A Highly Secure Identity-Based Authenticated Key-Exchange Protocol for Satellite Communication

Zhong Yantao and Ma Jianfeng

Abstract: **In recent years, significant improvements have been made to the techniques used for analyzing satellite communication and attacking satellite systems. In 2003, a research team at Los Alamos National Laboratory, USA, demonstrated the ease with which civilian global positioning system (GPS) spoofing attacks can be implemented. They fed fake signals to the GPS receiver so that it operates as though it were located at a position different from its actual location. Moreover, Galileo in-orbit validation element A and Compass-M1 civilian codes in all available frequency bands were decoded in 2007 and 2009. These events indicate that cryptography should be used in addition to the coding technique for secure and authenticated satellite communication. In this study, we address this issue by using an authenticated key-exchange protocol to build a secure and authenticated communication channel for satellite communication. Our protocol uses identity-based cryptography. We also prove the security of our protocol in the extended Canetti-Krawczyk model, which is the strongest security model for authenticated key-exchange protocols, under the random oracle assumption and computational Diffie-Hellman assumption. In addition, our protocol helps achieve high efficiency in both communication and computation and thus improve security in satellite communication.**

Index Terms: **Authenticated key exchange (AKE), computational Diffie-Hellman (CDH) assumption, extended Canetti-Krawczyk (ECK) security, identity-based cryptography (IBC), random oracle model, satellite communication.**

I. INTRODUCTION

Satellite networks support communication among satellites, terrestrial networks, airplanes, spacecraft, and other network nodes in the sky or on the ground. Hence, such networks offer the advantage of widespread coverage. For this reason, satellite networks are suitable for providing communication services for various space missions such as meteorological studies, environmental monitoring, military reconnaissance, and space exploration.

Despite the notable improvements made to the techniques used for preventing attacks to satellite systems, poor security in satellite communication remains a problem. In 2003, a research team at Los Alamos National Laboratory, USA, demonstrated the ease with which civilian global positioning system (GPS) spoofing attacks can be implemented. The team accomplished this spoofing by using a GPS satellite simulator that is unregulated and widely available. The simulator broadcasted a fake signal that caused the GPS receiver to operate as though it were located at a position different from its actual location [1]. Moreover, in 2007 and 2009, a research team at Stanford University decoded Galileo in-orbit validation element A (GIOVE-A) and Compass-M1 civilian codes in all available frequency bands [2]–[4]. These events show that merely using a coding technique is not sufficient for secure communication in satellite networks. More attention should be paid to the fundamental security issues in satellite communications, namely authentication and privacy. Both these issues can be addressed by establishing a secure channel between two nodes through an authenticated key-exchange (AKE) protocol, at the end of which the two parties share a common session key.

AKE protocols can be based on symmetric or asymmetric cryptography. In the symmetric approach, a large number of pairwise keys are required, and hence, it is impractical for scalable satellite networks. More specifically, in the symmetric approach, a long-term symmetric key should be distributed to a pair of nodes before the key-exchange protocol between them is executed. With the dynamic growth of satellite networks, generating and managing these keys has become highly problematic. In the certificate-based asymmetric approach, each node should obtain the public key of the other and verify it with a trusted third party. Unfortunately, one of the key characteristics of satellite communication-long time delay in data transmission due to the long distance between network nodes-makes deployment of certificate-based AKE protocols in satellite networks inefficient. More specifically, before executing an AKE protocol in a certificate-based method, one network node must obtain the certificate of the other node from a certificate authority (CA) and then access the certificate revocation list so that the two nodes extract each other's valid public keys. These operations impose an extra burden on communication. The identity-based cryptosystem (IBC) introduced by Shamir [5] helps eliminate this burden because the nodes' public key can be an arbitrary string, which is typically an identity string, without the intervention of a CA.

The aim of the present research is to establish an efficient identity-based AKE protocol with adequate security guarantees for use in satellite communication. In particular, we concentrate on ease of deployment, computational complexity, and communication rounds. However, before discussing our proposed protocol, we address some existing identity-based AKE protocols.

A few identity-based AKE protocols have been proposed till date. The early schemes [6]–[8] do not take into account the possibility of active attacks and hence do not provide adequate

Manuscript received April 14, 2010.

These authors would like to thank the anonymous reviewers for their valuable and constructive comments.

This work was supported by grants from 863 Project of China (2007AA01Z429), the National Science Foundation of China (No. 60573036, 60702059, 60503012, 60872041), and the Key Project of the National Science Foundation of China (No. 60633020).

