Chapter 5

Pairing-free Identity-based Authenticated Group Key Agreement Protocol

The secure and reliable group communication gains popularity in imbalanced mobile networks due to the increase demand of group-oriented applications such as teleconferences, collaborative workspaces, etc. For acquiring the group security objectives, many authenticated group key agreement (AGKA) protocols exploiting the CA-PKC/PKI have been proposed, which require additional processing and storage space for validation of the public keys and the certificates. In addition, the most of the AGKA protocols have high computation cost due to the bilinear pairing and MTP hash function. Owing the limitation of communication bandwidth, computation ability, and storage space of the low-power mobile devices, these protocols are not suitable for insecure imbalanced mobile networks. To cope with the aforementioned problems, we proposed a pairing-free and identity-based authenticated group key agreement (ID-AGKA) protocol using ECC. It is found that our protocol, compared with others, not only improves the computational efficiencies but, also enhances the security features.

5.1 Introduction

The AGKA protocol allows a group of entities to agree upon a common contributory group session key which can subsequently be used to encrypt/decrypt all the transmitted message among them through an open network. There are two types of group key establishment protocols: authenticated group key distribution (AGKD) protocol [196, 197, 198, 199] and authenticated group key agreement (AGKA) protocol [43, 44, 85, 200, 201, 202]. In the former protocol, an entity determines the group key and distributes it to other entities securely, whereas in the latter protocol, all the entities cooperatively determine the group key, i.e., neither entity should be able to force the group key to be a preselected value. That is, the AGKA protocol achieves the *no key control* (NKC) property [95] of a session key agreement protocol whereas the AGKD protocol does not. The wireless networks can be divided in two categories in general: balanced wireless network and imbalanced wireless network. In an imbalanced wireless network, an authentication server (Home Network) authenticates the users of mobile devices. Note that the mobile devices have limited resource and low computing ability, whereas the authentication server has stronger computing capability with fewer restrictions. Compared with wired networks and balanced wireless networks, there are only a few AGKA protocols [17, 19, 91, 203, 204] for imbalanced wireless networks.

Due to the public key certificate management problem and security vulnerabilities, the traditional PKI-based AGKA protocols [17, 19, 20, 21, 43, 85, 91, 200, 203] are less efficient for low-power mobile devices. Furthermore, some of the PKI-based AGKA protocols [19, 20, 21, 91] using bilinear pairing and MTP hash function [34] are too costly and they cannot be applicable in imbalanced mobile networks. These cause the computation loads and the energy consumptions of mobile devices very high, and thus, the protocols mentioned earlier are not suitable for limited bandwidth, computation ability and storage space environments such as imbalanced mobile network. Therefore, the problems of traditional GKA protocols motivated us to devise an efficient and

secure contributory AGKA protocol which will be more suitable for imbalanced mobile networks. Here, we proposed an efficient pairing-free ID-AGKA protocol for such networks using ECC in which the session key is derived as a function of contributions provided by all mobile nodes. The proposed protocol thus eliminates CA for the public key authentication and mitigates the computation cost by using IBC and ECC.

This chapter is organized as follows. In Section 5.2, we present our ECC-based pairing-free ID-AGKA protocol. The security analysis of the proposed protocol is given in Section 5.3. We compare the efficiency of the proposed protocol with other related protocols in Section 5.4. Finally, Section 5.5 concludes the chapter.

