Theoretical and Applied Informatics
ISSN 1896–5334
Vol.24 (2012), no. 4
pp. 293–312
DOI: 10.2478/v10179-012-0018-z

An improved ID-based client authentication with key agreement scheme on ECC for mobile client-server environments

SK HAFIZUL ISLAM, G. P. BISWAS

Department of Computer Science and Engineering, Indian School of Mines,
Dhanbad-826004, Jharkhand, India
hafi786@gmail.com, hafizul.ism@gmail.com, gpbiswas@gmail.com

Received 9 October 2012, Revised 23 November 2012, Accepted 1 December 2012.

Abstract: In wireless mobile networks, a client can move between different locations while staying connected to the network and access the remote server over the mobile networks by using their mobile devices at anytime and anywhere. However, the wireless network is more prone to some security attacks, as it does not have the ingrained physical security like wired networks. Thus, the client authentication is required while accessing the remote server through wireless network. Based on elliptic curve cryptosystem (ECC) and identity-based cryptography (IBC), Debiao et al. proposed an ID-based client authentication with key agreement scheme to reduce the computation and communication loads on the mobile devices. The scheme is suitable for mobile client-server environments, is secure against different attacks and provides mutual authentication with session key agreement between a client and the remote server as they claimed. Unfortunately, this paper demonstrates that Debiao et al.'s scheme is vulnerable some cryptographic attacks, and proposed an improved ID-based client authentication with key agreement scheme using ECC. The proposed scheme is secure based on Elliptic Curve Discrete Logarithm Problem (ECDLP) and Computational Diffie-Helmann Problem (CDHP). The detail analysis shows that our scheme overcomes the drawbacks of Debiao et al.'s scheme and achieves more functionality for the client authentication with lesser computational cost than other schemes.

Keywords: Elliptic curve cryptography, identity-based cryptosystem, mutual authentication, session key, users' anonymity, client-server environment

1. Introduction

In wireless environments, the mutual authentication between the mobile clients and the remote server gains popularity due to the rapid development of mobile communication and easy portability of handheld mobile devices such as smart phone, PDA, notebook PC etc. In recent years, the numbers of mobile client increases exponentially and at any time and any place, the people are more interested for online transaction using their mobile devices. However, the wireless mobile networks do not have the ingrained physical security like wired networks. Thus, an efficient and secure authentication technique for mobile client is required in order to provide the flexibility and robustness of online transaction. The authentication of the server and the client both are equally important when a client wants to acquire various services from the remote server to protect server's spoofing attack and impersonation attack from the outsiders. Some schemes although proposed for remote client mutual authentication, they are not usable in some applications such as e-voting, online-order placement, pay-TV etc., where a session key agreement is also necessary for exchanging the confidential information between the server and the clients over an open network. The main contribution of this paper is that design of a secure and computational efficient mutual authentication with key agreement scheme for mobile client-server environments.

In order to offer the strong securities, earlier client authentication schemes are usually implemented by means of public key infrastructure (PKI) [1, 2]. However, the real application of PKI-based remote authentication scheme brings heavy management burden of public key and certificates. Besides, the computation cost of PKI-based remote authentication scheme is very high due to modular exponentiation, making it unsuitable for mobile environments since the mobile devices have low-power battery, low storage space and low computation ability. Recently, ECC [3, 4]-based public key cryptosystem has attracted a great attention due to its shorter length key such as 160 bit ECC-based key provides same level of security as of 1024 bit RSA-based key, low storage and faster computation. Thus, several ECC-based client authentication schemes [5-15] have been proposed by the researchers to reduce the computation and communication loads on mobile devices. Despite the suitability of ECC-based scheme for low-power mobile devices, ECC-based cryptosystem has some other limitations like PKI-based cryptosystem. It needs extra storage space to store clients' public keys and certificates. The client must have additional computation ability to verify the other's public key certificate.

In 1984, Shamir [16] introduce the notion of identity-based cryptography (IBC) in which client's public key is an easily computable function of his email address, physical IP address etc., where the corresponding private key is generated by binding the client's identity with the master secret key of the trusted authority, called private key generator (PKG). The client's private key is given using a secure channel and known to only client and PKG, but its legitimacy can be verified publicly. The IBC avoids the use of public key certificates, so it can save system resources and improve the system efficiency. Thus, IBC seems to be an alternative solution to the PKI-based cryptosystem. However, most of the IBC scheme suffers from the private key escrow problem since PKG knows the private key of the user and thus, a malicious PKG can easily impersonate any legitimate client and the present work has been carried out in the direction for the private key escrow problem. After Shamir's work, several ID-based schemes have been proposed, but Boneh and Franklin first proposed the scheme satisfying ID-based encryption scheme (IBE) [17] in 2001 using bilinear pairing [18, 19] over an elliptic curve group.

1.1. Related works

Recently, several ID-based client authentication schemes [5, 9-11, 19- 23] have been found in literature. However, these are vulnerable to various attacks such as replay attack [5, 11, 20, 21], privileged-insider attack [14, 23-25], impersonation attack [5, 9], lost/stolen smartcard attack [23, 24], known session-specific temporary information attack [5, 11, 12, 14, 25], and many logged-in users' attack [5, 11, 12, 14, 25]. In addition, some of these schemes are faces from the problem of users' anonymity [5, 11, 14, 25], perfect forward security [5, 9] and clock synchronization [9-11, 25]. In 2011, Debiao et al. [14] proposed an ID-based client authentication with key agreement scheme on ECC for mobile client-server environments. They claimed that their scheme provides remote mutual authentication and session key agreement with low computation cost and is secure against various attacks. However, [25] showed that Debiao's scheme cannot withstand the clock synchronization problem, many logged-in users' attack, known session-specific temporary information attack, impersonation attack, privilege-insider attack, incapable to provide users' anonymity and no provision for changing/updating the leaked private key. Thus, aforementioned problems inspired us to design an efficient and secure ID-based client authentication with key agreement scheme for mobile client-server environments.

1.2. Contribution and organization

In this paper, we propose an improved ID-based client authentication with key agreement scheme using ECC, which is secure under the CDHP and ECDLP. The proposed scheme removes the security pitfalls and weaknesses of Debiao et al.'s scheme while keeping the merits of earlier scheme, and protects other attacks as well. Compared with other works, our scheme is efficient and secure, and thus suitable for mobile client-server environments.

