Anonymous authentication protocol for multi-services in wireless environments

JIANG Nan (✉), WANG Jian

1. College of Computer Science, Beijing University of Technology, Beijing 100124, China
2. School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
3. Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract

To provide mutual authentication among users, wireless networks, and service providers in roaming wireless environments, this article presents an anonymous authentication and access control protocol. Bases on this protocol, entities of different trusted domains can mutually authenticate each other and preserve the anonymity of users. Hybrid cryptosystem, secret splitting, and hash chains are used in the protocol, which decrease computational loads and establish trusted relations for both entities. The proposed protocol has the least computation complexity compared with other protocols, whereas, the security has been significantly improved.

Keywords authentication, wireless networks, trusted domain, secret splitting, hash chain

1 Introduction

Roaming users access different wireless networks and get application services in the visited networks. The security of access control for both wireless networks and services is the basic security requirement for roaming service and application service, and the security is based on mutual authentication [1–7]. On the access security of wireless networks [1–4], every network has its own authentication center, and the authentication center provides the authentication service for home users. When the user roams into other networks, the authentication center of the visited network authenticates the roaming user with the help of the user’s home network. On the service authentication [5–7], the application service server authenticates the roaming user through his own authentication center and the user’s home authentication center. As roaming users are in the visited networks, first, they need to be authenticated in the visited networks. After accessing the networks, they also need to authenticate service providers to get services. In the visited networks, for the security of roaming and application services, which are dependent on the security services, the basis for security requirements of security service includes:

1) Mutual authentication among the roaming user, home network, visited networks, and the application service server;
2) Session key agreement and key renewal;
3) Anonymity and privacy of user;
4) Data confidentiality and data integrity.

A combination of different wireless networks to build a personal communication network is the trend of developing wireless networks. Therefore, although there are many trusted domains in the wireless networks, the entities of different domains have no trusted relation. How to effectively build a trusted relation that becomes the keystone of roaming authentication research is a problem. The main difficulties of this research are as follows:

1) Authentication between different trusted domains: The roaming user has no trusted relation with the visited networks or application service server, although symmetric cryptosystem and public key cryptosystem are used to design the authentication protocols. However security features and computational load need to be improved.
2) Efficiency and computational load of authentication protocol: In wireless environments, there are two things to consider when authentication protocols are being designed. One is that mobile devices have a low computational power.
The other is that wireless networks have a lower communication bandwidth than wire networks.

The authors present an anonymous authentication and access control protocol, and thus entities of different trusted domains can mutually authenticate each other. The proposed protocol works for the security requirements of roaming services and application services. Hybrid cryptosystem is used in this proposed protocol, and a smart card is used for the security module of the user. This protocol is based on two cryptographic techniques, secret splitting and hash chain, which support the anonymity of the user and establish a trusted relation between the user and the visited network. The key renewal mechanism renew the session key for each session; therefore, this reduces the risk of using a compromised session key to communicate with visited networks or application service servers. The protocol requires a lower computational power and communication bandwidth than other existing protocols, and improves the security features significantly.

The remainder of the article is organized as follows. In Sect. 2, previous authentication protocols are reviewed, and the two cryptographic techniques used in this protocol are introduced. In Sect. 3 the proposed protocol is presented. Subsequently, the security and performance of the proposed protocol is discussed in Sect. 4, and finally, in Sect. 5 conclusions are given.

## 2 Background

### 2.1 Related work

Many protocols have been presented to solve the authentication of roaming services and application services. These protocols are based on a symmetric cryptosystem or a public key cryptosystem. Suzuki [1] presents and analyzes a protocol, which is the same as the protocol used in the global system for mobile communication (GSM). Symmetric cryptosystem and challenge/response mechanism are used in the protocol, in which, however, the user lacks authentication to the networks, thus there are many potential attacks as the intruder can impersonate the visited networks. Zhu [2] presents a new authentication protocol with anonymity, and the protocol is based on a smart card. Lee [3] improves the security features for the Zhu [2] protocol. Jiang [4] proposes two authentication and key exchange protocols with anonymity for roaming services, while these protocols require similar computational power as the other protocols. Horn [5] defines the authentication and payment initialization protocol’s security goals and analyzes many candidate protocols, most of which are based on the public key cryptosystem, while in practice, the computational complexity and communication bandwidth should be decreased. Ren [6] utilizes the blind signature technique in the present application service authentication protocol for pervasive computing environments, and allows differentiated service access control. The main shortcoming in Ren’s protocols is that the session key never renews after mutual authentication between the user and the server. Peng [7] presents an application service authentication protocol based on identity public key cryptosystem for different trusted domains, but this protocol is not fit for wireless devices because of high computational loads. All the previous studies do not consider the authentication of roaming service and application service. Thus, the authors designed a new authentication protocol for both roaming service and application service. This proposed protocol is more effective than the two independent protocols for roaming service and application service.

### 2.2 Cryptographic techniques

The two cryptographic techniques used in the proposed protocol are depicted as follows.

A one-way hash function takes a message of arbitrary size as its input, and outputs a fixed length string of digits. It is computationally unfeasible to derive the original input from the output. \( H() \) denotes a one-way hash function, \( m \) denotes message, and \( n \) is an integer. By using \( H() \) repeatedly on \( m \), a chain of outputs \( \{H^1(m), H^2(m), \ldots, H^n(m)\} \) [8] can be computed. A hash chain has many properties.

1) One is the one-way property. \( H^1(m) = H[H^{j-1}(m)] \), \( j \leq n \), otherwise, if one only knows \( H^1(m) \), one cannot compute \( H^{j-1}(m) \).

2) Another is the authentication property. \( H^{j-1}(m) \) can be proven to be authentic if \( H^1(m) \) has been proven to be authentic. Lamport [8] first proposes the concept of hash chain as a password authentication scheme. Recently, hash chain has been widely used in micropayment [9–10]. In the protocol proposed by the authors, hash chain is used for secrets between users and visited networks after \( H^1(m) \) is authenticated by the visited networks, although there is mutual authentication between users and visited networks. The computational load is decreased by using the hash chain technique.

The secret splitting technique divides the message into pieces, where each piece alone has no meaning, and it is only when all pieces are put together that the original message can be restored [11]. Secret splitting has a lower computational load and absolute data confidentiality. Secret splitting is widely used for user’s anonymity [2–4] and electronic cash [12]. The authors have used the secret splitting technique to design a new
3 System model and protocol

The system model in wireless environment has three entities. The first entity is the user (U) who uses various terminals to transfer data. The second entity is the authentication center (AC). The third entity is the application service (AS).

Figure 1 represents the relation of entities in the system model. There are two goals of this authentication protocol. The first is that the user establishes a trusted relation with the VAC and a session key is agreed upon. The second is that the user authenticates the roaming user through the user’s home authentication center (HAC). After successful validation, the user establishes a trusted relation between the VAC and the HAC and establishes a session key after successful validation.

3.1 Notations

Notations are used as follows:

- \( I_X \): Identity of an entity \( X \);
- \( (K_{\text{PKES}}, K_{\text{PKS}}) \): The pair of public key and secret key of an entity \( X \);
- \( K_{\text{AB}} \): Session key between an entity \( A \) and an entity \( B \);
- \( T_X \): A timestamp generated by the entity \( X \);
- \( R_X \): A random nonce generated by the entity \( X \);
- \( E_K ( ) \): Symmetric encryption / decryption using a key \( K \);
- \( S_k ( ) \): Message signed by using a key \( K \);
- \( C_K \): Certificate of an entity \( X \);
- \( D_K \): A temporary identity of an entity \( X \);
- \( P \): The secret parameter between AC and U.

3.2 Anonymous authentication and access control protocol

In this section, the authors present an anonymous authentication and access control protocol (AAACP).

According to the system model, the protocol includes two phases. In phase I, the VAC mutually authenticates a roaming user through the user’s HAC, and after successful validation, a user establishes trusted relation with the VAC and a session key is agreed upon. In the subsequent communication sessions, the one-time session key renewal mechanism assures the mutual authentication and freshness of the session key. In phase II, using the trusted relation established in phase I, a roaming user mutually authenticates the AS through the VAC, and establishes a session key after successful validation. The user can renew the session key with the AS and get services from the AS directly.

