Intelligent and Converged Networks
2021, 2(2): 108–114

ISSN 2708-6240
DOI: 10.23919/ICN.2021.0009

Blockchain-enabled fog resource access and granting

Gang Liu, Jinsong Wu, and Ting Wang*

Abstract: Fog computing is a new computing paradigm for meeting ubiquitous massive access and latency-critical applications by moving the processing capability closer to end users. The geographical distribution/floating features with potential autonomy requirements introduce new challenges to the traditional methodology of network access control. In this paper, a blockchain-enabled fog resource access and granting solution is proposed to tackle the unique requirements brought by fog computing. The smart contract concept is introduced to enable dynamic, and automatic credential generation and delivery for an independent offer of fog resources. A per-transaction negotiation mechanism supports the fog resource provider to dynamically publish an offer and facilitates the choice of the preferred resource by the end user. Decentralized authentication and authorization relieve the processing pressure brought by massive access and single-point failure. Our solution can be extended and used in multi-access and especially multi-carrier scenarios in which centralized authorities are absent.

Key words: fog computing; edge computing; access control; blockchain; smart contract

1 Introduction

Fog computing, as shown in Fig. 1, is an extension of traditional cloud-based computing model, which selectively moves computing, storage, communication, control, and decision making closer to the network edge where data are being generated. Similar with edge computing concept, fog computing can relieve the limitations in current network infrastructure and better support mission-critical, data-dense use cases.

Fog computing is often erroneously called edge computing, but there are key differences. As its name suggests, fog is geographically distributed with uncertainty and instability, similar to the real fog that floats everywhere without a fixed shape. As for the edge concept, it always means relatively static or stable resources, which are typically deployed at certain places, e.g., a central office.

In a word, federating and floating are the key differences between the fog and edge nodes rather than physical location. In fog cases, the geo-distributed fog resources are federated as a ubiquitous resource pool. In addition, different fog nodes may work together to support collaborative tasks, e.g., Augmented Reality (AR) mobility, robot teamwork, and distributed storage. Services or applications can be unaware of any specific fog node that provides resources and where it is. To some extent, edge computing can also be considered as a type of fog computing.

2 Technical challenge

In contrast with traditional cloud computing, several

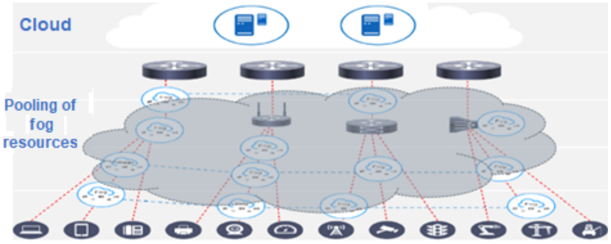


Fig. 1 Fog computing.

• Gang Liu is with the Nokia Shanghai Bell Co., Ltd., Shanghai 201206, China. E-mail: gang.i.liu@nokia-sbell.com.

• Jinsong Wu is with the School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin 541004, China. E-mail: wujs@ieee.org.

• Ting Wang is with the Shanghai Key Laboratory of Trustworthy Computing and the School of Software Engineering, East China Normal University, Shanghai 200062, China. E-mail: twang@sei.ecnu.edu.cn.

∗ To whom correspondence should be addressed.

Manuscript received: 2020-12-22; accepted: 2021-03-16

unique features bring quite different problems in fog computing scenarios. First, each fog node must authenticate the requestor and verify its right to access the fog resources. In traditional cloud computing, normally an access entrance is responsible for all authentication and authorization actions. This centralized access entrance performs a unified authentication and authorization procedure, which means that each subscriber usually uses a static password/key to access corresponding resources until they intend or are required to change the password/key. In addition, the price of a resource usually is static and unified no matter when the subscriber requests that resource unit. A similar phenomenon occurs in a telecom network: all resources for upper-layer applications are deployed behind an access gateway that is responsible for performing authentication/authorization/accounting functions. All the resources belong to the operator and a uniform access offer is provided.

