Hindawi
Security and Communication Networks
Volume 2018, Article ID 7202806, 14 pages
https://doi.org/10.1155/2018/7202806

Research Article

BAVP: Blockchain-Based Access Verification Protocol in LEO Constellation Using IBE Keys

Songjie Wei[1], **Shuai Li**[2], **Peilong Liu**[3] and **Meilin Liu**[4]

[1]*School of Computer Science and Engineering, Nanjing University of Science & Technology and State Key Laboratory of Air Traffic Management System and Technology, Nanjing 210094, China*
[2]*School of Computer Science and Engineering, Nanjing University of Science & Technology, Nanjing 210094, China*
[3]*Shanghai Engineering Center for Microsatellites, Shanghai 201203, China*
[4]*Shanghai Institute of Satellite Engineering, Shanghai 200240, China*

Correspondence should be addressed to Shuai Li; 116106000732@njust.edu.cn

Received 28 December 2017; Accepted 5 April 2018; Published 14 May 2018

Academic Editor: Guojun Wang

LEO constellation has received intensive research attention in the field of satellite communication. The existing centralized authentication protocols traditionally used for MEO/GEO satellite networks cannot accommodate LEO satellites with frequent user connection switching. This paper proposes a fast and efficient access verification protocol named BAVP by combining identity-based encryption and blockchain technology. Two different key management schemes with IBE and blockchain, respectively, are investigated, which further enhance the authentication reliability and efficiency in LEO constellation. Experiments on OPNET simulation platform evaluate and demonstrate the effectiveness, reliability, and fast-switching efficiency of the proposed protocol. For LEO networks, BAVP surpasses the well-known existing solutions with significant advantages in both performance and scalability which are supported by theoretical analysis and simulation results.

1. Introduction

This paper is based on the conference paper [1]. Low-Earth-Orbit (LEO) satellite network systems as represented by the Iridium system and Globalstar system have become one of the most heated areas of research. Because of the low orbits, LEO networks have the advantages of short delay and low path-loss compared with traditional satellite networks. In addition, a constellation of multiple satellites in a LEO satellite network system brings true global coverage and efficient frequency reuse. LEO satellite systems play an important role in mobile satellite communications and are supposed to be one of the most important components in future global communications.

Due to the open nature of satellite networks, communications can be easily intercepted by unauthorized or malicious attackers. Mechanisms for ensuring secure communication within satellite networks are key for achieving security within satellite network systems. In these communications systems, the use of encryption algorithms to maintain confidentiality is a common and effective method. There is significant difference between satellite networks and terrestrial networks in many respects, such as computing capability, storage space, high packet loss rate, and dynamic topology. Consequently, the terrestrial authentication protocols represented by a series of protocols with certificates are less applicable in such scenarios with satellites. On the other hand, the existing public key infrastructure (PKI) must ensure dependability of a third party such as a certificate authority (CA) in general. Certificates and key management overhead are not negligible. Thus, when considering the design of authentication protocols, we ensure secure communication with concern about computation and storage overhead and the number of steps and nodes involved.

Unlike traditional satellite networks, LEO satellite networks have the characteristics of dynamic topology and frequent connection switching. The authentication protocol running on satellite nodes has to be as light-weighted and cost-effective as possible in premise of ensuring security. This means cryptography used in authentication has to be

carefully selected and customized for satellites onboard. A short response time during authentication is also preferred. However, there are a lot of concerns within the centralized authentication protocols in satellite network, such as complex computation, central bottleneck, and long response time, which make the above desires not easily achievable. This paper proposes a Blockchain-based Access Verification Protocol (BAVP) by combining identity-based encryption (IBE) and decentralized blockchain technology. IBE brings in the advantage of fast key generation with specified identity string provided by users, which eliminates the cost of certificates used in traditional authentication protocols. Blockchain contributes to the decentralizing of both data storage and computation.

2. Related Work

Regarding the related literature on centralized authentication protocols used in existing satellite network, Cruickshank proposes an authentication protocol [2] that uses asymmetrical encryption algorithms. However, the operations involved in his protocol are too complicated to implement. Hwang et al. redesign the authentication protocol without a public key cryptosystem [3], but the shared secret key still needs to be updated every time when a user is authenticated. Y. F. Chang and C. C. Chang propose a mutual authentication protocol that requires only XOR and hash function [4], where, during every authentication procedure, a network control center (NCC) need not generate a private key and a temporary identity for user. However, the NCC is involved in every authentication session as critical bottleneck and single-point-of-failure resource which may bring in higher delay during authentication. The performance of the authentication protocol is restricted by NCC. Zheng et al. propose an authentication protocol avoiding these weaknesses by involving a gateway in authentication [5]. Their proposed protocol involves not only users and satellites but also the gateway and NCC during authentication. The number of interactive steps is inflated resulting in a variant response time of authentication. Lin's paper compares and summarizes the characteristics of symmetric encryption, asymmetric encryption, and the certificate system used in satellite network [6].

Additionally, traditional centralized authentication protocols are designed mainly for MEO (Medium Earth Orbit) and GEO (Geosynchronous Earth Orbit). There is less consideration on distributed handover authentication which is unavoidable in LEO satellite networks with frequent link switching and narrow single-satellite coverage. By simply applying the existing centralized authentication protocols in LEO satellite networks, each handover authentication in a LEO satellite network requires a new complete authentication. This magnifies the disadvantages with these protocols discussed above and thus is inappropriate for LEO satellite networks.

There are several schemes focusing on LEO satellite network as noted in papers [4, 7, 8]. In paper [7], the author proposes an efficient and secure anonymous authentication scheme that requires only XOR and hash function and improves the disadvantages such as user's privacy not being kept confidential compared to paper [4]. However, it still has the NCC bottleneck during authentication. Wu et al. propose a lightweight authentication and key agreement (AKA) scheme [8] based on the synchronization mechanism of user's temporary identity which fixes the security problems found in paper [9]. All these papers utilize the XOR and hash function for efficient computation, but none of them is optimized for LEO satellite network with NCC still involved.

In summary, PKI is still the most fundamental for implementing key management and not appropriate for LEO with resource constraint. In addition, referring to decentralized authentication protocol used in satellite network, previous researches are relatively lacking. In other resource constraint scenarios similar to satellite networks, such as wireless sensor networks, the authentication protocols investigated intensively focus on mainly cluster and mostly centralized ones.

3. Protocol Design

In the proposed Blockchain-based Access Verification Protocol (BAVP) for LEO authentication, Key Generation Center (KGC) generates public and private keys of all roles (users and satellites) with its private key and these roles' identities. Meanwhile, based on blockchain, a trust chain consisting of KGC, satellites, and users is the core base for rapid handover authentication. With distributed storage in blockchain, this protocol records users' registration, cancellation, login, logout, handover, and other related logs as plugin.

Authentication is divided into two parts: access authentication and handover authentication. During access verification, users and satellites can implement mutual authentication through their public and private keys, and a user's authority is checked against his token. Meanwhile, the relevant authentication logs are recorded in a form of blocks which would be merged and distributed between satellites and the KGC. We describe the logical structure of this system as in Figure 1. A satellite in each orbit is selected as a logical root responsible for the interaction of blocks with KGC. This logical structure is also the basis of blocks' merging and distribution. Before presenting the detailed design, we first briefly review IBE and blockchain technology as background knowledge.

3.1. Background

3.1.1. Identity-Based Encryption. In IBE [10], a user's public key can be derived directly using his unique identity string, such as a phone number and email. IBE eliminates the computation and storage overhead with certificates. In this way, we can create the mapping between identity and public key.

