Secure anonymous authentication scheme without verification table for mobile satellite communication systems

Jia-Lun Tsai¹, Nai-Wei Lo¹,∗,*† and Tzong-Chen Wu¹,²

¹Department of Information Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan
²Taiwan Information Security Center, National Taiwan University of Science and Technology, Taipei 106, Taiwan

SUMMARY

An authentication scheme is one of the most basic and important security mechanisms for satellite communication systems because it prevents illegal access by an adversary. Lee et al. recently proposed an efficient authentication scheme for mobile satellite communication systems. However, we observed that this authentication scheme is vulnerable to a denial of service (DoS) attack and does not offer perfect forward secrecy. Therefore, we propose a novel secure authentication scheme without verification table for mobile satellite communication systems. The proposed scheme can simultaneously withstand DoS attacks and support user anonymity and user unlinkability. In addition, the proposed scheme is based on the elliptic curve cryptosystem, has low client-side and server-side computation costs, and achieves perfect forward secrecy. Copyright © 2013 John Wiley & Sons, Ltd.

Received 19 August 2012; Revised 1 September 2013; Accepted 18 September 2013

KEY WORDS: authentication scheme; denial of service attack (DoS attack); mobile satellite communication systems; user anonymity; user unlinkability

1. INTRODUCTION

With the rapid development of satellite communication systems [1–7], an increasing number of mobile users are employing these systems for interpersonal communication. For traditional satellite communication systems, a geostationary satellite is responsible for establishing communication channels among mobile users in the geosynchronous equatorial orbit (GEO). However, because the distance between the geostationary satellite and the earth is approximately 22,300 miles, it has a signal latency problem. Thus, several non-GEO satellite communication systems have been developed such as low-earth-orbit (LEO) satellite communication systems [3, 6, 7]. Although LEO satellite communication systems are convenient for mobile users, serious problems result from this convenience. Specifically, the following two satellite communication system problems require resolution: (1) Security is a basic requirement for satellite communication systems. Messages that are transmitted using satellite communication systems are always transmitted via insecure wireless channels. Thus, adversaries with sufficient resources might be able to modify or intercept these messages before they reach the intended destination. (2) Because mobile devices have low computational capacity compared with personal computers, adaptive protocols cannot place high computation demands on the client side. Thus, the majority of security protocols are unsuitable for satellite communication systems because they have heavy computation costs on the mobile user side.

An authentication scheme is very crucial for satellite communication systems because it prevents illegal access by an adversary. In LEO satellite communication systems, users are required to register on the network control center (NCC). Figure 1 shows the general authentication system architecture.
using NCC in mobile satellite framework. When one mobile user wants to communicate with another mobile user, they must be authenticated by the NCC. After authentication, a secure communication link is established between the NCC and the mobile user. The security and efficiency problems of satellite authentication schemes have motivated us to develop a secure and efficient authentication protocol that is more suitable for mobile satellite communication systems. In general, a secure authentication scheme for satellite communication systems should satisfy the following security requirements [8–17]: (1) resistance against impersonation attack. A server or a user must have the ability to verify whether the login request is sent from an adversary; (2) resistance against denial of service (DoS) attack. A DoS attack is an attempt to make a legitimate user incapable of accessing a remote server; (3) resistance against smart card loss attack. If an adversary obtains a legitimate smart card, he or she may attempt to retrieve sensitive information from the compromised smart card. Thus, a strong authentication scheme must withstand a smart card loss attack; (4) resistance against stolen-verifier attack. Even if a server verification table is stolen by an adversary, this malicious adversary cannot obtain any secret key or user password from the compromised verification table. To avoid such an attack, several authentication schemes without verification table have been proposed; and (5) resistance to replay attack. An adversary who has learned all of the previous authentication messages may attempt to resend these messages during additional authentication processes. In addition, a strong authentication scheme should provide the following functional requirements [8–17]: (1) provision of mutual authentication. A strong authentication scheme must allow users and servers to authenticate each other; (2) provision of user anonymity. This prevents an adversary from retrieving any identity information from the authentic messages; (3) provision of user unlinkability [15–17]. An adversary cannot know whether the same user logs into a server twice; and (4) provision of perfect forward secrecy. Perfect forward secrecy prevents an adversary from compromising the previous session keys, even if all user secret keys are compromised by the adversary.