In fog scenarios, fog resources are physically located on the network edge, which is beyond the traditional access gateway, e.g., broadband network gateway and mobile management entity entities. Basically, traffic must undergo a round trip through an access gateway, because the edge/fog node cannot perform access control. In addition, different fog nodes may belong to different vendors, and even an individual can share spare resources on a residential gateway or WLAN AP and make these entities work as fog nodes. These different fog providers may have to perform autonomous access control and independent charging actions. To some extent, fog providers must build their own access control infrastructure. On the other hand, fog node providers may decide their own price for a resource according to the cost, time, node status, and even personal preference. The subscribers have more choices on fog nodes/providers, and therefore may choose more cost-effective nodes to fulfill their requirements. During the entire fog resource access and granting procedure, several issues emerge.

2.1 Fog node identity and target node selection

How can a subscriber or application client be helped at performing authentication with target fog nodes, given that these fog nodes are geographically distributed? In cloud computing, we usually use one uniform and centralized authentication entrance for a subscriber to apply a cloud resource. The cloud subscribers normally do not care about where the cloud resource is. However, in fog computing, subscribers must choose the proper fog node to fulfill their requirements (computing, storage, latency, price, etc.). In fog computing, subscribers need to know which fog node they chose and its location. This means the subscriber or application client needs to access and submit requests to the targeted fog, and the fog node should perform an authentication procedure and then grant corresponding resources. The procedure is quite different from traditional cloud computing.

2.2 Dynamic credential for each transaction

In fog computing, one subscriber may submit resource requests to different fog nodes. This means the subscriber should submit different credential keys when requesting different fog nodes. While in a cloud computing scenario, a subscriber usually only maintains one credential for a cloud resource request. When many potential fog nodes are available, how can a flexible/reliable authentication methodology be designed for a subscriber to access a targeted fog node?

2.3 Dynamic independent offer provision

In fog computing, different fog nodes may provide different resource offers to requesters. For example, fog node A may offer its 1 GB storage space and charge 1 dollar per hour, while fog node B may charge more or less for the same resource offer. How to support independent fog resource granting is a unique problem in fog computing.

2.4 Authentication and resource offering

In typical fog scenarios, the resource access and granting may be performed with per-transaction granularity. When a subscriber intends to access specific resources from fog node A, he must trigger a new transaction and obtain a specific credential that is only valid during the duration indicated by the transaction profile. After the expiration time, the subscriber should trigger a new transaction process and submit new credentials for resource access. Each time, the fog node must verify different access credentials, even for requirements from the same subscriber. In this sense, the authentication/validation process and resource offering process must be integrated to guarantee per-transaction level fog resource access control and grant, instead of the single subscriber authentication procedure. How can the authentication and resource offering or granting be combined into one procedure?

3 Related work

As mentioned above, a cloud infrastructure provider uses one uniform authentication portal/entrance for all requests. The resource consumers usually do not (need not) know the specific location, e.g., which blade server eventually provides the computing or storage resources. Therefore, no authentication requirement between the resource requestor and the specific server provides the resources. In addition, concurrent requests from millions of subscribers bring great pressure to the central server, which suffers a potential risk of high processing latency and a single point of failure. To handle these issues, operators always must invest more to improve the capability of the central infrastructure.

Many researchers have investigated how to use blockchain technologies to tackle access control issues during specific scenarios.

Fernández-Caramés and Fraga-Lamas[1] gave a thorough review on how to adapt blockchain to the specific needs of the IoT to develop Blockchain-based IoT (BIoT) applications. After describing the basics of blockchain, the most relevant BIoT applications were described to emphasize how blockchain can affect traditional cloud-centered IoT applications. Current challenges and possible optimizations are also detailed regarding many aspects that affect the design, development, and deployment of BIoT applications. Pinno et al.[2] designed an architecture that includes a secure way of creating relationships, assigning attributes for them, and using them in the access control.

Smart-contract methodologies are naturally adopted by many researchers and used to publish IoT access offer as presented in Refs. [3–6]. In order to implement access permission propagation, Xu et al.[7] proposed the BlendCAC methodology. A robust, identity-based capability token management strategy was used to exploit a smart contract for registration, propagation, and revocation of the access authorization. In the proposed BlendCAC scheme, IoT devices are their own master to control their resources instead of being supervised by a centralized authority.

There are also some initiatives on leveraging blockchain to support distributed access or multi-access to different carrier networks, as described in Refs. [8–11]. Aiming to provide finer access control, Chen et al.[12] proposed a location-aware authentication scheme using smart contracts to ensure that IoT devices can securely perform Wi-Fi network authentication. The scheme adopts the concept of secondary authentication and consists of two phases: the registration phase, which is mainly designed to complete the generation of the public and private keys and to link the device information with its related device information; and the authentication phase, which is mainly designed to determine whether the requesting device is within a legal location range. The smart contract is used to ensure the credibility and irreparability of the authentication process.

Using the blockchain methodology to facilitate the transaction of digital assets, including electronic tickets, valuable data, and knowledge, is a straightforward idea. Lin et al.[13] proposed a Peer-to-Peer (P2P) knowledge market to make knowledge tradable in the edge-AI-enabled IoT. Moreover, they developed a knowledge consortium blockchain for secure and efficient knowledge management and trading for the market, which includes a new cryptographic currency knowledge coin, smart contracts, and a new consensus mechanism proof of trading. In addition, a noncooperative, game-based knowledge pricing strategy with incentives for the market was proposed.

From our perspective, fog resources, including computing and memory, are also a type of network resource and face various security threats. The geo-distributed, autonomy features of fog nodes bring the unique requirements of decentralized and independent access control, which means that a fog node may perform the access control and resource granting locally and independently, rather than using a conventional centralized authentication point with a unified price/policy as well.

4 Blockchain-enabled fog resource access

The existing access authentication and resource granting solutions in cloud computing are not adaptive to the unique requirements of geographically distributed fog nodes. For example, it cannot support peer-to-peer authentication between a requestor/subscriber and a provider/fog node. In addition, it lacks the capability of secure access with a traceable and irreversible record.

As a core component of the Bitcoin system[14], the blockchain maintains a distributed database that records all transaction data using blocks of special data structures and prevents tampering with historical data. Blockchain attracts considerable attention from academics and practitioners, such as those in finance, healthcare, and government, because of its unique attributes, such as distribution, security, anonymity, privacy, and tamper resistance.

The smart contract is the key concept of the blockchain 2.0 system. A smart contract is an executable code that runs on the blockchain to facilitate, execute, and enforce the terms of an agreement between untrusted parties. It can be considered a system that releases digital assets to all or some of the involved parties once the predefined rules have been met. Compared to traditional contracts, smart contracts do not rely on a trusted third party to operate, resulting in low transaction costs.

Addressing the unique challenges and requirements presented above, a smart-contract-based fog resource access and granting mechanism is proposed to enable decentralized authentication and independent resource offering and granting for each fog node. The entire procedure is shown in Fig. 2.

A smart contract is actually a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code. In our case, a typical smart contract consists of a fog resource offer with a segment of self-executing code that can generate and deliver a secret access key automatically when the contract is triggered. Every fog node submits its smart contract onto the distributed, decentralized blockchain network.

If a subscriber tries to apply the resources of a fog node, he can check the contract from the blockchain. When the subscriber fulfills the terms of the contract and triggers the execution of the contract (for example, enforcing a digital transaction), the self-executing code in the contract will generate a unique secret access key for that subscriber, encrypt it with the subscriber's public key, and then deliver it to the subscriber. All the transaction information, including the access profile, will be recorded on the blockchain, which is traceable and irreversible.

The subscriber decrypts the secret access key and uses it to access the targeted fog node resource. The access profile indicates the number of leased resources, their duration, and their scale. The resource entrance of the target fog node (e.g., the IP address of the VM instance created and reserved for tenants/subscribers) may also be delivered to the subscriber, along with the access key.