IBE requires a trusted third-party KGC to provide key generation services for different roles in this system. When registering, a user needs to provide his identity to the KGC; then the KGC uses its private and public key together with related system parameters to calculate a pair of public and private keys for this user and also securely transmit them to the user. When sending confidential information, a user needs no certificate but the public key which corresponds

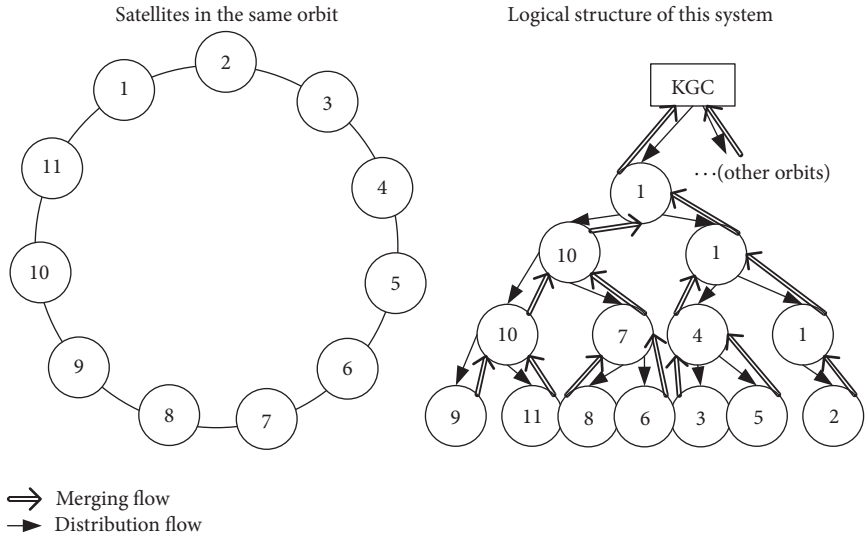

Figure 1: Logical structure of this system.

to the receiver's identity in order to encrypt messages before sending.

The most efficient IBE schemes are based on bilinear pairings of elliptic curves, and currently IBE based on pairing is mainly divided into three categories: exponent inversion, full domain hash, and commutative blinding. The full-domain-hash mechanism requires much computation for the mapping between user's identity and a point on elliptic curve, which is not suitable for resource constrained scenarios such as satellite networks. Thus, in a scenario with limited computing power like in satellite networks, the other two schemes are more suitable to be adopted.

3.1.2. Blockchain. Blockchain [11] is the underlying technology that supports Bitcoin. It is essentially a distributed ledger secured by cryptography. Its core strength is that trust is built among distributed nodes and data ensured for integrity without being tampered or forged. Furthermore, blockchain supports customization with smart contracts according to diverse demands.

Data integrity and distributed consensus on trust are the two main advantages of blockchain. The former is guaranteed when each node in the network stores a complete copy of data. And the latter primarily depends on the effectiveness of consensus mechanism with no need for Trusted Third Party (TTP) among nodes. According to different scenarios, blockchain can be classified into three types, namely, public blockchain, private blockchain, and consortium blockchain. The major differences are found in their adopted consensus mechanisms. In the case of LEO satellite network system, consortium blockchain would be more suitable in terms of architecture and various demands like being controllable and manageable. Fabric (a consortium blockchain platform) supported by Hyperledger (a global open source collaboration hosted by the Linux Foundation) is a representation of consortium blockchain with a modular architecture delivering high degrees of confidentiality, resiliency, flexibility, and scalability. Additionally, there are also some new blockchain technologies emerging like IOTA which takes directed acyclic graph (DAG) instead of linked list as its underlying architecture. Generally speaking, the most popular public blockchain platforms are still Bitcoin and Ethereum. Blockchain technology is still under continuous development and evolution.

3.1.3. Smart Contract. Smart contract is a program protocol intended to verify, facilitate, or enforce the performance of a contract. In this paper, smart contract refers particularly to a contract program running on blockchain as the greatest achievement in blockchain 2.0. Taking Ethereum as an example, smart contract is implemented by EVM (Ethereum Virtual Machine) which is Turing-complete. When a smart contract being programmed by solidity or other smart contract programing languages and deployed on blockchain, it is encoded as EVM bytecode and executed by all mining full nodes. Full node refers to those with a complete copy of data of the blockchain while light node refers to nodes with only partial data in the blockchain.

Due to its programmability, atomicity, consistency, and unambiguity, smart contract contributes greatly to blockchain technology. Users can verify the correctness of smart contract by comparing the bytecode of source code provided by promulgator with the bytecode stored in blockchain. And access control is supported based on accounts within smart contract. Accordingly, smart contract can implement specific business logic on blockchain which makes blockchain more promising and practical in various applications.

3.2. BAVP Principles and Processes. BAVP has two major parts: key management implemented with IBE; authentication and records of related logging which is based on both blockchain and IBE. In describing this protocol, we use the symbolic conventions as shown in Table 1.

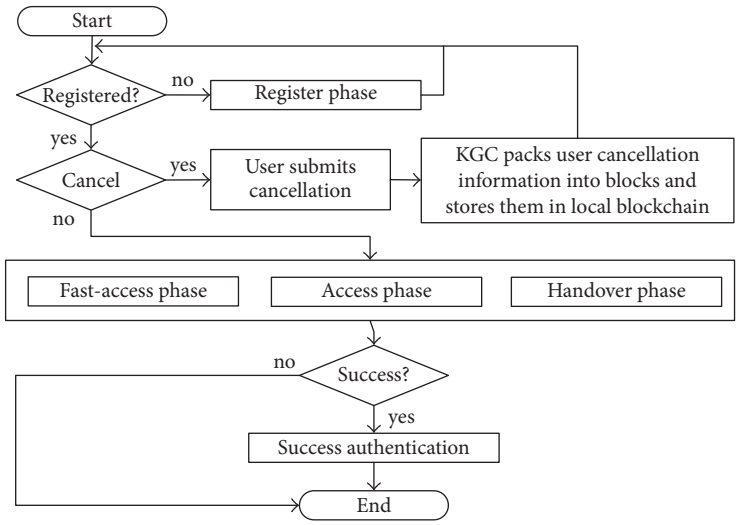

Figure 2: Authentication control flow of the proposed protocol.

Table 1: Symbols and meanings.

Symbol	Meaning
ID_A	User A's ID
ID_S	Satellite S's ID
P_A	User A's public key
P_S	Satellite's public key
P_{KGC}	KGC's public key
d_A	User A's private key
d_S	Satellite S's private key
d_{KGC}	KGC's private key
$Encry_x()$	Encryption with x as key
Stime	Time of handover
U_authority	User's authority
Start_time	Authority's beginning time
Stop_time	Authority's ending time
XX_Sign	XX's signature
$Auth_{Token}$	User's authorization token
UserInfo	User's related info
Service	Service that user applies for
result	Result of authentication
$Sign_x()$	Signing with x as key
Splace	Place of handover

When explaining the principles of each phase, all messages included in this protocol are timestamped by default, and nodes receiving the messages always check the timestamp. The BAVP control procedure is shown in Figure 2.

3.2.1. Registration Phase. A KGC is a trusted authority which is responsible for calculating a user's public key, private key, and user token for authority. A registered user is allowed to access the satellite system at any time during the token's period of validity. An authorized user submits his identity to KGC to obtain a pair of public and private keys calculated by the KGC, together with a token signed by the KGC.

The calculation is as follows: user A provides his identity ID_A (such as *user*: *Alice@gmail.com*, where *user* means the role of user). KGC uses hash function and P_{KGC} to calculate P_A. Next, the KGC calculates d_A with d_{KGC}. Satellites register themselves in the same way before issuance.

Meanwhile, KGC constructs user token of A and signs it with d_{KGC}. And $ID_A \parallel U\_authority \parallel Start\_time \parallel Stop\_time \parallel KGC\_Sign$ is the format of $Auth_{Token}$ where *KGC_Sign* means signature of the first four fields in this token. After finishing, KGC returns the pair of public and private keys, along with the token, to user A safely (e.g., via secure email). Afterwards, KGC packs this user's registration log into blocks which would be stored in local blockchain. At this point, user A has completed the steps necessary for accessing the satellite system. The diagram of the registration phase is shown in Figure 3.