1.1. Related works

For satellite communication systems, implementing a secure authentication system is one of the fundamental requirements. Several authentication schemes have recently been proposed. In 1996, Cruickshank [8] first proposed an authentication scheme for satellite communication systems. However, the computational costs of this scheme of Cruickshank are high, making it unsuitable for mobile devices with limited computational power. In 2003, Hwang et al. [9] proposed an efficient authentication scheme using secret-key cryptosystems for mobile satellite communication systems. However, Chang and Chang [11] observed that their scheme does not provide perfect forward secrecy and efficiency. They subsequently proposed a solution to enhance these shortcomings. Unfortunately, the authentication scheme by Chang and Chang is vulnerable to impersonation attacks and fails to provide user anonymity.
In 2009, Chen et al. [12] first proposed a self-verification authentication scheme for satellite communication systems. The scheme does not require heavy computational costs for authentication from mobile users because it does not employ a public key infrastructure. In addition, it does not require any sensitive information to be stored on the NCC verification table. However, Lee et al. [10] observed that the scheme by Chen et al. is vulnerable to stolen-verifier attacks. If NCC verification table parameters are stolen by an adversary, the private keys of registered users and servers can be obtained using Euclidean algorithms [18]. They subsequently proposed a simple and efficient authentication scheme for mobile satellite communication systems that is based on a one-way hash function and exclusive-OR operations; therefore, this scheme is more efficient than that by Chen et al. They claimed that their scheme achieves all security requirements and functional requirements. However, in the schemes of Chen et al. and Lee et al., the NCC needs to create a verification table to store each user’s identity and temporary identity. Obviously, the size of registration table and the time for searching the corresponding identity increase with the total number of registered users.

1.2. Our contribution

This paper shows that the authentication scheme by Lee et al. is vulnerable to DoS attacks and does not provide perfect forward secrecy. Therefore, we propose a novel secure authentication scheme without verification table for satellite communication systems that is based on elliptic curve cryptosystems (ECCs). The proposed scheme achieves user anonymity and user unlinkability and resists all attacks. In addition, the proposed scheme does not require an NCC verification table and has low client-side computational costs. Thus, the proposed scheme is more suitable for mobile satellite communication systems.

2. REVIEW OF THE SCHEME OF LEE ET AL.

This section briefly reviews the authentication scheme proposed by Lee et al., which comprises three phases: registration, login, and authentication. First, we state the following notations that are used throughout this study.

- \( p, q \) the prime numbers satisfying \( q(p-1) \)
- \( g \) a generator in \( \mathbb{Z}_p \)
- \( ID_i, TID_i \) the identity and the temporary identity of mobile user \( U_i \)
- \( LEOID \) the identity of the LEO satellite
- \( G \) a cyclic additive group
- \( h() \) a one-way hash function such that \( h(): \mathbb{Z}_p \rightarrow \mathbb{Z}_p \)
- \( H() \) a one-way hash function such that \( H(): \mathbb{Z}_p \rightarrow G \)
- \( P \) the generator of the cyclic additive group
- \( P_i \) the secret key of the user

Figure 2 shows the login and authentication phases of the authentication scheme proposed by Lee et al. Each phase is described in detail as follows.

**Registration phase:** If user \( U_i \) wants to join the system, he or she must transmit identity \( ID_i \) to NCC for registration through a secure channel. Then, the NCC computes \( P_i = h(ID_i, x) \) and \( R_i = P_i \oplus h(ID_i, k) \), where \( k \) is a random number and \( x \) is a long-term private key selected by the NCC. Next, the NCC selects temporary identity \( TID_i \) for the user and then stores \( ID_i \) and \( TID_i \) in the NCC verification table. Next, the NCC stores \( \{ TID_i, R_i, k, h() \} \) on the smart card and then transmits this smart card to the user through a secure channel.

