Peer-to-Peer Networking and Applications (2023) 16:727–748
https://doi.org/10.1007/s12083-022-01437-x

Blockchain-based differentiated authentication mechanism for 6G heterogeneous networks

Zhe Tu[1,2] · Huachun Zhou[1,2] · Kun Li[1,2] · Haoxiang Song[1,2] · Wei Quan[1,2]

Received: 26 May 2022 / Accepted: 9 December 2022 / Published online: 9 January 2023

Abstract
It is well known that the Sixth Generation (6G) communication system integrating multiple access networks promotes the internet of everything world-widely. However, due to the differentiated underlying network protocols, it is difficult to find a general authentication solution to support various authentication methods in different access networks. Blockchain is a new technology that supports network heterogeneity, which provides a potential solution for differentiated authentication. In this paper, we propose a blockchain-based differentiated authentication mechanism for 6G Heterogeneous Networks (HetNets), which can efficiently authenticate user identities through scheduling different authentication methods. Particularly, we analyze the authentication architecture of 6G HetNets and put forward a blockchain-based differentiated authentication framework. Besides, to improve the scalability of user authentication, it is the first time to use various blockchain authentication contracts to represent different authentication methods. Meanwhile, a differentiated authentication management contract is proposed to uniformly manage different authentication contracts to realize differentiated identity authentication. Based on the evaluation of the prototype system, the proposed mechanism can dynamically provide differentiated authentication services (e.g. EAP-MD5, 5G-AKA) with low additional time (milliseconds levels) cost.

Keywords 6G · Blockchain · Heterogeneous networks (HetNets) · Identity authentication

1 Introduction

With the advancement of mobile communication, 6G networks with intelligent communication and ubiquitous interconnection have gradually become popular [1, 2]. The 6G network integrates multiple access networks in multiple spatial dimensions (ocean, land, air, and space), and has ultra-high heterogeneity [3, 4]. The authentication method can identify user identities and block the access of malicious users, which is an important method to improve network security [5–7]. Compared with the traditional network architecture, the 6G Heterogeneous Networks (HetNets) puts forward new security requirements for the authentication methods, which are as follows:

(i) **Reliability**. The deep integration of various access networks has spawned a variety of application scenarios. The 6G authentication methods need to be able to provide secure and reliable authentication services for different scenarios.
(ii) **Scalability**. In 6G HetNets, there are differences in the service capabilities of different access networks. Authentication methods need to be scalable and universal to be deployed in different networks.
(iii) **Efficiency**. The 6G HetNets has further promoted the interconnection of everything, and the number of network users and devices has increased dramatically. How to achieve fast and efficient authentication

✉ Huachun Zhou
hchzhou@bjtu.edu.cn

Zhe Tu
zhe_tu@bjtu.edu.cn

Kun Li
kun_li@bjtu.edu.cn

Haoxiang Song
20120099@bjtu.edu.cn

Wei Quan
weiquan@bjtu.edu.cn

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

[2] National Engineering Research Center of Advanced Network Technologies, Beijing 100044, China

in the large-scale interconnected 6G network is also a key problem that needs to be solved urgently.

The trusted data sharing mechanism constructed by Blockchain effectively solves the problem of the lack of mutual trust among HetNets nodes and provides a new evolution direction for 6G HetNets [8–13]. In 6G HetNets, the blockchain-based authentication method can well meet the new security requirements brought by network heterogeneity. In terms of reliability, the anonymity and non-tampering characteristics of blockchain can effectively prevent the leakage and tampering of authentication data, and ensure the security and reliability of identity authentication [14]; In terms of scalability, the decentralized characteristics of blockchain can alleviate the impact of network heterogeneity, which is conducive to the construction of a scalable and extensible authentication method [15]; In terms of efficiency, the trusted data sharing mechanism promotes the exchange of massive authentication data, reduces the consumption of cross-domain authentication signaling, and improves the efficiency of cross-domain authentication [16].

However, the existing blockchain-based authentication methods cannot be directly applied to 6G HetNets for the following two reasons. On the one hand, considering the diversity of authentication methods for 6G HetNets, the existing methods lack unified management of various authentication methods. On the other hand, the requirements of 6G HetNets users are diverse, and the existing methods are difficult to provide differentiated authentication services for different authentication requirements of users. In this paper, the differentiated authentication services refers to providing different authentication protocols for users with different authentication requirements (such as authentication methods, security levels, response speed, etc.). Therefore, in 6G HetNets, there is an urgent need for an authentication mechanism that can meet different authentication requirements and manage authentication methods in a unified manner.

In this paper, a Blockchain-based Differentiated Authentication Mechanism (BDAM) for 6G HetNets is proposed. The proposed authentication mechanism stores various authentication methods in the form of smart contracts, which realizes the unified management of HetNets authentication methods. In addition, the proposed mechanism can analyze user authentication requests, and provide differentiated authentication services for 6G HetNets users.

The contributions of this paper are summarized as follows:

- We propose a Blockchain-based Differentiated Authentication Framework (BDAF) for 6G HetNets. This framework stores the authentication methods on the blockchain in the form of authentication contracts, giving flexibility and scalability to the deployment of HetNets authentication methods.
- We put forward a BDAM mechanism based on BDAF, implementing an integrated description of different blockchain-based authentication processes in HetNets. In the proposed BDAM, we design two special smart contracts: Differentiated Authentication Management Contract (DAMC) and Identity Authentication Record Contract (IARC). DAMC stores authentication method information to realize unified management and differentiated authentication; IARC records historical authentication behaviors, reduces the signaling consumption of cross-domain authentication and improves the efficiency of identity authentication.
- We evaluate BDAM in a prototype system through two common authentication methods (EAP-MD5, 5G-AKA). The evaluation result shows that the proposed BDAM can realize the dynamic deployment of authentication methods and provide differentiated authentication services with low additional time (10ms level) cost.

The remainder of this paper is organized as follows. In Section 2, we summarize the existing research on HetNets authentication. In Section 3, we introduce the proposed BDAF and Authentication Contract System (ACS), respectively. In Section 4, we design BDAM based on the BDAF. Then, in Section 5, we present the evaluation setting of the proposed BDAM prototype system. In Section 6, we evaluate the performance of the proposed BDAM. In Section 7, we analyze the security of the BDAM. Finally, in Section 8, we conclude the paper.

2 Related works

With the deep integration of satellite networks, mobile networks, and other access networks, 6G HetNets further realize the coordination of various communication resources, thus meeting the requirements of intelligent connection and ubiquitous interconnection.

2.1 HetNets authentication methods

For different HetNets scenarios, there are many kinds of research on HetNets authentication. Xiong et al. [17] proposed an authentication protocol for identify (ID)-based system and certificateless (CL)-based system in heterogeneous Industrial Internet of Things (IIoT). The proposed protocol can meet the needs for privacy protection between sensors in ID-based systems and users in CL-based systems. In the heterogeneous beyond 5G network, Cui et al. [18] proposed an authentication framework supporting edge computing, which is different from the existing 5G authentication standards. They divide the procedures into three stages: the offline register stage, the primary authentication

stage, and the transparent authentication stage. The unified authentication of Identity-Based Cryptography (IBC) is realized among the access nodes. Cao et al. [19] proposed an authentication protocol IEAP-AKA based on a hybrid cryptography system and certificate-free signature encryption in LTE-WLAN HetNets, and verified that the proposed method has higher security compared with the original EAP-AKA protocol. Liu et al. [20] proposed authentication and key agreement protocol applied to Public Key Infrastructure (PKI) and CertificateLess Cryptography (CLC) environments, which implements secure communication between legally authorized users in heterogeneous IoT scenarios. In 4G/5G HetNets, Alezabi et al. [21] used the standard AKA protocol to improve the authentication protocols of LET, WLAN, and WiMAX and proposed a re-authentication mechanism in HetNets for vertical handover and horizontal handover scenarios respectively. In heterogeneous wireless sensor networks, Athmani et al. [22] proposed authentication and key division scheme EDAK based on the dynamic key matrix. This method realizes lightweight authentication and key distribution and optimizes the memory consumption of sensor nodes in the authentication process, which effectively reduces the overhead of nodes.

The above authentication methods focus on HetNets with one or a limited number of access networks, and can not be applied in 6G HetNets with multiple access networks, nor can they achieve unified management of different authentication methods. In addition, most of the proposed authentication methods are centralized deployed, which can not effectively prevent a single point of failure, and it is difficult to quickly respond to cross-domain user authentication.

2.2 Blockchain-based authentication methods

In recent years, with the development of the blockchain, more and more scholars tend to use blockchain to construct HetNets authentication methods. The distributed and decentralized characteristics of blockchain can well solve the single problem of authentication centers in traditional HetNets. At the same time, the blockchain-based authentication method realizes rapid cross-domain authentication through the global sharing of authentication information.

In the key-based user identity authentication system, user authentication and security key management are inseparable. Nowadays, many researchers have put forward a variety of HetNets security key management methods based on blockchain. In the heterogeneous vehicle network, Lei et al. [23] designed a security key management framework based on blockchain and proposed a key transmission method between security managers, realizing dynamic updates of vehicle keys in different security domains. In heterogeneous Flying Ad-Hoc Network (FANET), Tan et al. [24] proposed a distributed key management scheme based on blockchain, which realized the distribution and update of keys among UAV clusters and improved the communication security of UAV clusters in different scenarios.

Currently, there are many blockchain-based authentication methods for HetNets. In the heterogeneous IoT, Zhang et al. [25] designed a hybrid blockchain model consisting of a global blockchain and a local blockchain, and proposed a method for mutual authentication between nodes with different capabilities. Khalid et al. [26] proposed a decentralized authentication and access control mechanism based on fog computing and blockchain. The decentralized mechanism proposed satisfies the security requirements of IoT. By authenticating and authorizing devices in IoT, the communication between devices in different IoT systems is realized. Panda et al. [27] proposed a blockchain-based distributed IoT architecture composed of the device layer, fog layer, and cloud layer. Based on the proposed architecture, they use one-way hashing technology to propose an efficient key generation and management scheme to achieve mutual authentication between heterogeneous IoT communication entities. To realize the interconnection of multiple trust domains in heterogeneous mobile edge computing scenarios, Lin et al. [28] proposed a zero-knowledge proof authentication system based on blockchain. The paper divides the MEC server into light nodes and consensus nodes based on the computing capacity of nodes. Among them, the light node authenticates users based on the non-interactive Schnorr Zero-knowledge Proof identity authentication method, and the consensus node runs a consensus algorithm to store the authentication information on the chain to realize fast switching authentication of users in HetNets. In the scenario of distributed large-scale HetNets, Shi et al. [29] proposed an Authentication, Authorization and Accounting (AAA) scheme for blockchain authorization to access HetNets data. By storing access control permissions on the chain, the paper redesigned a blockchain-based AAA process that is decentralized, tamper-free, and reliable.

However, there are still many problems in the above-mentioned blockchain-based HetNets authentication research. The proposed authentication method is static and cannot dynamically adjust authentication method according to the heterogeneous access network conditions, and the authentication method has poor scalability. Besides, the above authentication methods lack the analysis of user authentication requirements, so it is difficult to provide differentiated authentication services according to different user authentication requirements.

2.3 Differentiated authentication methods

To address the problem of differentiated authentication, Zhang et al. [30] designed a cross-domain authentication method based on blockchain. By storing domain encryption algorithm information and user authentication information on the blockchain, differentiated authentication among users with different hash algorithms and different signature algorithms between different domains is achieved. In addition, Luo et al. [31] proposed a flexible and secure composable authentication and service authorization framework for differentiated authentication requirements. The paper designs a combination of three-factor authentication protocols for different applications in 5G networks and implements a differentiated authentication scheme corresponding to four different security levels. Although the above-differentiated authentication methods can achieve differentiated authentication according to user needs, they still cannot meet the authentication requirements of 6G HetNets in terms of scalability.