5.2 Proposed ID-AGKA Protocol based on ECC

In this section, the proposed ID-AGKA protocol for imbalanced mobile networks using ECC is described. Note that an imbalanced mobile network comprises with an authentication server (powerful node) that has a fixed infrastructure with unlimited resource (e.g., power, computing, storage, etc.) and a number of mobile devices (low-power node powered by battery), which have limited power without any fixed infrastructure. The proposed protocol can be implemented easily for practical application in mobile networks as it is free from pairings and MTP hash function. Similar to the works proposed in [19, 20, 21, 202], the following three assumptions have been considered in our protocol. Firstly, let $U=\{U_1, U_2, \cdots, U_{n-1}\}$ be the set of low-power mobile nodes and U_n be the powerful node of the network, however, each $U_i(1\leq i\leq n)$ can execute the proposed protocol. Secondly, each group member at beginning must know the identity of other group members by some sort of other mechanism so that secure group is formed. Thirdly, the node U_n has the authorization of adding and removing the low-power nodes from the group. Our protocol consists of five phases: setup phase, key extraction phase, authenticated group key agreement phase, removal phase and joining phase, where setup and key extraction phases are based on the works proposed in [15, 205]. The notations used throughout the chapter are elaborated in Table 5.1.

Table 5.1: The notations used in the proposed protocol

Notation	Description
U_i	A low-power mobile node ($1 \leq i \leq n-1$)
U_n	The powerful node
ID_i	Identity of the node U_i ($1 \leq i \leq n$)
d_i	The private key of the node U_i ($1 \leq i \leq n$)
P_i	The public key of the node U_i ($1 \leq i \leq n$), where $P_i = d_i P$
q	A large prime number such that $q \geq 2^k$, k is a security parameter
G_q	An additive elliptic curve group of prime order q
P	The generator of the elliptic curve group G_q
H_0, H_1	Two one-way and secure general cryptographic hash functions
n	The number of participants involved in generating a group session key
$\|$	Concatenation operation

5.2.1 Setup Phase

For given a security parameter $k \in Z^+$, the PKG runs this algorithm and generates the system's parameter and a master key as follows:

(a) Choose a k-bit prime q and determine the tuple $\{F_q, E/F_q, G_q, P\}$, where the point P is the generator of G_q.

(b) Choose $x \in_R Z_q^*$ as master key and compute the system public key $P_{pub} = xP$.

(c) Choose two cryptographic secure and one-way hash functions $H_0: \{0,1\}^* \times G_q \longrightarrow Z_q^*$ and $H_1: \{0,1\}^* \longrightarrow \{0,1\}^k$.

(d) Publish $\Omega = \{F_q, E/F_q, G_q, P, P_{pub}, H_0, H_1\}$ as system's parameter and keep the master key x secret.

5.2.2 Key Extraction Phase

This algorithm takes PKG's master private key, identity of a user, and the system's parameter as input, and then returns the identity-based long-term private key of a user as explained below. For a user i with identifier ID_i, the PKG executes the following operations:

(a) Choose a number $r_i \in_R Z_q^*$, compute $R_i = r_i P$ and $h_i = H_0(ID_i \| R_i)$.

(b) Compute $d_i = (r_i + h_i x) \bmod q$.

The PKG sends (d_i, R_i) via a secure and confidential channel to the user ID_i. The corresponding public key of ID_i is computed as $P_i = R_i + H_0(ID_i \| R_i) P_{pub}$ and the private/public key pair (d_i, P_i) can be verified by checking whether the equation $P_i = R_i + H_0(ID_i \| R_i) P_{pub} = d_i P$ holds as

$$\begin{aligned} P_i &= R_i + H_0(ID_i \parallel R_i) P_{pub} \\ &= r_i P + H_0(ID_i \parallel R_i) x P \\ &= (r_i + H_0(ID_i \parallel R_i) x) P \\ &= (r_i + h_i x) P \\ &= d_i P \end{aligned} \tag{5.1}$$

5.2.3 Authenticated Group Key Agreement Phase

Step 1. In this round, each low-power node U_i (identified by the identity ID_i) for $(1 \leq i \leq n-1)$ picks a number $a_i \in_R Z_q^*$ and performs the following:

(a) Compute $T_i = a_i P$ and $S_i = (d_i + a_i)(a_i H_1(ID_i \| T_i) + d_i)^{-1}$.

(b) Send the message (ID_i, T_i, S_i, R_i) to the powerful node U_n.

Step 2. Upon receiving each message (ID_i, T_i, S_i, R_i) from each low-power node U_i $(1 \leq i \leq n-1)$, the powerful node U_n (identified by the identity ID_n) selects a number $a_n \in_R Z_q^*$ and executes the following operations:

(a) U_n verifies whether the equation $S_i[H_1(ID_i \| T_i)T_i + P_i] = (T_i + P_i)$ holds for $(1 \leq i \leq n-1)$, where $P_i = (R_i + H_0(ID_i \| R_i)P_{pub})$. If it holds, U_n can ensure that $(ID_i, T_i, S_i, R_i)(1 \leq i \leq n-1)$ are sent by $U_i(1 \leq i \leq n-1)$ and each of them are authentic.

(b) U_n computes $T_n = a_n P$ and $T = \sum_{i=1}^{n}(T_i + P_i)$

(c) U_n computes $\overline{Z} = (a_n + d_n)T$ and $Z_i = (a_n + d_n)^2(T_i + P_i)$ for the low-power node $U_i(1 \leq i \leq n-1)$.

(d) U_n computes $ID = ID_1 \| ID_2 \| \cdots \| ID_n$, $Z = Z_1 \| Z_2 \| \cdots \| Z_{n-1} \| \overline{Z} \| T_n$ and $S_n = (a_n + d_n)(a_n H_1(ID_n \| \overline{Z} \| Z \| T_n) + d_n)^{-1}$.

(e) U_n computes the partial session key as $K = (a_n + d_n)(T - T_n - P_n)$ and the final session key as $SK = H_1(ID \| Z \| K)$.

(f) Then the powerful node U_n broadcast the message $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ to other low-power node $U_i(1 \leq i \leq n-1)$.

Step 3. After receiving the message $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ from the powerful node U_n, each low-power node $U_i(1 \leq i \leq n-1)$ computes $ID = ID_1 \| ID_2 \| \cdots \| ID_n$, $Z = Z_1 \| Z_2 \| \cdots \| Z_{n-1} \| \overline{Z} \| T_n$ and $P_n = [R_n + H_0(ID_n \| R_n)P_{pub}]$ and then verifies whether the equation $S_n[T_n H_1(ID_n \| \overline{Z} \| Z \| T_n) + P_n] = (T_n + P_n)$ holds. If it holds, each $U_i, (1 \leq i \leq n-1)$ can ensure that the message $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ is authenticated and is sent by the powerful node U_n. Then each $U_i(1 \leq i \leq n-1)$ computes the partial session key $K_i = \overline{Z} - (a_i + d_i)^{-1} Z_i$ and the contributory group session key $SK = H_1(ID \| Z \| K)$, where $K_1 = K_2 = \cdots = K_{n-1} = K$. The proposed pairing-free ID-AGKA protocol based on ECC is shown in Fig. 5.1.

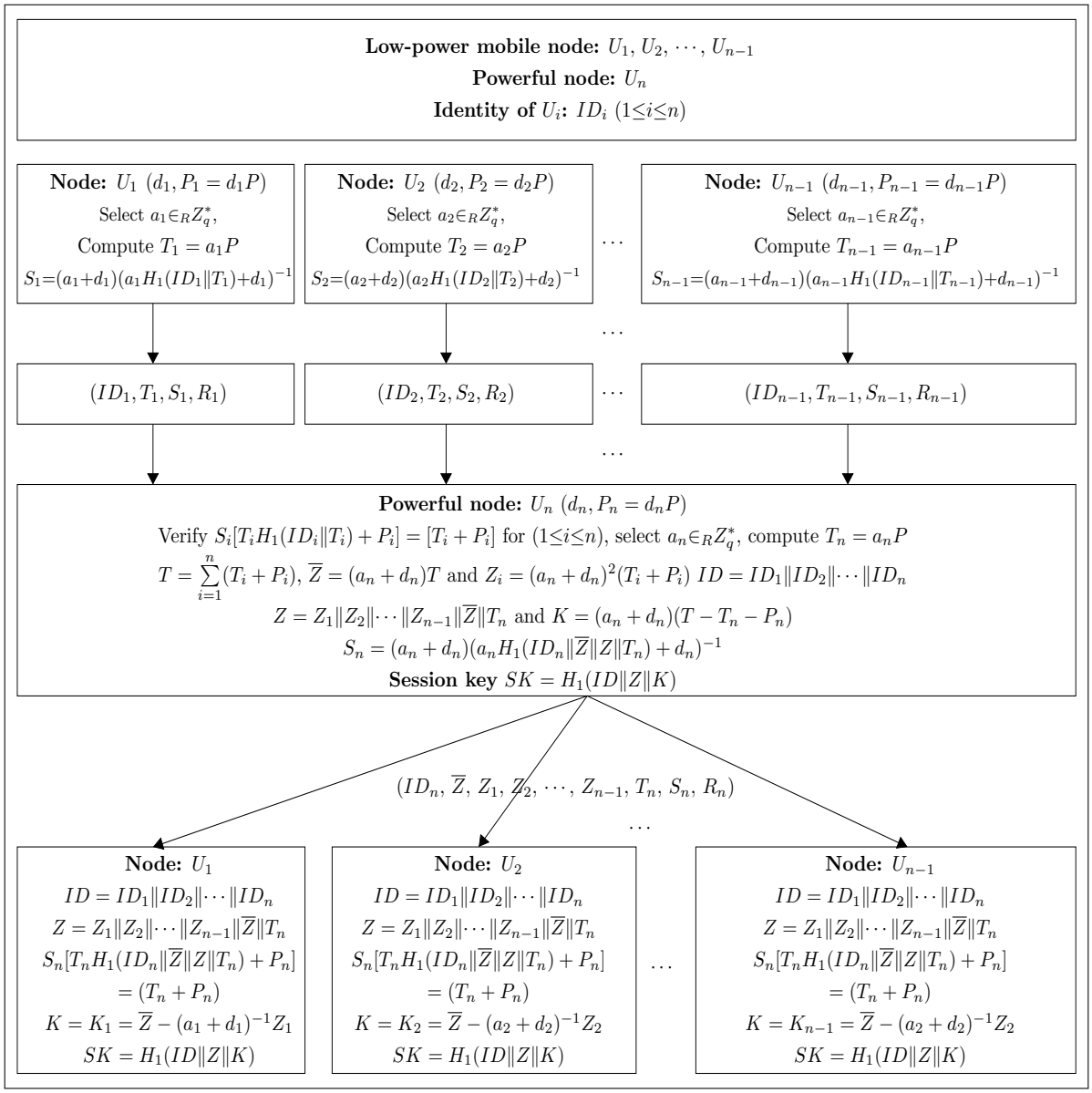

Figure 5.1: Proposed ID-AKGA protocol for imbalanced mobile networks.

In our work, we assume that each low-power node U_i ($1 \leq i \leq n-1$) and the powerful node U_n as well properly send the required messages and execute the protocol to establish a secure group key. Now in a session, if a node U_t (say) did not send its contribution to U_n, then U_n computes Z_i ($1 \leq i \leq n-1$, $i \neq t$) and S_n without considering the node U_t and broadcasts the message (ID_n, ID_t, $\overline{Z}$, Z_1, Z_2, $\cdots$, Z_{t-1}, Z_{t+1}, $\cdots$, Z_{n-1}, T_n, S_n, R_n) to each U_i ($1 \leq i \leq n-1$) including U_t as shown, to inform other low-power nodes not to include U_t in the group key formation. Then, each node U_i ($1 \leq i \leq n-1$, $i \neq t$) computes the session key in the same fashion as described earlier, where the identity ID_t of U_t is excluded.

5.2.4 Removal Phase

When a user or a set of users wish to leave the group, the removal phase occurs and in this case, either a new group key or the modification of the existing group is necessary for the protection of the group. In this section, we proposed a modification of the existing group key in such a way that none of the leaving user can compute the subsequent group key generated. Suppose that a set of low-power mobile nodes $\overline{U} = \{U_{j+1}, U_{j+2} \cdots, U_{n-1}\}$ wish to leave the group, the proposed protocol for implementing the removal phases is given below.

Step 1. **(a)** Each U_k ($j+1 \leq k \leq n-1$) informs U_n as they wants to leave the group.

(b) U_n then updates the group $U' = U \backslash \overline{U}$.

(c) U_n selects $a'_n \in_R Z_q^*$, computes $T'_n = a'_n P$, $T' = \sum_{i=1}^{j}(T_i + P_i) + (T'_n + P_n)$.

(d) U_n computes $\overline{Z}' = (a'_n + d_n)T'$, $Z'_i = (a'_n + d_n)^2 T_i$, for the low-power node U_i ($1 \leq i \leq j$).

(e) U_n computes $ID' = ID_1 \| ID_2 \| \cdots \| ID_j \| ID_n$, $Z' = Z'_1 \| Z'_2 \| \cdots \| Z'_j \| \overline{Z}' \| T'_n$ and $S'_n = (d_n + a'_n)(a'_n H_1(ID_n \| \overline{Z}' \| Z' \| T'_n) + d_n)^{-1}$.

(f) U_n computes the partial session key as $K' = (a'_n + d_n)(T' - T'_n - P_n)$ and the final session key as $SK = H_1(ID'\|Z' \|K')$.

(g) Then U_n broadcast the message $(ID_n, \overline{Z}', Z'_1, Z'_2, \cdots, Z'_j, T'_n, S'_n, R_n)$ to each U_i $(1 \leq i \leq j)$.

Step 2. On receiving the message $(ID_n, \overline{Z}', Z'_1, Z'_2, \cdots, Z'_j, T'_n, S'_n, R_n)$, each low-power node U_i $(i \leq i \leq j)$ computes $ID' = ID_1\|ID_2\| \cdots \|ID_j\|ID_n$, $Z' = Z'_1\|Z'_2\| \cdots \|Z'_j\|\overline{Z}'\|T'_n$, $P_n = [R_n + H_0(ID_n\|R_n)P_{pub}]$ and verifies whether the equation $S'_n(T'_nH_1(ID_n\|\overline{Z}'\|Z'\|T'_n) + P_n) = (T'_n + P_n)$ holds. If it holds, each node U_i $(i \leq i \leq j)$ authenticates the message $(ID_n, \overline{Z}', Z'_1, Z'_2, \cdots, Z'_j, T'_n, S'_n, R_n)$ and its sender U_n. Subsequently, each U_i $(1 \leq i \leq j)$ computes the partial session key as $K'_i = \overline{Z}' - (a_i + d_i)^{-1}Z'_i = K'(1 \leq i \leq j)$ and the contributory group session key $SK' = H_1(ID'\|Z'\|K')$.

5.2.5 Joining Phase

This phase occurs when a new user or a set of new users want to join the existing group. In order to provide the fairness in the group key formation, the existing group key in this case should also be updated by including the contributions of the new members, however, it should be done in such a manner that none of new members can compute any of the previous group session keys. Suppose that a set of low-power mobile nodes $\widehat{U}$=$\{\mathrm{U}_{n+1}, U_{n+2}, \cdots, U_m\}$ wish to join an existing group. The new contributory group session key in this phase for the mobile nodes $U'' = U \cup \widehat{U}$ with U_n will be computed as follows:

Step 1. **(a)** Each members of $\widehat{U}$ send their identity to U_n.

(b) U_n updates the group $U'' = U \cup \widehat{U}$.

(c) Each U_k $(n + 1 \leq k \leq m)$ picks a number $a_k \in_R Z_q^*$, computes $T_k = a_kP$, $S_k = (a_k + d_k)(a_kH_1(ID_k\|T_k) + d_k)^{-1}$ and sends (ID_k, T_k, S_k, R_k) to the powerful node U_n.

Step 2. Upon receiving the messages (ID_k, T_k, S_k, R_k) ($n+1 \leq k \leq m$), U_n executes the following operations:

(a) U_n verifies the integrity of each (ID_k, T_k, S_k, R_k) ($n+1 \leq k \leq m$) as discussed earlier.