3.2.1 Initialization of protocol

This protocol uses the hybrid cryptosystem, and all entities are initialized before the protocol is executed. It is assumed that every network has a centralized AC. The AC can issue certificates to other entities and issue smart cards to users in its network. Another assumption is that each of the different ACs shares a secret key for protection of messages among them. The key \( K_{\text{IV}} \) is shared between the HAC and the VAC.

The AC issues \( I_{\text{AS}}, K_{\text{PKES}}, K_{\text{PKS}}, C_{\text{AC}}, I_{\text{AC}}, \) and \( K_{\text{AC}} \) to every AS, for which \( K_{\text{PKES}} \) is the secret key of encryption, and \( K_{\text{PKS}} \) is the secret key of sign. \( K_{\text{PKES}} \) and \( K_{\text{PKS}} \) should be stored safely.

An AC issues a smart card to the user through secure channels. In the smart card, it stores \( I_U, P, I_{\text{AC}}, C_{\text{AC}}, \) and \( K_{\text{AC}} \). \( P \) is computed by \( H\left(R_{\text{AC}} \oplus I_{\text{AC}}\right) \oplus \) \( I_U \), and \( I_U, P, \) and \( K_{\text{AC}} \) should be stored safely.

3.2.2 Phase I: anonymous access control protocol

The goals of anonymous access control protocol (AACP) are to provide mutual authentication between the roaming user and the VAC and to establish a trusted relation between them. The authors use the hash chain technique for the authentication secret, and then the VAC can authenticate a user directly. Secret splitting is used for designing a new temporary identity for roaming users’ anonymity. The AACP is shown as follows:

**Step 1** \( U \rightarrow \text{VAC} : D_U, R_U, E_{\text{KUAC}}(R_U, I_{\text{VAC}}, H^*(R_{U0})), I_{\text{VAC}}, T_U \).

**Step 2** \( \text{VAC} \rightarrow \text{HAC} : D_U, R_{\text{VAC}}, R_U, T_{\text{VAC}}, E_{\text{KUAC}}(R_U, R_{U0}, I_{\text{VAC}}, H^*(R_{U0})), H^*(R_{U0}), H^*(R_{U0}), H^*(R_{U0}), H^*(R_{U0}) \).

**Step 3** \( \text{HAC} \rightarrow \text{VAC} : E_{\text{KIV}}(R_{\text{VAC}}, H^*(R_{U0}), H(I_U \oplus I_{\text{VAC}})), D_U, H_{\text{KIV}}, D_U, stE_{\text{KIV}}[E_{\text{KUAC}}(R_{\text{VAC}}, H^*(R_{U0})), H(I_U \oplus I_{\text{VAC}})] \).

**Step 4** \( \text{VAC} \rightarrow U : R_{\text{VAC}}, E_{\text{KUAC}}(R_{\text{VAC}}, R_{U0}, I_{\text{VAC}}, K_{\text{UVAC}}) \).

The authors describe the proposed AACP according to the exchanged messages as follows:

1) In step 1, when \( U \) enters a new visited network, \( U \) generates random nonce \( R_{U0}, R_{U0} \), and \( R_{U1} \), and computes a hash chain \( [H'(R_{U0}), H''(R_{U0}), \ldots, H^{n-1}(R_{U0}), H^n(R_{U0})] \). \( D_U \) is
the temporary identity of U, \( D_U = P \oplus H(R_{U1} \oplus I_{HAC}) \). \( D_U \) encrypts \( R_{U1}, R_{U2}, I_{VAC} \) and \( H^J(R_{U0}) \) with \( K_{VAC} \), and then sends the access request to the VAC to identify himself as a legal subscriber.

2) After the VAC receives the access request, it checks whether the time stamp \( T_v \) is within allowable range compared with its current time. If the VAC validates \( T_v \), then it generates a random nonce \( R_{VAC} \) and stores \( D_U \) and \( R_{U0} \). Finally, the VAC computes the message authentication code \( H_{K_{VAC}}(.) \) by using \( K_{VAC} \), and then sends the authentication request to the HAC.