The authors are with the Key Laboratory of Computer Networks and Information Security (Ministry of Education), Xidian University, Xi'an, China, email: zhongyantao@126.com, jfma@mail.xidian.edu.cn.

security. Most of the more recent identity-based AKE protocols [9]–[11], which are secure against active attacks, employ a pairing technique in which a session key is established through a bilinear pairing operation. These protocols are computationally inefficient because a greater number of multiplications in the underlying finite field would be required for bilinear pairing operations than for elliptic curve point scalar multiplication in the same finite field [12], [13]. In addition, in most of these schemes, a participant's identity must be mapped to a point on an elliptic curve, and this is computationally expensive. Another modular approach for designing identity-based AKE protocols involves the application of identity-based signature schemes to the construction proposed in [14]. Informally speaking, when running a Diffie-Hellman (DH) key-exchange protocol [15], both participants authenticate each other through an identity-based signature scheme. An AKE protocol constructed in this manner (also called an authenticated DH protocol) is Canetti-Krawczyk (CK) secure [16], but requires at least three rounds of message transmission, while many AKE protocols require only two rounds. An example of an authenticated DH protocol has been provided in [17].

Protocol security is also one of our goals, particularly resistance to various active and passive attacks. Analysis of security of AKE protocols has been demonstrated to be a nontrivial task. Bellare and Rogaway [18] presented the first formal security model (Bellare-Rogaway (BR) model) for AKE protocols. Since then, several extensions have been made to the BR model. The most famous extended BR model is the CK model [16] which was proposed in 2001. However, all these models are based on the strong assumption that the adversary is not allowed to obtain certain secret information about the session that is being attacked. As a result, a secure AKE protocol in these models might still be vulnerable to the leakage of an ephemeral secret or a private key. Recently, an extended CK (ECK) model [19], that is resistant to the above mentioned attack has been reported. The only attacks that are not allowed for in the ECK model are those that would trivially break an AKE protocol, i.e., an attack in which the adversary reveals the ephemeral secret and the static private key of one node in the protocol so that this node can be impersonated. Thus, the ECK model is currently regarded as the strongest security model for AKE protocols. ECK security also implies key generation center (KGC) forward secrecy [9], which helps ensure the security of previously established session keys after the master key of the KGC has been compromised. Many ECK-secure AKE protocols have been proposed till date [19]–[24].

With this background, we have developed an efficient ECK identity-based AKE protocol. Our protocol uses DH values and hash functions to combine the long-term private key and the ephemeral key of both nodes so that the security requirement matches that in the case of other ECK-secure protocols. We prove the security of our protocol in the ECK model under the random oracle assumption and the computational Diffie-Hellman (CDH) assumption. Furthermore, as opposed to other identity-based AKE protocols, our protocol requires only two sent messages and three exponentiations per node in each session. Therefore, our protocol affords the highest possible communication and computation efficiency. Suffice it to say that our protocol is well suited for communication in satellite networks.

The rest of the paper is organized as follows. Section II presents the preliminaries of the protocol. An overview of the protocol is given in Section III. A detailed description of the protocol and a security analysis are presented in Section IV. Comparisons between the proposed protocol and those proposed in related studies are listed in Section V. Section VI includes concluding remarks.

II. PRELIMINARIES

In this section, we briefly review some preliminaries upon which the security of our protocol is based, including the CDH assumption and the ECK security model. For more details of the ECK model, we refer the reader to [20].

A. CDH Assumption

Let λ be a security parameter. Let $G = \langle g \rangle$ be a cyclic group of prime order q, where $q = O(2^{\lambda})$. Let $a, b \in Z_q^*$. For any probabilistic polynomial time algorithm M, there exists a negligible $\varepsilon(\lambda) > 0$, such that $\Pr[M(q, g, g^a, g^b) = g^{ab}] < \varepsilon(\lambda)$.

B. ECK Security Model

In the ECK model there are n nodes each modeled by a probabilistic polynomial time turing machine. Each node has a static public-private key pair.

Session. A particular instantiation of an AKE protocol executed by one node is called an AKE session. Legitimate execution of an AKE protocol by two nodes A and B consists of two AKE sessions. These two sessions are called matching sessions. A session identifier sid consists of the identities of the two nodes and the information exchanged between two nodes, i.e., $sid = (role, ID, ID^*, comm_1, \cdots, comm_n)$. Here $role \in \{initiator, responder\}$ is the role in the protocol; ID, the identity of the node executing the session; ID^*, the identity of the peer; and $comm_i \in \{0,1\}^*$, the ith message sent in the protocol execution.

Adversary. The adversary M is modeled as a probabilistic polynomial time Turing machine and has full control of the communication network and may eavesdrop, delay, replay, alter, or insert messages at will. The adversary's capability is modeled by accessing the following oracles.

- Send(sid, m). The adversary sends a message m to the session sid.
- EphemeralKeyReveal (sid). The adversary obtains the ephemeral private key of the session sid.
- SessionKeyReveal (sid). The adversary obtains the session key of a complete session sid.
- EstablishNode (ID). The adversary arbitrarily registers a new node with identity ID. In this way, the adversary has total control of this node.
- PrivateKeyReveal (ID). The adversary obtains the private key of node ID.
- Test(sid). This oracle can be queried by the adversary only once in an experiment. Provided that the session key has been computed in the session sid, the adversary obtains the session key with probability 1/2 and a uniformly chosen random value in $\{0,1\}^{\lambda}$ with probability 1/2.