The rest of the paper is organized as follows. In Section 2, a brief review of Debiao et al.'s scheme is given. Section 3 describes the weaknesses of Debiao et al.'s scheme. Section 4 presents the details of the proposed scheme. The security analysis of the proposed scheme is given in Section 5 and the Section 6 compares the proposed scheme with other works in terms of computation cost. Finally, some concluding remarks are made in the last section.

2. Review of Debiao et al.'s scheme

In this section, we briefly reviewed the Debiao et al.'s scheme proposed for client-server environments. The scheme has three phases: system initialization phase, client registration phase and mutual authentication with key agreement phase. The Table 1 includes the notations used in the Debiao et al.'s scheme.

Notations	**Descriptions**
C_i	The client
S	The remote server.
ID_{C_i}	Identity of the client C_i.
p, n	Two large prime numbers.
F_p	A finite field.
$E_p(a,b)$	An elliptic curve defined on finite field F_p with prime order n.
$G_p(a,b)$	An additive cyclic group of elliptic curve points on $E_p(a,b)$.
P	A base point of the group $G_p(a,b)$ with order n.
$H_1 : \{0,1\}^* \to Z_n^*$, $H_2 : \{0,1\}^* \to Z_p^*$, $H_3 : \{0,1\}^* \to Z_p^*$	Three secure one-way hash function (i.e. SHA-1).
$MAC_K(m)$	Secure message authentication code of message m under the key K.
(x, P_S)	Private/public key pair of the server S, where$P_S = xP$.

Tab. 1. Notations used in Debiao's scheme

2.1. System initialization phase

In this phase, the remote server S generates the system parameters as follows.

(1). ***S*** chooses an elliptic curve equation $E_p(a,b)$.

(2). ***S*** selects a base point P with the order n over $E_p(a,b)$.

(3). ***S*** selects $x \in_R Z_n^*$ (it means x randomly selected from Z_n^*) as master key and computes public key as $P_S = xP$.

(4). ***S*** chooses three secure and one-way hash functions $H_1(\cdot), H_2(\cdot), H_3(\cdot)$ and a message authentication code $MAC_K(m)$. The server S keeps x in private and publishes $\{F_p, E_p, n, P, P_S, H_1(\cdot), H_2(\cdot), H_3(\cdot), MAC_K(m)\}$ as system's parameter.

2.2. Client registration phase

(1). The client C_i submits his identity ID_{C_i} to the remote server S for registration.

(2). S computes $h_{C_i} = H_1(ID_{C_i})$ and clients private key/public key pair $D_{C_i} = (x + h_{C_i})^{-1}P$ and $P_{C_i} = (x + h_{C_i})P$. Then, S returns the private key D_{C_i} with ID_{C_i} to C_i through a secure channel.

(3). C_i validates his private/public key pair (D_{C_i}, P_{C_i}) by checking whether the equation $P_{C_i} = D_{C_i}P = P_S + h_{C_i}P$ holds.

The Debiao et al.'s scheme uses two approaches to deliver the private key D_{C_i} to the client C_i. One is off-line approach, where S stores the identity ID_{C_i}, the private key D_{C_i} into a smartcard and returns it to C_i. Therefore, to deliver the smartcard a secure channel is necessary; otherwise, someone can tamper the smartcard. Second, the on-line approach, where C_i connects to S through Internet, then S may use the Secure Socket Layer (SSL) channel in the https mode to deliver the private key D_{C_i} to C_i. Note that, Debiao et al.'s scheme uses the secure channel in the registration phase, however, the scheme suffer from the private key escrow problem (here we call it as privileged-insider attack) since the private key D_{C_i} is completely known to S and of course its privileged-insider.

2.3. Mutual authentication with key agreement phase

This phase is executed by between a client C_i and the server S for the mutual authentication and the agreement of common session key. Initially, C_i sends a login request to S and then the server verifies the client's request. If it is valid, S returns a response message to C_i, which helps C_i to validate S. Subsequently, both the client and the server generate a common session key. The mutual authentication and the session key agreement phase is stated as follows:

(1). C_i selects a number $r_{C_i} \in_R Z_n^*$, computes $M = r_{C_i}P$, $M' = r_{C_i}D_{C_i}$ and $K = H_2(ID_{C_i}, T_{C_i}, M, M')$, and sends the login message $M_1 = \{ID_{C_i}, T_{C_i}, M, MAC_K(ID_{C_i}, T_{C_i}, M)\}$ to S, where T_{C_i} is the current timestamp.

(2). Upon receiving the login message $M_1 = \{ID_{C_i}, T_{C_i}, M, MAC_K(ID_{C_i}, T_{C_i}, M)\}$ at time T_{Ci}', S checks the validity of ID_{C_i} and the timestamp T_{C_i}. S rejects the login request if either ID_{C_i} or $T_{C_i}' - T_{C_i} \leq \Delta T_{C_i}$ is invalid, where ΔT_{C_i} means the acceptable time interval. Otherwise, S computes $h_{C_i} = H_1(ID_{C_i})$, $M' = (x + h_{C_i})^{-1}M$ and $K = H_2(ID_{C_i}, T_{C_i}, M, M')$. The integrity of the message $(ID_{C_i}, T_{C_i}, M,)$ is checked with $MAC_K(ID_{C_i}, T_{C_i}, M)$ that is computed by using the key K. S rejects the request if the integrity check fails; otherwise chooses a number $r_S \in_R Z_n^*$, computes $W = r_SP$, $K_S = r_SM$ and the session key as $SK = H_3(ID_{C_i}, T_{C_i}, T_SM, W, K_S)$. Then S replies the client C_i with the message $M_2 = \{ID_{C_i}, T_S, W, MAC_K(ID_{C_i}, T_S, W)\}$, where T_S is timestamp of S.

(3). On receiving the message M_2, client C_i validates the timestamp T_S and the integrity of $\{ID_{C_i}, T_S, W\}$ to check the authenticity of S. C_i computes $K_{C_i} = r_{C_i}W$ and the session key $SK = H_3(ID_{C_i}, T_{C_i}, T_S, M, W, K_{C_i})$ if all the conditions are satisfied.

3. Weaknesses of Debiao et al.'s scheme

This section shows that Debiao et al.'s scheme is vulnerable to privileged-insider attack [25, 13], many logged in users' attack [11, 12, 25], impersonation attack [5, 9, 13, 25, 29] and known session-specific temporary information attack [11, 12, 25-27]. Besides, the scheme has the problems of clock synchronization [11, 13, 25] and users' anonymity [11, 25]. In addition, Debiao et al.'s scheme has no provision for changing/updating leaked authentication key [11, 25] with previous identity. It means, if the authentication key is leaked accidentally, the client cannot get a new authentication key with the previous identity. The client has to choose new identity each time for fresh private key. Thus, Debiao et al.'s scheme is inefficient to offer flexibility of changing/updating the leaked private key.

3.1. Privileged-insider attack

The private key escrow problem is an inherent problem of the most of the identity-based cryptography (IBC) since its inception. In the setting of IBC, a third party (PKG) generates the private key and the assumption that PKG is fully trusted is a strong assumption, which is very crucial for real-life application. Therefore, a malicious PKG may impersonate any user by using client's private key since it is completely known to PKG. In this paper, we call this attack as privileged-insider attack [11-13, 15, 25], which is now described here. Note that Debiao's scheme is based on IBC and thus, it has the similar problem since the remote server knows the private key of each client. In the registration phase, S computes the private key D_{C_i} and returns it to C_i. Therefore, there is a chance to expose the private key D_{C_i} to the privileged-insider E of S. If E learns the private key D_{C_i} of C_i during the registration phase, then E may of course successfully impersonate C_i to login S by using D_{C_i}. Thus, Debiao et al.'s scheme is vulnerable to this kind of privileged-insider attack.

3.2. Many logged-in users' attack

It is always preferable that the remote server gives permission one person at a time to reach the account of a legitimate client. Otherwise, the inconsistency of information may occur while updating or accessing the information stored into the remote server. However, Debiao et al.'s scheme cannot protect the situation where more than one person can get access to the same account concurrently [11, 12]. Assume that if C_i's private key D_{C_i} is leaked to more than one person then all who know the pair (ID_{C_i}, D_{C_i}), may attempt to use C_i's account at the same time by originating the individual login requests. Therefore, each adversary can get access to C_i's account simultaneously just by selecting a random number $r_{C_i} \in_R Z_n^*$ and executing the following the mutual authentication phase of Debiao et al.'s scheme, because all of them employed the same authentication process using C_i's valid private key D_{C_i}. The server S is unable to stop all of them to get access to C_i's account concurrently. Thus, the many logged-in users' attack with same login-id and leaked private key [11, 25, 12, 15] is possible in Debiao et al.'s scheme.

3.3. Impersonation attack

In general, the server stores minimal information about each client in the database based upon which the server verifies the legitimacy of clients during mutual authentication phase, but Debiao et al.'s scheme does not store any information about the clients. Suppose an adversary E steals the identity ID_{C_i} of an authorized client C_i and asks for the private key to S corresponding to ID_{C_i}. Then S returns the private key $D_{C_i} = (x + H_1(ID_{C_i}))^{-1}P$ to E, which is nothing but the private key of C_i. Therefore, E can attack the authentication system by impersonating the client C_i easily and can get access to the remote server S using client's private key D_{C_i} [5, 9, 13, 25].

3.4. Known session-specific temporary information attack

Canetti and Krawczyk [26] investigated the known session-specific temporary information attack in 2001. Later on, Cheng et al. [27] pointed out that the secrecy of the generated session key should not be affected even if the session ephemeral secrets are leaked to an adversary anyway. We can show that Debiao's scheme fails to protect this kind of attack. In mutual authentication phase, the client C_i and the server S generate the common session key $SK = H_3(ID_{C_i}, T_{C_i}, T_S, M, W, K_S)$, where all of $(ID_{C_i}, T_{C_i}, T_S, M, W)$ are public information other than K_S, and the security of the session key SK depends only on the confidentiality of $K_S = r_S r_{C_i} P$. According to [26, 27], if the session ephemeral secrets r_{C_i} and r_S are exposed to an outsider, then he can compute K_S easily and the resulting session key *SK* as well. Thus, the Debiao et al.'s scheme is not secure against the known session-specific temporary information attack [11, 12, 25].

3.5. Inability to protect users' anonymity

The police and security of mobile services allow the client to be anonymous while doing the transaction over the public channel. In some applications such as e-voting, secret online-order placement, online-shopping, pay-TV etc., it is important to maintain the user secrecy, because from the identity ID_{C_i} some personal secret information may be leaked about the client C_i or without employing any effort an adversary recognizes the particular transaction being performed by the client C_i [11]. Therefore, a well sound remote login scheme should preserve the user anonymity in all respects. However, Debiao et al.'s scheme does not preserve the users' anonymity [25]. In mutual authentication phase, C_i's original identity ID_{C_i} is transmitted with the message

$M_1 = \{ID_{C_i}, T_S, W, MAC_K(ID_{C_i}, T_S, W)\}$ through the public network. Therefore, an adversary can identify the client who is trying to login the remote server.

3.6. No provision for changing/updating leaked private key

Debiao et al.'s client authentication scheme does not offer the revocation of leaked private key with old identity [25] that is, if the authentication key is leaked to an outsider however, the server cannot compute new authentication key different from previous key with the same identity. Note that, for real-life applications, clients are interested to change their private key while keeping the identity same. In the registration phase, the private key $D_{C_i} = (x + H_1(ID_{C_i}))^{-1}P$ of the client C_i is generated using the identity ID_{C_i} and the server's secret key x. This shows the uniqueness of the authentication key depends on the identity only. Therefore, the client has to choose dissimilar identity every time to get distinct authentication key. Therefore, Debiao's scheme does not have flexibility for changing/updating the leaked private key with same identity [11].

3.7. Clock synchronization problem

In any mutual authentication scheme, timestamp is used to prevent the replay attack and man-in-the-middle attack. However, the timestamp raises the problem of clock synchronization [11, 25] in large networks, such as wide area networks, mobile communication networks, and satellite communication networks. The schemes based on the timestamp can withstand the replay attack using systems' timestamp provided the system clock must be synchronized; otherwise, the scheme will not work properly. Since network environment and transmission delay is unpredictable [28], a potential replay attack exists in all schemes that use the timestamp. The Debiao et al.'s is based on timestamp, so it faces the problem of clock synchronization.

4. Proposed scheme

This section proposes an improved ID-based client authentication with key agreement protocol based on ECC for mobile client-server environments. Similar to Debaio et al.'s scheme, two entities are involved in our scheme, namely a client C_i and the remote server S. The proposed scheme consists of four phases: system initialization phase, client registration phase, mutual authentication with key agreement phase and changing/updating the leaked private key phase. We explain the proposed scheme by the following steps.

4.1. System initialization phase

The remote server S, to setup the overall system parameters executes this phase once. In this phase, given a security parameter $k \in \mathrm{Z}^+$, the server S generates the following system parameters.

(1). S chooses an elliptic curve group $G_p(a, b)$.

(2). S selects a base point P with the order n over $G_p(a, b)$.

(3). S selects its master key $x \in_R Z_n^*$ and computes public key $P_S = xP$.

(4). S chooses three secure one-way hash functions $H_1(\cdot), H_2(\cdot), H_3(\cdot)$. The server S keeps x in private and publishes $\{F_p, G_p(a, b), n, P, P_S, H_1(\cdot), H_2(\cdot), H_3(\cdot)\}$.

4.2. Client registration phase

This phase is executed once for registration by the client C_i before login the remote server S to obtain his authentication key, which is used to verify the identity of the client or the remote server. The steps of this phase are given as follows.

(1). The client C_i chooses a number $t_{C_i} \in_R Z_n^*$, computes $T_{C_i} = t_{C_i}P$ and submits (ID_{C_i}, T_{C_i}) to S.

(2). S checks the uniqueness of the identity ID_{C_i}. If the identity is not unique, S requests C_i to submit another fresh identity. Otherwise, S checks the registration details and computes the private/public key pair for C_i.

(3). S chooses a number $x_{C_i} \in_R Z_n^*$, computes $R_{C_i} = x_{Ci}P$, $V_{Ci} = R_{C_i} + T_{C_i}$, $h_{C_i} = H_1(ID_{C_i}, V_{C_i})$, $D^*_{C_i} = x_{C_i} + h_{C_i}x$ and $P_{C_i} = V_{C_i} + h_{C_i}P_S$. Now S and returns $(D^*_{C_i}, V_{C_i})$ to C_i through a secure channel. The server S stores the information $(ID_{C_i}$, *status-bit*) about the client C_i to his own database. S sets the *status-bit* to one if the user is logged in, otherwise sets to zero.

(4). On receiving $(D^*_{C_i}, V_{C_i})$, the client C_i computes his final private key $D_{C_i} = D^*_{C_i} + t_{C_i}$ and checks the validity of the private/public keys by the equation $P_{C_i} = D_{C_i}P = V_{C_i} + H_1(ID_{C_i}, V_{C_i})P_S$. Since,

$$\begin{aligned}
P_{C_i} &= V_{C_i} + h_{C_i}P_S \\
&= T_{C_i} + R_{C_i} + h_{C_i}P_S \\
&= t_{C_i}P + x_{C_i}P + H_1(ID_{C_i}, V_{C_i})xP \\
&= (t_{C_i} + x_{C_i} + H_1(ID_{C_i}, V_{C_i})x)P \\
&= (t_{C_i} + D^*_{C_i})P \\
&= D_{C_i}P
\end{aligned}$$

If the above verification is satisfied then the private key (D_{C_i}, V_{C_i}) and the public key P_{C_i} are valid. After validating the tuple $(D_{C_i}, V_{C_i}, P_{C_i})$, client C_i stores $(ID_{C_i}, V_{C_i}, P_{C_i})$ into his mobile device. The detailed description of the registration phase is given in Fig. 1.

Client C_i (ID_{C_i})	Server $S(x, P_S = xP)$
Choose ID_{C_i} and a number $t_{C_i} \in_R Z_n^*$ Compute $T_{C_i} = t_{C_i}P$ and send (ID_{C_i}, T_{C_i})	
(ID_{C_i}, T_{C_i}) $\longrightarrow$	

	If (ID_{C_i} is not unique)
	Request C_i to submit another identity
	Else
	Check registration details and compute the key as
	Choose $x_{C_i} \in_R Z_n^*$, compute $R_{C_i} = x_{C_i}P$, $V_{C_i} = R_{C_i} + T_{C_i}$,
	$h_{C_i} = H_1(ID_{C_i}, V_{C_i})$ and $D^*_{C_i} = x_{C_i} + h_{C_i}x$, $P_{C_i} = V_{C_i} + h_{C_i}P_S$

$\longleftarrow$	$(D^*_{C_i}, V_{C_i})$
Compute the private key $D_{C_i} = D^*_{C_i} + t_{C_i}$ If $(D_{C_i}P \neq V_{C_i} + H_1(ID_{C_i}, V_{C_i})P_S)$ $(D^*_{C_i}, V_{C_i})$ is invalid Else $(D^*_{C_i}, V_{C_i})$ is invalid Store (ID_{C_i}, V_{C_i}, P_{C_i}) into his mobile device.	

Fig. 1. Registration phase of the proposed scheme

Note that, in Debiao et al.'s scheme the private key D_{C_i} is completely known to S and of course its privileged-insider. In our scheme, S only knows the partial key $D^*_{C_i}$, but not the complete key $D_{C_i} = D^*_{C_i} + t_{C_i}$, because $t_{C_i} \in_R Z_n^*$ is unknown to him. Here, t_{C_i} is only known to C_i and hence, the private key D_{C_i} is unknown to S. Therefore, any privileged-insider of S cannot impersonate C_i and as a result, the proposed scheme removes the privileged-insider attack (key-escrow problem).

4.3. Mutual authentication with key agreement phase

In this phase, both the client C_i and the server S mutually authenticate each other using clients' mobile device, and then generate a common session key. The proposed client authentication scheme is based on the three-way challenge-response handshake technique instead of two-way challenge-response technique as used in Debaio et al.'s scheme.

(1). The client C_i keys his identity ID_{C_i} and the private key D_{C_i} into the mobile device and then the device checks whether $P_{C_i} = D_{C_i}P$ holds. If it is invalid, the mobile device asks the client for exact identity-private key pair, otherwise, chooses a number $r_{C_i} \in_R Z_n^*$ and then computes the followings: $M = r_{C_i}P$, $M' = r_{C_i}P_S$, $M'' = D_{C_i}P_S$, $DI_{C_i} = ID_{C_i} \oplus H_2(M)$ and $H_{C_i} = H_2(ID_{C_i}, M', M'', V_{Ci})$. Subsequently, the device sends the login message $M_1 = \{DI_{C_i}, M', V_{C_i}, H_{C_i}\}$ to S.

(2). On receiving $M_1 = \{DI_{C_i}, M', V_{C_i}, H_{C_i}\}$, S computes $M = x^{-1}M' = x^{-1}xr_{C_i}P = r_{C_i}P$, $ID_{C_i} = DI_{C_i} \oplus H_2(M)$ and $P_{C_i} = V_{C_i} + H_1(ID_{C_i}, V_{C_i})P_S$. S also computes $M'' = xP_{C_i}$, $H_{Ci} = H_2(ID_{C_i}, M', M'', V_{C_i})$ and checks whether the computed H_{C_i} is equal to received H_{C_i}. If so, the integrity of the received message M_1 is preserved, otherwise S rejects the login request. Afterward, S selects a number $r_S \in_R Z_n^*$, computes $W = r_SP$, $H_S = H_2(ID_{C_i}, M, M', M'', W)$ and then sends $M_2 = \{W, H_S\}$ to C_i.

(3). Upon receiving $M_2 = \{W, H_S\}$, the client C_i computes $H_S = H_2 (ID_{C_i}, M, M', M'', W)$ and checks whether the received H_S is equal to computed H_S. If the result is negative then C_i rejects the transaction, otherwise authenticates the server S and computes the session key as $SK = H_3(ID_{C_i}, M, M', M'', W, K)$, where $K = r_{C_i}W = r_{C_i}r_SP$. The client C_i computes $H_{CS} = H_2(ID_{C_i}, M, M', M'', W, SK)$ and sends it to S.

(4). S computes $K = r_SM = r_{C_i}r_SP$, $SK = H_3(ID_{C_i}, M, M', W, K)$ and $H_{SC} = H_2(ID_{C_i}, M, M', M'', W, SK)$. After that, S accepts *SK* as the session key if the received H_{CS} is equal to the computed H_{SC}, otherwise, rejects the transaction.

It is to be noted that both S and C_i mutually authenticates each other securely and hold the same session key *SK*. The client authentication with session key agreement phase also is shown in Fig. 2.

4.4. Changing/updating the leaked private key with same identity

The proposed scheme has the flexibility of changing/updating the leaked private key with the old identity. Assume that the private key D_{C_i} of the client C_i is leaked by some means to an adversary. Thus, to get a new private key with the same login-id, C_i performs the following steps (See Fig. 3.):

(1). C_i chooses a fresh number $t_{C_i}' \in_R Z_n^*$, computes $T_{C_i}' = t_{C_i}'P$ and makes a request with (ID_{C_i}, T_{C_i}') to S.

(2). On receiving (ID_{C_i}, T_{C_i}'), S verifies the authorization and the registration details of C_i and if it is positive, S chooses a number $x_{C_i}' \in_R Z_n^*$, computes $R_{C_i}' = x_{C_i}'P$, $V_{C_i}' = R_{C_i}' + T_{C_i}'$ and $D_{C_i}' = x_{C_i}' + H_1(ID_{C_i}, V_{C_i}')x$ for C_i. Then S sends the tuple (D_{C_i}', V_{C_i}') to C_i through a secure and authenticated channel.

(3). Upon receiving (D_{C_i}', V_{C_i}'), C_i computes his final private key as $D_{C_i}'' = D_{C_i}' + t_{C_i}'$ and cheeks the validity of the private-public key pair as stated in the registration phase of the earlier section. If the result is positive, C_i updates the mobile device $(ID_{C_i}, V_{C_i}, P_{C_i})$ with $(ID_{C_i}', V_{C_i}', P_{C_i}')$.

Client C_i (ID_{C_i}, D_{C_i})	**Server S** $(x,\ P_S = xP)$

Insert ID_{C_i} and D_{C_i} into the mobile device

Mobile device performs:

If $(P_{C_i} \neq D_{C_i} P)$

  Ask for exact $(ID_{C_i},\ D_{C_i})$

Else

  Choose $r_{C_i} \in_R Z_n^*$, compute $M = r_{C_i} P$, $M' = r_{C_i} P_S$,

  $M'' = D_{C_i} P_S$, $DI_{C_i} = ID_{C_i} \oplus H_2(M)$,

  and $H_{Ci} = H_2(ID_{C_i}, M', M'', V_{C_i})$

$M_1 = \{DI_{C_i}, M', V_{C_i}, H_{C_i}\}$ $\longrightarrow$

Compute $M = x^{-1} M' = r_{C_i} P$, $ID_{C_i} = DI_{C_i} \oplus H_2(M)$

$P_{C_i} = V_{C_i} + H_1(ID_{C_i}, V_{C_i}) P_S$, $M'' = xP_{C_i}$

and $H_{Ci} = H_2(ID_{C_i},\ M',\ M'',\ V_{C_i})$

If (computed $H_{C_i} \neq$ received H_{C_i})

  Reject the client's request

Else

  Select $r_S \in_R Z_n^*$, compute $W = r_S P$ and

  $H_S = H_2(ID_{C_i},\ M,\ M',\ M'',\ W)$

  $M_2 = \{W,\ H_S\}$

$\longleftarrow$

Compute $H_S = H_2(ID_{C_i}, M, M', M'', W)$

If (received $H_S \neq$ computed H_S)

  Reject the server's challange

Else

  Compute $K = r_{C_i} W = r_{C_i} r_S P$, the session key

  $SK = H_3(ID_{C_i}, M, M', M'', W, K)$ and

  $H_{SC} = H_2(ID_{C_i}, M, M', M'', W, SK)$

$\{H_{CS}\}$ $\longrightarrow$

Compute $K = r_S M = r_{C_i} r_S P$,

$SK = H_3(ID_{C_i}, M, M', W, K)$

and $H_{SC} = H_2(ID_{C_i}, M, M', M'', W, SK)$

If $(H_{SC} \neq H_{CS})$

  Reject the client's response

Else

  Accept client's response and SK as session key

Fig. 2. Mutual authentication with session key agreement phase of the proposed scheme

5. Security analysis of the proposed scheme

In this section, we analyzed the proposed scheme, which can resist all attacks, and compares with other related schemes from security point of view. The proposed scheme not only eliminates the weaknesses of Debaio et al.'s scheme, but also resists all other

Client C_i (ID_{C_i})		Server $S(x,\ P_S = xP)$
Choose a number $t_{C_i}' \in_R Z_n^*$ Compute $T_{C_i}' = t_{C_i}'P$ and send (ID_{C_i}, T_{C_i}')		
	$\xrightarrow{(ID_{C_i}, T_{C_i}')}$	
		Verify the authorization and registration details of C_i. If the result is positive, choose $x_{C_i}' \in_R Z_n^*$, compute $R_{C_i}' = x_{C_i}'P$, $V_{C_i}' = R_{C_i}' + T_{C_i}'$ and the new key $D_{C_i}' = x_{C_i}' + H_1(ID_{C_i}, V_{C_i}')x$ with the old identity ID_{C_i}. Send (D_{C_i}', V_{C_i}') to C_i through a secure channel.
	$\xleftarrow{(D_{C_i}', V_{C_i}')}$	
Compute the private key as $D_{C_i}'' = D_{C_i}' + t_{C_i}'$ If $(D_{C_i}''P = V_{C_i}' + H_1(ID_{C_i}, V_{C_i}')P_S)$ Update the mobile device $(ID_{C_i},\ V_{C_i},\ P_{C_i})$ to $(ID_{C_i}', V_{C_i}', P_{C_i}')$. Else Reject the key.		