In Table 1, we summarize the related work of blockchain-based and differentiated authentication methods and analyze whether they meet the new security requirements of 6G HetNets. Although the above-mentioned blockchain-based authentication methods can meet the requirements of reliability, scalability, and efficiency to a certain extent, they cannot provide differentiated authentication services for HetNets users. However, the related work of differentiated authentication mainly focuses on how to provide differentiated authentication services, and it is difficult to fully meet the new security requirements proposed by 6G HetNets. Therefore, aiming at the requirements of differentiated authentication in 6G HetNets, this paper designs a reliable, scalable, and efficient BDAM by using distributed blockchain technology. The proposed authentication mechanism can be deployed in heterogeneous access networks with different service capabilities, ignoring the impact of network heterogeneity on authentication. In addition, the mechanism can deploy different authentication methods according to network security requirements and has strong scalability. More importantly, the authentication mechanism proposed in this paper can provide differentiated authentication services by analyzing authentication requirements, can meet the endless authentication requirements in massive scenarios, and achieve fast and efficient authentication in 6G HetNets.

Table 2 Notations and Parameters of BDAM

Nation	Description
U_{id}	The identity of UE.
U_l	The unique identity label.
U_p	The password of UE.
M	The authentication method.
V	The version of the UE authentication method.
T_{ur}	The registration time of UE.
T_{ua}	The authentication time of UE.
T_{ue}	The expiration time of the UE password.
T_{ar}	The registration time of the authentication method.
T_{ae}	The expiration time of the authentication method.
R_{ua}	The result of UE authentication.
NoS	The number of successful UE authentication.
NoF	The number of failed UE authentication.
$NoAP$	The number of UE authentications per unit time.
$ContrInf$	The content of UIAC.
$ContrI/F$	The interface of UIAC.

3 Model definition and preliminaries

In this section, we first analyze the authentication architecture of 6G HetNets and then describe the proposed system model. The main notations and parameters used in the proposed system model are listed in Table. 2.

3.1 6G HetNets authentication architecture

As illustrated in Fig. 1, 6G HetNets Authentication Architecture (HAA) considered in this paper consists of a large number of access networks, authentication centers, HetNets users/devices, and HetNets gateway. In the authentication process, HetNets users/devices send authentication requests to the HetNets gateway. Then, the HetNets gateway forwards the authentication request to the authentication center to

Table 1 Analysis of Related Work

Ref.	Year	Reliability	Scalability	Efficiency	Unified Management	Differentiated Authentication
[25]	2020	✓	✓	✓	✗	✗
[26]	2020	✓	✓	✓	✗	✗
[27]	2021	✓	✓	✓	✗	✗
[28]	2021	✓	✓	✓	✗	✗
[29]	2021	✓	✓	✓	✗	✗
[30]	2020	✓	✗	✓	✓	✓
[31]	2021	✓	✗	✓	✗	✓
Ours	2022	✓	✓	✓	✓	✓

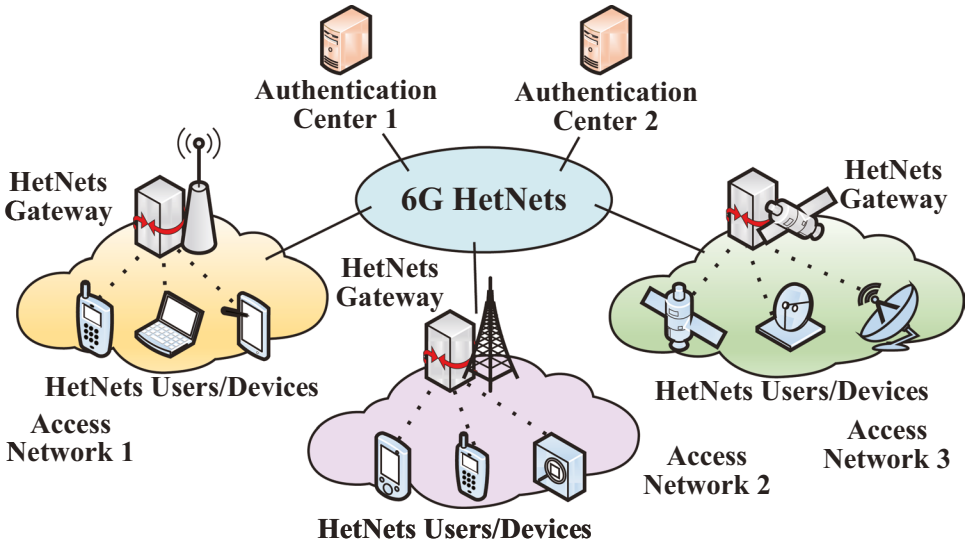

Fig. 1 The illustrate of 6G HetNets Authentication Architecture (6G HAA)

authenticate the user/device identity. The main roles of the components in 6G HAA are explained as follows.

Access Networks 6G HetNets is composed of a variety of access networks (mobile network, satellite network, WLAN, etc.). Since the underlying protocols and service capabilities of each access network are different, the identity authentication methods in different access networks are different.

Authentication Center The authentication center is the entity that implements HetNets user and HetNets device identity authentication in the 6G HAA. The authentication center can be built based on different technologies (such as PKI, CLC, IBC) to meet different authentication needs.

HetNets Users/Devices The HetNets Users/Devices are the entities that initiate the authentication requests and need to be authenticated. In 6G HAA, HetNets Users/Devices consist of a large number of users and equipment (such as sensors, mobile phones, laptops, smart grid devices, smart home devices, etc.).

HetNets Gateway Each HetNets Gateway in the 6G HAA connects to a large number of HetNet Users/Devices in a different way (e.g., WiFi, LAN, Bluetooth). The HetNets Gateway forwards and processes authentication packets between HetNets Users/Devices and authentication centers.

3.2 System model

To differentially authenticate user identities in 6G HetNets, we design a system model BDAF based on the 6G HAA, which consists of User Equipment (UE), Authentication Agent (AA), Access Domain (AD), Blockchain Networks (BN), and Network Administrator (NA), as shown in Fig. 2. BDAF can be divided into two parts: 6G HetNets and BN. The 6G HetNets consist of different ADs. Each AD contains AAs and UEs. BN consists of different AAs, which are deployed in different ADs. It should be noted that the proposed BDAF is built on 6G HAA. Unlike the 6G HAA, we use AA and BN to replace the function of the Authentication Center of 6G HAA. In the following, we present the functions of the different authentication entities in the proposed BDAF.

AD We divide *Access Networks* in 6G HAA into several ADs according to the type of access networks. As shown in Fig. 2, WLAN and satellite network are represented as AD_1 and AD_3 respectively. The access networks of the same type can also be divided into different ADs due to other factors such as region and scale. As shown in Fig. 2, AD_1 and AD_2 represent two WLANs of the same type. In addition,

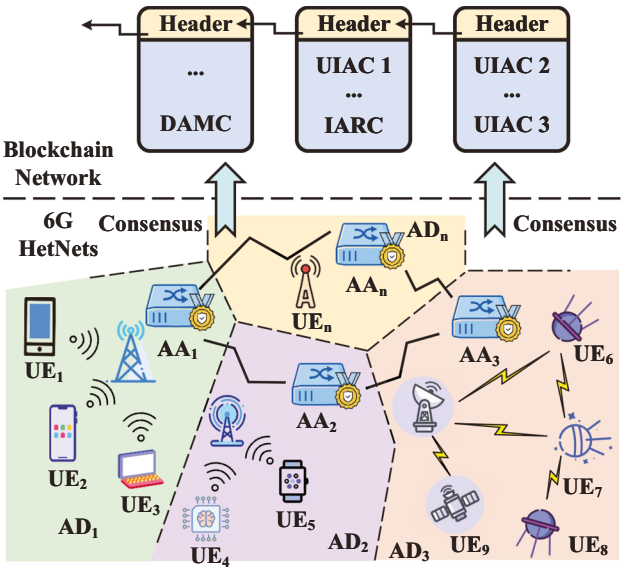

Fig. 2 Blockchain-Based Differentiated Authentication Framework (BDAF)

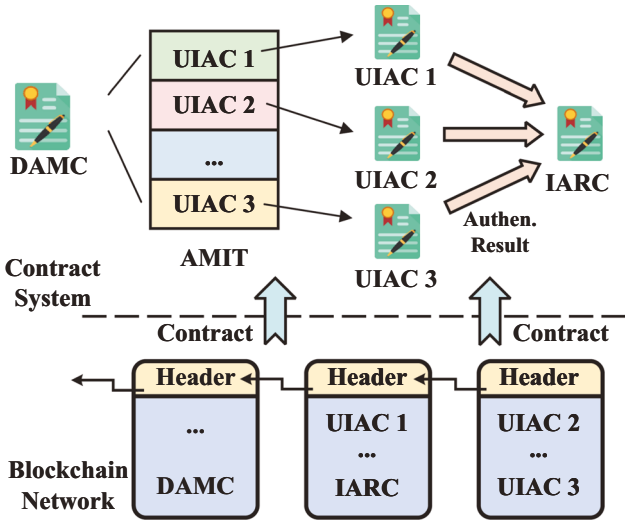

Fig. 3 The Authentication Contract System (ACS) in BDAF

to differentially authenticate user identities, we deploy AA entities in each AD for differential authentication operations. NA can dynamically deploy the number of AA entities based on the network scale and requirements.

UE In BDAF, UE is used to uniformly represent *HetNets Users/Devices* in 6G HAA. As shown in Fig. 2, UE in AD_1, AD_2, and AD_3 are represented as UE_1- UE_9 respectively.

AA For differentiated identity authentication, AA performs all functions of *HetNets Gateway* and part functions of the *Authentication Center* in 6G HAA. It has the following four functions: 1) Respond to differentiated authentication requests to conduct differentiated authentication. 2) Play the role of *Authentication Center* in 6G HAA together with BN for UE registration and authentication; 3) As the interface of BN, it provides storage, query, and update services for UEs; 4) Provide a management interface for NAs to register and update authentication methods, and implements dynamic updating of HetNets authentication methods. As shown in Fig. 2, AA is represented as AA_1- AA_n in the ADs. In actual deployment, AA is deployed on the edge access gateway of a 6G HetNets to improve the efficiency of identity authentication.

BN AAs in different ADs run the same consensus algorithm to form a BN, and BN has most of the functions of the *Authentication Center* in 6G HAA. In BDAF, BN mainly has the following three functions. Firstly, BN stores the authentication information (registration information, authentication credentials, authentication records, etc.) to provide identity authentication services for UEs; Secondly, multiple authentication methods are stored in the form of authentication contracts, which entrusts the scalability of the BDAF; Thirdly, BN implements unified management of different authentication methods in 6G HetNets. To solve the processing bottleneck caused by blockchain data synchronization, the blockchain can be set to periodic synchronization, and AA responds to authentication requests, to improve the authentication processing efficiency. Smart contracts [32] in the BDAF (such as DAMC, UIAC, IARC, etc.) will be introduced in the next subsection.

NA To manage authentication methods manually, we introduce the role of NA in the BDAF. The NA can adjust (deploy, delete or update) the authentication methods in the blockchain through AA's management interface according to the network conditions.

3.3 Authentication contract system

The structure of the Authentication Contract System (ACS) in BDAF is shown in Fig. 3. ACS consists of multiple UE Identity Authentication Contracts (UIAC), 1 Identity Authentication Record Contract (IARC), and 1 Differentiated Authentication Management Contract (DAMC). The function of each contract is introduced as follows.