Experiment. The adversary M is given a set of honest nodes. M can make any sequence of the oracle queries described above except the test query. At any time in the experiment, M makes a query Test(sid) to a complete session sid owned by an honest node. At the end of the experiment, M guesses whether the value he obtained is a random value or the session key.

Definition 1 (Cleanness). Let sid be a completed session owned by an honest node A, and sid^* is its matching session owned by an honest node B, if it exists. sid is said to be clean if none of the following three conditions holds:

- Adversary makes a SessionKeyReveal query to sid or sid^*, if it exists.
- sid^* exists and the adversary either makes queries:
  - EphemeralKeyReveal(sid) and PrivateKeyReveal(A); or
  - EphemeralKeyReveal(sid^*)and PrivateKeyReveal(B).
- sid^* does not exist and the adversary either makes queries:
  - EphemeralKeyReveal(sid) and PrivateKeyReveal(A); or
  - PrivateKeyReveal(B).

Definition 2 (ECK security). The adversary M wins the experiment if the test session is clean and M makes a correct guess. The advantage of M in the experiment above is defined as

$$Adv^{AKE}(M) = |\Pr[Mwins] - 1/2|.$$

An AKE protocol is called secure in the ECK model (ECK-secure) if matching sessions compute the same session keys and $Adv^{AKE}(M)$ is negligible for every probabilistic polynomial time M.

III. PROTOCOL OVERVIEW

In identity-based AKE, a trusted authority called KGC is required to derive private keys from arbitrary public keys for nodes, and it publishes this public system information. In our scheme, we place the KGC on the ground as a satellite network control center. We do this because the ground control center is controlled by personnel, giving it stronger computational power and a greater ability to resist attack. Furthermore, all ground stations in a satellite network can be considered a distributed system, so that the stations are logically regarded as one node - the KGC.

In addition to ground stations, a satellite network usually includes a number of satellites as fixed members, and many other kinds of nodes that can temporarily access the satellite network. Since ground stations are more capable of resisting attack, the secure communication between ground stations or between one ground station and one satellite can be easily achieved using existing techniques. Therefore, we focus on the secure communication between two satellites or between a satellite and another kind of node.

Our protocol consists of three stages labeled I, II, and III. They are also called the setup stage, the private key extraction stage, and the key-exchange stage, respectively. Stage II and stage III are shown in Figs. 1 and 2 respectively.

Stage I is performed by KGC, and the main goal is to determine all system parameters for the satellite network. In addition, the KGC chooses a system master secret, and computes a system public key. Stage I can be executed before building the satellite network, thus all of these parameters and the system public key can be preloaded into the network nodes' memory.

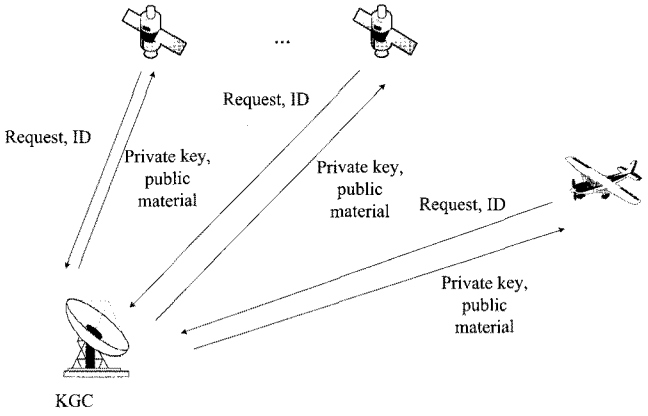

Fig. 1. Private key extraction stage of our protocol.

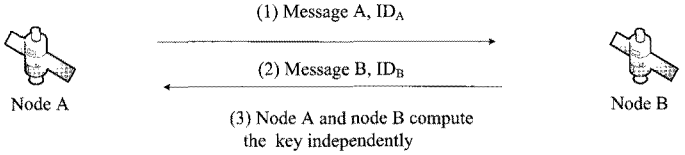

Fig. 2. Key exchange stage of our protocol.

The goal of stage II is to issue a private key to each node. In this stage, each node sends a request for a new private key together with its identity to the KGC. The KGC then calculates a corresponding private key and a public material and sends it to the node. As with the usual method, the KGC uses the system's master key, the node's identity, and the validity period of the key to calculate the private-key-public-material pair. A node receives the private key from the KGC only once before it participates in any key exchange. Therefore, stage II can be executed before launching the node, and the private key can be preloaded into the network node's memory before launching it.

Before engaging in a secure conversation, two nodes should execute stage III first. In an execution of stage III, the two nodes send a message containing their identity, a DH value, and the public material to each other, and both nodes compute the same session key after exchanging the message.

IV. PROPOSED PROTOCOL

A. Detailed Description of Our Protocol

(1) Setup stage. Upon receiving a security parameter λ, KGC determines the following parameters:

q: A large prime number such that $q = O(2^{\lambda})$.

G: A cyclic group of order q.

g: An element of order q in group G.