3) On receiving the message from the VAC, the HAC checks the time stamp \( T_{VAC} \) and \( H_{K_{VAC}}(.) \). The HAC decrypts \( E_{K_{VAC}}(R_{VAC}, R_{U0}, I_{VAC}, H^J(R_{U0})) \) and then compares if \( R_{U0} \) is the same in the receiving message. At the same time, The HAC obtains \( U \)'s real identity by computing: \( H(R_{U0} \oplus I_{HAC}) \), then \( I_2 = D_2 \oplus H(R_{U0} \oplus I_{HAC}) \oplus H(R \oplus I_{HAC}) \). The HAC stores \( R_{VAC}, R_{U0}, H^J(R_{U0}) \) safely. Subsequently, the HAC encrypts \( R_{VAC}, H^J(R_{U0}), \) and \( H(I_2 \oplus R_{U0}) \) with \( K_{VH} \) and computes a message authentication code \( H_{K_{VH}}(.) \) of a response message by using \( K_{VH} \), and finally replies with a response message to the VAC.

4) After receiving the response message from the HAC, the VAC checks \( H_{K_{VH}}(.) \), and decrypts \( E_{K_{VH}}(R_{VAC}, H^J(R_{U0}), H(I_2 \oplus R_{U0})) \), and then compares if \( R_{VAC} \) is the same in the receiving message. Finally, the VAC stores \( H^J(R_{U0}), H(I_2 \oplus R_{VAC}) \). Next, the VAC computes \( K_{VUAC} = H^J(R_{U0}) \oplus H(I_2 \oplus R_{VAC}) \) and encrypts \( R_{U0}, R_{VAC}, I_{VAC}, K_{VUAC} \). It then sends the message to \( U \). When \( U \) has received the message, it computes new \( K_{VUAC} \), \( K_{VUAC} = H^J(R_{U0}) \oplus H(I_2 \oplus R_{VAC}) \) and decrypts \( E_{K_{VUAC}}(R_{VAC}, R_{VAC}, I_{VAC}, K_{VUAC}) \).

After the AACP is executed, \( U \) establishes the trusted relation with the VAC, and \( H^J(R_{U0}) \) is authenticated by the VAC and stored safely in it. Subsequently, \( U \) uses a hash chain to mutually authenticate VAC without the help of the HAC.

Consequently, when \( U \) authenticates the VAC again, the exchanged messages are shown as follows:

**Step 5** \( U \rightarrow \text{VAC}: D_U, R_{U0}, E_{K_{VUAC}}(R_{VAC}, H^J(R_{U0})) \).

**Step 6** \( \text{VAC} \rightarrow U: E_{K_{VUAC}}(R_{U0}, K_{VUAC}) \).

In step 5, \( U \) uses the current \( K_{VUAC} \) to encrypt \( R_{VAC}, H^J(R_{U0}) \), in which \( R_{U0} \) is a new random nonce generated by \( U \), and \( H^J(R_{U0}) \) is a future value of a hash chain. The VAC decrypts the message. It then computes \( H[H^J(R_{U0})] \) if the value is the same as \( H^{J+1}(R_{U0}) \). Next, \( H^J(R_{U0}) \) is substituted for \( H^{J+1}(R_{U0}) \). In step 6, the VAC computes the new \( K_{VUAC}, K_{VUAC} = H^J(R_{U0}) \oplus H[I_2 \oplus R_{VAC}] \) and sends a response message to \( U \). After receiving the message, \( U \) computes the new \( K_{K_{VUAC}} \), decrypts the message, and checks if \( K_{VUAC} \) is the same as the computed value.

### 3.2.3 Phase II: authentication for multi-services protocol

When the AACP is successfully executed, the VAC mutually authenticates \( U \) and they establish a trusted relation with each other. As \( U \) tries to get the application services in the visited networks, the AS needs to authenticate \( U \) first. Using the trusted relation between the VAC and \( U \), the AS can authenticate \( U \) through the VAC. This scheme decreases the complexity of entity authentication in different trusted domains.

The authentication for multi-services protocol (AMSP) is described as follows:

**Step 1** \( U \rightarrow \text{AS}: D_U, R_{U0}, E_{K_{VUAC}}(R_{VAC}, I_{AS}, H^J(R_{U0})), T_{U0} \).

**Step 2** \( \text{AS} \rightarrow \text{VAC}: D_U, R_{U0}, E_{K_{VUAC}}(R_{VAC}, I_{AS}, H^J(R_{U0})), R_{AS}, T_{AS} \).