**Login Phase:** Assume that mobile user \( U_i \) wants to communicate with another mobile user or access the resources. The user first inserts the smart card and then inputs identity \( ID_i \) into the smart card. The smart card then computes

\[
P_i' = R_i \oplus h(ID_i, k),
\]

\[
Q = P_i' \oplus r,
\]

\[
S = h(ID_i, r),
\]

where \( r \) is a random number. Next, \( U_i \) transmits login request \( \{ Q, S, TID_i \} \) to the LEO. After receiving \( \{ Q, S, TID_i \} \), the LEO forwards the login request \( \{ Q, S, TID_i \} \) and its identity \( LEOID \) to the NCC.
Authentication phase: After receiving the login request, the NCC and \( U_i \) perform the following steps to authenticate each other.

Step 1. The NCC retrieves \( ID_i \) according to \( TID \) from its verification table and then computes

\[
P_i = h(ID_i, x),
\]

(4)

\[
r' = Q \oplus P_i,
\]

(5)

\[
S' = h(ID_i, r').
\]

(6)

The NCC then verifies whether the computed \( S' \) matches the received \( S \). If it holds, \( U_i \) is authenticated. Otherwise, NCC rejects the login request.

Step 2. The NCC computes

\[
V_1 = P_i \oplus t,
\]

(7)

\[
V_3 = h(r', t),
\]

(8)

where \( t \) is a random number. Next, the NCC selects a new temporary identity \( TID_{\text{new}} \) and then computes

\[
V_4 = V_3 \oplus TID_{\text{new}}.
\]

(9)

Next, the NCC computes

\[
V_2 = h(P_i, r', t, V_4)
\]

(10)

and then replaces the old temporary identity \( (TID) \) with a new temporary identity \( (TID_{\text{new}}) \) in the verification table. Finally, the NCC computes session key \( SK = h(ID_i, r', t, P_i) \) and then transmits \( \{V_1, V_2, V_4, LEOID\} \) to the LEO.

Step 3. After receiving \( \{V_1, V_2, V_4, LEOID\} \), the LEO forwards \( \{V_1, V_2, V_4, LEOID\} \) to \( U_i \).

Step 4. After receiving the response message, \( U_i \) computes

\[
t' = V_1 \oplus P_i',
\]

(11)

\[
V_2' = h(P, r, t', V_4)
\]

(12)
and then verifies whether the computed $V_3'$ matches the received $V_2$. If they are identical, NCC is authenticated. Thereafter, $U_i$ computes

$$V_3' = h(r, t'),$$

(13)

and then replaces the old temporary identity $T_{ID}$ with a new temporary identity $T_{ID_{new}}$ in the user's smart card. Finally, the smart card computes session key $SK = h(ID_i, r, t', P_i')$. This session key is then used to encrypt/decrypt all communication messages between the $U_i$ and the NCC.

### 3. CRYPTANALYSES OF THE AUTHENTICATION SCHEME OF LEE ET AL.

This section shows that the authentication scheme by Lee et al. is vulnerable to DoS attacks and fails to provide perfect forward secrecy.

#### 3.1. Denial of service attack

The authentication scheme by Lee et al. is vulnerable to DoS attacks because a user’s temporary identity on the NCC’s verification table must be synchronized with the user’s temporary smart card identity after the user has been authenticated by the NCC. Assume that the NCC authenticates the user and the LEO then forwards $\{V_1, V_2, V_4, LEOID\}$ to the user during authentication. It is easy to show that any additional login requests are rejected by the NCC if the adversary interrupts response message $\{V_1, V_2, V_4, LEOID\}$ in Step 3. This is because the user’s temporary smart card identity differs from that in the NCC verification table. Thus, the authentication scheme by Lee et al. cannot resist the DoS attacks.

#### 3.2. Lack of perfect forward secrecy

Based on [18–20], the definition of perfect forward secrecy is that a previous session key cannot be compromised, even if an adversary obtains all of the entities’ private keys. In the authentication scheme by Lee et al., session key $SK = h(ID_i, r, t, P_i)$ comprises four parameters: user identity $ID_i$, random number $r$, random number $t$, and the user’s secret key $P_i = h(ID_i, x)$. Detailed analysis for perfect forward secrecy in the scheme of Lee et al. is described as follows.