Fig. 3. Changing/updating the leaked private key with same identity of the proposed scheme

attacks. Our scheme is secure provided that following two computational problems are infeasible on the elliptic curve group.

Definition 1.
Elliptic curve discrete logarithm problem (ECDLP): Given $(P, Q) \in G_p$, find an integer $a \in_R Z_n^*$ such that $Q = aP$.

Definition 2.
Computational Diffie-Hellman problem (CDHP): Given $(P, aP, bP) \in G_p$ for any $a, b \in_R Z_n^*$ computation of abP is hard in the group G_p.

5.1. Many logged-in users' attack

The proposed scheme can protect the many logged-in users' attack. Assume that C_i's private key D_{C_i} is exposed to the outsiders A_1 and A_2. Now A_1 and A_2 know (ID_{C_i}, D_{C_i}) and thus, they can try to get access C_i's account to the server S concurrently. However, the proposed scheme allows one person at a time to get access to C_i's account. Suppose that A_1 gets logged in to S, then S sets the *status-bit* to one and meanwhile if A_2 try to get login S, then the server S refuses A_2's request since the *status-bit* indicates still someone is logged in. Thus, the proposed scheme is robust against many logged-in users' attack.

5.2. Known session-specific temporary information attack

The proposed scheme can withstand the known session-specific temporary information attack. According to [11, 12, 25], the disclosure of the session short-term secrets r_{C_i} and r_S in a session does not expose the session key of that session. In our scheme, the client C_i and the remote server S mutually authenticate each other and then compute the session key $SK = H_3(ID_{C_i}, M, M', W, K)$. If r_{C_i} and r_S are exposed to an outsider E, then E can compute $K = r_{C_i} r_S P$ using r_{C_i} and r_S, but he cannot compute $M'' = D_{C_i} xP$ without the secret key of either the client or the server. The secret M'' can be computed directly from the pair $(P_{C_i}, P_S) = (D_{C_i} P, xP)$ if a polynomial-time algorithm solves the CDH problem. However, there is no such polynomial-time algorithm, which can break the CDH problem. Therefore, the proposed scheme can protect the known session-specific temporary information attack.

5.3. Lost/Stolen mobile device attack

In our scheme, client C_i stores the information $(ID_{C_i}, V_{C_i}, P_{C_i})$ into his mobile device, which can help both the client C_i and the server S for mutual authentication. Suppose an adversary E steals C_i's mobile device, extracts $(ID_{C_i}, V_{C_i}, P_{C_i})$ from the device and then try to get login S by using the extracted information. However, from $(ID_{C_i}, V_{C_i}, P_{C_i})$ the adversary cannot extract $(t_{C_i}, x_{C_i}, D_{C_i}, x)$ due to the difficulties of ECDLP problem. Therefore, E cannot get any valuable information from the stolen/lost mobile device that can help him to impersonate the client C_i. Thus, the lost/stolen mobile device attack is infeasible to the proposed scheme.

5.4. Mutual authentication and session key agreement

The proposed scheme achieves the mutual authentication and secret session key agreement between a client and the remote server by means of three-way challenge-response handshake technique. In our scheme, the client C_i sends the login request message $M_1 = \{DI_{C_i}, M', V_{C_i}, H_{C_i}\}$ to the server S, where $H_{C_i} = H_2(ID_{C_i}, M', M'', V_{C_i})$ and $M'' = D_{C_i} xP$. Then S verifies the received message M_1 by using the secret M'', which can be computed by the real server S and the legal client C_i only. After validating M_1, S sends a challenge message $M_2 = \{W, H_S\}$ to C_i, where $H_S = H_2(ID_{C_i}, M, M', M'', W)$. Next, C_i checks that the received H_S is valid or not by computing H_S using (ID_{C_i}, M, M', W), $M'' = D_{C_i} xP$ and accept or reject the server S depending on the verification result. Finally, C_i sends the response message $\{H_{CS}\}$ to S. On receiving $\{H_{CS}\}$, S computes the session key SK and $H_{SC} = H_2(ID_{C_i}, M, M', M'', W, SK)$, and subsequently whether $H_{CS} =? H_{SC}$ holds. If so, S authenticates the client C_i and allows him to get access to the resource of S. Thus, the proposed scheme supports a secure session key agreement and mutual authentication between a client and the remote server.

5.5. Users' anonymity problem

The proposed scheme offers users' anonymity [11] when any communication have been made by a user over the public networks. In our scheme, instead of C_i's original identity ID_{C_i} a dynamic identity $DI_{C_i} = ID_{C_i} \oplus H_2(M)$ is sent from which an outsider E cannot obtain ID_{C_i} since $M = r_{C_i}P$ is unknown to him. Noted that, E can compute M from M' by executing $M = x^{-1}M'$ only if the secret key x of S is known. However, E has no knowledge about the secret key x. On the other hand, E also cannot derive the secret key x from public key $P_S = xP$ due to the hardness of ECDLP problem. Thus, the users' anonymity is preserved in our scheme.

5.6. Replay attack and clock synchronization problem

The proposed scheme does not employ the timestamp and thus, clock synchronization problem is removed. Besides, an outsider E cannot make a replay attack in our scheme. Assume that E replays a previous session valid message $M_1 = \{DI_{C_i}, M', V_{C_i}, H_{C_i}\}$ to S for the current session. Then, S replies with fresh message $M_2 = \{W', H_S'\}$ to E, where $W = r_S'P$ and $H_S' = H_2(ID_{C_i}, M, M', M'', W')$. However, E cannot compute $M'' = D_{C_i}xP$, $K' = r_{C_i}r_S'P$ and $SK' = H_3(ID_{C_i}, M, M', M'', W', K')$ as well since he is unaware about the short-term secret r_{C_i} and the private key D_{C_i}. Therefore, E replies with the wrong message $\{H_{CS}''\}$ that can be detected by the server S by comparing it with the computed $H_{SC}' = H_2(ID_{C_i}, M, M', M'', W', SK')$ and rejects E's login request.