DAMC: DAMC is the core component of ACS, which uniformly manages UE identity and authentication contracts. In ACS, the DAMC has the following two functions. First, unified management of UIACs to meet the requirements of secure sharing of authentication methods on HetNets; Second, provide AAs with an authentication method query interface to provide UEs with differentiated authentication services. Therefore, two lookup tables are maintained in the DAMC according to the two functions described above: Authentication Method Information Table (AMIT), and Authentication Contract Information Table (ACIT). AMIT stores authentication method information, and ACIT stores the registered authentication contract information. Table 3 is an illustration of AMIT, in which each row represents an authentication method

Table 3 Illustration of Authentication Method Information Table (AMIT)

U_l	U_{id}	M	V	T_{ur}
14f7b73ab7e5fda85a86bbfdc5d0d966	5453e43b22d95a547dfc5f72594831f4	EAP-MD5	1.0	2021-09-03 16:24
f5edc81797c3f50065ee8133a5e702ed	5453e43b22d95a547dfc5f72594831f4	EAP-MD5	2.0	2021-11-10 18:32
...	...	...	...	...
cb177ecd1834e2c958fa3e2ddadfafcf	b2c78cfe1b52426c246e96067ee21eec	5G-AKA	2.0	2021-12-25 14:23

Table 4 Illustration of Authentication Contract Information Table (ACIT)

M	V	T_{ar}	T_{ae}	*ContrInfo*	*ContrI/F*
EAP-MD5	1.0	2021-09-01 12:20	2023-10-31 15:20	{Channel: mychannel; Chaincode: eap_auth1; ...}	{Query: 'query'; Auth: 'authen'; Update: 'update'; ...}
EAP-MD5	2.0	2021-10-10 08:12	None	{Address: 0x6b17...1d0f; ...}	{Get: 'get'; Auth: 'auth1'; Delete: 'delete'; ...}
...	...	...	...	...	...
5G-AKA	2.0	2021-12-25 12:53	None	{Channel: mychannel; Chaincode: mycc2; ...}	{Query: 'query'; Auth: 'authen2'; Delete: 'del'; ...}

registered by a UE. Table 4 gives an illustration of the ACTI, in which each row represents the information of a UE authentication method.

UIAC: UIAC (such as UIAC 1-UIAC 3 in Fig. 3) is an important component of ACS. In ACS, each UIAC represents a UE authentication method. The same authentication method can be registered as a different UIAC due to the version or other factors. In each UIAC, information for authentication (such as identity credentials, time information, etc.) is stored. Through the interface with BN, AA invokes the UIAC to authenticate UEs. There are two deployment methods for UIAC. Firstly, NA can register the authentication method when the system is initialized; secondly, UIAC can be dynamically deployed according to the authentication requirements. The NA can register, update and delete the UIAC by calling the management interface of AA. UIAC maintains a UE identity information table (UIIT), which stores UE authentication information such as UE credentials. We present the UIIT based on the EAP-MD5 V1.0 authentication method, as shown in Table 5. In Table 5, each row represents a UE authentication information.

IARC: The IARC stores UE authentication records in ACS. After UE completes authentication, AA invokes the IARC to store UE authentication behavior on the chain. The deployment of the IARC can not only realize the traceability of user authentication behavior but also meet the requirements of cross-domain rapid authentication. An Authentication Record Information Table (ARIT) is maintained in the IARC. Table 6 is an example of ARIT, in which each row represents a UE authentication behavior record.

3.4 Assumptions

We believe it is appropriate and necessary to clarify the assumptions made before the BDAM is proposed.

1. Blockchain-enabled AA nodes are legitimate and trusted.
2. Before differentiated identity authentication, the UE and NA have negotiated the key with the AA through the secure channel, and UE has obtained the public key pk of the AA.
3. The transactions are initiated between AAs and BN through the secure channel

4 Blockchain-based differentiated authentication mechanism

In this section, based on the proposed BDAF, we design the BDAM as shown in Fig. 4. It should be noted that, in this paper, to simplify the authentication process, the encryption and decryption process of the messages passed between UE and AA is not shown in Fig. 4.

As shown in Fig. 4, BDAM is composed of four processes: authentication method registration process (steps 1-6), UE

Table 5 Illustration of EAP-MD5 V1.0 UE Identity Information Table (UIIT)

U_{id}	U_p	T_{ur}	T_{ue}
5453e43b22d95a547df-c5f72594831f4	6c04bbdd4d2f3587	2021-09-03 16:24	2022-05-03 16:24
...	...	...	...
5c4cf69866ac22f1973e-ca3a50b4742c	113461a0739de7b6	2021-12-08 12:53	2022-12-08 12:53

Table 6 Illustration of Authentication Record Information Table (ARIT)

U_{id}	M	V	T_{ua}	R_{ua}	NoS	NoF	$NoAP$
5453e43b22d95a547dfc5f72594831f4	EAP-MD5	1.0	2021-09-04 17:34	Success	1	0	1
5c4cf69866ac22f1973eca3a50b4742c	EAP-MD5	1.0	2022-01-09 15:54	Failed	0	1	1
...	...	...	...	...	...	...	...
b2c78cfe1b52426c246e96067ee21eec	5G-AKA	2.0	2022-01-11 08:12	Success	8	1	3

registration process (steps 7-12), UE initial authentication process (steps 13-21), and UE re-authentication process (steps 19-24). In the following subsections, we present the detailed steps of the above four processes.

4.1 Authentication method registration process

In the proposed BDAF, the different authentication methods are represented by different UIACs. To uniformly manage the different UIACs, we deploy DAMC in the ACS. In the authentication method registration process, NA registers the authentication method in the DAMC through the secure management interface of AAs. The detailed processes are described in the following.

1. NA sends an Authentication Method Registration (AMR) request which is encrypted by the AA_1's public key pk_1 to the authentication agent AA_1. The AMR includes authentication method M, authentication method version V, authentication method expired time T_{ae} and authentication method information I_{ar}. I_{ar} includes contract interface *ContrI/F*, contract information *ContrInf* and other content.
2. The AA_1 decrypts the AMR using its own private key sk_1 and then invokes the *VerifyAuthMethod()* function of DAMC to verify whether the authentication method that needs to be registered exists in the blockchain. The content of smart contract DAMC is shown in Algorithm 1.

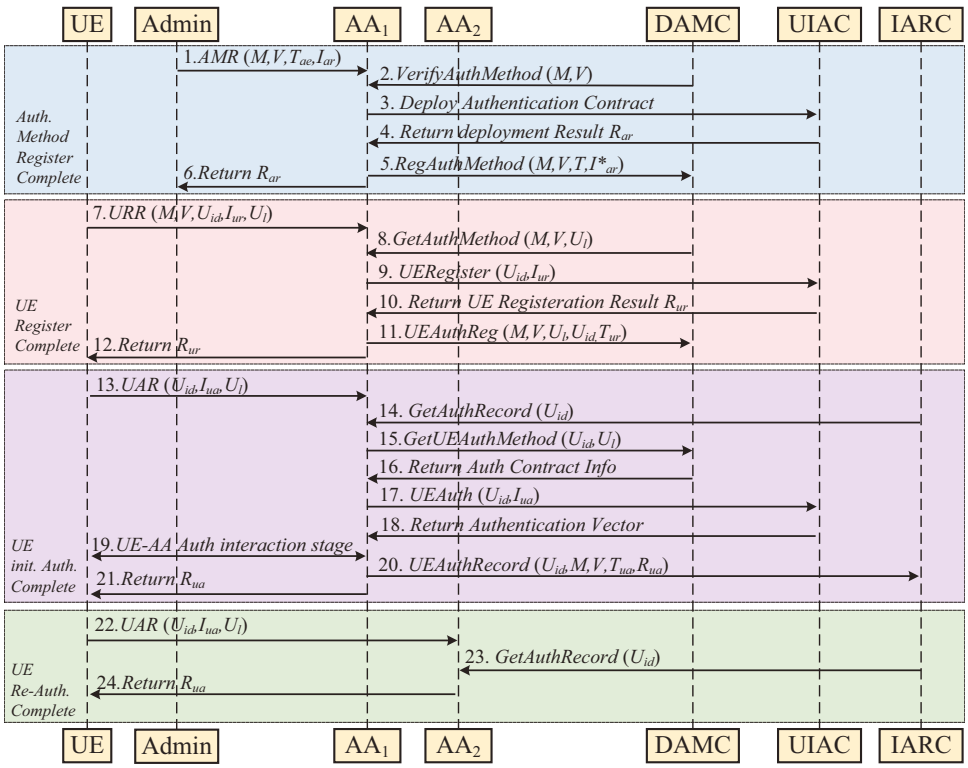

Fig. 4 The Blockchain-Based Differentiated Authentication Mechanism (BDAM)

3. If the authentication method does not exist, AA_1 initiates a contract deployment transaction and deploys the corresponding UIAC in the blockchain.
4. The blockchain returns the deployment result R_{ar} and the smart contract information of UIAC to AA_1.
5. After the UIAC is deployed, AA_1 invokes the *RegAuthMethod()* function to register M, V, T and the updated authentication method information I^*_{ar} in the blockchain. I^*_{ar} is the updated authentication method information based on the contract registration result in I_{ar}, for example, adding contract address information and other contents to the *ContrInf* in I_{ar}. T contains expiration time T_{ae} and registration time T_{ar} of the authentication method.
6. The AA_1 returns the authentication method registration result R_{AM} to NA.

4.2 UE registration process

Before UE authentication, the identity information of UE needs to be registered in the UIAC through AA. In the UE registration process, the UE first needs to query the information of the corresponding authentication method in the DAMC. Then, the AA calls the registration interface to register its identity in the UIAC. The steps of the UE registration process are given as follows.

7. UE sends a UE Registration Request (URR) with authentication method M, authentication method version V, user identity U_{id}, user registration information I_{ur} and user identification U_l to AA_1. The URR is encrypted by the public key pk_1 of AA_1. The UE credentials are stored in I_{ur}, and the content stored in I_{ur} varies according to authentication methods. Besides, the U_l characterizes the unique identity of UE and is used to construct the mapping relationship between UE and the authentication method.
8. AA_1 first uses its own private key sk_1 to encrypted the request. And then, it invokes *GetAuthMethod()* function to obtain the authentication contract information stored in the DAMC.
9. After obtaining the information of the corresponding UIAC, AA_1 invokes the *UERegister()* function to register UE identify on UIAC. The smart contract of UIAC is shown in Algorithm 2.
10. UIAC returns the UE registration result R_{ur} to AA_1.
11. After UE registration is successful, the AA_1 invokes the *UEAuthReg()* function to store the registration information in the DAMC.
12. Finally, AA_1 returns the registration result R_{ur} to UE through the secure channel.

Algorithm 1 The Smart Contract of DAMC.

```
function VerifyAuthMethod(M,V)
    if stub.GetState(M,V) != nil then
        return "Auth. Method exists"
    else
        return "Auth. Method does not exist"
    end if
end function
function RegAuthMethod(M,V,T,I*_ar)
    if stub.PutState(M,V,T,I*_ar) != nil then
        return "Auth. Method Reg. Suc."
    else
        return error()
    end if
end function
function GetAuthMethod(M,V,U_l)
    if stub.GetState(M,V,U_l) != nil then
        return ContrInf,ContrI/F
    else
        return "Auth. Method not Reg."
    end if
end function
function UEAuthReg(M,V,U_l,U_id,T_ur)
    if stub.PutState(M,V,U_l,U_id,T_ur) != nil then
        return "UE Auth. Method Reg. Suc."
    else
        return error()
    end if
end function
function UEAuthUpd(M*,V*,U*_l,U_id,T_ur)
    if stub.PutState(M*,V*,U*_l,U_id,T_ur) != nil
then
        return "UE Auth. Method upd. Suc."
    else
        return error()
    end if
end function
function GetUEAuthMethod(U_id,U_l)
    if stub.GetState(U_id,U_l) != nil then
        return ContrInf, ContrI/F
    else
        return error()
    end if
end function
```

It should be noted that the process of UE information updating is generally the same as the user registration process, so it is not shown in Fig. 4. The main differences between the two processes are as follows. First, in the update process, the UE sends a UE Update Request (UUR) including the updated authentication method M^*, the updated authentication method version V^*, the updated

identification U_l^*, and the updated authentication credentials I_{ur}^*. In addition, AA needs to determine whether the authentication credential is updated or the authentication method is updated according to the identification bit in UUR. If the credentials need to be updated, AA calls the *UEUpdate()* function to update the UE credentials; if the authentication method need to be updated, the *UEAuthUpd()* function is invoked to update the authentication information stored in the DAMC.

4.3 UE initial authentication process

After the UE registration process, when UE accesses the network, identity authentication is required to verify the identity of UE. The complete process of the UE initial authentication is shown in steps 13-21 in Fig. 4.

13. UE sends a UE Authentication Request (UAR) to AA_1. The UAR contains the UE identity U_{id}, UE authentication information I_{ua} and UE identification U_l. In order to prevent information from being leaked, the UAR is encrypted with AA_1's public key pk_1.

14. AA_1 decrypts the UAR with the private key sk_1, and invokes the *GetAuthRec()* function to query UE history authentication record in IARC. The smart contract of IARC is shown in Algorithm 3.

15. If there is no historical authentication record, AA_1 invokes *GetUEAuthMethod()* function to query the authentication method information registered in the DAMC.

16. DAMC returns the authentication contract information which is stored in the blockchain to AA_1.

17. Then, AA_1 invokes the *UEAuth()* function to generate the Authentication Vector (AV) in UIAC.

18. UIAC returns the generated AV to AA_1

19. AA_1 uses the returned AV to interact with UE to verify UE's identity.

20. After UE's authentication is successful, AA_1 invokes the IARC *UEAuthRec()* function to record the authentication behavior information in IARC. The authentication behavior information includes U_{id}, M, V, authentication time T_{ua} and authentication result R_{ua}. Thereafter, IARC updates *NoS*, *NoF*, and *NoAP* while storing UE authentication behaviors.

21. In the end, AA_1 returns authentication result R_{ua} to UE through secure channel.

Algorithm 2 The Smart Contract of UIAC.

```
function UERegister(U_id,I_ur)
    if stub.PutState(U_id,I_ur) != nil then
        return R_ur="UE Reg. Suc."
    else
        return R_ur="UE Reg. Failed"
    end if
end function
function UEAuth(M,V,T,I*_ar)
    if stub.GetState(M,V,T,I*_ar) != nil then
        Generate Authentication Vector AV
        return AV
    else
        return error()
    end if
end function
function UEUpdate(U_id,I*_ur)
    if stub.PutState(U_id,I*_ur) != nil then
        return R_up="UE Upd. Suc."
    else
        return R_up="UE Upd. Failed"
    end if
end function
```

To improve the security of the 6G HetNets, we introduce three parameters NoS, NoF, and NoAP to evaluate the reputation of the UE in the authentication process.

Algorithm 3 The Smart Contract of IARC.

```
function GetAuthRec(U_id)
    if stub.GetState(U_id) != nil then
        Get Authentication Result R_ua
        return R_ua
    else
        return error()
    end if
end function
function UEAuthRec(U_id,M,V,T_ua,R_ua)
    if stub.PutState(U_id,M,V,T_ua,R_ua) != nil
then
        return "UE Auth. Record Stored Suc."
    else
        return error()
    end if
end function
```

NoS and NoF can characterize UE authentication reputation to a certain extent. The higher the proportion of NoS in the total number of authentications, the better UE's reputation; conversely, the higher the proportion of NoF,

the worse UE's reputation. During user re-authentication, the AA will block the authentication behavior of UEs with poor reputation according to the preset reputation threshold, so as to prevent the authentication behavior of malicious UEs.

Besides, to prevent malicious UEs from consuming system resources through frequent authentication in a short period, we also use the NoAP to record the authentication behavior of UEs within a period. AA can control over-frequent UE authentication behaviors according to the preset NoAP threshold.

4.4 UE re-authentication process

In order to rapidly authenticate massive UEs in 6G HetNets, we design a fast authentication method in BDAM. It is assumed that UE has been authenticated in AA_1, and when UE accesses AA_2, UE Re-authentication is required. This process provides a new solution for UE fast authentication, which eliminates the need for complete UE authentication, reduces signaling overhead, and improves authentication efficiency. The process of UE Re-authentication is shown as follows.

22. UE sends AA_2 a UAR encrypted with AA_2's public key pk_2.

23. AA_2 decrypts the UAR with the private key sk_2, and calls IARC *GetAuthRecord()* function to obtain UE history authentication records stored in the blockchain.

24. AA_2 analyzes the authentication record for fast authentication and returns authentication result R_{ua} to UE through a secure channel.

5 Evaluation setting

Based on the proposed BDAM, we deployed a prototype system for performance evaluation. In this section, we first describe the configuration of the blockchain-based differentiated authentication prototype system. Subsequently, we introduce several comparison methods in the evaluation.

5.1 Evaluation environment

As shown in Fig. 5, We deploy the VMware VSphere virtualization platform in the ESXi cluster and installed 10 servers for evaluation. To reduce the evaluation error, we set the same configuration for each server. Each server was configured with 40G disk size, 8G memory, and the Ubuntu 20.04 system. The deployed 10 servers are divided into two parts: 9 servers are deployed as AAs to authenticate UEs, and 1 server is deployed as UE to initiate UAR and URR to AA. In order to evaluate the BDAM in HetNets, we divide the 9 AAs into 3 ADs (AD_1, AD_2 and AD_3), and each AD contains 3 AAs. The blockchain client is installed in each AA, and the blockchain clients run the same consensus algorithm to form BN.

In the prototype system, we build a consortium blockchain to satisfy the security requirements in the registration and authentication process. Deploying BDAM in consortium blockchain has the following advantages over deploying it in public blockchain and private blockchain [33]. On the one hand, compared with the public blockchain, the consortium blockchain has the advantages of low transaction cost and high privacy and can avoid the authentication data from being obtained by malicious nodes; on the other hand, compared with the private blockchain, consortium blockchain has better scalability and can provide the dynamic authorization and management of the blockchain nodes.

HyperLedger is a consortium blockchain architecture initiated by the Linux Foundation (https://www.hyperledger.org/). HyperLedger Fabric [34], as a highly modular, scalable, and extensible architecture in HyperLedger, has been widely and maturely used in various scenarios. Therefore, in this paper, we deploy a differentiated authentication approach based on Fabric.

There are three common consensus algorithms in Fabric: Solo, Kafka, and Raft. Compared with Solo and Kafka, Raft is concisely configured, highly decentralized, and can enable strong consistency in distributed systems. So, in this paper, we choose Raft as the consensus algorithm in Fabric.

In Fabric, the Components (CO.) can be divided into Client, Peer, Orderer, and CA. The client is employed to interact with the blockchain; the Peer process transactions, and maintain the ledger and smart contracts; Orderer is used to sort the transactions initiated by Peers and pack the sorted transactions to form blocks; CA is the authentication center of Fabric to authenticate and authorize blockchain nodes. In the proposed prototype system, BN is divided into three organizations (org1, org2, and org3). In each org, one CA and 3 Peers are deployed. org1, org2, and org3 are mapped to AD_1, AD_2, and AD_3, respectively. The client component is deployed in each Peer node. In this paper, we use Fabric SDK (fabric-sdk-py) (https://github.com/hyperledger/fabric-sdk-py) to enable the Client function. Besides, we deploy the Orderer cluster with 3 Orderers (orderer0, orderer1, and orderer2). The configuration of the prototype system is shown in Table 7. Figure 5 shows the prototype system of differentiated authentication. We build HetNets with three ADs based on the configuration in Table 7. In each AA server, we deploy Fabric components (Peer, CA, and Orderer).

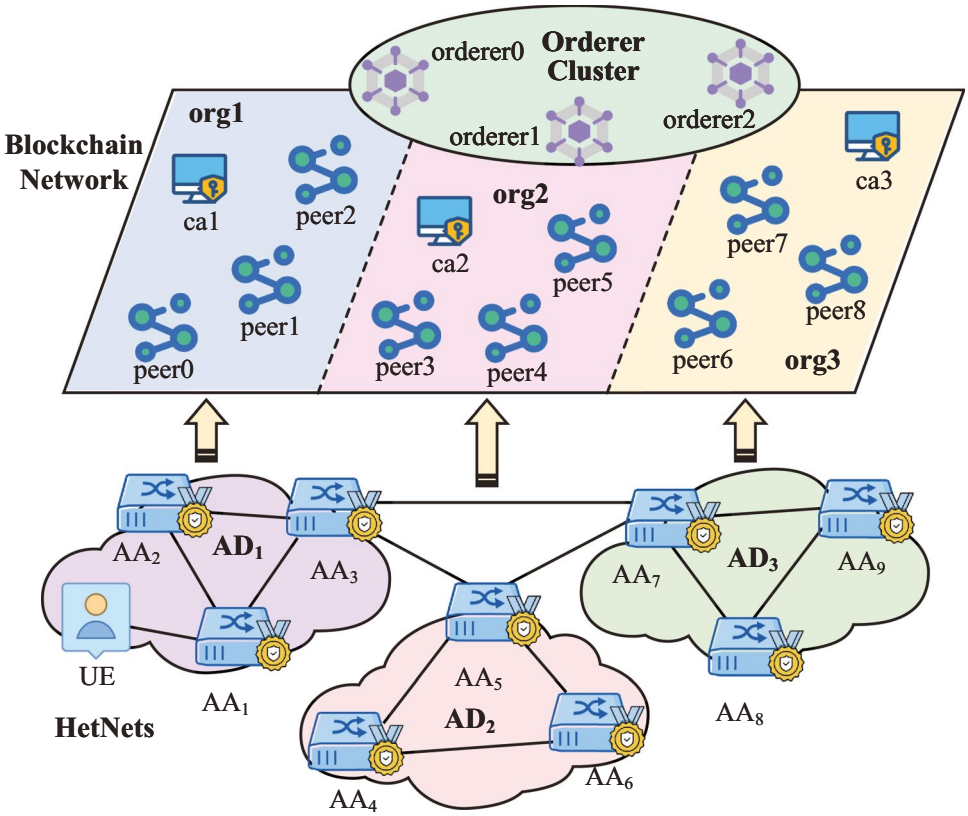

Fig. 5 Blockchain-Based Differentiated Authentication Prototype System

Figure 6 shows the relationship between UE, AA, Fabric components, and BN. UE and AA are deployed in two different servers (Node10 and Node1), and UE communicates with AA by packets for UE registration and authentication. In Node1, peer0, orderer0, and ca1 are also deployed. The Raft consensus algorithm is run between different Peers that make up BN. The ACS (DAMC, UIAC, and IARC) are deployed in BN in the form of chaincodes. AA interacts with the Peer component through fabric-sdk-py to invoke the chaincodes.

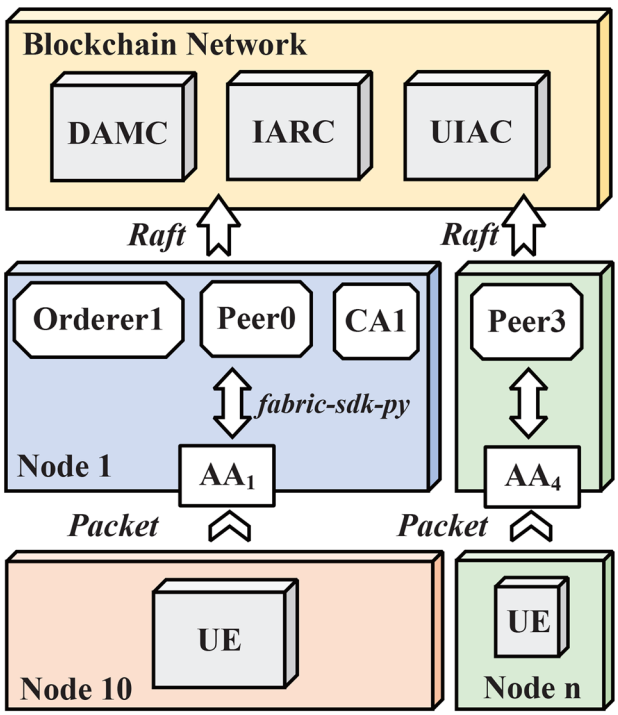

Fig. 6 The Relationship between UE, AA, Fabric Components and BN

5.2 Comparison methods

In this subsection, we present several authentication methods for comparative evaluation.

5.2.1 Non-BDAM

Before introducing several other authentication methods, we present the UE registration and authentication process in the None Blockchain-based Differentiated Authentication Mechanism (Non-BDAM). The Non-BDAM is shown in Fig. 7.

As shown in Fig. 7, Non-BDAM is essentially built on the blockchain-based authentication framework too. Entities in Non-BDAM are consistent with BDAM and are also composed of UE, AA, AD, and BN. It should be noted that the AA in Non-BDAM does not provide differentiated authentication services. Compared with BDAM, the ACS

Table 7 The configuration of Differentiated Authentication Prototype System

Name	*CO.*	*Org*	*AD*	*AA*	*Server*
orderer0	Orderer	org1	AD1	/	Node1
ca1	CA	org1	AD1	/	Node1
peer0	Peer	org1	AD1	AA1	Node1
peer1	Peer	org1	AD1	AA2	Node2
peer2	Peer	org1	AD1	AA3	Node3
UE	/	/	AD1	/	Node10
orderer1	Orderer	org2	AD2	/	Node4
ca2	CA	org2	AD2	/	Node4
peer3	Peer	org2	AD2	AA4	Node4
peer4	Peer	org2	AD2	AA5	Node5
peer5	Peer	org2	AD2	AA6	Node6
orderer2	Orderer	org3	AD3	/	Node7
ca3	CA	org3	AD3	/	Node7
peer6	Peer	org3	AD3	AA7	Node7
peer7	Peer	org3	AD3	AA8	Node8
peer8	Peer	org3	AD3	AA9	Node9

in Non-BDAM BN contains only UIACs and IARC, and no DAMC provides a differentiated authentication service.

In the authentication method registration process of Non-BDAM, NA sends the AMR to the AA_1. Then, AA_1 deploys the corresponding UIAC in the blockchain. The authentication method registration time is the time from the initiation of the authentication method installation transaction by the NA to the completion of the transaction.

Steps 5-8 in Fig. 7 are the UE registration process for Non-BDAM. The process of UE registration in the Non-BDAM is given as follows. UE first sends the URR to the AA_1, and then AA_1 invokes the function of UIAC to register the identity in the blockchain. The UIAC returns the registration result to the AA_1, and the AA_1 forwards the result to UE through the secure channel.

Steps 9-15 are the UE authentication process. The Non-BDAM UE authentication process lacks steps 15 and 16 compared to the BDAM authentication process in Fig. 4. UE first sends the UAR to the AA_1. The AA_1 calls the function to verify whether the UE is authenticated in the network. If UE is the user who authenticates for the first time, AA_1 invokes the function to authenticate the identity in the blockchain. UIAC returns the authentication vector to the AA_1, and then AA_1 interacts with UE through the secure channel by using the returned authentication vector. After the UE-AA authentication interaction stage ends, the AA_1 stores the authentication record in the blockchain, and sends the authentication result R_{ua} to the UE.

To realize the rapid authentication of UE identity, there is also the UE Re-authentication process (steps 16-18) in the Non-BDAM. The UE Re-authentication process in the Non-BDAM is the same as that in BDAM.

5.2.2 BDAM-NR

Furthermore, to evaluate the performance of the proposed rapid authentication method, we design a Non-rapid authentication method based on the proposed BDAM (BDAM-NR) for comparative experiments.

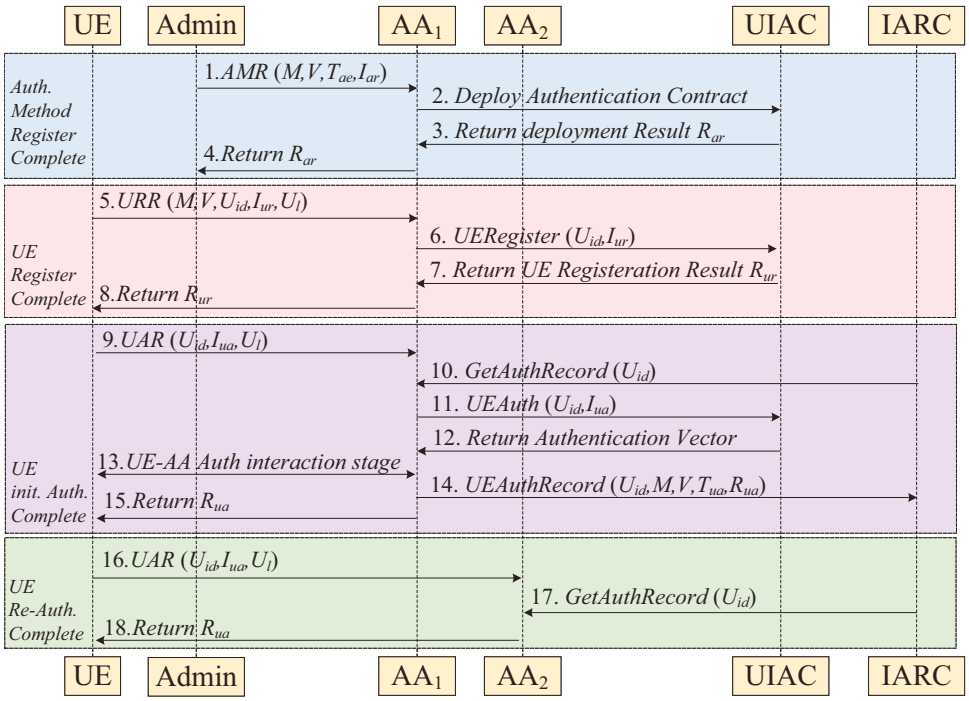

Fig. 7 The None Blockchain-based Differentiated Authentication Mechanism (Non-BDAM)

Compared with the UE initial authentication process, BDAM-NR is consistent with BDAM except for the lack of steps 14 and 20 in Fig. 4. In addition, due to the lack of storage and query steps for UE authentication behavior, in BDAM-NR, the UE re-authentication process needs to re-authenticate UE identity in UIAC smart contract, the same as the UE initial authentication process.

5.2.3 5G-AKA

In 5G networks, 5G-AKA and EAP-AKA' are the main authentication methods for user authentication [35]. Given that the authentication protocol of 5G-AKA and EAP-AKA' are similar, this subsection uses 5G-AKA as an example to demonstrate the use of mobile network authentication methods in the proposed BDAM. It should be noted that this subsection focuses on the user authentication process in 5G mobile networks and does not describe the key negotiation process.

Authentication entities in 5G-AKA can be divided into the following four categories: User Equipment (UE), SEcurity Anchor Function (SEAF), AUthentication Server Function (AUSF), and Unified Data Management/Authentication credential Repository and Processing Function (UDM/ARPF). UE is the user terminal, which stores SUbscription Permanent Identifier (SUPI), public key *pk* of Home Network (HN), sequence number *sqn* and long-term shared key *K*. SEAF is the authentication participation entity in Service Network (SN), which is used to provide services to UE after successful authentication. AUSF is the authentication server in HN, responsible for discriminating SEAF authority and verifying the authentication response of UE; UDM/ARPF stores the processing of subscriber authentication credentials. UDM stores HN private key *sk*, for Subscription Identifier De-concealing Function (SIDF) provides SUbscription Concealed Identifier (SUCI) resolution into SUPI. ARPF stores the shared key *K*, SUPI, and root sequence number *SQN*, which is used to generate the authentication vector.

To deploy the 5G-AKA in BDAM, we adapt the entities in 5G-AKA. First, AA_1 is used instead of SEAF to enable the forwarding and processing of 5G-AKA authentication messages. In addition, we unified the authentication functions such as AUSF, ARPF, UDM, and SIDF into an authentication contract (UIAC 5G-AKA) to realize 5G-AKA identity authentication in distributed scenarios. The 5G-AKA authentication entity in BDAM consists of UE, AA_1, and ACS (DAMC, UIAC 5G-AKA, and IARC).

In UIAC 5G-AKA (U5A), the Authentication Data (AD) to generate the AV is stored. AD consists of Authentication Initialization Data (AID) and Authentication Process Data (APD). AID is the authentication information negotiated during registration phase, and contains *K*, *SQN*, registered SUPI, and *sk*. APD is the authentication information obtained in the process of generating the AV and is used to authenticate UE. APD includes XRES*, the freshness *F* of AV, RAND, etc.

The 5G-AKA in the proposed BDAM (5AB) is shown in Fig. 8. To simplify the authentication process, we make two assumptions as follows. First, it is assumed that AA_1 has completed the authorization in U5A. Second, it is assumed that UE has already registered in the U5A.

Steps 1-9 in Fig. 8 are consistent with the UE authentication process (steps 13-21) in Fig. 4. Different from Fig. 4, in Fig. 8 we add three specific descriptions of authentication preparation (step x), authentication contract response (steps i-k), and authentication interaction (steps a-h).

In authentication preparation phase, UE generates SUCI using the stored *pk*, *enc()* is the encryption function. When generating SUCI, we add a random number *R* to protect against replay attacks. In addition, to construct a mapping between UE and the authentication methods, the unique identity label U_l is generated by SUPI, with *f()* as the identity label generation function. When constructing the UAR message, the U_{id} content is set to SUCI, the U_l content is set to the generated identity label, and the I_{ua} content is set to *R*.

In authentication contract response phase, U5A first decrypts the received U_{id} with *sk* to obtain the SUPI and *R**, and *dec()* is the decryption function. Subsequently, the accuracy of the received message and the legitimacy of UE are verified in turn. The accuracy of the received message is obtained by comparing *R** with I_{ua}; the legitimacy of UE is obtained by verifying whether the SUPI is the registered legitimate UE. After the message accuracy and the identity legitimacy are verified, the authentication contract generates an AV containing RAND, AUTN, XRES*, and HXRES* based on AID.

In the authentication interaction phase, UE first generates AUTN* based on the RAND, and then compares the AUTN* with the received AUTN to verify the identity of the network. After the network identity verification is completed, UE generates RES* and sends it to AA_1. Then, AA_1 calculates HRES* and compares it with HXRES*. If the two are equal, it means that SN has authenticated the UE; if they are not equal, UE authentication fails. After the verification of HRES* and HXRES* is completed, AA_1 invokes *authVerify()* function to forward RES* to U5A; U5A first verifies the freshness *F* of AV and compares RES* with XRES*. If they are equal, it means that HN has authenticated UE; otherwise, authentication fails; finally, U5A returns the R_{ua} to AA_1.

5.2.4 EAP-MD5

In WLAN, EAP-MD5 is one of several common authentication methods. The authentication entity in EAP-MD5 consists of Client, Device, and Server [36]. EAP-MD5 authentication

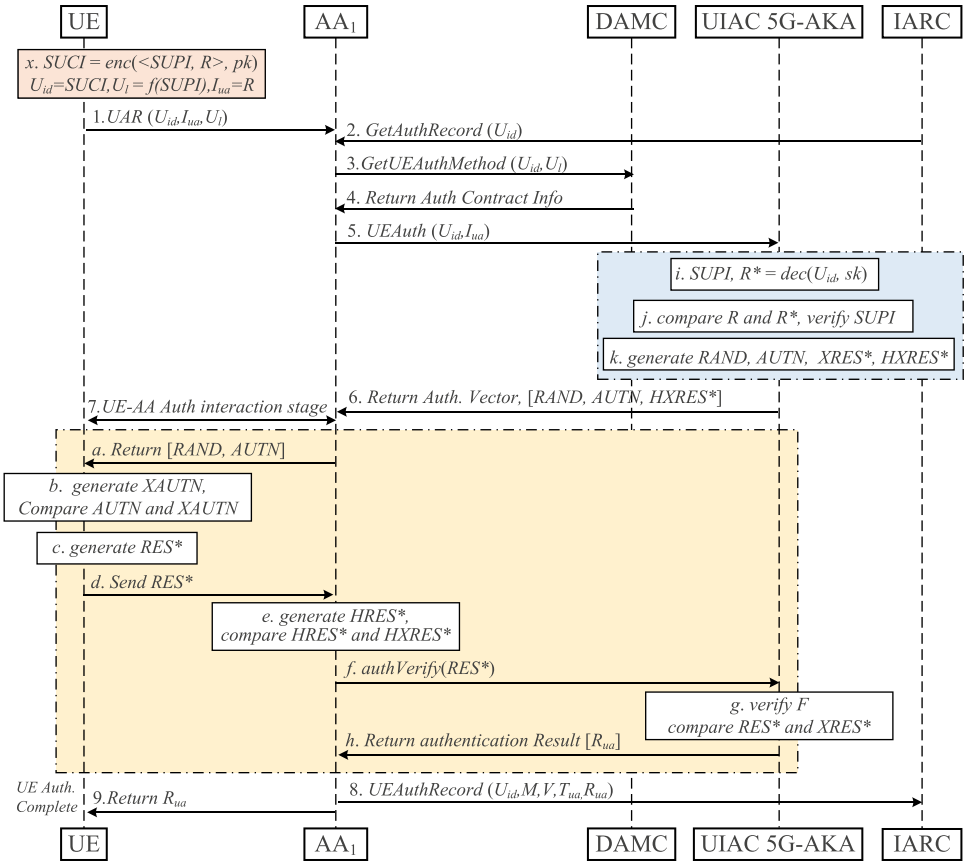

Fig. 8 5AB Authentication Protocol

method consists of two phases: registration and authentication. In the registration phase, the Client needs to register the identity and password in the Server; in the authentication phase, the Device verifies the MD5-Challenge (MC) generated by the Server and Client to authenticate the identity of the Client.

In the proposed BDAM, we adapt the authentication entity in EAP-MD5. First, we use UE to uniformly characterize the Client in BDAM; second, to construct a distributed authentication method applicable to BDAM, we represent the authentication function of the Server in EAP-MD5 with the smart contract (UIAC EAP-MD5). Finally, AA performs the function of the Device and forwards the authentication messages of UE and UIAC EAP-MD5 (UEM).

Figure 9 shows the EAP-MD5 in the proposed BDAM (EMB). We assume that UE has a registered identity in UEM. In addition, we also improve the EAP-MD5 to achieve mutual authentication between UE and the network.

In Fig. 9, we add three phases based on Fig. 4, which are the authentication preparation phase (step x), authentication contract response phase (steps i-k), and authentication interaction phase (steps a-e).

In the authentication preparation phase, UE needs to generate a unique identifier label U_l for mapping the authentication method and set the I_{ua} content in UAR as the generated random challenge R_1. The U_{id} content in UAR is set to the identity of UE.

In the authentication contract response phase, UEM first verifies the identity based on the received U_{id}. After successfully verifying UE identity, UEM generates MC MC_1 based on the received I_{ua} and the password U_p registered on the chain. In addition, we add the random challenge R_2 and MC MC_2 to achieve mutual authentication. MC_2 is generated by R_2 and U_p. *md5()* is the MC generation function.

In authentication interaction phase, UE first generates MC MC_1^* based on R_1 and password u_p, and compares it with the received MC_1 to authenticate the network. After the successful authentication of the network, UE sends the generated MC MC_2^* to AA_1. MC_2^* is generated by R_2 and u_p. AA_1 compares MC_2^* with MC_2 to authenticate UE.

6 Evaluation analysis

In this section, we first analyze the performance of the proposed BDAM under different BN configurations (network scales and block size). Subsequently, we compare the two proposed authentication methods in registration

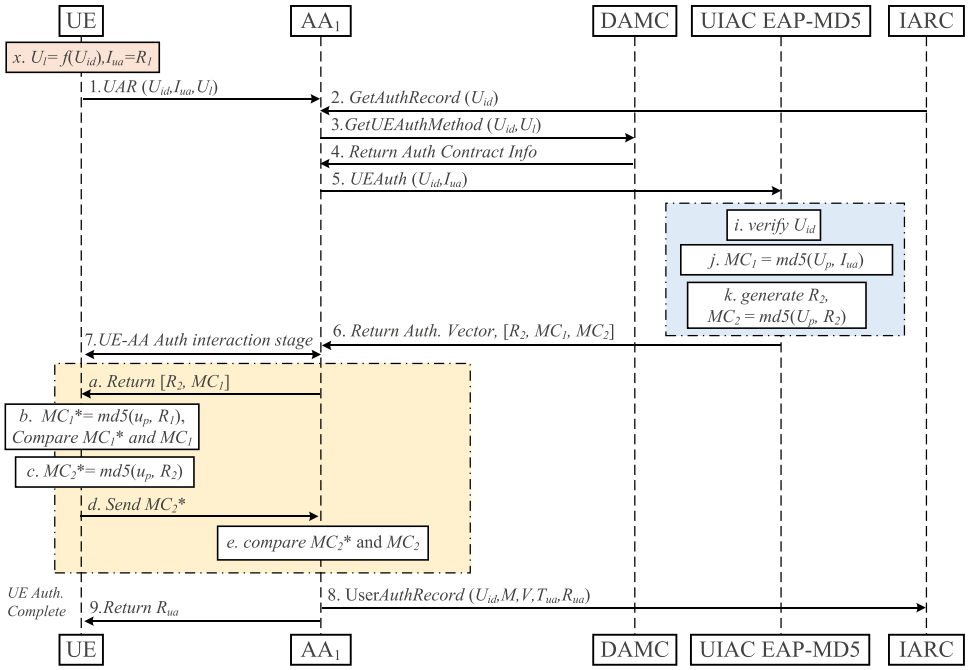

Fig. 9 EMB Authentication Protocol

and authentication processes with those in the Non-BDAM. In the end, we verify the scalability and differential authentication service capability of the proposed BDAM.

6.1 Evaluation under different network scals

Network scales refer to the number of peer nodes contained in the BN. To evaluate the performance of BDAM at different network scales, we first analyze the time to register authentication methods in BDAM for BNs of 3 Peers, 6 Peers, and 9 Peers, respectively. For better visualization of the impact of the network scale, we set the block size to 1, i.e., BN generates a new block for every 1 registration transaction submitted by the AA. As can be seen from Fig. 10, as the network scale increases, the time spent for registration in BDAM increases. Because the increased network scale increases the time for consensus generation of new blocks among blockchain nodes, which affects the authentication method registration time.

In addition, we also compare the registration time of authentication methods in Non-BDAM. As can be seen from Fig. 10, the registration time of authentication methods in Non-BDAM is slightly lower than that in BDAM. Because in BDAM, the registration of authentication methods needs to invoke DAMC to store the authentication method information for unified management, and it takes some time to establish consensus among blockchain nodes. In contrast, in Non-BDAM, the authentication method only needs to be installed on a single Peer, and consensus among Peers is not required, so it takes less time. The evaluation analysis shows that the designed BDAM enables the unified management of authentication methods with a lower increase (milliseconds) in time spent.

6.2 Evaluation under different block sizes

We also analyze the impact of different block sizes on BDAM. Block size refers to the number of transactions contained in a block. Since UE registration and authentication will generate a number of transactions, the block size has a certain impact on blockchain performance. Figure 11 illustrates the time spent to register authentication methods in BDAM for block sizes

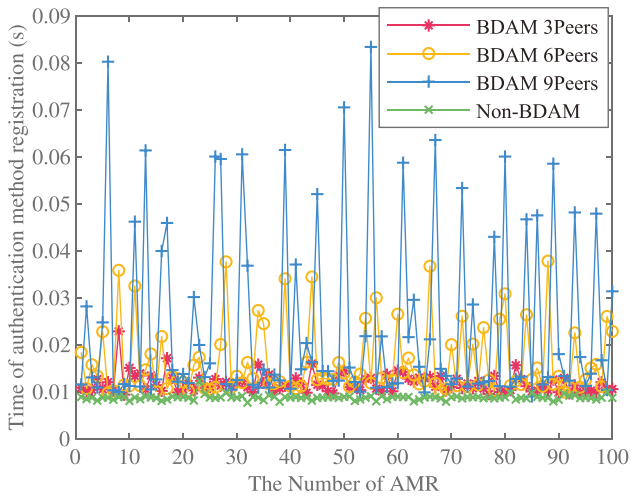

Fig. 10 The Time to Register Authentication Methods for 100 UEs under Different Network Scales

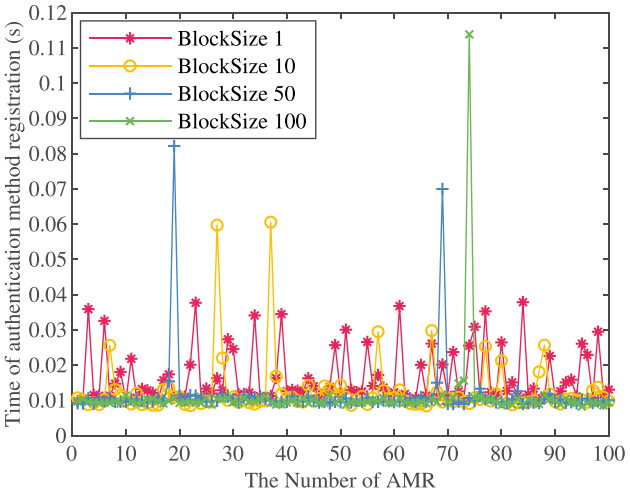

Fig. 11 The Time to Register Authentication Methods for 100 UEs under Different Block Sizes

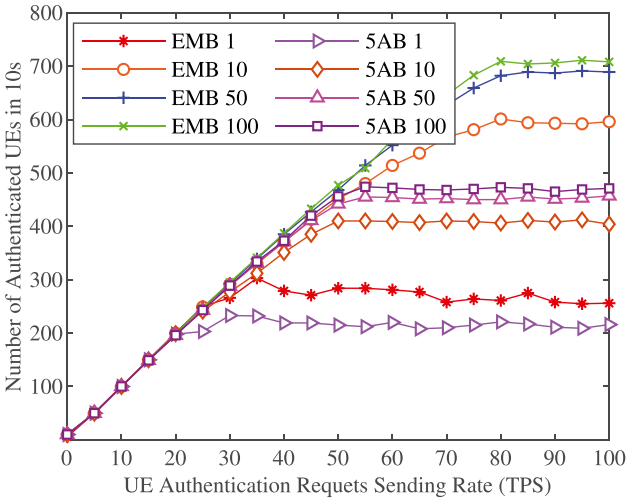

Fig. 12 The Number of Authenticated UEs at Different UAR Sending Rates within 10s

of 1, 10, 50, and 100, respectively. As shown in Fig. 11, the larger the block size, the smaller the fluctuation in registration time. Since the block size affects the consensus speed between blockchain nodes. The smaller the block size, the faster the node generates new blocks and the longer it takes on average to register the authentication method.

Figure 12 illustrates the impact of UAR sending rate (transactions per second, TPS), authentication method, and block size on the performance of BDAM in a single Peer. As can be seen from Fig. 12, as the UAR sending rate increases, the number of UE authenticated per unit time increases. After the UAR sending rate arrives at a certain amount, the number of completed authentication stabilizes, which is caused by the saturation of the number of UEs that one Peer can authenticate per unit time. Then, we evaluate the performance under two proposed authentication methods, EMB, and 5AB. As it can be seen from Fig. 12, the number of authenticated UEs of EMB is higher than that of 5AB per unit time. The difference in the number of authentication UEs depends on the complexity and security of 5AB and EMB, as can be analyzed in Figs. 8 and 9.

Besides, in Fig. 12, we can draw a conclusion consistent with Fig. 11, that is, as the block size increases, the number of completed authentication per unit time increases. The increase in the block size can increase the number of authentications per unit time, but the impact of the block size is not linear. As can be seen in the figure, when the block size is 50 and 100, the number of authenticated UE is almost the same. For this reason, the larger block size implies that the block contains more information, and the time taken by blockchain nodes to synchronize large blocks will increase. Therefore, considering the network scale and block size comprehensively, we set the blockchain scale to 6 Peers and the block size to 50 in the subsequent evaluation.

6.3 Evaluation of UE registration and authentication

Subsequently, we evaluated the time for UE Registration (UR) and UE Authentication (UA) in BDAM and Non-BDAM. Figure 13 shows UR and UA time in EMB and EAP-MD5 in Non-BDAM (EMNB), and Fig. 14 shows UR and UA time of 5AB and 5G-AKA in Non-BDAM (5ANB).

It can be seen from Figs. 13 and 14 that UR and UA time in Non-BDAM is slightly lower than that in BDAM. The average UR and UA time of EMNB is 2.924ms and 3.173ms less than that of EMB, and the average UR and UA time of 5ANB is 3.189ms and 4.31ms less than that of 5AB.

As can be seen from Figs. 13 and 14, the Non-BDAM method spends less time than the BDAM method in UR and UA processes. Considering that the deployment of BDAM can not only realize the unified management of authentication methods but also provide differentiated authentication services for different UE requirements, it is worth spending a small amount of extra time compared to Non-BDAM.

Furthermore, we also evaluate the performance of the fast authentication mechanism in the proposed BDAM. Figure 15 shows the UA time in 5AB/EMB and 5G-AKA/EAP-MD5 in BDAM-NR (5ABNR/EMBNR). As can be seen from Fig. 15, the authentication time of both 5G-AKA and EAP-MD5 methods with a rapid authentication mechanism in BDAM is lower than that without the rapid authentication mechanism. The average authentication time of 5AB is 11.16ms shorter than that of 5ABNR, and that of EMB is 8.09ms shorter than that of EMBNR. Therefore, it can be concluded that compared with the authentication method without fast authentication

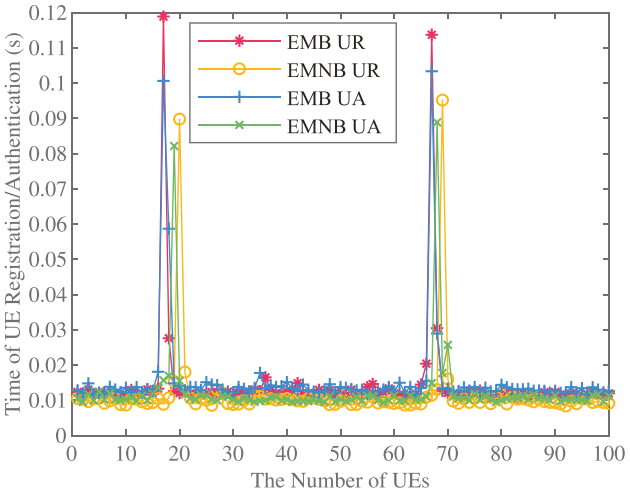

Fig. 13 UR/UA Time in EMB and EMNB by a Single UE

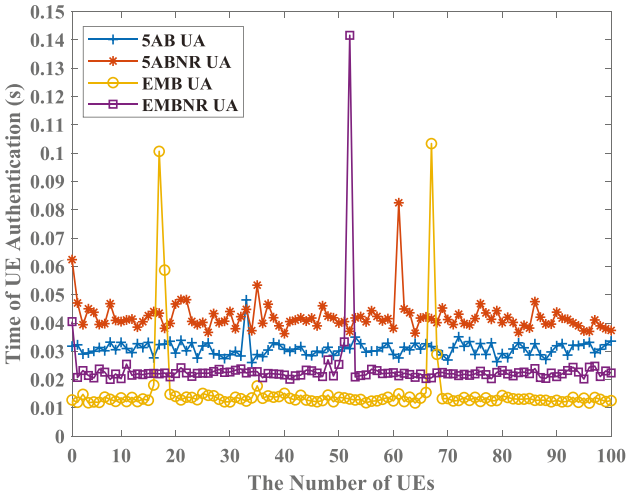

Fig. 15 UA Time in BDAM and BDAM-NR by a Single UE

mechanism, the BDAM proposed in this paper can effectively reduce the UE authentication time by reducing the interaction of authentication signaling packets.

6.4 Evalutaion of differentiated authentication and scalability

Finally, we verified the differentiated authentication capabilities and scalability of BDAM. In Non-BDAM, we deploy different authentication methods in different ADs. In AD_1, EMNB is deployed; in AD_2, 5ANB is deployed; in AD_3, both EMNB and 5ANB are deployed. In BDAM, since the authentication methods information is shared between the AAs in the three domains, both the 5AB method and EMB method are deployed on the blockchain nodes of the three ADs.

In addition, we designed the continuous UAR flow. In different periods from 0 to 280s, UE requiring different authentication services sends different UARs to AA to verify the scalability and differentiated authentication capability of BDAM. In 0-40s and 120-160s, UE sends 5G-AKA UARs; During 40-80s and 160-200s, UE sends EAP-MD5 V1 UARs; In 80-120s and 200-240s, UE sends EAP-MD5 V2 UARs. In 240-280s, UE randomly sends the above three UARs. EAP-MD5 V1 and EAP-MD5 V2 are two authentication methods based on EAP-MD5. In ACS, they are characterized as UEM V1.0 and UEM V2.0. The above two authentication methods are consistent with the authentication process except for the version number of the authentication contract and the function name in the contract.

It can be seen from Fig. 16 that in Non-BDAM, each AD can only respond to the UARs for which authentication methods have been deployed. If the authentication methods are not deployed in the domain, the expected authentication

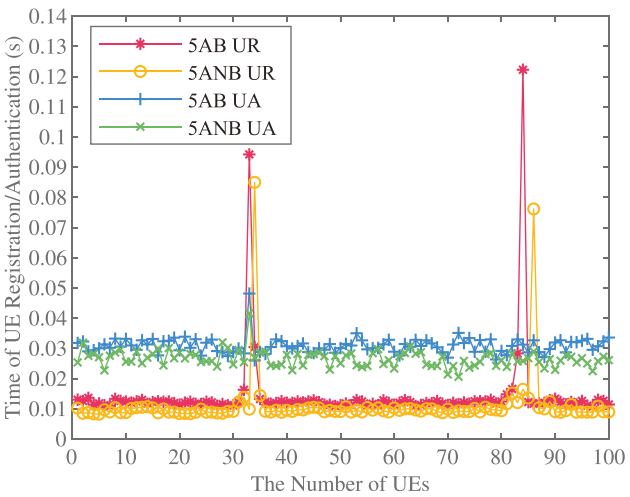

Fig. 14 UR/UA Time in 5AB and 5ANB by a Single UE

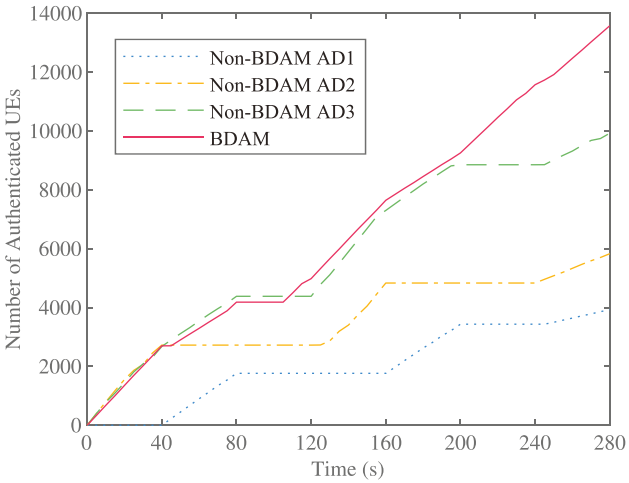

Fig. 16 The Number of Authenticated UEs in BDAM and Non-BDAM in 0-280s

service cannot be provided. For example, AD_1 can only provide EMNB V1 authentication services, and cannot provide 5ANB and EMNB V2 authentication services. In BDAM, since the authentication methods between each AD are synchronized, as long as one authentication method is deployed, other ADs can also provide corresponding authentication services. BDAM can overcome network heterogeneity, by sharing the deployed authentication methods in different heterogeneous ADs, and can provide differentiated authentication services for UEs.

To further evaluate the scalability of BDAM. In the 110s, we deploy the EMB V2 authentication method into other ADs. As can be seen in Fig. 16, within 80-110s, AA cannot provide the EMB V2 authentication services. In the time period of 110-120s and 200-240s, after the installation and deployment of the authentication contract and the synchronization of authentication information, the authentication node can respond to the EMB V2 authentication request. In contrast, compared to BDAM, Non-BDAM still cannot respond to EMNB V2 authentication requests due to the lack of updates and synchronization of the authentication methods. It can be concluded from the above evaluation that BDAM can deploy authentication methods flexibly and dynamically, realize unified management of authentication methods, and has high scalability.

7 Security analysis

In this section, we analyze the security requirements of the proposed BDAM. The main security requirements in BDAM include reliability, availability, anonymity, integrity, non-repudiation, and scalability. Subsequently, we also analyze several common attacks that BDAM can resist.

Reliability The proposed BDAM is deployed in a distributed blockchain, which can avoid the impact of a single point of failure and can effectively improve the reliability of the authentication system.

Availability The availability of BDAM is reflected in the ability to provide UEs with differentiated authentication services and improve the security capabilities of the network. BDAM needs to focus on preventing replay attacks and Denial of Service (DoS) attacks. We give the analysis for the above two attacks in the following.

Anonymity In BDAM, preventing the leakage of stored UE information is the most important aspect of enhancing the confidentiality of the system. The confidentiality in BDAM is embodied in the following two aspects: First, AA executing differentiated authentication is a trusted authentication entity authorized by the system, which can effectively prevent malicious nodes from posing as a AA to obtain authentication data; Secondly, AA node will hide UE identity when UE identity information is stored on the chain to prevent the risk of the authentication data leakage caused by the information on the chain being obtained by malicious nodes.

Integrity The integrity of BDAM is embodied in two aspects: data integrity and message integrity. In terms of data integrity, the BDAM is deployed on the blockchain, and unauthorized devices cannot join the blockchain network to obtain UE's authentication data; in terms of message integrity, the registration (or authentication) transaction carries the signature of AA, and only transactions with the correct signature verification can be published in the blockchain.

Non-Repudiation The authentication method deployment, authentication information update, and other operations in BDAM are stored in the blockchain in the form of transactions, and once the transactions are published by the blockchain nodes, they cannot be tampered with and have non-repudiation.

Scalability BDAM is scalable. On the one hand, the designed BDAM is applicable to different HetNets, and the authentication methods can be dynamically deployed for different access domains; on the other hand, the authentication methods are deployed in the form of smart contracts, and the authentication methods can be dynamically adjusted according to the requirements of the network, which is highly scalable.

Message Replay Attack The differential authentication communication process can be divided into two parts: UE-AA and AA-BN. The interaction between AA and BN is carried out through the interface of the smart contract, and the registration and authentication process is carried out in the blockchain, so there is no message replay attack; in the process of interaction between UE and AA, the random numbers and timestamps can effectively resist message replay attack.

DoS Attack The proposed BDAM is constructed based on blockchain. The distributed architecture is more flexible and redundant than the centralized architecture, which can effectively avoid the situation that DoS attacks lead to the inability to provide differentiated authentication services to UEs. On the other hand, the blockchain nodes that join the blockchain network are pre-authorized, and unauthorized nodes cannot send a large number of transaction requests to overload the blockchain, which is one of the ways to

effectively resist DoS attacks. In addition, we stored the registration and authentication record on DAMC and IARC, so that the service of the same UE who registers and authenticates multiple times within a short period will be denied, which also resists the DoS attack to a certain extent.

MITM Attack The differential authentication method proposed is established after the completion of the key negotiation between UE and AA. After the key negotiation is completed, the interaction messages between UE and AA are encrypted with the negotiated key, so the information leakage caused by the Man-In-The-Middle (MITM) attack is not discussed in this paper. In addition, in the process of differentiated authentication, the interaction messages between UEs and AAs are signed by their private keys, and if there is a malicious middleman to tamper with the interaction messages, the receiver cannot verify the signed messages, thus effectively preventing information tampering caused by a MITM attack.

8 Conclusion

In this paper, we have proposed a BDAM to efficiently authenticate user identities. The proposed mechanism can dynamically provide differentiated authentication services for different user requirements. We have implemented the proposed mechanism in the prototype system and evaluated its performance compared to the Non-BDAM method. Evaluation has demonstrated the advantages of the proposed mechanism, which can realize flexible and dynamic deployment of authentication methods with low (milliseconds level) additional time cost.

In future work, based on the existing authentication framework, we will verify and improve BDAM in different actual network scenarios to further optimize the performance of user authentication. In addition, we will also further investigate how to combine new technologies such as artificial intelligence and digital twin with the proposed BDAM to achieve intelligent authentication.

Acknowledgements The authors would like to thank the reviewers for their valuable comments.

Author contribution The manuscript was written entirely by the authors. All authors made an equal contribution to the development of the paper.

Funding This paper was supported in part by the National Key R &D Program of China under Grant No. 2018YFA0701604, and in part by the Fundamental Research Funds for the Central Universities under Grant No. 2021YJS012, No. 2021YJS008.

Data availability All data generated or analyzed during this study are included in this published article.

Declarations

Ethical approval and consent to participate Not applicable.

Human and animal ethics Not applicable.

Consent for publication Not applicable.

Competing interests The authors declare that they have no competing interests.

References

1. Nguyen DC, Ding M, Pathirana PN, Seneviratne A, Li J, Niyato D, Dobre O, Poor HV (2022) 6G internet of things: a comprehensive survey. IEEE Internet Things J 9(1):359–383
2. Giordani M, Polese M, Mezzavilla M, Rangan S, Zorzi M (2020) Toward 6G networks: Use cases and technologies. IEEE Commun Mag 58(3):55–61
3. Zhu X, Jiang C (2020) Integrated satellite-terrestrial networks toward 6G: Architectures, applications, and challenges. IEEE Internet Things J 9(1):437–461
4. Zhang Z, Xiao Y, Ma Z, Xiao M, Ding Z, Lei X, Karagiannidis G, Fan P (2019) 6G wireless networks: Vision, requirements, architecture, and key technologies. IEEE Veh Technol Mag 14(3):28–41
5. El-Hajj M, Fadlallah A, Chamoun M, Serhrouchni A (2019) A survey of internet of things (IoT) authentication schemes. Sensors 19(5):1141
6. Wang W, Han Z, Alazab M, Gadekallu TR, Zhou X, Su C (2022) Ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps. IEEE Trans Ind Appl 58(5):5616–5623
7. Yang X, Yang X, Yi X, Khalil I, Zhou X, He D, Huang X, Nepal S (2021) Blockchain-based secure and lightweight authentication for internet of things. IEEE Internet Things J 9(5):3321–3332
8. NakamotoS (2008) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf
9. Dai HN, Zheng Z, Zhang Y (2019) Blockchain for internet of things: a survey. IEEE Internet Things J 6(5):8076–8094
10. Mohanta BK, Jena D, Ramasubbareddy S, Daneshmand M, Gandomi AH (2020) Addressing security and privacy issues of IoT using blockchain technology. IEEE Internet Things J 8(2):881–888
11. Khowaja SA, Khuwaja P, Dev K, Lee IH, Khan WU, Wang W, Qureshi NMF, Magarini M (2022) A secure data sharing scheme in community segmented vehicular social networks for 6G. IEEE Trans Ind Inf 19(1):890–899
12. Wang W, Yang Y, Yin Z, Dev K, Zhou X, Li X, Qureshi NMF, Su C (2022) BSIF: Blockchain-based secure, interactive, and fair mobile crowdsensing. IEEE J Sel Areas Comm 40(12):3452–3469

13. Wang J, Ling X, Le Y, Huang Y, You X (2021) Blockchain-enabled wireless communications: a new paradigm towards 6G. Natl Sci Rev 8(9):nwab069
14. Mohanta BK, Jena D, Panda SS, Sobhanayak S (2019) Blockchain technology: a survey on applications and security privacy challenges. Internet Things 8:100107
15. Pohrmen FH, Das RK, Saha G (2019) Blockchain-based security aspects in heterogeneous internet-of-things networks: a survey. Trans Emerg Telecommun Technol 30(10):e3741
16. Shen M, Liu H, Zhu L, Xu K, Yu H, Du X, Guizani M (2020) Blockchain-assisted secure device authentication for cross-domain industrial IoT. IEEE J Sel Areas Comm 38(5):942–954
17. Xiong H, Wu Y, Jin C, Kumari S (2020) Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT. IEEE Internet Things J 7(12):11713–11724
18. Cui Q, Zhu Z, Ni W, Tao X, Zhang P (2021) Edge-intelligence-empowered, unified authentication and trust evaluation for heterogeneous beyond 5G systems. IEEE Wirel Commun 28(2):78–85
19. Cao L, Liu Y, Cao S (2019) An authentication protocol in LTE-WLAN heterogeneous converged network based on certificateless signcryption scheme with identity privacy protection. IEEE Access 7:139001–139012
20. Liu J, Ren A, Zhang L, Sun R, Du X, Guizani M (2019) A novel secure authentication scheme for heterogeneous internet of things. In: Proceedings of 2019 IEEE International Conference on Communications (ICC), pp 1–6
21. Alezabi KA, Hashim F, Hashim SJ, Ali BM (2020) Jamalipour A (2020) Efficient authentication and re-authentication protocols for 4G/5G heterogeneous networks. EURASIP J Wirel Comm 1:1–34
22. Athmani S, Bilami A, Boubiche DE (2019) EDAK: an efficient dynamic authentication and key management mechanism for heterogeneous WSNS. Future Gener Comput Syst 92:789–799
23. Lei A, Cruickshank H, Cao Y, Asuquo P, Ogah CPA, Sun Z (2017) Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. IEEE Internet Things J 4(6):1832–1843
24. Tan Y, Liu J, Kato N (2020) Blockchain-based key management for heterogeneous flying Ad-Hoc network. IEEE Trans Industr Inform 17(11):7629–7638
25. Zhang S, Cao Y, Ning Z, Xue F, Cao D, Yang Y (2020) A heterogeneous IoT node authentication scheme based on hybrid blockchain and trust value. KSII Trans Internet Inf 14(9):3615–3638
26. Khalid U, Asim M, Baker T, Hung PC, Tariq MA, Rafferty L (2020) A decentralized lightweight blockchain-based authentication mechanism for IoT systems. Cluster Comput 23(3):2067–2087
27. Panda SS, Jena D, Mohanta BK, Ramasubbareddy S, Daneshmand M, Gandomi AH (2021) Authentication and key management in distributed IoT using blockchain technology. IEEE Internet Things J 8(16):12947–12954
28. Lin W, Zhang X, Cui Q, Zhang Z (2021) Blockchain based unified authentication with zero-knowledge proof in heterogeneous MEC. In: Proceedings of 2021 IEEE International Conference on Communications Workshops (ICC Workshops), pp 1–6
29. Shi N, Tan L, Li W, Qi X, Yu K (2021) A blockchain-empowered AAA scheme in the large-scale HetNet. Digit Commun Netw 7(3):308–316
30. Zhang H, Chen X, Lan X, Jin H, Cao Q (2020) BTCAS: a blockchain-based thoroughly cross-domain authentication scheme. J Inf Secur Appl 55:102538
31. Luo Y, Li H, Ma R, Guo Z (2021) A composable multifactor identity authentication and authorization scheme for 5G services. Secur Commun Netw 2021:6697155
32. Szabo N (1997) Formalizing and securing relationships on public networks. First Monday 2(9)
33. Li Z, Kang J, Yu R, Ye D, Deng Q, Zhang Y (2017) Consortium blockchain for secure energy trading in industrial internet of things. IEEE Trans Ind Inf 14(8):3690–3700
34. Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, Caro AD, Enyeart D, Ferris C, Laventman G, Manevich Y, Muralidharan S, Murthy C, Nguyen B, Sethi M, Singh G, Smith K, Sorniotti A, Stathakopoulou C, Vukolic M, Cocco SW, Yellick J (2018) Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, pp 30:1–30:15
35. Hojjati M, Shafieinejad A, Yanikomeroglu H (2020) A blockchain-based authentication and key agreement (AKA) protocol for 5G networks. IEEE Access 8:216461–216476
36. Tu Z, Zhou H, Li K, Song H, Wang W (2021) A blockchain-based user identity authentication method for 5G. In: Proceedings of International Symposium on Mobile Internet Security, pp 335–351

Zhe Tu received the B.S. degree in telecommunications engineering from Beijing Jiaotong University (BJTU), China, in 2016, where he is currently pursuing the Ph.D. degree in information and communication engineering. He joined the National Engineering Research Center for Mobile Private Networks, BJTU. His main research interests include the architecture of next-generation internet, network service management, and network security. His other research interests include satellite networks and artificial intelligence.

Huachun Zhou received the B.S. degree from the People's Police Officer University of China, in 1986, and the M.S. degree in telecommunication automation and the Ph.D. degree in telecommunications and information system from Beijing Jiaotong University (BJTU), in 1989 and 2008, respectively. In 1994, he joined the Institute of Automation Systems, BJTU, where he is currently a Lecturer. From 1999 to 2009, he was a Senior Engineer with the School of Electronics and Information Engineering, BJTU, and with the Network Management Research Center, BJTU. Since 2009, he has been a Professor with the National Engineering Research Center for Mobile Private Networks, BJTU. He has authored over 40 peer-reviewed articles. He holds 17 patents. His main research interests include the area of mobility management, mobile, secure computing, routing protocols, network management, and satellite networks.

Kun Li received the B.S. degree in telecommunications engineering from Beijing Jiaotong University (BJTU), China, in 2018, where he is currently pursuing the Ph.D. degree in information and communication engineering. He joined the National Engineering Research Center for Mobile Private Networks, BJTU. His main research interests include the architecture of next-generation internet, network service management, network security, satellite networks, and mobile internet.

Haoxiang Song received his B.S. degree from Beijing Jiaotong University in June 2020. He joined the National Engineering Research Center for Mobile Private Networks, BJTU. His main research is directed to blockchain and trusted protocol.

Wei Quan received the Ph.D. degree in communication and information system from Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing. He has published more than 20 papers in prestigious international journals and conferences, including IEEE Communications Magazine, IEEE WIRELESS COMMUNICATIONS, IEEE NETWORK, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS LETTERS, IFIP Networking, IEEE ICC, and IEEE GLOBECOM. His research interests include key technologies for network analytics, future Internet, 5G networks, and vehicular networks. Dr. Quan is a TPC Member of IEEE ICC in 2017 and 2018, IEEE INFOCOM (NewIP Workshop) in 2020, ACM MOBIMEDIA in 2015, 2016, and 2017, and IEEE CCIS in 2015 and 2016. He serves as an Associate Editor for the Journal of Internet Technology, Peer-to-Peer Networking and Applications, and IEEE ACCESS, and as a technical reviewer for many important international journals. He is also a member of ACM and a Senior Member of the Chinese Association of Artificial Intelligence.