When a fog node is receiving requests with a secret access key, it will check and verify the key, as well as its

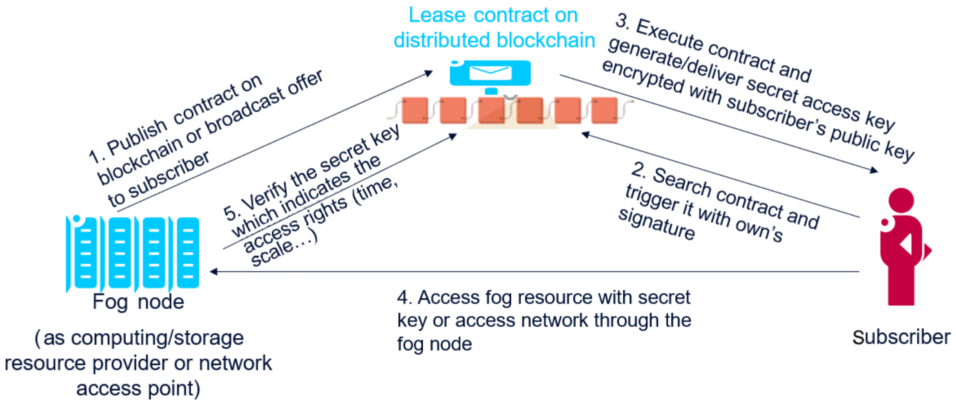

Lease contract on
distributed blockchain
1. Publish contract on blockchain or broadcast offer to subscriber
3. Execute contract and generate/deliver secret access key encrypted with subscriber's public key
5. Verify the secret key which indicates the access rights (time, scale…)
2. Search contract and trigger it with own's signature
4. Access fog resource with secret key or access network through the fog node
Fog node
(as computing/storage resource provider or network access point)
Subscriber

Fig. 2 Working procedure of fog resource access.

access profile locally or on the blockchain. The access authentication and resource granting procedure can be implemented simultaneously.

Alternatively, as shown in Fig. 3, when the subscriber cannot perform a blockchain relative operation and interact with the target fog node, an agent may be deployed on the network access point devices (typically a residential gateway, wireless access point, or base station) to represent the subscriber and interact with the blockchain system and fog nodes. In this case, the access network can help the subscriber to request fog resources, and the subscriber may be unaware of any specific fog node that provides the resources, as well as its location.

5 Smart contract design and implementation

The smart contract concept is introduced to enable dynamic, automatic credential generation and delivery for independent fog resource access and granting. In our design, a smart contract is a series of software codes intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract on fog resource access and granting. A private blockchain was built up with several Go-Ethereum clients[15], which serve as miner nodes and transform the devices into an Ethereum node[16]. When a fog node is willing to share its resources, it can submit an independent lease contract on the Ethereum-based blockchain.

The lease contract is a segment of executable code, and it defines how the contract is to be performed. In Fig. 4, a simple storage lease contract is given as an example. In this contract, when a caller sends a certain number of digital coins to the provider, which means the terms of the agreement are fulfilled, corresponding events are triggered automatically, and a unique secret key is generated based on the identity/address of the caller. Then, the secret key is encrypted with the caller's public key to enable secure delivery, as shown in Step 3 of Fig. 2.

The lease contract is posted on the blockchain with the signature of the resource provider (fog node in this case). Each step of the contract performance must be confirmed with the executor's signature, which can guarantee the non-repudiation. As with the classic blockchain concept, all the contract records are added and stored in the linear linked block.

In addition, the entire blockchain is maintained by thousands/millions of miners, and any miner can verify each transaction or contract execution using the corresponding counterparty's public key. Therefore, all the transactions and contracts are traceable and irreversible.

Each resource provider can have its own contract, which may be different from those of other fog nodes. Unlike common cloud policy, each fog node can independently provide its own resource offer on the blockchain.

During the execution of the smart contract, a unique secret key for the subscriber is automatically generated, and it can only be decrypted by the subscriber. Compared with the traditional cloud solution in which usually only one key is given to a subscriber, a smart-contract-based solution supports each subscriber to use different secret keys when accessing different fog nodes. The entire procedure is performed reliably without the need of a trusted authority or central server.

The smart contract is maintained on the blockchain by all the miner nodes, and the subscriber can invoke or trigger the performance of a contract by external interfaces, e.g., Web API and Wallet client. As shown in Fig. 5, one miner node can submit the update request of

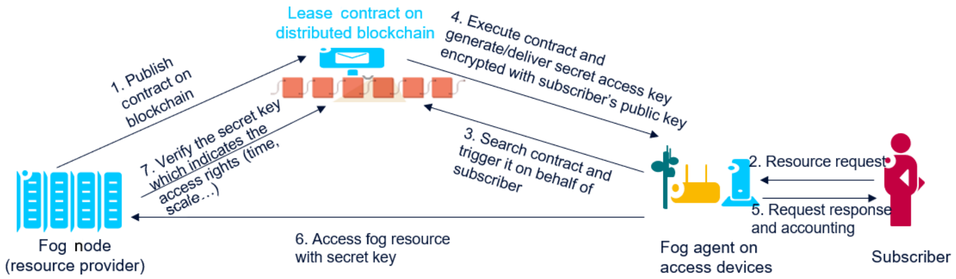
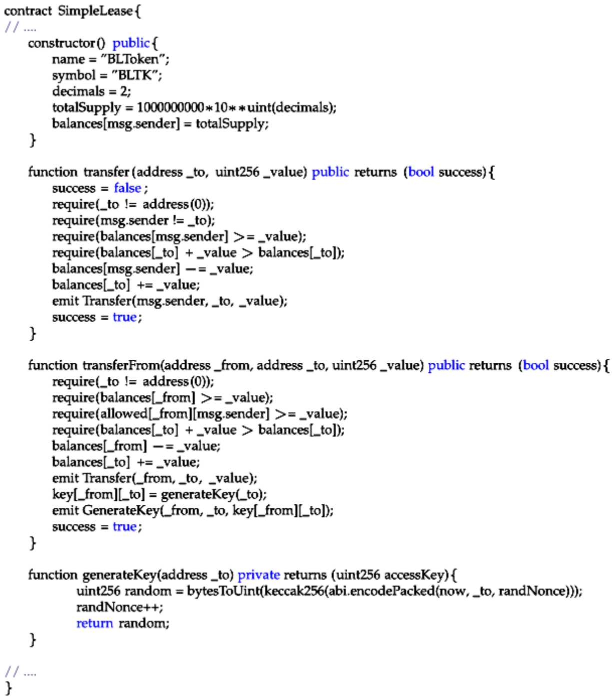
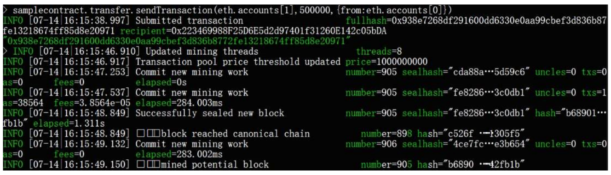

Lease contract on
distributed blockchain
1. Publish contract on blockchain
4. Execute contract and generate/deliver secret access key encrypted with subscriber's public key
7. Verify the secret key which indicates the access rights (time, scale…)
3. Search contract and trigger it on behalf of subscriber
2. Resource request
5. Request response and accounting
6. Access fog resource with secret key
Fog node
(resource provider)
Fog agent on
access devices
Subscriber

Fig. 3 Fog agent deployed on access devices.



```
contract SimpleLease{
// ....
    constructor() public{
        name = "BLToken";
        symbol = "BLTK";
        decimals = 2;
        totalSupply = 1000000000*10**uint(decimals);
        balances[msg.sender] = totalSupply;
    }

    function transfer(address _to, uint256 _value) public returns (bool success){
        success = false;
        require(_to != address(0));
        require(msg.sender != _to);
        require(balances[msg.sender] >= _value);
        require(balances[_to] + _value > balances[_to]);
        balances[msg.sender] -= _value;
        balances[_to] += _value;
        emit Transfer(msg.sender, _to, _value);
        success = true;
    }

    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success){
        require(_to != address(0));
        require(balances[_from] >= _value);
        require(allowed[_from][msg.sender] >= _value);
        require(balances[_to] + _value > balances[_to]);
        balances[_from] -= _value;
        balances[_to] += _value;
        emit Transfer(_from, _to, _value);
        key[_from][_to] = generateKey(_to);
        emit GenerateKey(_from, _to, key[_from][_to]);
        success = true;
    }

    function generateKey(address _to) private returns (uint256 accessKey){
            uint256 random = bytesToUint(keccak256(abi.encodePacked(now, _to, randNonce)));
            randNonce++;
            return random;
    }

// ....
}
```

Fig. 4 Simple storage lease contract.



```
> samplecontract.transfer.sendTransaction(eth.accounts[1],500000, {from:eth.accounts[0]})
INFO [07-14|16:15:38.997] Submitted transaction                    fullhash=0x938e7268df291600dd6330e0aa99cbef3d836b87
fe13218674ff85d8e20971 recipient=0x223469988F25D6E5d2d97401f31260E142c05bDA
"0x938e7268df291600dd6330e0aa99cbef3d836b8772fe13218674ff85d8e20971"
> INFO [07-14|16:15:46.910] Updated mining threads                   threads=8
INFO [07-14|16:15:46.917] Transaction pool price threshold updated price=1000000000
INFO [07-14|16:15:47.253] Commit new mining work                   number=905 sealhash="cda88a…5d59c6" uncles=0 txs=0
as=0       fees=0            elapsed=0s
INFO [07-14|16:15:47.537] Commit new mining work                   number=905 sealhash="fe8286…3c0db1" uncles=0 txs=1
as=38564   fees=3.8564e-05   elapsed=284.003ms
INFO [07-14|16:15:48.849] Successfully sealed new block            number=905 sealhash="fe8286…3c0db1" hash="b68901…
fb1b" elapsed=1.311s
INFO [07-14|16:15:48.849] □□□block reached canonical chain            number=898 hash="c526f …1305f5"
INFO [07-14|16:15:49.132] Commit new mining work                   number=906 sealhash="4ce7fc…e3b654" uncles=0 txs=0
as=0       fees=0            elapsed=283.002ms
INFO [07-14|16:15:49.150] □□□mined potential block                    number=905 hash="b6890 …42fb1b"
```

Fig. 5 Transaction submitted and committed.

the specific contract, and then the transaction is executed and committed by the entire blockchain. Similarly, when a fog node wants to verify a secret key from a specific subscriber, it can request any miner node of the blockchain. In contrast with the traditional centralized solution, our decentralized solution relieves the issues brought by a single point of failure, e.g., massive concurrent requests, high processing latency, data monopoly, and privacy violation.

6 Conclusion and future work

Fog computing plays a crucial role in satisfying the requirements of delay-sensitive applications, such as VR, AR, and industrial production lines. Fog nodes are also a type of network resource, but they have unique characteristics: geo-distributed, autonomous, and independent offerings. Traditional network control and resource granting methodologies usually assume that the relevant resources are placed behind a certain authentication/accounting point, which may not be suitable for fog computing cases. Furthermore, in some cases, fog nodes may belong to different owners or individuals who would sell the resources at a very different price and may dynamically adjust the resource offer. Addressing these new requirements, we propose a smart-contract-based fog resource access and granting mechanism to enable decentralized authentication and independent resource offering/granting for each fog node.

In the next stage, our solution will be extended to network access, especially multi-carriers and multi-access scenarios, to support independent network selection and access across different carriers/operators without any centralized authority.

References

[1] T. M. Fernández-Caramés and P. Fraga-Lamas, A review on the use of blockchain for the internet of things, *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[2] O. J. A. Pinno, A. R. A. Gregio, and L. C. E. De Bona, ControlChain: Blockchain as a central enabler for access control authorizations in the IoT, in *Proc. 2017 IEEE Global Communications Conf.*, Singapore, 2017, pp. 1–6.

[3] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, FairAccess: A new blockchain-based access control framework for the Internet of Things, *Security and Communication Networks*, vol. 9, pp. 5943–5964, 2016.

[4] P. Wang, Y. L. Yue, W. Sun, and J. J. Liu, An attribute-based distributed access control for blockchain-enabled IoT, in *Proc. 2019 Int. Conf. Wireless and Mobile Computing, Networking and Communications* (*WiMob*), Barcelona, Spain, 2019, pp. 1–6.

[5] C. Dukkipati, Y. P. Zhang, and L. C. Cheng, Decentralized, Blockchain based access control framework for the heterogeneous Internet of Things, in *Proc. 3^{rd} ACM Workshop on Attribute-Based Access Control*, Tempe, AZ, USA, 2018, pp. 61–69.

[6] Y. Y. Zhang, S. Kasahara, Y. L. Shen, X. H. Jiang, and J. X. Wan, Smart contract-based access control for the Internet of Things, *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.