5.7. Privileged-insider attack

The privileged-insider attack is not possible in the proposed scheme. In the registration phase, C_i sends the tuple (ID_{C_i}, T_{C_i}) to S and then S computes the partial private key as $D^*_{C_i} = x_{C_i} + h_{C_i}x$, but not the actual private key D_{C_i}. The complete private key $D_{C_i} = D^*_{C_i} + t_{C_i}$ is unknown to S and its privileged-insider E (say), because t_{C_i} is only known to C_i. To compute t_{C_i} from $T_{C_i} = t_{C_i}P$, E has to solve the ECDLP problem, which is hard to break by any polynomial time algorithm to date. Due to the incomplete knowledge of D_{C_i}, E cannot impersonate C_i successfully and thus cannot access to get the remote server S.

5.8. Known session key security

The known session key security states that an outsider cannot compute the current session key even he knows some previous session keys. In our scheme, the session key $SK = H_3(ID_{C_i}, M, M', M'', W, K)$ is computed by using a one-way hash function on the session secrets. Due to the one-way property of hash function, an outsider cannot takeout (M, M'', K) from SK. Moreover, *SK* is distributed uniformly in $\{0,1\}^k$ thus, each session has separate *SK* as it is depends on the short-term secrets r_{C_i} and r_S, which are generated independently and both will be different in each session. Therefore, disclosure of some previous session keys does not expose the current session key.

5.9. Perfect forward secrecy

The property of perfect secrecy states that the compromise of the private keys of both the participating entities does not affect the security of the previous session keys. The proposed scheme supports the perfect forward security in all respects. Since, the client C_i and the server S compute the session key $SK = H_3(ID_{C_i}, M, M', M'', W, K)$, where $M'' = D_{C_i}xP$ and $K = r_{C_i}r_SP$. If C_i's private keys, or S's private key is compromised to an adversary E, then E can compute M'', but not K due to the difficulties of CDH problem. Thus, our scheme satisfies the perfect forward security.

5.10. No key control

In the proposed scheme, after mutual authentication, both C_i and S compute the session key $SK = H_3(ID_{C_i}, M, M', M'', W, K)$ that depends on the ephemeral secrets r_{C_i} and r_S contributed by C_i and S. Therefore, neither S nor C_i can force the session key *SK* to a pre-selected value or lies with in a set containing the small number of elements. Hence, our protocol satisfies the no key control property.

In order to analyze the efficiency of the proposed scheme, we list some of the security requirements and make comparisons of our scheme with other related schemes [5, 9, 14] in Table 2, which shows that our scheme is more efficient than others and can be applicable in mobile client-server environments.

6. Comparison with other schemes

This section evaluates the performance of the proposed scheme in terms computation cost with competitive schemes [5, 9, 14]. To estimate the computation cost of our scheme, we define the following notations: **PM** is the time complexity to execute elliptic curve scalar point multiplication, **H** is the time complexity to execute hash operation and **X** is the time complexity to execute XOR operation. It is to be noted that the XOR operation needs very few computations; it is usually neglected considering its computational

Schemes/Attacks	**Yang-Chang**	**Yoon-Yoo**	**Debaio et al.**	**Our scheme**
Known session-specific temporary information attack	No	No	No	Yes
Many logged-in users' attack	No	No	No	Yes
Impersonation attack	No	Yes	No	Yes
Revocation of leaked private key	No	No	No	Yes
Users' anonymity problem	No	No	No	Yes
Clock synchronization	No	No	No	Yes
Session key forward secrecy	No	No	Yes	Yes
Mutual authentication and session key agreement	Yes	Yes	Yes	Yes
Yes: Resists the attack; **No:** Vulnerable to the attack.				

Tab 2. Security comparison

cost. The computation cost of a scheme is defined by the time spent by the client and the server for registration phase and mutual authentication with session key agreement phase. The Yang-Chang scheme needs **9PM+9H**, Debaio et al. needs **7PM+11H** where as our scheme needs **7PM+7H+2X**. Besides, our scheme avoids the problem of clock synchronization, and achieves users' anonymity as well, which requires two extra XOR operations. In addition, the proposed scheme can offer resilience against various attacks such as many logged-in users' attack, lost/stolen mobile device attack, impersonation attack, known session-specific temporary information attack, privileged-insider attack, replay attack, etc. We summarize the computation cost of our scheme and carried out a comparison with other schemes [5, 9, 14] in Table 3, which shows that our scheme is efficient ID-based client authentication scheme for mobile client-server environments.

Schemes/Computation cost	Yang-Chang	Yoon-Yoo	Debaio et al.	Proposed
Registration phase	1PM+1H	1PM+1H	1PM+1H	1PM+1H
Mutual authentication phase	8PM+8H	7PM+12H	6PM+10H	6PM+6H+2X
Total computation cost	9PM+9H	8PM+13H	7PM+11H	7PM+7H+2X
PM: Elliptic curve scalar point multiplication; **H:** hash operation; **X:** XOR operation.				

Tab 3. Efficiency comparison

7. Conclusions and future works

In this paper, we analyzed the Debaio et al.'s ID-based client authentication with session key agreement scheme and found that it is unable to resist privileged-insider attack, many logged in users' attack, impersonation attack and known session-specific temporary information attack. Besides, their scheme does not provide users' anonymity and the leaked key revocation phase with the previous identity. In addition, Debaio et al.'s scheme has the problem of clock synchronization. As a remedy, we proposed an improved ID-based client authentication with session key agreement scheme for mobile client-server environments using elliptic curve cryptosystem. The rigorous analysis of security and efficiency of the proposed scheme has been made, which shows that our scheme resists more attacks than the Debaio et al.'s scheme with lesser computational overheads.

An improved dynamic identity-based remote user mutual authentication with session key agreement scheme for mobile client-server environments is proposed. It can be noted that, our scheme is secured against all possible attacks known; however, the security analysis of it in a computational model like random oracle model may be carried out to have provable security features.