Assume that an adversary has obtained a user’s identity $ID_i$ and his/her corresponding secret key $P_i = h(ID_i, x)$ and all previous authentication messages. The adversary first selects an authentication message $\{Q, S, T_{ID}, V_1, V_2, V_4, LEOID\}$ and then retrieves $r$ and $t$ by computing $r = P_i \oplus Q$ and $t = P_i \oplus V_1$ from the previous authentic messages by using $P_i = h(ID_i, x)$. The adversary then computes $V_2 = h(P_i, r, t, V_4)$. If the computed $V_2$ is the same as the compromised $V_2$, the adversary believes that the authentication message is sent from the same user with the identity $ID_i$. Finally, the adversary can compute the session key $SK = h(ID_i, r, t, P_i)$. Thus, the authentication scheme by Lee et al. does not provide perfect forward secrecy.

### 4. THE PROPOSED SCHEME

This section presents the proposed ECC-based scheme, which is divided into three phases: registration, login, and authentication. First, the NCC selects random number $x$ as its secret key and then computes its corresponding public key $P_{pub} = xP$. Figure 3 shows the login and authentication phases of the F3 proposed authentication scheme. Details of each phase are described as follows.

**Registration phase:** During this phase, the mobile user sends his or her identity $ID_i$ to the NCC for registration. After receiving identity $ID_i$, the NCC computes $d_i = h(ID_i, x)P$ and then stores $d_i$ and $P_{pub}$ on a smart card. The NCC then delivers the smart card to $U_i$ through a secure channel. When mobile user $U_i$ receives the smart card from the NCC, mobile user $U_i$ computes $Q_i = h(ID_i, x)P + H(PW)$ and then replaces $Q_i$ with $Q_i$.
Login phase: When mobile user $U_i$ wants to communicate with another mobile user, he or she enters identity $ID_i$ and password $PW_i$ into the smart card. Then, the smart card computes

$$h(ID_i, x)P = Q_i + H(PW_i).$$  \hspace{1cm} (15)$$

$$C_1 = r_1P.$$ \hspace{1cm} (16)$$

$$C_2 = (ID_i, N_c) \oplus h(r_1P, ID_i).$$ \hspace{1cm} (17)$$

$$C_3 = h(h(ID_i, x)P, r_1P).$$ \hspace{1cm} (18)$$

where $r_1$ and $N_c$ are two random numbers. The smart card then transmits $C_1$ and $C_2$ to the LEO for authentication. After receiving the login request \{C_1, C_2, C_3\}, the LEO forwards \{C_1, C_2, C_3\} and its identity $LEOID$ to the NCC.

Authentication phase: After receiving \{C_1, C_2, C_3, LEOID\} from mobile user $U_i$, the following steps are performed to authenticate the mobile user and the NCC.

Step 1. NCC obtains $(ID_i, N_c, r_1P_{pub})$ by computing

$$V = xC_1 = r_1P_{pub},$$ \hspace{1cm} (19)$$

and then verifies whether the following equation holds:

$$C_3 = h(h(ID_i, x)P, V).$$ \hspace{1cm} (20)$$

If the equation does not hold, the login request is rejected and terminates this session. Otherwise, $U_i$ is authenticated by the NCC and continues the following steps.

Step 2. After authentication, the NCC computes

$$C_4 = r_2P + h(ID_i, x)P,$$ \hspace{1cm} (22)$$

$$SK = r_2C_1 = r_2r_1P,$$ \hspace{1cm} (23)$$

$$C_5 = h(N_c, r_1P, r_2P, SK)$$ \hspace{1cm} (24)$$

and then returns \{C_4, C_5, LEOID\} to the LEO.
Step 3. After receiving \([C_4, C_5, LEOD_i]\), the LEO forwards \([C_4, C_5]\) to mobile user \(U_i\).

Step 4. After receiving \([C_4, C_5]\) from the LEO, \(U_i\) computes

\[
\begin{align*}
r_2P &= C_4 - h(ID_i, x)P, \\
SK &= r_1r_2P
\end{align*}
\]

and then verifies whether the following equation holds:

\[
C_5 = h(NC, r_1P, r_2P, SK).
\]

If the equation holds, the NCC is authenticated. Otherwise, the response message is rejected.