$H_1, H_2 : \{0,1\}^* \rightarrow Z_q^*$: Two one-way hash functions mapping an identity of a node to a number belonging to Z_q^*.

$H_3, H_5 : \{0,1\}^* \rightarrow Z_q^*$: Two collision-resistant one-way hash functions modeled as random oracles.

$H_4 : \{0,1\}^* \rightarrow Z_q^*$: A collision-resistant one-way hash func-

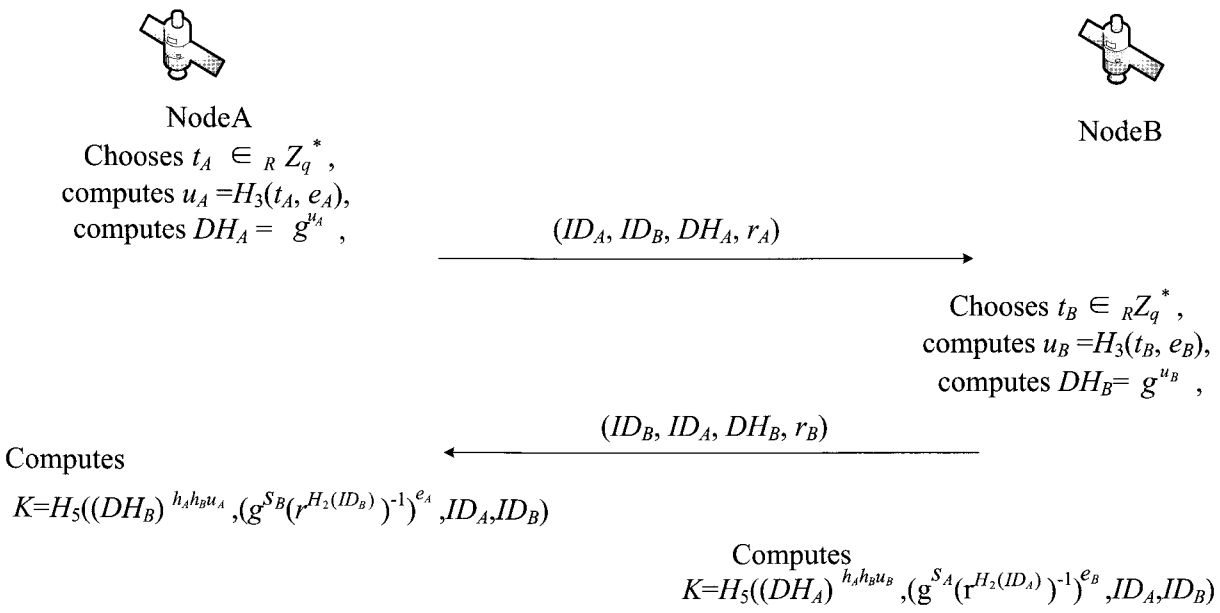

Fig. 3. Detailed description of key-exchange stage.

tion mapping a string of arbitrary length to a number belonging to Z_q^*.

In addition, the KGC chooses a random number $x \in Z_q^*$ as the system master secret, and computes $y = g^x$ as the system public key.

(2) Private key extraction stage. Given a satellite node identity ID, the KGC constructs a private key for this node as follows. The KGC randomly chooses a number $k \in Z_q^*$ and computes $s = H_1(ID)$, $r = g^k$, and $e = [s - kH_2(ID)][x^{-1} \bmod (q-1)]$. Then the private key is e, and r is the public material of this node.

(3) Key exchange stage. Assume that satellite node A with identity ID_A and satellite node B with ID_B want to share a secret session key. Both of them have already got their private key e_A, e_B and public material r_A, r_B, respectively. Then they perform the following steps, including two rounds of message sending as shown in Fig 3.

Step 1. Node A chooses a random number $t_A \in Z_q^*$ as his ephemeral private key, computes $u_A = H_3(t_A, e_A)$, and $DH_A = g^{u_A}$, and sends (ID_A, ID_B, DH_A, r_A) to node B.

Step 2. Node B chooses a random number $t_B \in Z_q^*$ as his ephemeral private key, computes $u_B = H_3(t_B, e_B)$, and $DH_B = g^{u_B}$, and sends $(ID_B, ID_A, DH_B, r_B)$to node A.

Step 3. Upon receiving the message from node B, node A computes $h_A = H_4(ID_A, ID_B, DH_A)$, $h_B = H_4(ID_A, ID_B, DH_B)$, $s_A = H_1(ID_A)$, $s_B = H_1(ID_B)$, and the session key

$$K = H_5((DH_B)^{h_A h_B u_A}, (g^{s_B}(r^{H_2(ID_B)})^{-1})^{e_A}, ID_A, ID_B)$$
$$= H_5(g^{u_A h_A h_B u_B}, y^{e_A e_B}, ID_A, ID_B).$$

Step 4. Upon receiving the message from node A, node B computes $h_A = H_4(ID_A, ID_B, DH_A)$, $h_B = H_4(ID_A, ID_B, DH_B)$, $s_A = H_1(ID_A)$, $s_B = H_1(ID_B)$, and the session key

$$K = H_5((DH_A)^{h_A h_B u_B}, (g^{s_A}(r^{H_2(ID_A)})^{-1})^{e_B}, ID_A, ID_B)$$
$$= H_5(g^{u_A h_A h_B u_B}, y^{e_A e_B}, ID_A, ID_B).$$

Note. The use of public material and a private key in our protocol makes it resemble a certificate-based protocol. However, there are distinct differences between our protocol and a certificate-based AKE protocol. In our protocol, the public material of a node is generated by the KGC, while in a certificate-based protocol the public-private key pair can be assigned by the node. More importantly, public material need not be authenticated by CA in our protocol, because the computation of public material and a private key for a node involves the master secret of the system and the identity of the node, and only a KGC can complete this task. Thus our protocol is indeed an identity-based AKE protocol.

B. Security Analysis

In this section, a formal security proof is given, which shows that our AKE protocol is ECK-secure under the CDH assumption in the random oracle model.

Lemma 1. If two nodes complete matching sessions, then they obtain the same session key.

Proof: This lemma follows immediately from the definition of matching sessions and the description of the protocol. That is, if two nodes, assumed to be nodes A and B, complete the matching sessions, they both compute the same following key

$$K = H_5((DH_B)^{h_A h_B u_A}, (g^{s_B}(r^{H_2(ID_B)})^{-1})^{e_A}, ID_A, ID_B)$$
$$= H_5((DH_A)^{h_A h_B u_B}, (g^{s_A}(r^{H_2(ID_A)})^{-1})^{e_B}, ID_A, ID_B)$$
$$= H_5(g^{u_A h_A h_B u_B}, y^{e_A e_B}, ID_A, ID_B).$$

□

Lemma 2. If H3 and H5 are random oracles, then there is no feasible adversary that succeeds in distinguishing the session key of a clean session with a non-negligible advantage.

Proof: Assume M is an AKE adversary against our protocol who succeeds with a non-negligible advantage in distinguishing the session key of a clean test session from a randomly chosen value. Since H_5 is a random oracle, M only has two ways to distinguish the session key:

- Key replication. M succeeds in forcing the establishment of another session that has the same session key as the test session but different from the matching session of the test session. In this case, M can make a SessionKeyReveal query to the session and get the same session key as the test session.
- Forging attack. At some point, M correctly computes the 4-tuple and queries the random oracle H_5 on the 4-tuple.

Since the input to the session key derivation includes the identities and ephemeral key of both nodes, the key replication is impossible when H_3, H_4, and H_5 are collision-resistant.

Now we consider the forging attack. We show that if M can mount a successful forging attack, then a simulator S can be constructed to solve the CDH problem by using M as a subroutine. S takes a 3-tuple (g, g^a,g^b) as input and is asked to compute the value of g^{ab}. Then after an execution of an ECK experiment with M, S can reveal the solution efficiently with non-negligible probability.

According to the definition of cleanness, we distinguish the following cases:

Case 1.There exists a matching session to the test session owned by another honest node, and M makes no PrivateKeyReveal query to the owner of the test session or the owner of its matching session. In this case, M may make EphemeralKeyReveal queries to both sessions.

Case 2. There exists a matching session to the test session owned by another honest node, and M makes one PrivateKeyReveal query to the owner of the test session or its matching session and one EphemeralKeyReveal query to the matching session of the previous one.

Case 3. There exists a matching session to the test session owned by another honest node, and M makes no EphemeralKeyReveal query to the test session or its matching session. In this case, M may make PrivateKeyReveal queries to both nodes.

Case 4. No honest node owns a matching session to the test session.

We now discuss these cases in turn as follows. Assume the adversary M activates at most n honest nodes and sc sessions in each node.

(1) Case 1. S prepares n nodes and executes the setup stage. Then S randomly selects two nodes, denoted by A and B, and assigns corresponding public material r_A and r_B as follows:

$$r_A = (g^{s_A}(g^a)^{-x})^{(H_2(ID_A))^{-1} mod(q-1)},$$
$$r_B = (g^{s_B}(g^b)^{-x})^{(H_2(ID_A))^{-1} mod(q-1)}$$

while the remaining n-2 nodes are assigned private keys and public material following the protocol.

When M asks node A or B to send a message to a node ID in some session executed with ephemeral key t_A or t_B , respectively, S randomly selects u_A or u_B and records in memory (t_A, u_A) or (t_B, u_B) , then sends (ID_A, ID, g^{u_A}, r_A) or (ID_B, ID, g^{u_B}, r_B) to M.

When M makes queries, the action of S is discussed below.

- $H_3(t, e)$: S checks whether $g^e = g^a$ or $g^e = g^b$. If true, then S submits $(g^b)^e$ or $(g^a)^e$ as the answer, respectively, to the CDH problem and thus completes its task, otherwise S acts as a random oracle.
- $H_4(ID_1, ID_2, DH)$: S acts in accordance with the protocol.
- H_5 : S acts as a random oracle.
- PrivateKeyReveal(C): If $C = A$ or $C = B$, S aborts its simulation, otherwise S follows the protocol.
- SessionKeyReveal(sid):
  1. If none of sid and its matching session, if it exists, is owned by A or B, then S computes the session key and returns it.
  2. If one of sid and its matching session, if it exists, is owned by A, S searches for the record (t_A, u_A) in the record in which t_A is the ephemeral key of node A in session sid, and uses u_A to compute the session key and return it. Note that in this situation, S is able to compute the session key.
  3. If one of sid and its matching session, if it exists, is owned by B, this situation is the same as the previous one.
  4. If both sid and its matching session, if it exists, are owned by A and B, S randomly selects a value k as the session key and returns it.
- Other oracles. S acts in accordance with the protocol.

If M wins in the forging attack, he must query H_5 with the following 4-tuple

$$(g^{u_A h_A h_B u_B}, y^{e_A e_B}, ID_A, ID_B).$$

On receiving this 4-tuple, S solves the CDH problem by computing the following value as the answer

$$g^{ab} = (y^{e_A e_B})^{x^{-1}}.$$

With probability $2/(n-1)n$, M selects A and B as the owner and the peer of the test session. When M succeeds with non-negligible probability ε in computing the correct 4-tuple, S solves the CDH problem with probability at least $(2\varepsilon)/n(n-1)$, noting that when M queries H_3 with some (t, e) satisfying $g^e = g^a$ or $g^e = g^b$, S also solves the CDH problem.

(2) Case 2. S prepares n nodes and executes the setup stage. Then S randomly selects a node B and a session, $session_a$, executed by some honest node A. In the key extraction stage, S assigns a randomly selected value r_B as the public material of node B and other $n-1$ nodes are assigned private keys and public material following the protocol.

When M asks node A to send a message in $session_a$ execution, S sets

$$DH_A = g^{u_A} = g^a$$

and sends (ID_A, ID, DH_A, r_A) to M.

When M asks node B to send a message to node A in some session execution with ephemeral key t_B, S sets

$$DH_B = g^{u_B} = g^b \quad (1)$$

and records in the memory (t_B, DH_B), then sends (ID_A, ID_B, DH_B, r_A) to M. When M makes queries, the action of S is discussed below.

- $H_3(t, e)$: S acts as a random oracle. Additionally, once a value $u = H_3(t, e)$ is computed, S checks whether $g^u = g^a$ or $g^u = g^b$. If true, then S submits $(g^b)^u$ or $(g^a)^u$ as the answer to the CDH problem and thus completes its task.
- $H_4(ID_1, ID_2, DH)$: S acts in accordance with the protocol.
- $H_5(DH_1, DH_2, ID_1, ID_2)$: S acts as a random oracle.
- PrivateKeyReveal(C): If $C = B$, S aborts its simulation, otherwise S follows the protocol.
- EphemeralKeyReveal(sid): If $sid = session_a$, S aborts its simulation, otherwise S follows the protocol.
- SessionKeyReveal(sid):
  1. If $sid = session_a$, S aborts the simulation.
  2. If none of sid and its matching session, if it exists, is owned by B, then S computes the session key and returns it.
  3. If one of sid and its matching session, if it exists, is owned by B, S searches for the record (t_B, DH_B) in the record in which t_B is the ephemeral key of node B in session sid, and uses DH_B to compute the session key and return it. Note that in this situation, S is able to compute the session key.
- Other oracles. S acts in accordance with the protocol.

If M wins in the forging attack, he must query H_5 with the following 4-tuple

$$(g^{u_A h_A h_B u_B}, y^{e_A e_B}, ID_A, ID_B).$$

On receiving this 4-tuple, S solves the CDH problem by computing the following value as the answer

$$g^{ab} = (g^{u_A h_A h_B u_B})^{(h_A h_B)^{-1}}.$$

With probability $2/((n-1)n \cdot sc)$, M selects A and B as the owner and the peer of the test session, and additionally $session_a$ as the test session or its matching session. When M succeeds with non-negligible probability ε in computing the correct 4-tuple, S solves the CDH problem with probability at least $(2\varepsilon)/(n(n-1)sc)$, noting that when M queries H_3 with some (t, e) satisfying $u = H_3(t, e)$, and $g^u = g^a$ or $g^u = g^b$, S also solves the CDH problem.

(3) Case 3. S prepares n nodes and executes the setup stage and key extraction stage. Then S randomly selects two sessions, $session_a$ and $session_b$, executed by some honest nodes A and B, respectively.

When M asks node A or B to exchange messages in these two sessions, S sets $DH_A = g^{u_A} = g^a$, and $DH_B = g^{u_B} = g^b$, then sends (ID_A, ID_B, DH_A, r_A), (ID_A, ID_B, DH_B, r_A) to M. When M makes queries, the action of S is discussed below.