(b) U_n selects a number $a_n'' \in_R Z_q^*$, and computes $T_n'' = a_n'' P$ and $T'' = \sum_{i=1, i \neq n}^{m} (T_i + P_i) + (T_n'' + P_n)$.

(c) U_n computes $\overline{Z}'' = (a_n'' + d_n) T''$, $Z_i'' = (a_n'' + d_n)^2 T_i$ for U_i ($1 \leq i \leq m$, $i \neq n$).

(d) U_n computes $ID'' = ID_1 \| ID_2 \| \cdots \| ID_n \| \cdots \| ID_m$, $Z'' = Z_1'' \| Z_2'' \| \cdots \| Z_{n-1}'' \| Z_{n+1}'' \| \cdots \| Z_m'' \| \overline{Z}'' \| T_n''$ and $S_n'' = (a_n'' + d_n)(a_n'' H_1(ID_n \| \overline{Z}'' \| Z'' \| T_n'') + d_n)^{-1}$.

(e) U_n computes the partial session key as $K'' = (a_n'' + d_n)(T'' - T_n'' - P_n)$ and the final session key as $SK'' = H_1(ID'' \| Z'' \| K'')$.

(f) Then U_n broadcast the message (ID_n, $\overline{Z}''$, Z_1'', Z_2'', $\cdots$, Z_{n-1}'', Z_{n+1}'', $\cdots$, Z_m'', S_n'', T_n'', R_n) to each U_i ($1 \leq i \leq m$, $i \neq n$).

Step 3. After receiving (ID_n, $\overline{Z}''$, Z_1'', Z_2'', $\cdots$, Z_{n-1}'', Z_{n+1}'', $\cdots$, Z_m'', S_n'', T_n'', R_n) from U_n, each U_i ($1 \leq i \leq m$, $i \neq n$) computes $ID'' = ID_1 \| ID_2 \| \cdots \| ID_n \| \cdots \| ID_m$, $Z'' = Z_1'' \| Z_2'' \| \cdots \| Z_{n-1}'' \| Z_{n+1}'' \| \cdots \| Z_m'' \| \overline{Z}'' \| T_n''$, $P_n = [R_n + H_0(ID_n \| R_n) P_{pub}]$ and then verifies the integrity of the received message as said earlier. If the integrity check satisfies, then each low-power node U_i ($1 \leq i \leq m$, $i \neq n$) confirms that the received message is really sent by the powerful node U_n. Thereafter, each U_i ($1 \leq i \leq m$, $i \neq n$) computes $K_i'' = \overline{Z}'' - (a_i + d_i)^{-1} Z_i'' = K''$ ($1 \leq i \leq m$, $i \neq n$) and the contributory group session key $SK'' = H_1(ID'' \| Z'' \| K'')$.

5.3 Security Analysis

This section provides an in-depth security analysis of the proposed pairing-free ID-AGKA protocol. According to [19, 91, 199, 200], an AGKA protocol is secure if

it satisfies the following requirements: *contributiveness*, *message integrity*, *resilience against passive attack* and *forward/backward secrecy* for joining/removing operation. Since the proposed protocol is a contributory group key agreement protocol, it provides resilience against other relevant known attacks also such as *known group key attack*, *key compromise impersonation attack*, *known session-specific temporary information attack*, *impersonation attack*, etc., as described in [12, 94, 95, 96, 97]. It can be noted that the security of the proposed protocol concerns both the privacy of the authenticated group session key and the integrity of the transmitted messages. The security of our group session key relies on the difficulties of OWHF, ECDLP, and CDHP assumptions. Now, in order to validate security claim of the proposed protocol, the following theorems are given.