3.2.2. Access Authentication Phase. The access authentication phase is shown in Figure 4, and the four steps are as follows:

(a) When user A wishes to access satellite S, he first checks the identity of S and then uses the hash function to calculate P_S with P_{KGC}. Afterwards, A sends his identity to S.

(b) While receiving this message, S checks the identity of A and searches for latest cancellations to check the validity of A. Then S calculates P_A accordingly, generates random number r together with session key k, and sends $m1$ to A as follows:

$$m1 = Encry_{P_A}\left(r, k, timestamp, Sign_{d_S}\left(r, k, timestamp\right)\right). \quad (1)$$

(c) After receiving this message, A decrypts it with d_A, verifies the signature of r and k, and then saves them. Thereafter, A sends $m2$ to S as follows:

$$m2 = Encry_k\left(r, Auth_{Token}, Service, UserInfo\right). \quad (2)$$

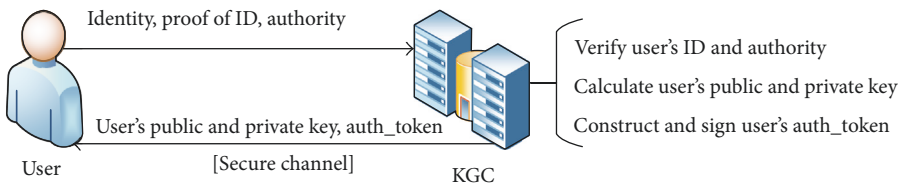

Figure 3: Diagram of registration phase.

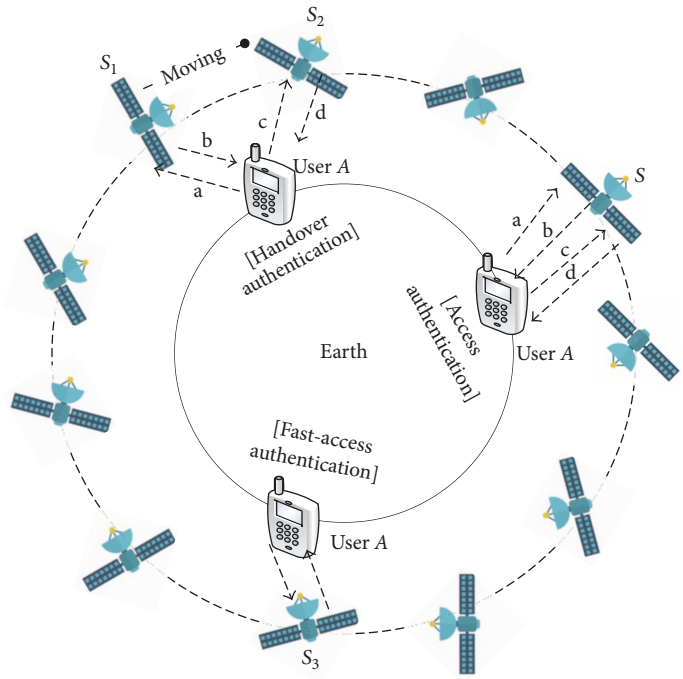

Figure 4: Diagram of the protocol.

UserInfo contains the location, time, and *A*'s identity when authentication starts.

(d) While receiving this message, *S* decrypts it with *k*, verifies the correctness of *r*, and searches for the latest cancellations to verify the validity of current user. If *A* is valid, then *S* verifies the signature of *A*'s $Auth_{Token}$ with P_{KGC}. The session key *k* uses symmetric encryption, such as the Rijndael algorithm. Next, *S* checks whether ID_A in $Auth_{Token}$ is consistent with the identity provided at the beginning or not.

With all the steps above without mistakes, *S* allocates the resources necessary to establish a secure connection with *A* and provides service according to the authority and expiration time in $Auth_{Token}$. Moreover, *S* packs *A*'s access log which contains *UserInfo* mainly and then stores it into *S*'s local blockchain. Otherwise, *S* disconnects from *A*.

The essence of this phase is to accomplish mutual authentication by IBE. A user needs not store the public key of each satellite in advance. Instead, only through the broadcasted network identification of a satellite, each user can calculate the corresponding public key directly. Authentication security is ensured of IBE. During authentication, a session key is negotiated, and a secure channel is established after each successful authentication.

3.2.3. Fast-Access Authentication Phase. Once a new user is successfully authenticated, his information would be stored in the access satellite. With data traceability in blockchain, when this user reconnects a satellite for service again, he only needs to send *m*3 to the satellite, calculated as

$$m3 = ID_A, Service, ID_{S_3}, timestamp, \\ Sign_{d_A}\left(ID_A, Service, ID_{S_3}, timestamp\right). \quad (3)$$

S_3 stands for the satellite user *A* wants to access. Next, after receiving this message, S_3 calculates P_A according to ID_A, verifies the signature, and then checks if ID_{S_3} in this message corresponds to its own. If there is no mistake, then the satellite can search for data related to *A* in its local blockchain, return a new session key which is signed with d_{S_3} and encrypted with P_A, and provide relevant service according to the relevant data.

Using this procedure, users can access satellites efficiently. The search time is $\log_2(n)$. However, if the satellite being

Table 2: Comparisons in authentication phase.

	Yoon et al.	BAVP: access	BAVP: fast-access	BAVP: handover
Hash operations	2/4	(1)/1	(1)/1	-/-/2
MAC operations	2/2	-	-	-
Symmetric operations	-	1/1	-/-	-/-/-
Asymmetric operations	-	1/1	1/1	-/-/-1
Signing operations	-	-/1	1/1	1/1/1
Signature verifications	-	1/1	1/1	1/-/2
Communication levels	2	1	1	1
Authentication center	NCC	None	None	None

( ) means only needed for the first time while users can cache satellite public key afterwards; x/y: x means the side of user; y means the side of satellite; $x/y/z$: x means the side of user; y means the side of first satellite; z means the side of second satellite.

accessed is not in the same orbit as the original satellite where the user is previously authenticated, then the user cannot take this fast-access way due to the lack of related data in this current satellite. User who needs the fast-access convenience should access at least one satellite in each orbit previously through regular access authentication procedure.

3.2.4. Handover Authentication Phase. The handover authentication phase is illustrated in Figure 4, and the four steps are explained as follows:

(a) Through the secure channel, user A informs the satellite (called S_1) of his leaving information including ID_A and ID_{S_2}.

(b) While S_1 receives such messages from A, it checks whether the satellite that A wants to switch to is a neighbor or not. For neighbor, S_1 will pack A's handover log as $(Stime, Splace, Service, ID_{S_1}, ID_{S_2}, ID_A)$ into block and store this in its local blockchain. The handover log can also be extended according to user needs. Instantly, S_1 calculates and returns $m4$ to A as

$$m4 = ID_{S_1}, ID_{S_2}, ID_A, timestamp, Service, \\ Sign_{d_{S_1}}\left(ID_{S_1}, ID_{S_2}, ID_A, timestamp, Service\right). \quad (4)$$

(c) After receiving $m4$, A disconnects from S_1, signs this message, and sends it to S_2.

(d) Subsequently, S_2 checks the timestamp of the message received from A and also checks out whether S_1 is its neighbor. If not, S_2 denies A's request. Otherwise, S_2 calculates P_{S_1} and P_A to verify the signature in this message. When verification succeeds, S_2 searches for the latest cancellations to check the validity of A. If A is valid, S_2 returns a new session key signed with d_{S_2} and encrypted with P_A to A. Later, S_2 officially allocates relevant resources and establishes secure connection with A by this new session key. Meanwhile, S_2 packs A's handover log which depends on packing the received message mainly into blocks and stores this in its local blockchain.