Acknowledgements

The authors are grateful to the anonymous reviewers for their valuable comments and suggestions, which help to improve this paper. This research work is supported by the Department of Science and Technology (DST), Govt. of India under the INSPIRE fellowship Ph.D program (Reg. No. IF10247) and the Department of Information Technology (DIT), Ministry of Communication and Information Technology, Govt. of India under the Information Security Education and Awareness (ISEA) program (Project No. MIT(2)/2006−08/189/CSE). The authors would also like to express their gratitude and heartiest thanks to the Department of Computer Science and Engineering, Indian School of Mines, Dhanbad, India for providing their research support, as without such help this work could not be carried out.

References

1. T. ElGamal: *A public key cryptosystem and a signature protocol based on discrete logarithms*, IEEE Transactions on Information Theory 31, 1985, 469-472.
2. R.L. Rivest, A. Shamir, L. Adleman: *A method for obtaining digital signatures and public key cryptosystems*, Communications of the ACM 21, 1978, 120-126.
3. V.S. Miller: *Use of elliptic curves in cryptography*, In: Proceeding of the Advances in Cryptology – Crypto'85, Springer-Verlag, New York, USA, 1985, pp. 417-426.
4. N. Koblitz: *Elliptic curve cryptosystem*, Mathematics of Computation 48, 1987, 203-209.
5. J.H. Yang, C.C. Chang: *An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem*, Computers & Security 28, 2011, 138-143.
6. P.E. Abichar, A. Mhamed, B. Elhassan: *A fast and secure elliptic curve based authenticated key agreement scheme for low power mobile communications*, In: Proceedings of the 2007 International Conference on Next Generation Mobile Applications, Services and Technologies, 2007, pp. 235-240.

7. Z. Jia, Y. Zhang, H. Shao, Y. Lin, J. Wang: *A remote user authentication scheme using bilinear pairings and ECC*, In: Proceedings of the Sixth International Conference on Intelligent System Design and Applications (ISDA), Jinan, China, 2006, pp. 1091-1094.
8. S.T. Wu, J.H. Chiu, B.C. Chieu: *ID-based remote authentication with smartcards on open distributed system from elliptic curve cryptography*, In: Proceedings of IEEE International Conference on Electro Information Technology, Lincoln, Nebraska, USA, May 22-25, 2005, pp. 5-9.
9. E. Yoon, K. Yoo: *Robust ID-based remote mutual authentication with key agreement protocol for mobile devices on ECC*, In: Proceedings of the International Conference on Computational Science and Engineering, Vancouver, Canada, 2009, pp. 633-640.
10. T.H. Chen, Y.C. Chen, W.K. Shih: *An Advanced ECC ID-Based remote mutual authentication scheme for mobile devices*, In: Proceedings of the Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, Xian, Shaanxi, China, 2010, pp. 116-120.
11. S.H. Islam, G.P. Biswas: *A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem*, The Journal of Systems and Software 84, 2011, 1892-1898.
12. S.H. Islam, G.P. Biswas: *Design of improved password authentication and update scheme based on elliptic curve cryptography*, Mathematical and Computer Modelling. http://dx.doi.org/10.1016/j.mcm.2011.07.001.
13. T-H. Chen, Y-C. Chen, W-K. Shih, H-W.Wei: *An efficient anonymous authentication protocol for mobile pay-TV*. Journal of Network and Computer Applications 34(4), 1131-1137, 2011.
14. H. Debiao, C. Jianhua, H. Jin: *An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security*, Information Fusion, 2011, <doi:10.1016/j.inffus.2011.01.001>.
15. A.K. Das, P. Sharma, S. Chatterjee, J.K. Sing: *A dynamic password-based user authentication scheme for hierarchical wireless sensor networks*, Journal of Network and Computer Applications. (In Press).
16. A. Shamir: *Identity-based cryptosystems and signature schemes*, In: Proceeding of the Advances in Cryptology – Crypto'84, Springer-Verlag, New York, USA, 1984, pp. 47-53.
17. D. Boneh, M. Franklin: *Identity-based encryption from the Weil pairing*, SIAM Journal of Computing 32, 2003, 586-615.
18. M.L. Das, A. Saxena, V. P. Gulati, D.B. Phatak: *A novel remote client authentication protocol using bilinear pairings*, Computers & Security 25, 2006, 184-189.
19. M.L. Das, A. Saxena, V. P. Gulati: *A dynamic ID-based remote user authentication scheme*, IEEE Transactions on Consumer Electronics 50, 2004, 629-631.
20. J.S. Chou, Y. Chen, J.Y. Lin: *Improvement of Das et al.'s remote user authentication scheme*, <http://eprint.iacr.org/2005/450.pdf>.

21. T. Goriparthi, M.L. Das, A. Saxena: *An improved bilinear pairing based remote user authentication scheme*, Computer Standards & Interfaces 31, 2009, 181-185.
22. Y.M. Tseng, T.Y. Wu, J.D. Wu: *A pairing-based client authentication protocol for wireless clients with smartcards*, Informatica 19, 2008, 285-302.
23. Y.Y. Wang, J.Y. Kiu, F.X. Xiao, J. Dan: *A more efficient and secure dynamic ID-based remote user authentication scheme*, Computer Communications 32, 2009, 583-585.
24. M.K. Khan: *Cryptanalysis and security enhancement of a 'more efficient & secure dynamic ID-based remote user authentication scheme'*, Computer Communications 34, 2011, 305-309.
25. S.H. Islam, G.P. Biswas: *Comments on ID-Based Client Authentication with Key Agreement Protocol on ECC for Mobile Client-Server Environment*, First International Conference on Advances in Computing and Communications (ACC 2011), part II, CCIS, Springer-Verlag, Berlin Heidelberg, vol. 191, 2011, pp. 628-635.
26. R. Canetti, H. Krawczyk: *Analysis of key exchange protocols and their use for building secure channels*, In: Proceeding of the Advances in Cryptology – Eurocrypt'01, LNCS, Springer-Verlag, Berlin Heidelberg, vol. 2045, 2001, pp. 451-472.
27. Z. Cheng, M. Nistazakis, R. Comley, L. Vasiu: *On the indistinguishability-based security model of key agreement protocols – simple cases*, <http://eprint.iacr.org/2005/129 >.
28. L. Gong: *A security risk of depending on synchronized clocks*, ACM Operating System Review 26, 1992, 49-53.
29. S. H. Islam, G. P. Biswas: *An improved remote login scheme based on ECC*, In: Proceedings of the International Conference on Recent Trends in Information Technology, 2011, pp. 1221-1226.