**Step 3** \( D_U \rightarrow \text{AS}: D_U, R_{U0}, E_{K_{VUAC}}(R_{VAC}, I_{AS}, H^J(R_{U0})), R_{AS}, T_{AS}, \rho_{K_{VUAC}} \).

**Step 4** \( U \rightarrow \text{AS}: E_{K_{VUAC}}(R_{VAC}, K_{UAS}, K_{VUAS}). \)

1) When \( U \) tries to get an application service, he sends a request message to the AS, to authenticate that he is a legal user in the visited networks by the VAC. In step 1, \( U \) generates a new random nonce \( R_{U0} \), encrypts \( R_{U0}, I_{AS} \), and \( H^J(R_{U0}) \) with the key \( K_{VUAC} \), and adds the time stamp \( T_{U0} \) in the sending message.

2) After receiving the message from \( U \), the AS checks whether the time stamp \( T_{AS} \) is within allowable range compared with its current time. If the AS validates \( T_{AS} \), then the AS records \( D_U, R_{U0}, \) and generates a random nonce \( R_{AS} \) and a time stamp \( T_{AS} \). The AS computes \( \rho_{K_{VUAC}} \) and \( E_{K_{VUAC}}(R_{VAC}, I_{AS}, H^J(R_{U0})), R_{AS} \). Finally, the AS sends the message to the VAC for identity authentication.

3) When the VAC receives the message, it checks the time stamp \( T_{AS} \) and \( D_U \) for correctness, then verifies the signature of the AS, \( \rho_{K_{VUAC}} \), and \( E_{K_{VUAC}}(R_{VAC}, I_{AS}, H^J(R_{U0})), R_{AS} \). The VAC decrypts \( E_{K_{VUAC}}(R_{VAC}, I_{AS}, H^J(R_{U0})), R_{AS} \) using \( K_{VUAC} \). If the decrypted \( R_{U0} \) is equal to the received value \( R_{U0} \), then the VAC can compute \( E_{K_{VUAS}}(R_{VAC}, K_{UAS}) \) using \( K_{VUAS} \), and encrypt \( R_{U0}, K_{UAS}, K_{VUAC} \) using \( K_{VUAS} \). The key \( K_{UAS} \) is a shared key between \( U \) and the AS that is generated by the VAC. \( K_{VUAS} \) is the session key between \( U \) and the AS, for which the renewal is the same as in the AACP.

4) In step 4, the AS verifies \( \rho_{K_{VUAC}} \) and \( E_{K_{VUAC}}(R_{VAC}, K_{UAS}, K_{VUAC}) \). If it is correct, then the AS decrypts \( E_{K_{VUAS}}(R_{VAC}, K_{UAS}) \). If the decrypted \( R_{AS} \) is the same as the original value, the AS records \( K_{UAS} \) safely. Subsequently, the AS transfers \( E_{K_{VUAC}}(R_{VAC}, K_{UAS}, K_{VUAC}) \) to \( U \). After receiving the new \( K_{VUAC} \), \( K_{VUAC} = H^J(R_{U0}) \oplus H[I_2 \oplus R_{VAC}] \). Also, \( U \) compares the received value with the expected value.
the response of HAC and authenticates to 
safely.
AMSP is successfully executed after the AS mutually authenticates U by VAC and establishes a shared session key 
UAS with U. The key renewal mechanism is shown as follows:
Step 5 \( U \rightarrow AS: DU, E_{KUAS}(R_i, K_UAS) \).
Step 6 AS \( \rightarrow U: E_{KUAS}(R_i, K_UAS) \).

In Step 5, to get service from the AS, U generates a new 
random nonce \( R_i \), and encrypts \((K_{UAS}, R_i)\), using the current 
K_UAS. After the AS has received the message from U, the AS 
decrypts \( E_{KUAS}(R_i, K_{UAS}) \) using K_UAS, which is indexed by \( DU \).
If the decrypted K_UAS is equal to the current K_UAS, then 
the message can be authenticated by the AS.