5. SECURITY ANALYSES

This section shows that our authentication scheme is secure against well-known attacks and achieves the discussed functional requirements. Before detailing our security analyses, we first introduce the discrete logarithm problem (DLP) and the computational Diffie–Hellman problem (CDHP) [14, 21, 22] that are used in our analysis.

Definition 1: DLP: Given \(P, Q \in G_1\), it is difficult to determine integer \(n \in \mathbb{Z}_q\) such that \(P = nQ\).

Definition 2: CDHP: Given \(xP \in G_1\) and \(yP \in G_1\), it is difficult to compute \(xyP \in G_1\).

The security analyses are detailed as follows:

Resistance against replay attack: A replay attack is when an adversary attempts to resend an interrupted authentication message to log onto the server. Assume that an adversary resends the previous authentic message \([C_1, C_2, C_3]\) to masquerade as a legitimate user and log on the server. The user cannot retrieve \(r_2P\) from the response message \([C_4, C_5]\) to generate the session key without knowing secret key \(h(ID_i, x)\) for user \(U_i\); therefore, the NCC detects adversary \(A\). Thus, our scheme resists replay attacks.

Resistance against impersonation attacks: Assume that an adversary attempts to masquerade as a legitimate user to log onto the NCC. The adversary must generate login request \([C_1, C_2, C_3]\) to deceive the NCC. However, he or she cannot compute \(C_2 = (ID_i, NC)@h(r_1P_{pub}, LEOD_i)\), \(C_3 = h(h(ID_i, x)P, r_1P_{pub})\), and \(SK = r_2r_1P\) without knowing the identity of user \(U_i\) and the smart card’s secret \(h(ID_i, x)\). Furthermore, the adversary may attempt to masquerade as a legitimate NCC. However, the adversary cannot compute \(C_2 = r_2P + h(ID_i, x)P\), \(SK = r_2V = r_2r_1P\), and \(C_3 = h(NC, r_2P, r_2P, SK)\) because the adversary does not have the smart card’s secret \(h(ID_i, x)\). Thus, the user detects the adversary, and our protocol resists impersonation attacks.

Resistance against DoS attacks: We have shown in Subsection 3.1 that the scheme by Lee et al. is vulnerable to DoS attacks because the scheme requires that the temporary smart card identity of user \(U_i\) be synchronized with the temporary identity of the NCC verification table. However, such an attack is impossible on our proposed scheme because the NCC does not require the storage of any parameters. Therefore, our scheme resists DoS attacks.

Resistance against smart card loss attacks: Assume that an adversary steals the smart card of legal user \(U_i\) and then masquerades as a legitimate user to log onto the NCC using the stolen smart card. However, the adversary fails because he or she does not know the password and identity of the mobile user. They can only randomly guess the identity and password of the mobile user. Thus, our scheme resists smart card loss attacks.

Resistance against stolen-verifier attack: In the proposed scheme, the NCC does not need a verification table. Thus, even if a malicious adversary breaks into the NCC, he or she cannot obtain any secret information of the mobile user from the NCC. Hence, our scheme resists stolen-verifier attacks.

Provision of mutual authentication: In the proposed scheme, mobile user \(U_i\) and the NCC can authenticate each other. If the user determines that the NCC is invalid, message \([C_4, C_5]\) is rejected. Similarly, if the NCC determines that mobile user \(U_i\) is invalid, login request \([C_1, C_2, C_3]\) is rejected. Thus, our authentication scheme provides mutual authentication.
**Provision of user anonymity:** Assume that an adversary has learned all authentication messages \(\{C_1, C_2, C_3, C_4, C_5\}\). The adversary may attempt to retrieve the identity of mobile user \(U_i\) from messages \(\{C_1, C_2, C_3, C_4, C_5\}\). However, the adversary cannot successfully obtain the identity of mobile user \(U_i\) because it is protected by \(h(r_1P_{pub}, LEOID)\). \(h(r_1P_{pub}, LEOID)\) is generated using random number \(r_1\). Without knowledge of the NCC’s private key \(x\), the adversary cannot retrieve \(r_1P\) from \(C_1 = r_1P_{pub}\) because the adversary must solve the DLP. Hence, our scheme provides user anonymity.