[7] R. H. Xu, Y. Chen, E. Blasch, and G. S. Chen, BlendCAC: A Blockchain-enabled decentralized capability-based access control for IoTs, in *Proc. 2018 IEEE Int. Conf. Internet of Things* (*iThings*) *and IEEE Green Computing and Communications* (*GreenCom*) *and IEEE Cyber, Physical and Social Computing* (*CPSCom*) *and IEEE Smart Data* (*SmartData*), Halifax, Canada, 2018, pp. 1027–1034.

[8] D. Di, F. Maesa, P. Mori, and L. Ricci, Blockchain based access control, in *Proc. Distributed Applications and Interoperable Systems*, Neuchatel, Switzerland, 2017, pp. 206–220.

[9] C. Lin, D. B. He, X. Y. Huang, K. K. R. Choo, and A. V. Vasilakos, BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0, *Journal of Network and Computer Applications*, vol. 116, pp. 42–52, 2018.

[10] T. Sanda and H. Inaba, Proposal of new authentication method in Wi-Fi access using Bitcoin 2.0, in *Proc. IEEE 5th Global Conf. Consumer Electronics*, Kyoto, Japan, 2016, pp. 1–5.

[11] X. Jiang, M. Z. Liu, C. Yang, Y. H. Liu, and R. L. Wang, A blockchain-based authentication protocol for WLAN mesh security access, *Computers, Materials and Continua*, vol.58, no. 1, pp. 45–59, 2019.

[12] Y. L. Chen, X. J. Wang, Y. L. Yang, and H. Li, Location-aware Wi-Fi authentication scheme using smart contract, *Sensors*, vol. 20, no. 4, p. 1062, 2020.

[13] X. Lin, J. H. Li, J. Wu, H. R. Liang, and W. Yang, Making knowledge tradable in edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6367–6378, 2019.

[14] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf, 2008.

[15] Geth client for building private blockchain networks, https://github.com/ethereum/go-ethereum, 2021.

[16] V. Buterin, A next-generation smart contract and decentralized application platform, https://ethereum.org/en/whitepaper/, 2021.

Gang Liu received the BEng and PhD degrees in computer science from Northwestern Polytechnical University, China in 2002 and 2008, respectively. Currently, he works as a research scientist in Nokia Shanghai Bell Co., Ltd. His research interests include fixed and mobile converged network, SDN/NFV, edge/cloud computing, and AI-assisted cellular network.

Jinsong Wu received the PhD degree from Queen’s University, Canada in 2006. He is now a professor at School of Artificial Intelligence, Guilin University of Electronic Technology. He was the founder (2011) and founding chair (2011–2017) of IEEE Technical Committee on Green Communications and Computing (TCGCC). He is also the co-founder (2014) and founding vice-chair (2015–present) of IEEE Technical Committee on Big Data (TCBD). He received both 2017 and 2019 IEEE System Journal Best Paper Awards. His co-authored paper won 2018 IEEE TCGCC Best Magazine Paper Award. He received IEEE Green Communications and Computing Technical Committee 2017 Excellent Services Award for Excellent Technical Leadership and Services in the Green Communications and Computing Community. He is the leading editor and co-author of the comprehensive book, entitled *Green Communications*: *Theoretical Fundamentals*, *Algorithms*, *and Applications*, published by CRC Press in September 2012. He has been an IEEE senior member since 2011. His research interests include green communication and computer network, big data, machine learning, IoT, and edge computing.

Ting Wang received the BS degree from University of Science and Technology Beijing, China in 2008, the MEng degree from Warsaw University of Technology, Poland in 2011, and the PhD degree in computer science and engineering from Hong Kong University of Science and Technology, China in 2015. He is currently an associate professor at Shanghai Key Laboratory of Trustworthy Computing and School of Software Engineering, East China Normal University (ECNU). Prior to joining ECNU in 2020, he worked at the Bell Labs as a research scientist from 2015 to 2016, and at Huawei as a senior engineer from 2016 to 2020. He is currently an associate editor of *IEEE Access*, the editor-in-chief of *IITCIB*, and a technical committee member of *Computer Communications*, Elsevier. His research interests include software defined networking, data center networking, AI-driven intelligent networking, cloud computing, edge computing, and network function virtualization.