Next, A decrypts the message received from S_2 with d_A. Then, A verifies the new session key's signature and continues to obtain service through new secure channel between him and S_2. If any step goes wrong, S_2 disconnects from A.

The core principle of implementing fast handover is its utilization of a trust chain consisting of satellites, users, and KGC. This also brings in consensus among all satellites. When a user is successfully authenticated by passing the check on one satellite in this system, other satellites should recognize the result of authentication as trust.

When it is time to synchronize data (depending on the update interval), each satellite sends its own latest blocks (i.e., blocks that have not been sent out) to adjacent nodes according to the logical organization of the constellation. KGC or satellites would merge these blocks received from other nodes with their own blockchain on the basis of timestamp. If the amount of data at satellite side reaches the threshold, each satellite removes those blocks in accordance with predefined rules, to keep only the latest and mostly queried records.

When a user cancels his identity, KGC packs the user's cancellation record into a block and stores the block in its local blockchain database. Blocks containing the newest cancellation records are periodically or proactively synchronized with P2P distribution as in a typical blockchain.

Regardless of merging or distribution, once a node receives blocks, it verifies the signature of blocks and then integrates these blocks with its local blockchain. The block structure in this protocol is consistent with blockchain. As for re-registration, a user should cancel his original identity and register with a new identity in the same way described in registration phase.

3.3. Performance and Advantages Analysis. As a theoretical analysis of the computational costs required in this proposed protocol, taking symmetric encryption/decryption as P, asymmetric encryption/decryption with IBE as E, signing as N, and signature verification as V, the access authentication phase requires $R_a(2P, 2E, 1N, \text{and } 2V)$. The fast-access authentication phase costs only $R_f(2E, 2N, \text{and } 2V)$, and the handover authentication phase needs $R_h(1E, 3N, \text{and } 3V)$. A comparison of authentication methods between Yoon et al.'s scheme and the protocol proposed in this paper is shown in Table 2.

Since Yoon et al.'s scheme [7] is far superior to those proposed in related works as shown in their paper, Table 2 shows the comparison between Yoon's protocol and the proposed BAVP. As there are only hash and mac operations involved

in Yoon et al.'s proposed scheme, this protocol appears less efficient in computation costs by comparison. Nevertheless, it is not only computation costs that decide whether an authentication protocol is efficient or not. Other factors like communication levels and existence of an authentication center would also affect the efficiency of authentication protocol. As mentioned in Section 2, in Yoon et al.'s scheme, NCC is still involved in authentication which may be the bottleneck of this whole authentication system. Meanwhile, there are two communication levels (user ↔ satellite and satellite ↔ NCC) during authentication in Yoon et al.'s scheme, while there is only one communication level with users and satellites in this proposed protocol. And considering the LEO satellite network that has the least network delay (10 ms–40 ms), the forward and backward delay of the extra communication level would bring at least 20 ms for response time of authentication protocol, which is far greater than the time for one operation of asymmetric encryption/decryption (in simulation environment with IBE, it is about 1.5 ms).

From the analysis, we can conclude that BAVP has the following extra merits:

(1) With IBE, this protocol eliminates certificate cost.

(2) Using IBE and blockchain, decentralized access authentication and fast handover among satellites can be implemented.

(3) Based on the trust chain consensus, the system stores information about users and satellites using blockchain technology which ensures the accuracy, completeness, consistency, and traceability of data within the block.

(4) Auditing is also made possible for protection of network resources and the implementation of security policies by unforgeable logging in blockchain.

3.4. Security Analysis. In the case of common attacks such as data tampering, eavesdropping, replay attacks, and man-in-the-middle attacks, this protocol has intrinsic resistance.

Key Security. A malicious attacker cannot get the plaintexts from the ciphertexts obtained by eavesdropping or sniffing, as long as he cannot get the private key of any user or satellite. Attack cannot tamper with the message, which is based on the security of IBE and AES algorithms. Session security after successful authentication is ensured by session key. Session key negotiation is secured by private keys of users and satellites. As clarified in the previous four sections, private keys of users and satellites are not included directly in various authentication messages, which means these keys cannot be obtained by eavesdropping. When the KGC is credible (keeping d_{KGC} secure and not storing or calculating user private keys illegally after users registered), users and satellites private keys are only known to themselves, which means users themselves are essentially responsible for security of their private keys.

Replay Attacks. The protocol uses timestamps, which can resist such attacks effectively. If there is an attacker who copies an encrypted message by eavesdropping and sends it at another moment, the receiver satellite will reject it after validating the timestamp in the message. Moreover, the random number r during access authentication phase actually implements a challenge/response method, and also during other phases, the attacker will fail to get session key without possessing private key of the user relevant with the message he replayed; therefore, the satellite will not allocate related resources officially. Thus, this BAVP protocol is resistant to replay attacks.

Man-in-the-Middle Attacks. The man in the middle cannot register with the role of a satellite or the role of an existing user in the system. Therefore, he cannot impersonate any existing role in this system. With IBE, user's identity and relevant public key are bound together, and the receiver can find out whether a message is signed by a specific user. An attacker cannot get the KGC's private key or those of satellites or registered users. Therefore, he cannot disguise himself as any role in the system in order to conduct man-in-the-middle attacks.

Impersonation Attacks. An attacker may attempt to impersonate an authorized user by forging an authentication request. As the first response to such authentication request from satellite during all three kinds of authentication phases should be encrypted using this user's private key, the attacker must know the exact content of right private key in order to be authenticated. However, he has no feasible way to know this private key. The attacker cannot even construct such authentication requests as he has no ability to forge the valid signature of an authorized user during fast-access authentication or handover authentication. At satellite side, if there is an attacker, who attempts to impersonate a satellite, he would fail to forge a valid signature for the session key without possessing the right satellite private key. Even if he replays a previous valid response, he would fail in the next steps of the authentication process due to the check of valid timestamp and the inability to decrypt the session key. Thus, the proposed protocol is secure against impersonation attacks.

Denial-of-Service Attacks. This protocol firstly checks whether the identity of current user is valid and then returns a response which is encrypted with the public key relevant to the identity during authentication. If there is a denial-of-service attack, it would not continue because the attacker has no corresponding private key, which means the satellite would not allocate relevant resources for service and the secure channel between this attacker and the satellite would not be established. Thus, this protocol can resist denial-of-service attacks.

Also, there is no threat of an attack using stolen verification tables or smart cards, as BAVP does not use verification tables or smart cards. Meanwhile, blockchain can ensure accuracy, completeness, consistency, and traceability of data which makes authentication more efficient and more secure. However, the KGC must be completely trusted as required by IBE, which may have potential safety problems hidden within it. It is reasonable to assume that KGC is trustworthy since users must register at KGC with their information to obtain services.

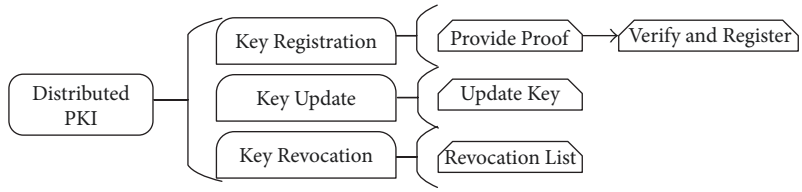

Figure 5: Structure of DPKI.