**Provision of user unlinkability:** In our scheme, login request \(\{C_1, C_2, C_3\}\) and response message \(\{C_4, C_5\}\) are generated using random number \(r_1\) or \(r_2\); therefore, none of the messages \(\{C_1, C_2, C_3, C_4, C_5\}\) are static. In other words, all messages cannot be used to identify mobile user \(U_i\). Hence, our scheme provides user unlinkability.

**Provision of perfect forward secrecy:** Perfect forward secrecy means that an adversary cannot compromise previous session keys, even if he or she obtains the private keys and the secret parameters of user \(U_i\) and the NCC. In our scheme, session key \(SK = r_1r_2P\) is constructed by user \(U_i\) and the NCC and cannot be compromised by the adversary because of the CDHP. Hence, our protocol provides perfect forward secrecy.

### 6. COMPARISONS

We first evaluate the computation cost of our authentication scheme. Let \(T_{SD}\) be the time required to search an identity \(ID_i\) according to \(T_{ID}\) from the NCC verification table, \(T_h\) be the time required to perform a one-way hash function, \(T_{HM}\) be the time required to perform a key hash function, \(T_s\) be the time required to perform a symmetric encryption/decryption operation, \(T_m\) be the time required to perform a multiplication operation, \(T_e\) be the time required to perform a modular exponentiation operation, and \(T_E\) be the time required to perform a multiplication point operation.

In the login phase, the mobile user requires \(3T_h + 2T_E\) to generate login request \(\{C_1, C_2, C_3\}\). In the authentication phase, the NCC requires \(4T_h + 3T_E\) to verify the login request and computes response message \(\{C_4, C_5\}\), and the mobile user requires \(T_h + T_E\) to verify the response message. Security properties and computational costs for the authentication schemes proposed by Chen et al., Lee et al., and this study are shown in Table I. Only the proposed scheme resists all potential attacks. Furthermore, the proposed scheme requires only slightly more computational costs than those by Chen et al., and Lee et al. in the client side. However, the authentication scheme by Chen et al. is vulnerable to stolen-verifier attacks, and the authentication scheme by Lee et al. is vulnerable to DoS attacks and fails to provide perfect forward secrecy. In addition, our scheme does not need to have NCC verification table, but the authentication schemes proposed by Chen et al. and Lee et al. do. The storage memory of the NCC verification table and the time for searching the user’s real identity from the NCC verification table increase with the total number of registered users. The total time complexity of the consumed searching time \(T_{SD}\) for finding real \(ID_i\) of the mobile user in the NCC verification table is \(O(n)\), where \(n\) indicates the total number of registered mobile users in the mobile satellite system. If 0.01 ms is required to complete user ID search in the NCC verification table with only

<table>
<thead>
<tr>
<th>Login and authentication phases</th>
<th>Chen et al.</th>
<th>Lee et al.</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>(T_h + T_{HM} + T_s)</td>
<td>(4T_h)</td>
<td>(4T_h + 3T_E)</td>
</tr>
<tr>
<td>NCC</td>
<td>(2T_h + T_{HM} + 2T_m + 3T_e + T_{SD})</td>
<td>(5T_h + T_{SD})</td>
<td>(4T_h + 3T_E)</td>
</tr>
</tbody>
</table>

C1: User can choose his/her identity; C2: mutual authentication; C3: user anonymity; C4: user unlinkability; C5: without leak out secret; C6: prevention of an impersonation attack; C7: prevention of a DoS attack; C8: The NCC needs to have verification table.

one record, consequently, for a mobile satellite system with 10 million users, the average searching time will be around 50 s, and it will take 100 s in the worst case. Thus, the authentication schemes by Chen et al. and Lee et al. require more computation costs and memory costs than our scheme in the server side if many users register on the NCC. Therefore, the authentication scheme proposed in this research is secure and efficient.