- $H_3(t, e)$: S acts as a random oracle. Additionally, once a value $u = H_3(t, e)$ is computed, S checks whether $g^u = g^a$ or $g^u = g^b$. If true, then S submits $(g^b)^u$ or $(g^a)^u$ as the answer to the CDH problem and thus completes its task.
- $H_4(ID_1, ID_2, DH)$: S acts in accordance with the protocol.
- $H_5(DH_1, DH_2, ID_1, ID_2)$: S acts as a random oracle.
- EphemeralKeyReveal(sid): If $sid = session_a$ or $sid = session_b$, S aborts its simulation, otherwise S follows the protocol.
- SessionKeyReveal(sid): If $sid = session_a$ or $sid = session_b$, S aborts its simulation, otherwise S follows the protocol.
- Other oracles. S acts in accordance with the protocol.

If M wins in the forging attack, he must query H_5 with the following 4-tuple

$$(g^{u_A h_A h_B u_B}, y^{e_A e_B}, ID_A, ID_B).$$

On receiving this 4-tuple, S solves the CDH problem by computing the following value as the answer

$$g^{ab} = (g^{u_A h_A h_B u_B})^{(h_A h_B)^{-1}}.$$

With probability at least $2/(n \cdot sc)^2$, M selects $session_a$ and $session_b$ as the test session and its matching session. When M succeeds with non-negligible probability ε in computing the correct 4-tuple, S solves the CDH problem with probability at least $(2\varepsilon)/(n \cdot sc)^2$, noting that when M queries H_3 with some (t, e) satisfying $u = H_3(t, e)$, and $g^u = g^a$ or $g^u = g^b$, S also solves the CDH problem.

(4) Case 4. According to the definition, for a clean test session, the adversary M cannot reveal both the ephemeral key and the private key of the owner and cannot reveal the private key of the peer. When M makes no PrivateKeyReveal query to the test session, the situation is the same as that of case 1. When M makes no EphemeralKeyReveal query to the test session, the situation is the same as that of case 2.

From the above, it can be concluded that if the adversary M succeeds with a non-negligible advantage, we can also solve the CDH problem with non-negligible probability, which contradicts the CDH assumption. So we complete the proof of Lemma 2. □

To summarize, the following theorem, which completes our security analysis, is also obtained.

Theorem 1. If H_3 and H_5 are random oracles, then the proposed protocol is ECK secure.

V. COMPARISON

In Table 1, we compare the efficiency, including security, computation load, and number of messages sent, of our protocol with other well-known identity-based AKE protocols. In addition, in Table 2 we compare our protocol with six other ECK-secure AKE protocols. In both tables, each number represents the number of calculation times of an operation in one execution of an AKE protocol. From Table 1, we can see that our protocol is the most secure—while others are unproven or CK-secure, our protocol is ECK-secure. In order to compare the computational requirements of these five protocols, we take into account three operations that require the most computation in each protocol, including modular exponentiation, hashing, and pairing. Note that in pairing-based protocols, the modular exponentiation operation is called point multiplication. While it does not require the pairing operation, our protocol is at a disadvantage with respect to modular exponentiation and especially the hashing operation. For communication efficiency, our protocol requires two rounds of message sending to reach the limit of the AKE protocol.

Table 1. Comparison with other identity-based AKE protocols.

Protocol	Security model	Modular exponentiation	Hashing	Pairing	Message sending
Okamoto's protocol [8]	unproven	3	0	0	2
Authenticated-DH	CK	2+	0+	0	3
Smart's protocol [11]	unproven	2	0	2	2
McCullagh's [10]	unproven	2	0	1	2
Our protocol	ECK	3	4	0	2

Table 2. Comparison with other ECK-secure AKE protocols.

Protocol	Security model	Modular exponent.	Hashing	Identity based
NAXOS [19]	ECK	4	3	no
E-NAXOS [20]	ECK	5	2	no
E-NETS [21]	ECK	3	2	no
Huang's protocol [22]	ECK	6	2	no
NETS [23]	ECK	3	3	no
NAXOS+ [24]	ECK	5	4	no
Ours	ECK	3	4	yes

From Table 2, it can be concluded that our protocol only requires three modular exponentiation operations, which is the least among these protocols. But the hashing operation that our protocol requires is slightly more than some of the other protocols. More importantly, our protocol is the only identity-based one among these protocols.

VI. CONCLUSION

In this paper, we present an identity-based authenticated key-exchange protocol for building a secure channel in satellite communication. In our protocol, the KGC is a ground control center and every network node should require a private key as well as public material from the KGC. In the key-exchange stage, two nodes compute a shared key after receiving a message from each other. We demonstrate that our protocol is suitable for satellite communication in two aspects. Unlike many previous identity-based AKE schemes, our protocol does not require any bilinear pairing operation, and thus, it helps achieve high computational efficiency; in addition, each session of our protocol requires only two rounds of message sending. We demonstrate that our protocol is ECK-secure in the random oracle model, under the CDH assumption. Therefore, our protocol improves communication security in satellite networks.

REFERENCES

[1] J. Warner and R. Johnston, "A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing," *J. Security Admin.*, pp. 19–28, 2002.

[2] G. X. Gao, D. Lorenzo, T. Walter, and P. Enge, "Acquisition and tracking of GIOVE-a broadcast L1/E5/E6 signals and analysis of DME/TACAN interference on receiver design," in *Proc. ENC Global Navig. Satellite Syst. Conf.*, Geneva, Switzerland, May 2007.

[3] G. X. Gao, D. Lorenzo, A. Chen, S. Lo, D. Akos, T. Walter, and P. Enge, "Galileo GIOVE-a broadcast E5 codes and their application to acquisition and tracking", *ION Nat. Tech. Meeting*, San Diego, California, Jan. 2007.

[4] G. X. Gao, A. Chen, S. Lo, D. Lorenzo, and P. Enge, "Compass-M1 broadcast codes in E2, E5b, and E6 frequency bands," *IEEE J. Sel. Topics. Signal Process.*, vol. 3, pp. 599–612, 2009.

[5] A. Shamir, "Identity-based cryptosystems and signature schemes.," in *Proc. Advances in Cryptology-Crypto*, Berlin: Springer-Verlag, 1984, pp. 47–53.

[6] M. Girault and J. C. Pailles, "An identity-based scheme providing zero-knowledge authentication and authenticated key exchange," in *Proc. European Symp. Research Comput. Security*, Oct. 1990, pp. 173–184.

[7] C. Gunther, "An identity-based key exchange protocol." in *Proc. EUROCRYPT*, 1989, pp. 29–37.

[8] E. Okamoto, "Distribution systems based on identification information." In *Proc. CRYPTO.*, vol. 293, 1987, pp. 194–202.

[9] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings," in *Proc. IEEE Comput. Security Found. Workshop*, 2003, pp. 219–233.

[10] N. McCullagh and P. S. L. M. Barreto, "A new two-party identity-based authenticated key agreement." in *Proc. CT-RSA*, 2005, pp. 262–274.

[11] N. P. Smart, "Identity-based Authenticated key agreement protocol based on weil pairing," *IET. Electron. Lett.* vol. 38, no. 13, pp. 630–632, 2002.

[12] P. S. L. M. Barreto, H. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing based cryptosystems." in *Proc. Advances in Cryptology-Crypto*, 2002, pp. 354–368.

[13] P. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," in *Sel. Areas in Crypt.*," SAC., pp. 17–25, 2003.

[14] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols", in *Proc. ACM Symp. on Theory Comput.*, 1998, pp. 419–428.

[15] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol.22, no. 6, pp. 644–654, 1976.

[16] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," *Lecture Notes Comput. Sci.*, Springer-Verlag, vol. 2045, pp. 453–474, 2001

[17] R. W. Zhu, G. Yang, and D. S. Wong, "An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices," *Theoretical Comput. Sci.*, pp. 198–207, 2007.

[18] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. CRYPTO*, 1993, pp. 232-249.

[19] B. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," *Lecture Notes Comput. Sci.*, vol. 4784, Heidelberg: Springer, pp. 1–16, 2007.

[20] Q. Cheng, C. Ma, and X. Hu, "A new strongly secure authenticated key exchange protocol," in *Proc. ISA*, 2009, pp. 135–144.

[21] Q. Cheng, G. Han, and C. Ma, "A new efficient and strongly secure authenticated key exchange protocol," in *Proc. ISA*, 2009, pp. 499–502.

[22] H. Huang and Z. Cao. (2008). Strongly secure authenticated key exchange protocol based on computational Diffie-Hellman problem. Cryptology ePrint Archive. [Online]. Available: http://eprint.iacr.org/2008/500

[23] J. Lee and C. S. Park. (2008). An efficient authenticated key exchange protocol with a tight security reduction. Cryptology ePrint Archive. [Online]. Available: http://eprint.iacr.org/2008/345.pdfl

[24] J. Lee and J. H. Park. (2008). Authenticated key exchange secure under the computational Diffie-Hellman assumption. Cryptology ePrint Archive. [Online]. Available: http://eprint.iacr.org/2008/344.pdfl

Zhong Yantao received the B.S. degree and the M.S. in Computer Science School of Sichuan University, Chengdu, China in 2001 and 2005, respectively. He is currently pursuing his Ph.D. degree in the Computer Science School of Xidian University, Xi'an, China. His research interests include information security and cryptography.

Ma Jianfeng received his B.S. degree in Mathematics from Shaanxi Normal University, Xidian, China, in 1985, and obtained his M.S. and Ph.D. degrees in Computer Software and Communications Engineering from Xidian University, China, in 1988 and 1995, respectively. Since 1995 he has been with Xidian University as a Lecturer, Associate Professor and Professor. From 1999 to 2001, he was with Nanyang Technological University of Singapore as a Research Fellow. Currently, he is the Director of the Key Laboratory of Computer Networks and Information Security (Ministry of Education). His research interests include information security, coding theory, and cryptography.