4. Distributed PKI

In the proposed protocol, although KGC is not involved during authentication, it is still the center for key management and is able to calculate all private keys of users. Once it is hacked, the security of the whole system would be threatened. In spite of some solutions that have partially solved this, there are no real all-around solutions. For example, in paper [12], a method based on (n, t) threshold secret sharing cryptography is designed to avoid this problem. The user's private key is split into n pieces and these key fragmentations are stored in different key privacy authority (KPA). Users only need to apply key fragmentation towards enough KPAs, and then they can restore their private key. Thus, this method can avoid the threat brought by centralized key management. Nevertheless, this solution brings additional costs for construction of KPAs, and also the number of KPAs should be large enough under individual owners for security. There are still concerns about KPA mechanism. For example, these KPAs need to take different strong safeguard procedures in order to increase the difficulty of breaking this system. Actually, IBE establishes mapping relations between identity and public key through mathematical methods which avoid the use of certificates, while we can realize this kind of mapping relations through smart contract on blockchain 2.0 like Ethereum. With such thought, we are actually building a distributed PKI (called DPKI).

4.1. Structure of DPKI. There are mainly three functional parts of DPKI: key registration, key update, and key revocation. The structure of DPKI is shown in Figure 5.

4.2. Methodology. DPKI is specifically built using smart contract with blockchain 2.0. Blockchain as a robust P2P network is able to ensure correctness of data stored in it. Thus, making centralized PKI distributed, which can overcome many weaknesses of traditional PKI, becomes possible. This section has the following structure: the first part explains how key registration works, followed by the principles of key update and key revocation and also the code template of DPKI smart contract.

4.2.1. Key Registration. For traditional PKI using certificates, users need to provide proof of their identity to get a valid certificate authorized by a trustworthy CA. With IBE, users need no certificates, but they simply provide a related identity string which can be an email address, ID number, or other strings, and then KGC calculates their private keys which are sent to users safely thereafter. In DPKI, users also submit proof for their identity and the authority they need. Users generate their public and private key pairs with any kind of asymmetrical encryption algorithm by themselves and then register their public key together with the standard name of algorithm used by invoking smart contract. After this, the administrator of the LEO system checks whether the identity is valid and afterwards passes their registration by invoking pass function of smart contract provided that nothing is wrong. In the satellite scenario, the asymmetrical encryption algorithm used should be limited to serval specified algorithms with consideration of resource constraints. The principles are shown in Figure 6.

4.2.2. Key Update. Out of security considerations, users should update their key pairs periodically. With a securely kept password, a user can update his key pairs proactively. During key registration, account address, public key, identity string, and algorithm used for key generation are bound together. Therefore, users willing to update their key pairs are able to do so through the smart contract update function at any time. The key update principles are shown in Figure 7.

4.2.3. Key Revocation. Generally, users need not revoke their key pairs. If their private keys are lost or stolen, they can generate new public and private key pairs and then update their key using the key update method. Nevertheless, when the passwords of users' blockchain accounts are lost or stolen, the users lose all control of key update and revocation. Assuming that there is one user A whose blockchain account is lost or stolen, he can revoke his original identity by submitting relevant proof and then the administrator checks validity of this proof. If this proof is right, the administrator adds A's original blockchain account to revocation list. Also, the administrator re-register A's new account with original authority and relevant remaining time after A executes key registration with a new blockchain account. The principles are shown in Figure 8.

4.2.4. Smart Contract of DPKI. With smart contract and blockchain, there is no need for a trustworthy CA, no cost on storage, and no overhead involved in key management. Figure 9 shows the smart contract structure for DPKI.

Multiple administrators can be enrolled to enhance the security of this satellite system. This can be realized by applying (n, t) threshold secret sharing or multisignature cryptography, which needs the majority of administrators to agree when taking an operation. We can also simply deploy DPKI on current public blockchain platform like Ethereum or implement DPKI on consortium blockchain constructed

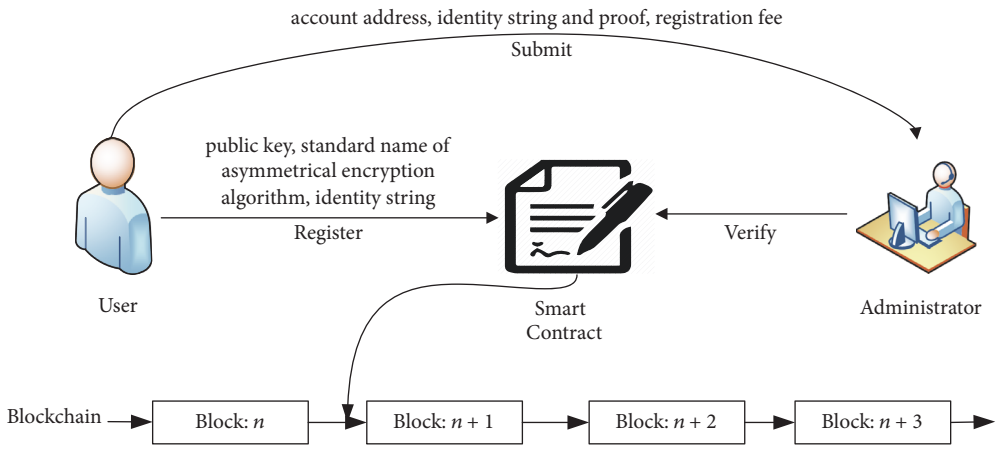

Figure 6: Principles of key registration.

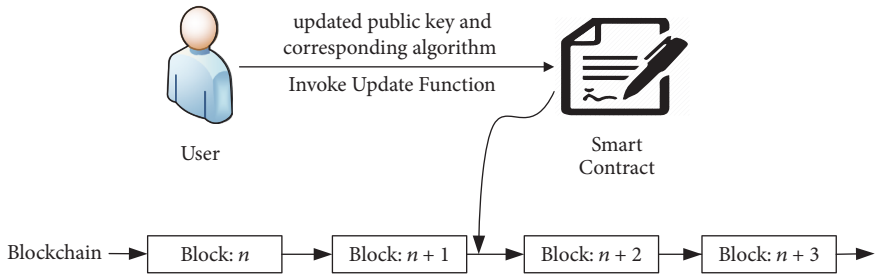

Figure 7: Principles of key update.

by the union of this satellite system. The two approaches with public- or consortium-based blockchain differentiates as follows:

(1) Public-based approach does not bring any storage overhead for users or the proprietor, and there is no money cost for construction of blockchain platform.

(2) Public-based approach must accept the current consensus mechanism by the adopted blockchain platform, while the consortium-based one can design an appropriate consensus mechanism for custom business needs, more customizable and controllable.

(3) A constructed consortium blockchain only contains data related to key management which makes it more efficient for query and other related respects.

4.3. Analysis of DPKI. For security analysis, traditional PKI is mature and adequate while CA is completely trustworthy. As for DPKI, the algorithm adopted for the generation of users' public and private key pairs should be adequately strong. Users themselves are responsible for choosing and maintaining such safety strength. In addition, each operation that invokes DPKI smart contract costs brokerage (known as gas price in Ethereum), which has a good resistance to denial-of-service attacks. With blockchain, there is no need for a trustworthy third party which avoids the potential threat in IBE. In addition, smart contract has the characteristics of atomicity and consistency.

In respect of overhead, traditional PKI has a huge cost for key management which is also complex, and every time a user wants to communicate with someone that has legal certificates, he needs to communicate with CA to verify the validity of certificates. For IBE, if the KGC is dependable, it does not store private keys of users, all of storage overhead is for public parameters and its own public and private key. Also, the KGC can be integrated with NCC at a low price, which means there is no need for a reliable third party in this system. Thus, it can be ignored. Referring to DPKI, if it is public-blocked based, then there is no storage overhead for users and satellites in LEO scenario. Only two communications with any full node in blockchain are needed for query of public key before authentication or other secure communication. If it is consortium-block based, then storage overhead of those full nodes will increase with the increasing amount of data.

In summary, considering that satellites cannot be set as a full node, the communication cost of querying public key is necessary. This is fatal for any efficient authentication protocol especially in high-delay scenarios like satellite networks. In future practice, consortium blockchain is also necessary because it is more controllable and customizable. With DPKI, users can cache the public keys of satellites they commonly connect to and save the calculation of satellites' public key in the proposed authentication protocol, but this is not suitable for satellites. Thus, IBE is still the best solution provided that the KGC is totally credible. We simulated the performance of this proposed protocol using IBE on OPNET. However, it

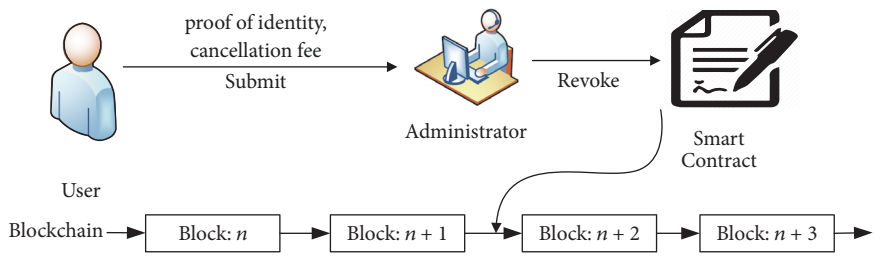

Figure 8: Principles of key revocation.

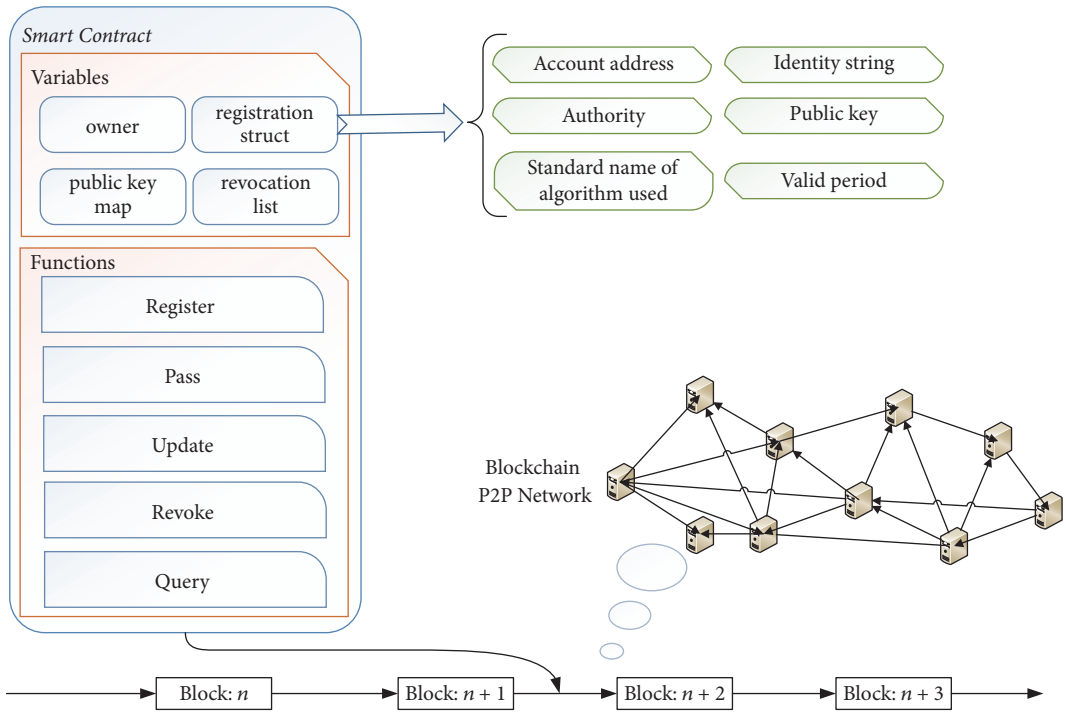

Figure 9: Smart contract of DPKI.

is also worth pointing out that DPKI is a promising scheme not only for satellite scenario but also for other scenarios with demands of secure communication.

5. Simulation and Evaluation

We evaluate the proposed protocol with simulation using IBE. With the OpenSSL, PBC, and GMP libraries, we implement an IBE algorithm and compare it to RSA which is recommended by the ISO as the asymmetric encryption standard. For example, in Cruickshank's paper, he uses RSA to implement the function of signature and encryption. In order to analyze the performance of the proposed protocol, we implement the protocol simulation on OPNET.

5.1. Comparison between IBE and RSA. To test whether IBE can be used in practice, we compared its performance to the RSA algorithm. While implementing IBE algorithm, we used the SHA1 algorithm that produces 160-bit digest as the hash function. We use the OpenSSL RSA algorithm.

The experimental environment used by the test program is Ubuntu 16.04LTS with 4 GB memory and 3.30 Ghz 4 Core i5-4590 processor. After running the test program for twenty times, the computational overhead of two algorithms is shown in Figure 10.

In this experiment, the bilinear pairing used by IBE is generated by the function whose prototype is *pbc_param_init_a_gen* (*pbc_param_t par*, *int rbits*, *int qbits*) in PBC, where *rbits* is 160 and *qbits* is 512 by default. The average time consumed for key generation, encryption, and decryption in IBE is 7.251 ms, 1.468 ms, and 1.369 ms, respectively. In the case of RSA, the time spent for key generation, encryption, and decryption is 37.817 ms, 3.753 ms, and 4.109 ms on average. It shows that IBE is superior to RSA, and this is mainly because IBE is based on bilinear pairings while RSA is based on the difficulty of decomposing a large number. Hence, the performance of IBE can satisfy the need for practical applications on satellite networks, and some advanced LEO satellite systems such as Iridium already have their own processors onboard which are superior in performance.

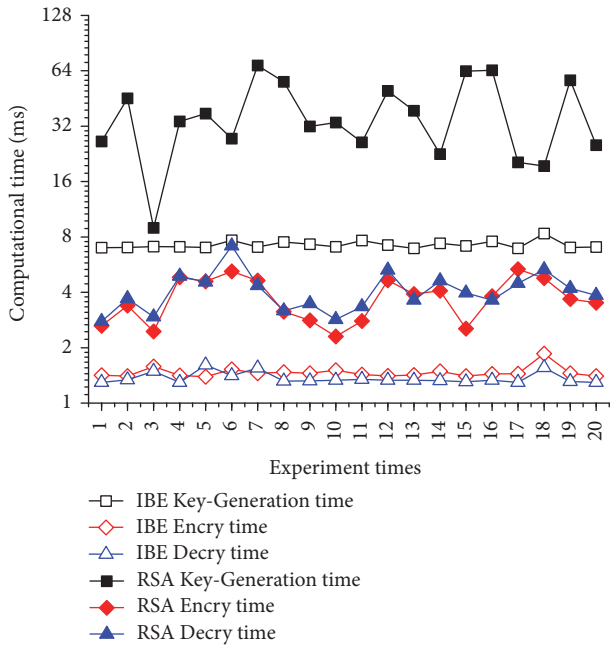

Figure 10: Comparison between IBE and RSA.

Moreover, the hash function, encryption, decryption, and other calculations involved in this protocol can be designed and implemented within particular hardware, so as to further reduce the demand for computing capability of satellite. In terms of the development with IBE, the Office of Chinese Security Commercial Code Administration issued the standard of SM9 algorithm which is one kind of identity-based encryption, and SM9 has entered the phase of promotion. For the security of IBE algorithm, paper [13] provides a rigorous demonstration.

5.2. OPNET Modeling and Simulation. Due to the low orbit of the satellites, handover is frequent in LEO satellite networks. Therefore, in order to ensure the communication persistence, the authentication protocol designed should be well adapted to this feature. In OPNET, we construct a LEO satellite network scenario [14] consisting of satellite nodes supporting applications attribute and analogous constellation of Iridium without backup satellites for simulation. The configurations of the satellite network include altitude: 780 km, inclination: 86.4°, period: 6027.14 s, and 6 orbits with 11 satellites per orbit.

We use *wlan_workstation_advanced* node as user node. Considering the relative motion between user and satellite, it is reasonable to set the user node to be immobile during simulation, and the satellites move in their own orbits. The process of this protocol is defined by *tasks_config*. There are mainly two phases: one is access authentication phase which is defined as *challenge_auth* and also fast-access authentication phase which is defined as *fast_access* in *task_config* object; the other one is the handover phase, which is defined as *switchsat*. The size and initialization time of message used during the simulation is based on the size of each field defined in each message and the performance of IBE together with the symmetric encryption (using the AES-192-ECB mode). For example, random number r used in the protocol is 4 bytes, identity string is no more than 30 bytes, timestamp is 15 bytes, and separator between different fields is 2 bytes. Of course, it is just a basic simulated setting which can be adjusted according to actual business needs. The bit error rate (BER) of the intersatellite link is 10^{-4}, and the BER of mobile link between mobile user and satellite is 10^{-5}. In addition, to build the entire LEO satellite network, it is also necessary to set IP addresses, routing protocols, signal-to-noise ratio of user, satellite nodes, and so on.

5.3. Interpretation of Result. In satellite constellation scenario, we simulate the performance of the protocol in a LEO satellite network by setting custom traffic between user and satellite nodes (based on *Application config*, *Task config*, and *Profile config* object).

We first simulated a complete flow of the protocol, the whole simulation lasts for 500 s, the access authentication occurs at 150 s, the handover authentication occurs at 300 s, and the fast-access authentication occurs at 400 s. The results of simulation are shown in Table 3.

From Table 3, we can see that the response time of each phase in this protocol is less than 500 ms which is far superior to the cost of authentication in paper [15] (10 s-level) and little superior to the cost of authentication in papers [5, 16] (500 ms-level). This protocol does not affect the quality of service (QoS) of satellites with such efficient performance. At the same time, the packet delay is basically between 50 ms and 70 ms. Compared to this, the average encryption and decryption time and other processing time can be ignored, this is also the feature that a practical authentication protocol should have. Moreover, it is easy to see that the handover authentication phase saves about 100 ms to 150 ms comparing with access authentication phase, and this proves the advantages of fast handover. In addition, we can see that the response time of fast-access authentication phase is shorter than other phases which benefits from the traceability and correctness of data in blockchain, and this is far superior to the performance of authentication protocol in paper [5, 16].

Next, we adjust the simulation to make it last for two hours. During this simulation, the average interval time of handover is about 10 min which is consistent with Iridium system. The results of simulation are shown in Figures 11 and 12.

From both figures, we can see that the response time of handover authentication is about 360 ms, which is 28 percent superior to the performance of authentication protocol in paper [5, 13], and the delay is about 53 ms, which is in accordance with the result of Table 3. This also demonstrates that our proposed protocol with IBE is stable, effective, and more suitable for LEO satellite network which has the characteristic of frequent link switching.

Next, we set an additional three application scenarios and ran them for 24 hours. The configurations are listed in Table 4.

In these three application scenarios where the arrival of users conforms to the Poisson distribution, we test the

Table 3: Response time and delay in each phase of the protocol.

Phase	Src	Dest	Response time	Delay
Access	User	S_1	0.17771 s	0.06737 s
Access	User	S_1	0.32246 s	0.07591 s
Fast-access	User	S_2	0.18039 s	0.05363 s
Handover	User	S_1	0.17816 s	0.05322 s
Handover	User	S_2	0.20689 s	0.05083 s

Table 4: Configuration of scenarios setting.

Application	Access (A)	Fast-access (FA)	Order
1	10	10	10 users/1 h (concurrent)
2	100	100	100 users/1 h (concurrent)
3	1000	1000	1000 users/1 h (concurrent)

Table 5: Simulation results of applications 1–3.

Application	Scale	Average response time (access)	Average response time (fast-access)
1	10 users/1 h	0.388472 s	0.122618 s
2	100 users/1 h	0.382054 s	0.174229 s
3	1000 users/1 h	0.414062 s	0.140895 s

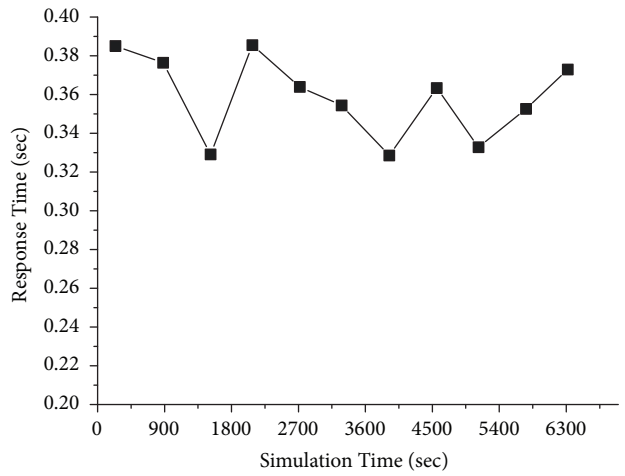

Figure 11: Response time.

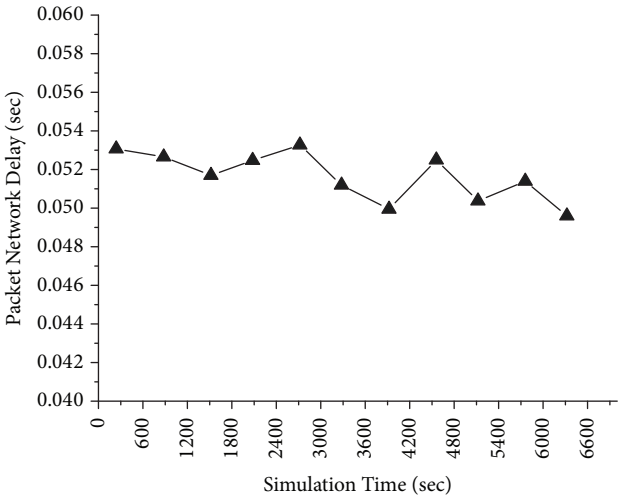

Figure 12: Packet delay.

efficiency and stability of this protocol with increased user-scale (10, 100, and 1000 per hour) while access and fast-access authentication phases are concurrently executed on different satellites.

From Table 5, we can find that the number of users who access the same satellite does not affect the performance of this protocol which proves this protocol stable performance. This is mainly due to the fact that there is no dependency or interference among different authentication methods. Meanwhile, the average time of these three application scenarios is about 400 ms for access authentication and 150 ms for fast-access authentication, which proves this protocol efficient performance. This is due to the low delay of LEO satellite networks compared to traditional satellite networks and the high efficiency of IBE together with the correctness and traceability of data in blockchain. The simulation results are a little superior to the performance in Table 3 and Figures 11 and 12 which are mentioned above. This is due to the different positions of users within the satellite coverage region during authentication, which emerges when the scale of users increases. The diagram of satellite coverage region is shown in Figure 13.

Obviously, authentication response time of the user at the edge of satellite cover region would be longer than that of the user right underneath the satellite. Also, the ratio of the shortest time divided by longest time during identical authentication should be $\cos(\alpha)$ theoretically. And this explains the range of fluctuations about simulation results.

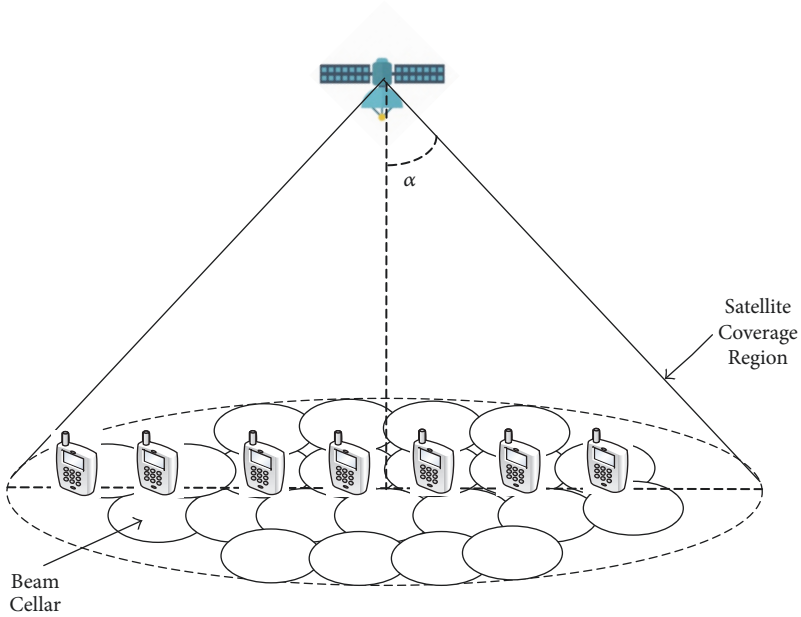

Figure 13: Diagram of multibeam satellite.

As for storage overhead, the cost consists of two parts: the first is used to store the public key of KGC and related system parameters; the second is for session key. Taking the number of users in Iridium system (which was 150,000 at its peak), as an example, the storage used for storing session keys is about 24 MB when 150,000 users are all online at the same time. Therefore, the cost of key storage is much lower than this for each satellite, which is acceptable. Furthermore, the logging function of this protocol also brings cost of storage, and its size is mainly determined by the threshold for storing blocks. When the number of blocks reaches the maximum, the satellite will delete all related blocks according to the certain rule. In this respect, the threshold specified is the cost of storage for each satellite (e.g., threshold can be set to 100 MB, but with the increasing number of users, it needs to be increased).

Assume that the arrival of user conforms to the Poisson distribution and the service time obeys negative exponent distribution. The computational overhead of access, fast-access, and handover authentication is R_a, R_f, and R_h, the average number of users per hour is λ, the average service time is $1/\mu$, and the average interval time of handover is t. Thus, for each satellite, the computational overhead brought by this protocol per hour is

$$x_1 \times \frac{e^{-\lambda}\lambda^{x_1}}{x_1!} \times R_a + x_2 \times \frac{e^{-\lambda}\lambda^{x_2}}{x_2!} \times R_f + \frac{e^{-\lambda}\lambda^{(x_1+x_2)}}{(x_1+x_2)!} \times \left(e^{-\lambda nt} - e^{-\lambda(n+1)t}\right) \times n \times R_h. \quad (5)$$

And x_1 represents the number of users who get authenticated by access authentication while x_2 is the number of users who get authenticated by fast-access authentication.

6. Conclusion and Future Work

Considering the dynamic topology and frequent link switching found in LEO satellite networks, this paper proposes a new decentralized access verification protocol: BAVP with IBE for authentication and blockchain for distributed computing and storage. For evaluation, we simulate this protocol in OPNET. The theoretical analysis and simulation result show that this protocol is secure, light-weighted, and efficient in LEO satellite network. Additionally, we also propose and analyze a distributed PKI scheme: DPKI which solves the problem of KGC single point-of-failure problem.

The proposed architecture and protocols will be further developed and optimized in several ways. Blockchain can ensure the stored data is accurate and tamper-resistant, but it cannot ensure data correctness and originality. That is why a third credible party is necessary. DPKI can avoid the defect of IBE where no user private key is owned by the KGC or such other centers, no matter whether these centers are reliable or not. However, the time for querying user/satellite public keys is limited by network latency which is usually high in satellite network and much longer than encryption/decryption time. Additionally, in actual deployment of blockchain on a satellite network, there are some future works like reforming blockchain technology according to particular satellite network routing algorithm and constellation needed to do. Besides, based on DPKI, many centralized application scenarios such as social applications and third-party payment can be innovated and reformed, which is also in our future research plan.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This material is based upon work supported by the China NSF Grant no. 61472189, the CASC Innovation Fund no. F2016020013, the State Key Laboratory of Air Traffic Management System and Technology no. SKLATM201703, and the Postgraduate Research & Practice Innovation Program of Jiangsu Province no. KYCX17_0369.

References

[1] S. Li, M. Liu, and S. Wei, "A distributed authentication protocol using identity-based encryption and blockchain for LEO network," in *Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, pp. 446–460, Springer, Cham, Switzerland, 2017.

[2] H. S. Cruickshank, "Security system for satellite networks," in *Proceedings of the 5th International Conference on Satellite Systems for Mobile Communications and Navigation*, pp. 187–190, IET, London, UK, May 1996.

[3] M. S. Hwang, C. C. Yang, and C. Y. Shiu, "An authentication scheme for mobile satellite communication systems," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 4, pp. 42–47, 2003.

[4] Y. F. Chang and C. C. Chang, "An efficient authentication protocol for mobile satellite communication systems," *ACM SIGOPS Operating Systems Review*, vol. 39, no. 1, pp. 70–84, 2005.

[5] G. Zheng, H. T. Ma, C. Cheng, and Y. C. Tu, "Design and logical analysis on the access authentication scheme for satellite mobile communication networks," *IET Information Security*, vol. 6, no. 1, pp. 6–13, 2012.

[6] L. Qi and L. Zhi, "Authentication and access control in satellite network," in *Proceedings of the 2010 Third International Symposium on Electronic Commerce and Security (ISECS)*, pp. 17–20, IEEE, Guangzhou, China, 2010.

[7] E.-J. Yoon, K.-Y. Yoo, J.-W. Hong, S.-Y. Yoon, D.-I. Park, and M.-J. Choi, "An efficient and secure anonymous authentication scheme for mobile satellite communication systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, no. 86, 2011.

[8] X. Wu, A. Zhang, J. Li, W. Zhao, and Y. Liu, "A lightweight authentication and key agreement scheme for mobile satellite communication systems," in *Proceedings of the International Conference on Information Security and Cryptology*, pp. 187–204, Springer, Cham, Switzerland, 2016.

[9] Y. Zhang, J. Chen, and B. Huang, "An improved authentication scheme for mobile satellite communication systems," *International Journal of Satellite Communications and Networking*, vol. 33, no. 2, pp. 135–146, 2015.

[10] J. Wu, Y. Long, Q. Huang, and W. Wang, "Design and application of IBE email encryption based on Pseudo RSA certificate," in *Proceedings of the 2016 12th International Conference on Computational Intelligence and Security (CIS)*, pp. 282–286, IEEE, Wuxi, China, 2016.

[11] D. Patel, J. Bothra, and V. Patel, "Blockchain exhumed," in *Proceedings of the Asia Security and Privacy (ISEASP)*, pp. 1–12, IEEE, Surat, India, 2017.

[12] R. Gangishetti, M. C. Gorantla, M. L. Das, and A. Saxena, "Threshold key issuing in identity-based cryptosystems," *Computer Standards & Interfaces*, vol. 29, no. 2, pp. 260–264, 2007.

[13] L. Chen and Z. Cheng, "Security proof of Sakai-Kasaharas identity-based encryption scheme," in *Proceedings of the IMA International Conference on Cryptography and Coding*, pp. 442–459, Springer, Berlin, Germany, 2005.

[14] H. Long, *OPNET Modeler and Computer Network Simulation*, Xi'an University of Electronic Science and Technology Press, Xi'an, China, 2006.

[15] Z. B. Xu and H. T. Ma, "Design and simulation of security authentication protocol for satellite network," *Computer Engineering and Applications*, vol. 42, pp. 130–132, 2007.

[16] X. Zhang, H. Liu, Y. Lu, and F. Sun, "A novel end-to-end authentication protocol for satellite mobile communication networks," in *Foundations and Applications of Intelligent Systems*, pp. 755–766, Springer, Berlin, Germany, 2014.