7. CONCLUSION

This study shows that the authentication scheme by Lee et al. is vulnerable to DoS attacks and fails to provide perfect forward secrecy. To address the discussed weaknesses, we propose a novel secure authentication scheme without verification table for mobile satellite communication systems. Because the proposed scheme’s security is based on ECC, it has low client-side and server-side computational costs. In addition, the proposed scheme resists DoS attacks and provides user anonymity, user unlinkability, and perfect forward secrecy.

REFERENCES

Dear Author,

During the copyediting of your paper, the following queries arose. Please respond to these by annotating your proofs with the necessary changes/additions.

- If you intend to annotate your proof electronically, please refer to the E-annotation guidelines.
- If you intend to annotate your proof by means of hard-copy mark-up, please refer to the proof mark-up symbols guidelines. If manually writing corrections on your proof and returning it by fax, do not write too close to the edge of the paper. Please remember that illegible mark-ups may delay publication.

Whether you opt for hard-copy or electronic annotation of your proofs, we recommend that you provide additional clarification of answers to queries by entering your answers on the query sheet, in addition to the text mark-up.

<table>
<thead>
<tr>
<th>Query No.</th>
<th>Query</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>AUTHOR: Please provide authors’ biographies and photos.</td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>AUTHOR: Please provide a suitable figure (abstract diagram or illustration selected from the manuscript or an additional eye-catching figure) and a short ‘GTOC’ abstract (maximum 80 words or 3 sentences) summarising the key findings presented in the paper for Table of Content (TOC) entry.</td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>AUTHOR: Sentences in Definitions 1 and 2 have been reworded for clarity. Please check if they are correct.</td>
<td></td>
</tr>
</tbody>
</table>
Required software to e-Annotate PDFs: Adobe Acrobat Professional or Adobe Reader (version 7.0 or above). (Note that this document uses screenshots from Adobe Reader X)

The latest version of Acrobat Reader can be downloaded for free at: http://get.adobe.com/uk/reader/

Once you have Acrobat Reader open on your computer, click on the Comment tab at the right of the toolbar:

This will open up a panel down the right side of the document. The majority of tools you will use for annotating your proof will be in the Annotations section, pictured opposite. We've picked out some of these tools below:

1. **Replace (Ins) Tool** – for replacing text.
   - Strikethrough (Del) Tool – for deleting text.
   - Add note to text Tool – for highlighting a section to be changed to bold or italic.
   - Add sticky note Tool – for making notes at specific points in the text.

### How to use it

- **Replace (Ins) Tool**
  - Highlight a word or sentence.
  - Click on the Replace (Ins) icon in the Annotations section.
  - Type the replacement text into the blue box that appears.

- **Strikethrough (Del) Tool**
  - Highlight a word or sentence.
  - Click on the Strikethrough (Del) icon in the Annotations section.

- **Add note to text Tool**
  - Highlight the relevant section of text.
  - Click on the Add note to text icon in the Annotations section.
  - Type instruction on what should be changed regarding the text into the yellow box that appears.

- **Add sticky note Tool**
  - Click on the Add sticky note icon in the Annotations section.
  - Click at the point in the proof where the comment should be inserted.
  - Type the comment into the yellow box that appears.
5. Attach File Tool – for inserting large amounts of text or replacement figures.

How to use it
- Click on the Attach File icon in the Annotations section.
- Click on the proof to where you’d like the attached file to be linked.
- Select the file to be attached from your computer or network.
- Select the colour and type of icon that will appear in the proof. Click OK.

6. Add stamp Tool – for approving a proof if no corrections are required.

How to use it
- Click on the Add stamp icon in the Annotations section.
- Select the stamp you want to use. (The Approved stamp is usually available directly in the menu that appears).
- Click on the proof where you’d like the stamp to appear. (Where a proof is to be approved as it is, this would normally be on the first page).

7. Drawing Markups Tools – for drawing shapes, lines and freeform annotations on proofs and commenting on these marks.

How to use it
- Click on one of the shapes in the Drawing Markups section.
- Click on the proof at the relevant point and draw the selected shape with the cursor.
- To add a comment to the drawn shape, move the cursor over the shape until an arrowhead appears.
- Double click on the shape and type any text in the red box that appears.

For further information on how to annotate proofs, click on the Help menu to reveal a list of further options: