with many sensors that collect data and transfer to other layers. Hence, terminals raise high standards to real-time responses and edge security due to their mobility and poor access control mechanism among terminals and edge nodes.
- **Blockchain Edge Layer:** Blockchain edge nodes contain two kinds of nodes: resolution nodes and cache nodes. Resolution nodes are responsible for resolving domain name, verifying transaction and committing a block to the Blockchain network. Cache nodes are used to cache required contents for terminals. Through the proposed authentication system, these edge nodes can provide edge authentication service and synchronize authentication data timely to monitor terminals' activity as Blockchain network clients.
- **Blockchain Network Layer:** The Blockchain network provides decentralized services of storing terminal information and creating smart contracts over Hyperledger Fabric. Hyperledger Fabric is a customizable consortium blockchain platform that supports smart contracts called "chaincode". As a distributed ledger, hyperledger stores authentication logs orderly with the optimized PBFT algorithm. Each recorder in the ledger acts as a time constraint and a unique cryptographic signature, realizing activity traceability of terminals.

Brief working of the proposed system is described as Fig. 3. Smart contracts are created, defining authentication

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2938001, IEEE Transactions on Industrial Informatics

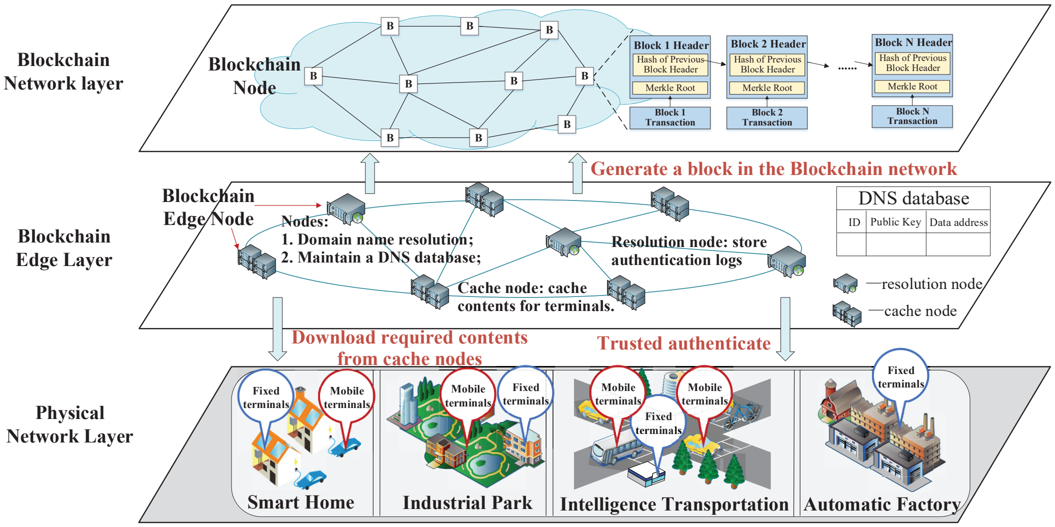

Fig. 2. System architecture

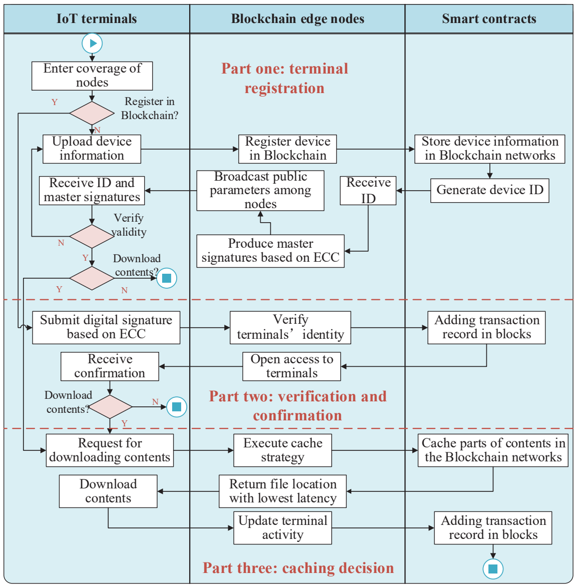
IoT terminals
Blockchain edge nodes
Smart contracts
Enter coverage of nodes
Register in Blockchain?
Y
N
Upload device information
Part one: terminal registration
Register device in Blockchain
Store device information in Blockchain networks
Receive ID and master signatures
Broadcast public parameters among nodes
Receive ID
Generate device ID
Verify validity
Produce master signatures based on ECC
Download contents?
Submit digital signature based on ECC
Verify terminals' identity
Adding transaction record in blocks
Receive confirmation
Open access to terminals
Part two: verification and confirmation
Download contents?
Request for downloading contents
Execute cache strategy
Cache parts of contents in the Blockchain networks
Download contents
Return file location with lowest latency
Update terminal activity
Adding transaction record in blocks
Part three: caching decision

Fig. 3. Authentication process

mechanism and integrating different protocols of IoT platforms [33]. By sending transaction to the address of smart contracts, resolution nodes can access to the smart contract and invoke its function. In terminal registration, terminals can be registered in the nearest cache nodes through the proposed name resolution strategy. Meanwhile, terminal identity anonymity and communication security can be guaranteed based on the cryptography algorithm.

Illegal and malicious terminals bring threats to data security and privacy confidentiality. In special scenarios like enterprise intranet, it is forbidden to open access to stranger terminals. Therefore, verification and confirmation on terminals are necessary. Through the authentication mechanism, Blockchain edge nodes can provide name resolution and edge authentication service, including verifying and confirming. At last, aiming at minimizing latency of downloading contents, a caching strategy is designed in Section V. To realize real-time monitoring and protect transaction transparency, authentication logs will be stored in the Blockchain network timely.

B. Optimized PBFT Consensus Algorithm

Considering that most alliance peers forming the consortium blockchain are trusted, an optimized PBFT algorithm is proposed, in which the primary peer is selected by round robin. The consensus algorithm is executed for storing authentication data and logs, to support data traceability and promote certification efficiency.

After receiving authentication results, alliance peers write authentication logs into the ledger through the optimized PBFT algorithm. It is assumed that there are N peers. For each round of consensus making, a peer will be selected as a speaker, while other peers play as congressmen. The speaker has no influence on consensus results, and it is allowed to host the consensus process for N times.

The speaker N_x is selected by $x = (h \bmod N) + 1$, where h is the current block height. Edge nodes can broadcast authentication results to alliance peers. Define t to present the time interval of generating a block. After t time interval, N_x broadcasts message: *pre_prepare* $< v, h, d, Sig_x >$, to all congressmen. v means view identity, d is the message digest and Sig_x is digest signature of the speaker.

Receiving *pre_prepare* messages from the speaker, a congressman, N_i, requires to verify the messages and signatures. If they are proved to be true, N_i broadcasts message: *prepare* $< v, h, d, Sig_i >$ among peers and continues to calculate messages received from other peers, where Sig_i denotes signatures of N_i.

If *prepare* messages from over $2f + 1$ different congressmen are received, N_i broadcasts messages: *commit* $< v, h, d, Sig_i >$ among peers. $f = \lfloor (N-1)/3 \rfloor$, which stands for the maximum number of malicious peers. When receiving over $f + 1$ *commit* messages, the speaker can confirm that a consensus is finished and generates a block in the ledger. The authentication logs are broadcast among peers and their ledgers are updated. Otherwise the block will be discarded and next round consensus will be executed.

Based on the optimized PBFT algorithm, a trusted and fault-tolerant Blockchain system is realized.

IV. Distributed Authentication Mechanism

A. Dynamic Name Resolution Strategy

A dynamic name resolution strategy is designed in this paper. As the foundation of Domain Name System (DNS), it provides edge authentication and data synchronization service. Both resolution nodes and cache nodes maintain a local DNS database which consists of terminal ID, public key and IP address.

Terminals will access Blockchain edge nodes through the same domain name. And the name resolver of resolution nodes translates domain name into corresponding IP address and delivers it to nearby cache nodes. Hence terminals can certificate in nearby nodes directly with lower delivery latency. Once the terminal's identity is confirmed by edge nodes, it can access to all resources among different platforms from one certification, achieving Single Sign on (SSO) and Identity and Access Management (IAM).

Edge nodes usually maintain a local DNS database for registered terminals to speed up verification. For newly coming terminals which have not registered, nodes will commit a block in the Blockchain network, gain a unique ID and corresponding public key as well. To avoid duplicate recordings, terminal information is transmitted to resolution nodes from cache nodes in one-way direction. Then cache nodes synchronize ID and accounts cached in resolution nodes at set intervals. Redundant logs will be stored in resolution nodes, in order to relieve storing pressure of the Blockchain network and enhance query efficiency.

B. Asymmetric Cryptographic Algorithm Based on ECC

To protect the identity anonymity and communication security of terminals and Blockchain edge nodes, an asymmetric cryptographic algorithm based on ECC is designed. The algorithm consists of four steps: setup, abstract, sign and verify.

1) Setup: Given a secure parameter k, an edge node selects two groups G_1 and G_2 with the same prime order q. Then it chooses a number $s \in Z_q^*$ as node private key and computes master public key $PK_{BE} = s \cdot P$. The pair of key is used to encrypt and decrypt messages, preventing malicious manipulation. After that two secure hash functions are selected: $H_1 : \{0,1\}^* \times G_1 \to Z_q^*$, $H_2 : \{0,1\}^* \times G_1 \times G_1 \to Z_q^*$.

2) Abstract: According to smart contracts, Blockchain assigns a unique ID id_j for V_j. According to the name resolution strategy, V_j can authenticate in the nearby cache node, $B_{c,n}$. $B_{c,n}$ will selects a random number $r_j \in Z_q^*$. Then formulations used to produce master signatures can be calculated:

$$R_j = r_j \cdot P, \tag{1}$$

$$h_j = H_1\left(id_j || R_j || PK_{BE}\right), \tag{2}$$

$$\delta_{j1} = \left(r_j + (h_j \cdot s) \bmod q\right)^{-1} \cdot P. \tag{3}$$

Public parameters $\{G_1, G_2, q, P, PK_{BE}, H_1, H_2, e(P,P)\}$ and cryptographic values in eq. (1)-(3) are broadcast among nodes, so that they can verify terminals' identity. With the asymmetric cryptography, Blockchain gurantees data integrity and security in the broadcasting.

Then V_j can obtain (R_j, δ_{j1}) from $B_{c,n}$ and checks whether $e(\delta_{j1}, R_j + h_j \cdot PK_{BE})$ equals to $e(P,P)$. When the equation is satisfied, V_j can confirm reliability of received messages and ensure that the 'block' recording information has been 'chained' in Blockchain networks. It is noted that these parameters should be kept confidentially as they play an important role in generating terminal signatures and verifying identity.

3) Sign: When V_j moves into the coverage of $B_{c,i}$, DDNS resolves its IP address and changes the certificating nodes. V_j need to get identified through delivering the digital signature. Selecting a random number $x_j \in Z_q^*$ as the private key, V_j calculates the correspondingly public key, $PK_j = x_j \cdot P$. Given a message m and public parameters, we have

$$X_j = H_2\left(id_j || PK_j || R_j || PK_{BE} || m\right), \tag{4}$$

$$\delta_{j2} = \left(X_j \cdot (r_j + h_j \cdot s \bmod q) + x_j\right)^{-1} \cdot P. \tag{5}$$

4) Verify: (X_j, δ_{j2}) will be uploaded to $B_{c,i}$ from V_j as signatures. Upon receiving data, $B_{c,i}$ verifies terminal identity via judging if $e\left(\delta_{j2}, X_j \cdot (R_j + h_j \cdot PK_{BE}) + PK_j\right) = e(P,P)$. If the equation is hold, $B_{c,i}$ will deliver the authentication results to the resolution node, which commits a block in the Blockchain network with consensus reaching according to the optimized PBFT algorithm. Examining equation can be deduced as below:

$$\begin{aligned}
&e\left(\delta_{j2}, X_j \cdot (R_j + h_j \cdot PK_{BE}) + PK_j\right) \\
&= e\Big(\left(X_j \cdot (r_j + h_j \cdot s) + x_j\right)^{-1} \cdot P, \\
&\quad \left(X_j \cdot (r_j + h_j \cdot s) + x_j\right) \cdot P\Big) \\
&= e(P,P)^{(X_j \cdot (r_j + h_j \cdot s) + x_j)^{-1} \cdot (X_j \cdot (r_j + h_j \cdot s) + x_j)} \\
&= e(P,P).
\end{aligned} \tag{6}$$

As demonstrated in [34], the decentralized nature of Blockchain can make distributed denial-of-service (DDoS) attacks against specific entities ineffective, since each alliance peer forming the consortium blockchain can verify data validity based on consensus algorithm. Furthermore, with the proposed cryptography defined in smart contracts, messages sent from terminals or edge nodes are encrypted. The inability of computing Nonce given ECC results makes it nearly impossible for attackers to compute correct digital signatures. Therefore, the system is trusted and DDoS-prevented.

C. Attack Model

There are several common attacks in the distributed and trusted authentication system employing the consortium Blockchain. And the following assumptions are used in the analysis of four possible attacks:

- Supported by governments, banks or large enterprises, peers in the consortium Blockchain are trusted and unforgeable.

- Alliance peers and edge nodes are curious about the real identity of smart terminals for gaining more profits.
- To improve response speed, smart terminals may intend to refuse authentication in the Blockchain edge nodes.
- Attackers enable to eavesdrop authentication results sent from terminals and tamper contents. They may access to the network through forging identity of terminals or edge nodes maliciously and destroy the system.

1) Denial of Service with False Signatures: The denial of service (DoS) with false signature is one of the most common attacks where data transferred from edge nodes to the Blockchain may be compromised by attackers. Specifically, attackers will eavesdrop authentication results transferred from edge nodes and sign them with their false signatures. After that attackers deliver the true authentication results with false signature to the consortium Blockchain. The polluted messages will be regarded as unreliable messages and be discarded by the alliance peers, even though the authentication results are valid.

2) Forgery Attack: On one hand, attackers can forge identity of terminals to access to the edge nodes, intending to obtain confidential contents or pollute authentication data. To produce a valid identity, a random number Err is selected and corresponding public key is computed $PK_{Err} = Err \cdot P$. Through computing or eavesdropping parameters used in following equations, forgery digital signatures can be calculated:

$$X_{Err} = H_2\left(id_j || PK_{Err} || R_j || PK_{BE} || m\right), \tag{7}$$

$$\delta_{Err} = \left(X_{Err} \cdot \left(r_j + h_j \cdot s \bmod q\right) + x_j\right)^{-1} \cdot P. \tag{8}$$

On the other hand, attackers can forge identity of edge nodes to steal or modify terminal information and distroy the authentication mechanism. To conduct forgery attacks, they need to eavesdrop messages delivered by the legal Blockchain edge nodes and compute ECC pairing mentioned above. Then massages may be tampered and delivered to the Blockchain or terminals with forgery signatures.

3) Man-in-the-Middle Attack: Man-in-the-middle attack is an attack that happens during message transmission between terminals and edge nodes, where attackers eavesdrop, intercept and manipulate information. Attackers may attempt to perform a man-in-the-middle attack using several approaches as below:

(1) Attackers block messages delivered from terminals to nodes, and then modify them before sending, acting as illegal terminals;

(2) Attackers block messages delivered from nodes to terminals, and then modify them, acting as malicious nodes;

4) Threats of privacy leakage: Attackers attempt to extract the real identity of terminals via eavesdropping multiple messages sent from the same terminals or edge nodes. If they make true, the system will face the threats of privacy leakage.

V. Caching Strategy Based on Belief Propagation Algorithm

A. Caching Model

It is assumed that the total number of contents that can be requested by terminals is Q. The contents are divided into H content groups (CG), which is denoted by $F = \{f_1, f_2, \ldots, f_H\}$. Without the loss of generality, all contents have the same size of s. Considering that some contents may be requested more frequently than others, they are modeled to have different popularities. Generally, the popularity of contents follows Zipf distribution [36]. The popularity that content q is requested can be modeled by:

$$e_q = \frac{1/q^{\tau}}{\sum_{a=1}^{Q} 1/a^{\tau}}, 0 < \tau < 1. \tag{9}$$

V_j can download contents from cache nodes or Blockchain networks. Assume that the dedicated bandwidth W_i is allocated from $B_{c,i}$ to terminals. For simplification, Blockchain network supplies a fixed download rate of R_0, which is lower than downloading rate supported by cache nodes. Define P_i as transmission power of $B_{c,i}$, and σ_j^2 denotes noise power of V_j. According to [35], path-loss between $B_{c,i}$ and V_j can be modeled as $d_{i,j}^{-\alpha}$, where $d_{i,j}$ is the distance between them and α is path-loss exponent. $f_{i,j}$ represents coefficient of Rayleigh fading between B_{c_i} and V_j. To eliminate interference among channels distributed from cache nodes to terminals, all the downlink channels are independent and identically distributed.

To describe relations between nodes and terminals, nodes and contents, terminals and contents, respectively, three matrices are defined correspondingly, denoted as $\mathbf{L}_{i,j} = [l_{i,j}]_{N \times M}$, $\mathbf{C}_{i,j} = [c_{i,j}]_{N \times Q}$, $\mathbf{R}_{\mathbf{i},\mathbf{j}} = [r_{i,j}]_{M \times Q}$, where

$$\begin{aligned} l_{i,j} &= \begin{cases} 1, & \text{if } V_j \text{ is within coverage of } B_{c,i}\ , \\ 0, & \text{otherwise,} \end{cases} \\ c_{i,j} &= \begin{cases} 1, & \text{if } B_{c,i} \text{ has cached } f_j, \\ 0, & \text{otherwise,} \end{cases} \\ r_{i,j} &= \begin{cases} 1, & \text{if } V_i \text{ requests for } f_j, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{10}$$

To evaluate efficiency of the caching strategy, hit ratio is used to present probability that edge node caches f_k that terminals need. Hit ratio of $B_{c,i}$ is:

$$hit_i = \frac{\sum\limits_j l_{i,j} \cdot \sum\limits_k c_{i,k} \cdot r_{j,k}}{\sum\limits_j l_{i,j}}. \tag{11}$$

B. Data Delivery Delay

When a terminal expects to download contents from higher layers, it faces three options and tolerates corresponding delivery delay. We give analysis of all options in next parts.

1) Delay to Connected Cache Nodes: V_j is within the coverage of $B_{c,i}$. If $B_{c,i}$ has cached f_k, V_j is able to download f_k directly from $B_{c,i}$. The transmission rate between V_j and $B_{c,i}$ can be calculated based on Signal to Interference Plus Noise ratio (SINR) [35],

$$R_{i,j} = W_i \log\left(1 + \frac{f_{i,j}^2 \cdot d_{i,j}^{-\alpha} \cdot P_i}{\sum\limits_{k \in V/i} f_{k,j}^2 \cdot d_{k,j}^{-\alpha} \cdot P_k + \sigma_j^2}\right). \tag{12}$$

Delivery delay from $B_{c,i}$ to V_j is:

$$del_{i,j} = \frac{s}{R_{i,j}}. \tag{13}$$

2) Delay to Other Cache Nodes: If f_k is not available in $B_{c,i}$, V_j can fetch contents from $B_{c,k}$ through $B_{c,i}$. Let $D_{i,k}$ and $bw_{i,k}$ denote the distance and average bandwidth between these nodes, respectively. Transmission delay from $B_{c,k}$ to $B_{c,i}$ can be calculated:

$$del_{B_i,B_k} = \frac{s}{bw_{i,k}} \cdot D_{i,k}. \tag{14}$$

Given weight factor $\beta_{i,j}$ related to network core congestion, transmission latency from other edge nodes to V_j is:

$$del_{i,j} = \beta \cdot \left(del_{i,j} + del_{B_i,B_k}\right). \tag{15}$$

3) Delay to the Blockchain Network: If f_k is not cached in any nodes, V_j has to submit requests to Blockchain networks. For simplification, the Blockchain network is regarded as $(N+1)_{th}$ node, denoted as B_{N+1}. Given weight factor γ, delay from Blockchain networks to V_j is:

$$del_{N+1,j} = \gamma \cdot \left(\frac{s}{R_0}\right). \tag{16}$$

Above all, we can calculate the transmission delay that V_j need to download contents from higher layer as follows:

$$Del_j = \sum_{i=1}^{N+1}\sum_{k=1}^{Q} r_{j,k} \cdot c_{i,k} \cdot l_{i,j} \cdot del_{i,j}. \tag{17}$$

To minimize the average delay of all terminals, the optimization problem model can be formulated as:

$$\begin{aligned} &\text{minimize} \quad Del\left(\mathbf{C}\right) = \tfrac{1}{M}\sum_{j} Del_j, \\ &s.t. \begin{cases} \sum\limits_{j} c_{i,j} \cdot \mathrm{s} \le Cap_i, \forall B_i \in \mathbf{B}, \forall V_j \in \mathbf{V} \\ \sum\limits_{i} l_{i,j} = 1 \\ \mathbf{L} \in \{0,1\}^{N \times M}, \mathbf{C} \in \{0,1\}^{N \times Q} \end{cases} \end{aligned} \tag{18}$$

where Cap_i is the cache capacity of $B_{c,i}$. Total caching size shouldn't exceed capacity of storage space of nodes. The problem is an integer programming problem, which is NP-hard. It can be optimized with Belief Propagation algorithm.

C. Algorithm Based on Belief Propagation Model

Standard BP model arrives at the scheduling decision by estimating marginal information of a certain joint probability distribution function [36]. Compared with other algorithm solving optimization problems, such like greedy maximal weight matching or CSMA-type method [37], [38], approximate BP algorithm has lower complexity and faster iteration speed. According to BP algorithm, a utility function is defined by $F\left(\mathbf{C}\right) = -Del\left(\mathbf{C}\right)$. And the optimization problem is:

$$\text{maximum} \quad F\left(\mathbf{C}\right). \tag{19}$$

Let $\mu > 0$ and the probability distribution is defined:

$$p\left(\mathbf{C}\right) = \frac{1}{Z}\exp\left(\mu F\left(C\right)\right), \forall B_{c,i} \in \mathbf{B},\ \forall V_j \in \mathbf{V}, \tag{20}$$

where Z is a partition function of μ and a normalization constant. According to large deviation, it's proved that if $\mu \to \infty$, $p\left(\mathbf{C}\right)$ concentrates to the maximum of $F\left(\mathbf{C}\right)$ with suitable condition. Let $E\left(\mathbf{C}\right)$ be the marginal expectation of $\mathbf{C}$. $E\left(\mathbf{C}\right)$ can be calculated once $p\left(\mathbf{C}\right)$ is given.

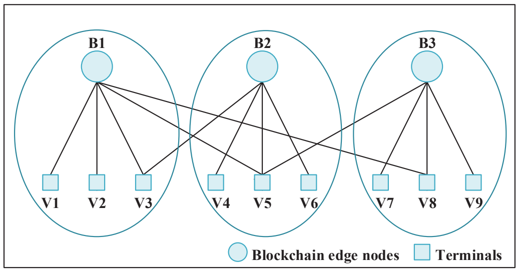

Fig. 4. Factor Graph

$$\lim_{\mu \to \infty} E\left(\mathbf{C}\right) = \arg\max_{\mathbf{C}} F\left(\mathbf{C}\right). \tag{21}$$

Therefore, BP algorithm provides us an optimized approach that we can recover a good approximate maximization of optimization problem from estimating marginal expectation of probability distribution.

To compute marginal distributions, a factor matrix $G = (V, E)$ is established in Fig. 4. Vertices V consists of N transmitter nodes TX associated with edge nodes, and M receiver nodes RX associated with terminals. An edge, $(i,j) \in E$, means that $B_{c,i}$ has influence on V_j. c_i^k implies that B_i has only stored f_k. BP algorithm iteratively passes beliefs which show estimates of marginal distributions along the edges of graph. We define $p_{i\to j}\left(t,\cdot\right)$ and $p_{j\to i}\left(t,\cdot\right)$ to represent belief messages passing from TX to RX and from reverse transmission direction in the t-th round. Therefore, $p_{i\to j}\left(t,\mathbf{c}_i\right)$ and $p_{j\to i}\left(t,\mathbf{c}_i\right)$ mean values of beliefs received in $B_{c,i}$. Procedures of applying BP algorithm for maximizing marginal distribution are as follows:

1) Initialization: Set $t = 0$ and let $p_{i\to j}\left(t,\mathbf{c}_i\right), \forall\left(i,j\right) \in E$ be initial distribution on $\mathbf{c_i}$. $p_{i\to j}\left(t,\mathbf{c}_i\right), \forall\left(i,j\right) \in E$ can be the popularity distribution of contents, E.

2) Mobile terminal nodes updating: In the $t-th$ iteration, V_j updates belief message $p_{j\to i}\left(t,\mathbf{c}_i\right)$ that be sent to $B_{c,i}$ according to beliefs received from other nodes except for $B_{c,i}$, denoted by $k \in \mathbf{B} \neq i$. With marginal distribution $\mathbf{c}_k \sim p_{k\to j}\left(t,\mathbf{c}_k\right), \forall k \in \mathbf{B} \neq i$, we update:

$$p_{j\to i}\left(t, c_{i,n}^*\right) = E\left\{\exp\left(\mu F_j\left(c_{i,n}^*, c_h, \forall h \in \mathbf{B} \neq i\right)\right)\right\}. \tag{22}$$

3) Blockchain edge node updating: In the $t-th$ iteration, $B_{c,i}$ updates belief message $p_{i\to j}\left(t+1,\mathbf{c}_i\right)$ to be sent to V_j, which is based on beliefs obtained from mobile terminals influenced by $B_{c,i}$ except for V_j, represented by $k \in B\left(i\right) \neq j$. Set $\frac{1}{Z_i}$ as normalization factor, and we have:

$$p_{i\to j}\left(t+1, c_{i,n}^*\right) = \frac{1}{Z_i}\prod_{k \in \mathbf{V} \neq j} p_{k\to i}\left(t, c_{i,n}^*\right). \tag{23}$$

4) Final decision: After T iterations, the algorithm is stopped and a selection of $\mathbf{c_i}$ is made by $B_{c,i}$. So the final estimation for marginal distribution of $\mathbf{c_i}$ is:

$$\Pr\left(\mathbf{c}_i\right) = \frac{1}{Z_i}\prod_{k \in B} p_{k\to i}\left(T,\mathbf{c}_i\right). \tag{24}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2938001, IEEE Transactions on Industrial Informatics

Based on final decision, a maximization of optimization problem can be estimated. Edge nodes can make caching decisions by selecting maximum posterior probability $Pr(\mathbf{c}_i)$. In this way, the caching matrix $\mathbf{C}_{i,j} = [c_{i,j}]_{N \times Q}$ can be determined and average delay can also be calculated. Therefore, an approximately optimization for minimum delivery delay is determined based on Belief Propagation algorithm.

VI. Security and Performance

A. Security Analysis

In this section, attack models defined in Section 3 are evaluated and analyzed correspondingly. The following preconditions are used in the security analysis:

- Attackers may be legal objects in the trusted authentication system, or illegal objects.
- For any x, y, if one of x or y is unknown, then $H(x||y)$ is unknown.
- Any private information, including secure parameters in cryptography, public parameters shared among edge nodes and ECC pairings, is unknown for attackers.
- Terminal information and digital signatures are unknown for attackers.

1) Prevention of Denial of Service with False Signature: According to the name resolution strategy, authentication results will be broadcast among Blockchain edge nodes. If attackers block authentication messages delivered from Blockchain edge nodes to the Blockchain and sign them with false signatures, alliance peers will detect polluted messages and discard them with the optimized PBFT algorithm. After that, the blocked messages will be resubmitted by a random resolution node in next rounds and the consensus algorithm can be executed for N times if failed. Therefore, DoS with false signature can only make true when massages from all edge nodes are blocked, which is difficult because of the widespread of edge nodes. Hence, the proposed authentication mechanism can prevent DoS with false signature.

2) Mutual Authentication: Prevention of terminal forgery: Based on ECC, it is computationally infeasible to calculate correct key R_j in eq. 7-8 for attackers even if eavesdropping transmission messages. Therefore, if illegal terminals submit requests with false signatures, their requests will be rejected by edge nodes since $e(\delta_{j2}, X_j \cdot (R_j + h_j \cdot PK_{BE}) + PK_j)$ and $e(P, P)$ are unequal in eq. 6.

Prevention of edge node forgery: To forge identity of nodes, attackers have to compute private keys and public parameters generated by ECC, which is also computationally infeasible as explained in terminal forgery.

To sum up, the distributed and trusted authentication system achieves mutual authentication and prevents forgery attacks.

3) Prevention of Man-in-the-Middle Attack: In man-in-the-middle attack, attackers must forge identity of edge nodes or terminals, while forgery attack is proved to be prevented as mentioned above. If they transfer messages without modifying, then the attack has no influence on the authentication system.

4) Conditional Privacy Preserving: On one hand, terminals communicate with edge nodes through digital signatures (X_j, δ_{j2}) generated by eq. 4-5, rather than the real identity id_j. The difficulty of computing id_j given (X_j, δ_{j2}) and P prevents attackers to extract real identity of terminals from multiple messages. On the other hand, privacy preserving is conditional since edge nodes need to record terminal information and synchronize transaction logs timely, in order to verify identity and trace data.

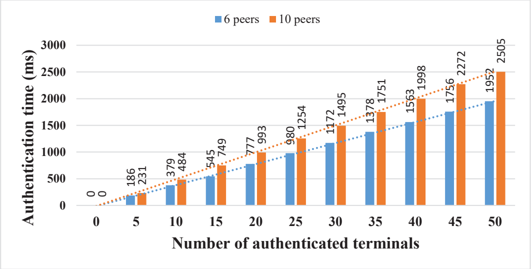

Fig. 5. Authentication time with different number of authentication terminals

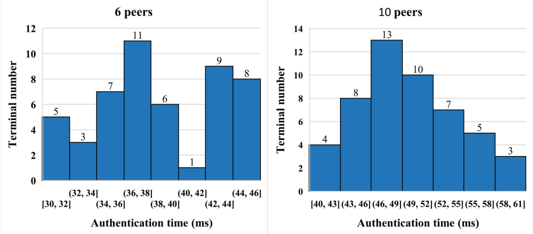

Fig. 6. Distribution of authentication time

B. Simulation Results and Analysis

The proposed system is evaluated with Matlab and Hyperledger Fabric.

1) Performance Evaluation of Trust Authentication: HyperLedger Fabric uses docker container technology to run Chaincode that contains the system application logic. The verification environment in this paper is carried out in Hyperledger Fabric version 1.4 of docker 18.06 container in ubuntu 16.04. Several nodes are virtually hosted on a single server machine, acting as alliance peers and reaching agreements with PBFT consensus algorithm. Each node is 2.0 GHz 8-vCPUS. VMs are interconnected through 1 Mbps virtual LAN cards. The proposed cryptography is used and typical orderer is deployed as a single ordering service.

Fig. 5 shows that authentication time increases with terminal amounts rising from 5 to 50, under different number of peers deployed. The reason is that as terminal amounts rises, their authentication requests wait longer to be handled by edge nodes. Fig. 6 shows distribution of authentication time with different peers. When peers number, N, is 6, the average latency is about 39 ms, and latency grows to 50 ms approximately when N is 10. According to PBFT consensus algorithm, more alliance peers in the network can improve security and fault-tolerant ability, at the cost of increasing authentication time of reaching consensus.

Communication cost: The communication cost is computed in the process of authentication. Initially, a terminal calculates the hash value which takes 80 bits (32 bits of identity, 32 bits

TABLE II
COMMUNICATION AND COMPUTATION COSTS FOR BLOCKCHAIN

Authentication Module	Cost[Units]
Random number generator (16 bits)	0.5 ms
Hash function (SHA-256 with input 160 bits)	3 ms
ECC pairing (176 bits)	10 ms
ECC point multiplication (160 bits)	4 ms
ECC point addition (160 bits)	2 ms
Node verifying	1 ms
PBFT consensus commitment	11 ms

TABLE III
SIMULATION PARAMETERS

Parameters[Symbols]	Value[Units]
Path loss exponent,α	4
Transmission power,P	6 Watt
Noise power,σ_j^2	10^{-10} Watt
Bandwidth, W	20 MHz
Size of each content,s	10 MB
Cache capacity of BE,Cap_i	10 GB
Zipf parameter, τ	0.6
Amounts of contents, Q	10000

of node identity, and 16 bit nonce value). Then the transaction is initialized through hash function which gives the output of 160 bits. And ECC pairing is produced in a size of 176 bits. At last, the final message digest for the terminal is computed using SHA-256 which outputs the digest of 256 bits. Therefore, the overall communication cost for a terminal to be authenticated is $256 + 176 + 16 + 160 = 608bits$, consisting of message digest, ECC pairing, time-stamp and transactions.

Computation cost: Random number generator, hash function and ECC pairing are executed once. ECC point multiplication and addition are conducted twice. When peer amount is 6, verifying and PBFT consensus commitment costs 1 ms and 15 ms, respectively. The overall computation cost for an authentication is about $0.5+3+10+4*2+2*2+1+11 = 37.5ms$.

2) Performance Evaluation of Caching strategy: Assume that Blockchain edge nodes distribute with intensity of λ_B, which represents the number of edge nodes per square kilometers. For simplification, all nodes have the same transmission power and channel fading coefficients. According to [39], parameters used in the simulation are set and list in Table. III. We study different scenarios where terminals submit requests for downloading contents with following methods:

- Popular caching [40]: According to the popularity distribution of files, k most popular contents should be cached to each edge nodes.
- Random caching [40]: The random caching strategy on edge nodes is that nodes ought to choose k contents randomly and cache into their capacity.
- BP caching: according to the proposed caching strategy based on Belief Propagation algorithm, edge nodes make cache decisions to improve allocation efficiency.

In simulation, experiments are conducted with different parameters. The proposed strategy is performed compared strategies based on popular caching and random caching in terms of average delivery delay and hit ratio.

Fig. 7 shows changes of average delay and hit ratio with different maximum tolerance of smart terminals, ranging from 100 to 500. Following rising of terminals, value of delay in BP caching increases gradually. Curves of popular caching and random caching fluctuate with amounts changing. Among them random caching strategy has the biggest latency. Additionally, with terminals getting denser, latency of popular caching is closer to that of BP caching. Reasons are that limited by caching capacity of edge nodes, strategy based on Belief Propagation algorithm tends to cache popular contents to meet demands of majority terminals. Therefore, BP caching and popular caching are converged to a similar result. In terms of hit ratio, hit ratio of BP caching is close to 1 in the beginning and decreases with amount rising. Values of random caching fluctuate at a low level. For popular caching, the hit ratio undulates in a small scope. And as terminals getting larger, it is converged to result of BP caching.

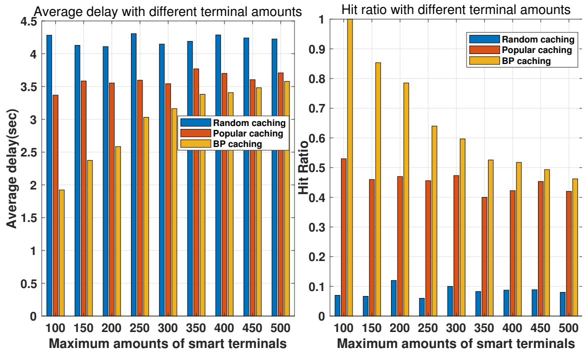

Fig. 7. Average delay and hit ratio with different amounts of terminals

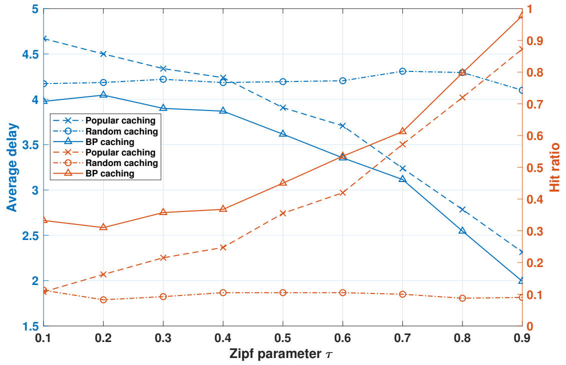

Fig. 8. Average delay and hit ratio with different Zipf parameters

Fig. 8 shows changes of average delay and hit ratio with different Zipf parameters, ranging from 0.1 to 0.9. The result demonstrates that the proposed algorithm can obtain the highest hit ratio compared with strategies based on popular caching and random caching. Besides, the hit ratio keeps increasing when the parameter of Zipf distribution increases. It is because that the percentage of requests for popular content goes up with Zipf parameters rising. Then replicas of the popular content in cache of edge nodes can satisfy more requests. And hit ratio of popular caching gets higher for the similar reasons.

On the contrary, average delay of the proposed strategy declines. Since a larger proportion of the popular contents arises, nodes are tended to cache popular contents. In this way, more terminals that request for the contents can download files

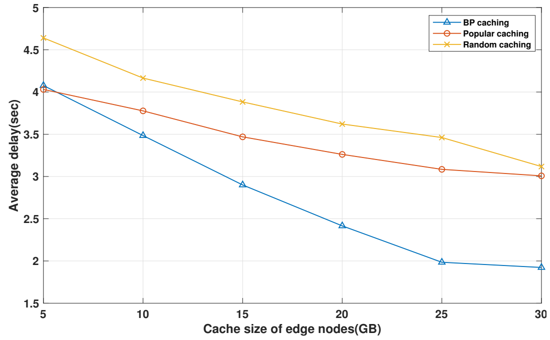

Fig. 9. Average delay with different cache capacity of nodes

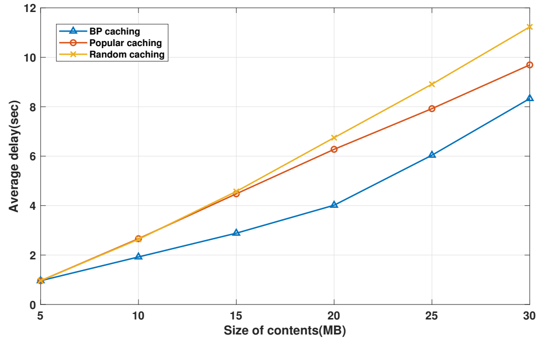

Fig. 10. Average delay with different size of contents

from connected nodes or nearby nodes with lower latency. In Fig. 9, we study the average delay of different algorithms with different cache capacity of nodes. When the cache size increases, a larger proportion of content delivered over the network can be stored. Therefore, average latency of all strategies declines with capacity growing. However, the BP caching strategy can still obtain the smallest delay compared with other strategies. The reason is that BP caching strategy can allocate cache capacity efficiently based on the analysis of content distribution among nodes and geological distribution of mobile terminals.

Fig. 10 describes that average delay increases as the size of contents grows. That is because terminals need to spend more time on downloading contents from nodes with size growing. Additionally, limited by cache capacity of edge nodes, a fewer proportion of content can be stored in nodes. In this way, average delay rises influenced by bigger contents. And value of BP caching strategy is the lowest among all strategies.

VII. Conclusion and Future Work

In this paper, a distributed and trusted authentication system combining edge computing and Blockchain is proposed to realize efficient authentication and information sharing among different IoT platforms. In the system, a hierarchical authentication architecture is established, consisting of physical network layer, Blockchain edge layer and Blockchain network layer. With the optimized PBFT consensus algorithm, the Blockchain stores authentication data and logs, guaranteeing trusted authentication and achieving activity traceability of terminals. To provide name resolution and edge authentication service, a distributed mechanism based on name resolution strategy and ECC is proposed. Evaluation on attack models proves that the mechanism is attack-prevented and fault-tolerant. Furthermore, we proposed a caching strategy based on BP algorithm which can realize cooperation among edge nodes and minimize downloading latency. In simulation, the authentication mechanism is evaluated in terms of communication and computation costs, demonstrating that the mechanism is applicable. Simulation results also prove that the proposed caching strategy has a higher hit ratio and lower delivery latency than other caching strategies based on popular caching and random caching.

In future work, we will apply this system to the blockchain based data sharing Platform for pilot verification, to further optimize performance and availability.

References

[1] LEVINE, D. “IIoT Challenges and Promises.” Machine Design, vol. 88, no. 7, pp. 20-23, July 2016.

[2] H. Cai, B. Xu, L. Jiang, at al.“IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges” in IEEE Internet of Things Journal, vol. 4, no. 1, pp. 75-87, Feb 2017

[3] Z. Zheng, S. Xie, H. Dai, et al. “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” 2017 IEEE International Congress on Big Data, Honolulu, HI, 2017, pp. 557-564.

[4] Nakamoto S. “Bitcoin: a peer-to-peer electronic cash system [Online],” available: https://bitcoin.org/bitcoin.pdf, 2009

[5] Z. Yang, K. Zheng, K. Yang, et al. “A blockchain-based reputation system for data credibility assessment in vehicular networks.” 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, 2017, pp. 1-5.

[6] A. Dorri, S. S. Kanhere, R. Jurdak, et al, “Blockchain for IoT security and privacy: The case study of a smart home,” 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, 2017, pp. 618-623.

[7] W. Shi, S. Hui, C. Jie, et al. “Edge Computing-An Emerging Computing Model for the Internet of Everything Era.” Journal of Computing Research and Development, vol. 54, no. 5, pp. 907-924, 2017.

[8] A. C. Baktir, A. Ozgovde and C. Ersoy. “How Can Edge Computing Benefit from Software-Defined Networking: A Survey, Use Cases and Future Directions.” IEEE Communications Surveys and Tutorials, vol. 19, no. 4, pp. 2359-2391, June 2017.

[9] P. K. Sharma, M. Chen and J. H. Park, “A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT,” in IEEE Access, vol. 6, pp. 115-124, 2018.

[10] M. Liu, F. R. Yu, Y. Teng, et al. “Computation Offloading and Content Caching in Wireless Blockchain Networks With Mobile Edge Computing,” in IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 11008-11021, Nov 2018.

[11] Chaudhary R, Jindal A, Aujla, et al. “BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system,” Computers and Security, vol. 85, pp. 288-299, August 2019.

[12] Jindal. A, Aujla. GS, Amar. N. “SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment,” Computer Networks, vol, 153, pp. 36-48, April 2019.

[13] F. Tang, S. Ma, Y. Xiang, et al. “An Efficient Authentication Scheme for Blockchain-Based Electronic Health Records,” IEEE Access, vol. 7, pp. 41678-41689, March 2019.

[14] S. Cha, J. Chen, C. Su, et al, “A Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things,” in IEEE Access, vol. 6, pp. 24639-24649, 2018.

[15] Y. Yao, X. Chang, J. Mii, et al. “BLA: Blockchain-Assisted Lightweight Anonymous Authentication for Distributed Vehicular Fog Services.” IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3775-3784, April 2019.

[16] S. Andreev, O. Galinina, A. Pyattaev, et al. “Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap[J].” IEEE Communications Magazine, vol. 53, no. 9, pp. 32-40, Sep 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2019.2938001, IEEE Transactions on Industrial Informatics

[17] P. Punithavathi, S. Geetha and S. Shanmugam. "Building cloud-based biometric services," 2017 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Bangalore, 2017, pp. 35-38..

[18] A. Dorri, S. S. Kanhere, R. Jurdak, et al. "Blockchain for IoT security and privacy: The case study of a smart home," 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, 2017, pp. 618-623.

[19] L. D. Xu, W. He and S. Li. "Internet of Things in Industries: A Survey[J]." IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, Nov 2014.

[20] W. Shen, L. Liu, X. Cao, et al. "Cooperative Message Authentication in Vehicular Cyber-Physical Systems [J]." IEEE Transactions on Emerging Topics in Computing, vol. 1, no. 1, pp. 84-97, June. 2018.

[21] H. Chien. "Group-Oriented Range-Bound Key Agreement for Internet of Things Scenarios [J]." IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1890-1903, June 2018.

[22] N. Li, D. Liu and S. Nepal. "Lightweight Mutual Authentication for IoT and Its Applications [J]." IEEE Transactions on Sustainable Computing, vol. 2, no. 4, pp. 359-370, Oct. 2017.

[23] M. Pustiek, A. Kos and U. Sedlar, "Blockchain Based Autonomous Selection of Electric Vehicle Charging Station," 2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI), Beijing, 2016, pp. 217-222.

[24] L. Li, J. Liu, L. Cheng, et al. "CreditCoin: A Privacy-Preserving Blockchain-Based Incentive Announcement Network for Communications of Smart Mobile terminals [J]." IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 7, pp. 1-17, July 2018.

[25] Khan S, Khan R. "Multiple Authorities Attribute-Based Verification Mechanism for Blockchain Mircogrid Transactions [J]." Energies, vol. 11, no. 5, 2018.

[26] C. Tselios, I. Politis and S. Kotsopoulos, "Enhancing SDN security for IoT-related deployments through blockchain," 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin, 2017, pp. 303-308.

[27] S. Ali, G. Wang, M. Z. A. Bhuiyan, et al. "Secure Data Provenance in Cloud-Centric Internet of Things via Blockchain Smart Contracts," 2018 IEEE SmartWorld, Guangzhou, 2018, pp. 991-998.

[28] H. Yao, T. Mai, J. Wang, et al. "Resource Trading in Blockchain-Based Industrial Internet of Things," IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3602-3609, June 2019.

[29] Z. Xiong, Y. Zhang, D. Niyato, et al. "When Mobile Blockchain Meets Edge Computing," IEEE Communications Magazine, vol. 56, no. 8, pp. 33-39, Aug 2018.

[30] Z. Chen, S. Chen, H. Xu, et al. "A Security Authentication Scheme of 5G Ultra-Dense Network Based on Block Chain," in IEEE Access, vol. 6, pp. 55372-55379, 2018.

[31] Y. Zhu, G. Zheng and K. Wong, "Blockchain-Empowered Decentralized Storage in Air-to-Ground Industrial Networks," in IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3593-3601, June 2019.

[32] W. Wang, N. Hu and X. Liu, "BlockCAM: A Blockchain-Based Cross-Domain Authentication Model," 2018 IEEE Third International Conference on Data Science in Cyberspace, Guangzhou, 2018, pp. 896-901.

[33] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," in IEEE Access, vol. 4, pp. 2292-2303, 2016.

[34] B. Rodrigues, L. Eisenring, E. Scheid, et al. "Evaluating a Blockchain-based Cooperative Defense," 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Arlington, VA, USA, 2019, pp. 533-538.

[35] H. S. Jo, Y. J. Sang, P. Xia, et al. "Heterogeneous cellular networks with flexible cell association: A comprehensive downlink SINR analysis," IEEE Trans. Wireless Communication, vol. 11, no. 10, pp. 3484-3495, Oct 2012.

[36] Rangan S, Madan R. "Belief Propagation Methods for Intercell Interference Coordination in Femtocell Networks[J]." IEEE Journal on Selected Areas in Communications, vol. 30, no. 3, pp. 631-640, April 2012.

[37] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," in Proc. 46th Ann. Allerton Conf. on Commun., Control and Comp., Monticello, IL, Oct 2008.

[38] C. Joo, X. Lin, and N. Shroff, "Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multihop Wireless Networks," IEEE/ACM Trans. Netw., vol. 17, no. 4, pp. 11321145, April 2009.

[39] J. Li, Y. Chen, Z. Lin, et al. "Distributed Caching for Data Dissemination in the Downlink of Heterogeneous Networks," in IEEE Transactions on Communications, vol. 63, no. 10, pp. 3553-3568, Oct 2015.

[40] Y. Hao, M. Chen, L. Hu, et al. "Energy Efficient Task Caching and Offloading for Mobile Edge Computing," in IEEE Access, vol. 6, pp. 11365-11373, 2018.

Shaoyong Guo received his B.S. degree in information and computing science from Hebei University in 2008, and the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications in 2013. He drafts an International standard as the first accomplisher and participates in three other International/industry standards. His research interests include blockchain application technology, mobile edge computing, and Internet of Things (IoT) in energy Internet. Dr. Guo was the recipient of two Provincial and Ministerial Second Prizes, one First Prize of the Power Innovation, and one Second Prize of Communication Society. Email: syguo@bupt.edu.cn.

Xing Hu received her B.S. degree in Communication Engineering from the Beijing University of Posts and Telecommunication, in 2014. And she is currently pursuing her MA.Sc. degree in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. Her main research interest lies in edge computing, Blockchain and Internet of Things (IoT) in energy Internet, including resource allocation and application of the Blockchain. Email: xinghu@bupt.edu.cn.

Song Guo received his PhD degree in computer science from University of Ottawa. He is currently a Full Professor with the Department of Computing, Hong Kong Polytechnic University. His current research interests include big data, cloud and edge computing, mobile computing, and distributed systems. Prof. Guo was a recipient of the 2019 TCBD Best Conference Paper Award, the 2018 IEEE TCGCC Best Magazine Paper Award, and six other Best Paper Awards from IEEE/ACM conferences. He was an IEEE Communications Society Distinguished Lecturer. He has served as an Associate Editor of IEEE TPDS, IEEE TCC, IEEE TETC, etc. He also served as the general and program chair for numerous IEEE conferences. He currently serves in the Board of Governors of the IEEE Communications Society. Email: song.guo@polyu.edu.hk

Xuesong Qiu received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2000. He is currently a Professor, a doctoral supervisor, and the deputy director of the State Key Laboratory of networking and switching technology, and vice president of Network Technology Research Institute of Beijing University of Posts and Telecommunications, China. He was editor of the two ITU-T standards and 4 industry standards of China. He has published more than 200 academic papers. He won twice the national scientific and technological progress prize of china. His major research interests include network and service management, information and communication technology of smart grid. Email: xsqiu@bupt.edu.cn.

Feng Qi is a Professor of Beijing University of Posts and Telecommunications, engaged in scientific research, teaching, and standardization research in information and communication. His research interests include communications software, network management, and business intelligence. He has won 2 National Science and Technology Progress Awards. He has also written more than 10 ITU-T international standards and Industry Standards. He is served as vice chairman of the ITU-T Study Group 4 and Study Group 12. Email: qifeng@bupt.edu.cn.