Subsequently, the AS computes a new \( K_{UAS}: K_{UAS} = R_i \oplus 
K_{UAS} \) and \( E_{KUAS}(R_i, K_{UAS}) \). Then the AS sends \( E_{KUAS}(R_i, K_{UAS}) \) 
to U. U also computes the new K_UAS and decrypts \( E_{KUAS}(R_i, 
K_{UAS}) \). If the decrypted K_UAS is equal to the computed value, 
then the new K_UAS substitutes for the current K_UAS.

4 Analysis for the protocol
4.1 Security analysis

This protocol has many security properties to provide 
mutual authentication, users’ anonymity, and one-time key 
renewal for roaming services and application services.

1) Anonymity of user

In this protocol, the anonymity of a user is obtained by 
secret splitting, one-way hash function, and smart card. Thus 
the VAC, AS or the malicious attackers cannot intercept the 
real identity of roaming users. The authors use \( DU \) to 
substitute for the real \( IU \) \( DU = P \oplus H(RU \oplus I_{HAC}) \). Otherwise, the 
HAC can obtain the real \( IU \) from \( DU \) to authenticate 
roaming users.

Secret splitting and one-way hash function are used to 
generate \( DU \). Because \( IU = H(RU \oplus I_{HAC}) \oplus H(RU \oplus I_{HAC}) \oplus DU \), 
to obtain \( IU \), one must have \( H(RU \oplus I_{HAC}) \), \( H(R \oplus I_{HAC}) \) and 
\( DU \) however, only \( U \) and HAC can obtain \( H(RU \oplus I_{HAC}) \), 
\( H(R \oplus I_{HAC}) \).

2) Mutual authentication

The AACP provides mutual authentication between a 
roaming user and the VAC, and establishes a trusted relation 
between them. In AMSP, the AS can mutually authenticate \( U 
\) by the VAC.

As shown in Sect. 3.2.2, in step 2 of the AACP, the VAC 
sends the authentication request to the HAC, the HAC 
decrypts \( E_{KUAC}(RU, RU, IU, VAC, H'(R_{10})) \) and checks \( RU \) and 
the message authentication code. If the HAC validates the request, 
it authenticates U and the VAC. In step 3, the VAC validates 
the response of HAC and authenticates U by the HAC. Finally, 
\( U \) validates \( E_{KUAC}(RU, RU, IU, VAC, K_{UAC}) \) and authenticates 
the HAC and the VAC.

As shown in Sect. 3.2.3, after receiving the request from 
the AS, the VAC verifies the signature of the AS and the 
request message, then it authenticates the AS, and the AS 
authenticates \( U \) by the VAC in step 3. Finally, \( U \) validates the 
received message that authenticates the VAC and the AS in 
step 4.

3) Session key agreement and key renewal

In AACP, \( K_{UAC} \) is the session key between \( U \) and VAC, 
which is established during the authentication of \( U \) and the 
VAC. The key renewal mechanism assures the freshness of 
\( K_{UAC} \). \( K_{UAC} = H'(RU) \oplus H(IU \oplus RU VAC) \), \( H'(RU) \) is the new 
hash chain value that will be used in the following 
authentication sequences.

In AMSP, the VAC generates the session key \( K_{UAS} \) between 
AS and \( U \). In subsequent communication, \( K_{UAS} \) renews 
every transaction between the AS and \( U \); \( K_{UAS} = R_i \oplus K_{UAS} \), 
for \( U \) generates the random nonce \( R_i \).

4) Prevention of replay attack and fraud attack

In the proposed protocol, random nonce challenge/response 
mechanism and time stamp mechanism are used to prevent 
replay attack. Old messages cannot be validated by entities 
for reusing random nonce and time stamps. At the same time, 
random nonce challenge/response mechanism, time stamp 
mechanism, and mutual authentication assure that fraud 
attacks are not successful.

In Table 1, the authors compare the security feature of this 
the partial anonymity of a user, because the temporary 
identity for a user is not changed during the lifetime of the 
authentication, because the mobile user and the visited 
network cannot authenticate each other. Ren [6] uses 
certificates in his protocol and the protocol does not have 
anonymity of users. Also, Ren [6] does not provide the key 
renewal mechanism. By comparison, this protocol has more 
security features than other protocols.

| Table 1 Security feature comparison with AAACP |
|-----------------|-----------------|------------------|-----------------|------------------|
| Anonymity of user | Partial | Partial | No | Yes |
| Mutual authentication | Yes | Partial | Yes | Yes |
| Session key agreement | Yes | Yes | No | Yes |
| and key renewal | Yes | Yes | Yes | Yes |
| Prevention of replay attack | Yes | Yes | Yes | Yes |

4.2 Performance analysis

In this section, the authors analyze the performance of 
the proposed protocol by comparing it with other protocols. The 
proposed AAACP is divided into two phase protocols, AACP
and AMSP. The two phase protocols have different goals of authentication, thus they compare the two phase protocols with different protocols that have the same goals of authentication.

The performance of access authentication without including cost of key renewal is compared. As shown in Table 2, the values in parentheses denote the cost of user computation. Table 2 shows that this protocol is much more practical and efficient in terms of computation cost and communication cost. The user’s cost is the least compared with other protocols. In this protocol, a user only uses two messages to mutually authenticate with other entities and establish the session key. Parameter \( n \) denotes the length of a hash chain that can change with security requirements, and the hash chain can be computed before using it.

### Table 2 Performance comparison with AACP

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponential operation</td>
<td>—</td>
<td>—</td>
<td>5 (1)</td>
<td>—</td>
</tr>
<tr>
<td>Public key encryption</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Public key decryption</td>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Signature operation</td>
<td>2</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Validation signature</td>
<td>2</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Validation certificate</td>
<td>2</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Symmetric encryption</td>
<td>2 (1)</td>
<td>5 (2)</td>
<td>3</td>
<td>3 (1)</td>
</tr>
<tr>
<td>Symmetric decryption</td>
<td>2 (1)</td>
<td>5 (1)</td>
<td>3 (1)</td>
<td>—</td>
</tr>
<tr>
<td>Hash operation</td>
<td>5 (2)</td>
<td>2</td>
<td>2</td>
<td>6+5*n (1)</td>
</tr>
<tr>
<td>Exchange messages</td>
<td>( U, V )</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>( H, V )</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

The authors compare the performance of application service authentication in Table 3. The cost of AMSP is lower than Ren’s protocol [6]. The AMSP works with the first phase protocol AACP, which decreases the computational its load.

### Table 3 Performance comparison with AMSP

<table>
<thead>
<tr>
<th>Operation</th>
<th>Ren [6]</th>
<th>AMSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public key encryption</td>
<td>2 (2)</td>
<td>1</td>
</tr>
<tr>
<td>Public key decryption</td>
<td>1</td>
<td>—</td>
</tr>
<tr>
<td>Symmetric encryption</td>
<td>1</td>
<td>2 (1)</td>
</tr>
<tr>
<td>Symmetric decryption</td>
<td>1 (1)</td>
<td>2 (1)</td>
</tr>
<tr>
<td>Signature operation</td>
<td>2 (1)</td>
<td>2</td>
</tr>
<tr>
<td>Validation signature</td>
<td>1</td>
<td>—</td>
</tr>
<tr>
<td>Validation certificate</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Hash operation</td>
<td>( n+5 (n+3) )</td>
<td>—</td>
</tr>
<tr>
<td>Exchange messages of user</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

### 5 Conclusions

Now, there has been a trend where various wireless networks gather in a large combination network. Subsequently, users belonging to different home networks can roam into other networks and visit their application services. Thus, entity authentication among different networks becomes important to protect the security of users, networks and application services. To protect the security of roaming services and application services, the authors present an anonymous authentication and access control protocol and thus the entities of different trusted domains can mutually authenticate each other and preserve the anonymity of users. Secret splitting and hash chain are used in the protocol, which decrease the computational loads and establish a trusted relation between both entities. The proposed protocol has the least computation complexity compared with other protocols, while the security has been significantly improved.

### Acknowledgements

This work was supported by the National Natural Science Foundation of China (60775010), the National Basic Research Program of China (2007CB311100), the Doctor Science Research Foundation of BJUT (52007016200701), the Excellent Person Development Foundation of Beijing (20061D0501500191), the Foundation for Talents of BJTU (2008RC028).

### References