Theorem 5.3.1 (**Contributiveness**). *Following the proposed ID-AGKA protocol, an identical contributory group session key is established by all mobile nodes, i.e., each mobile client confirms its contribution included in the group key.*

Proof. We know that an AGKA protocol is said to be a contributory group key agreement protocol if each and every member of a group contributes in forming the group session key. In the proposed protocol, each low-power node $U_i(1{\leq}i \leq n-1)$ unicast the message $(ID_i, T_i, S_i, R_i)(1{\leq}i \leq n-1)$ to U_n. The powerful node U_n validates $(ID_i, T_i, S_i, R_i)(1{\leq}i{\leq}n-1)$ and then computes $ID = ID_1\|ID_2\|\cdots\|ID_n$, $Z{=}Z_1\|Z_2\|\cdots\|Z_{n-1}\|\overline{Z}\|T_n$, the group session key $SK = H_1(ID\|Z\|K)$, where the partial session key is $K{=}(a_n+d_n)(T-T_n-P_n){=}a_nd_n\{(a_1+d_1)+(a_2+d_2)+\cdots+(a_{n-1}+d_{n-1})\}P$.

Further, U_n broadcast the message $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ to each $U_i(1 \leq i \leq n-1)$, where $Z_i = (a_n+d_n)^2 T_i = (a_n+d_n)^2(a_i+d_i)P$ for $(1 \leq i \leq n-1)$. After validating the message $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$, each $U_i(1 \leq i \leq n-1)$ generates the contributory group session key by computing $SK = H_1(ID \| Z \| K)$. Since all the mobile node $U_i(1 \leq i \leq n)$ wants to generate an identical contributory group session key SK, this means that the following equation must holds $K{=}K_i = \overline{Z} - (a_i + d_i)^{-1} Z_i$ for $(1 \leq i \leq n)$. Since $T = \sum_{i=1}^{n} (T_i + P_i) = \{(a_1+d_1)+(a_2+d_2)+\cdots+(a_n+d_n)\}P$, $\overline{Z} = (a_n + d_n)T = (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_n + d_n)\}P$ and $Z_i = (a_n + d_n)^2(T_i + P_i)$. Therefore, we have

$$Z_1 = (a_n + d_n)^2(T_1 + P_1) = (a_n + d_n)^2(a_1 + d_1)P$$

$$Z_2 = (a_n + d_n)^2(T_2 + P_2) = (a_n + d_n)^2(a_2 + d_2)P$$

$$\vdots$$

$$Z_{n-1} = (a_n + d_n)^2(T_{n-1} + P_{n-1}) = (a_n + d_n)^2(a_{n-1} + d_{n-1})P$$

Now each low-power node U_i $(1 \leq i \leq n-1)$ computes

$$\begin{aligned} U_1: \quad K_1 &= \overline{Z} - (a_1 + d_1)^{-1} Z_1 \\ &= (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_n + d_n)\}P - (a_n + d_n)^2 P \\ &= (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P \end{aligned}$$

$$
\begin{aligned}
U_2: \quad K_2 &= \overline{Z} - (a_2 + d_2)^{-1} Z_2 \\
&= (a_n + d_n)(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_n + d_n)\}P - (a_n + d_n)^2 P \\
&= (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P \\
&\vdots \\
U_{n-1}: \quad K_{n-1} &= \overline{Z} - (a_{n-1} + d_{n-1})^{-1} Z_{n-1} \\
&= (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_n + d_n)\}P - (a_n + d_n)^2 P \\
&= (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P
\end{aligned}
$$

Thus, we get $K = K_1 = K_2 = \cdots = K_{n-1} = (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P$. It is worth to note that each $K_i(1 \leq i \leq n-1)$ contains the ephemeral secrets $a_i(1 \leq i \leq n)$ and the private key $d_i(1 \leq i \leq n)$ of the node $U_i(1 \leq i \leq n)$. Therefore, SK is a contributory group session key and is identical for all nodes $U_i(1 \leq i \leq n)$. □

Theorem 5.3.2 (**Message integrity**). *If the equations* $S_i[T_i H_1(ID_i \| T_i) + P_i] = [T_i + P_i]$ *for* $(1 \leq i \leq n-1)$ *and* $S_n[T_n H_1(ID_n \| \overline{Z} \| Z \| T_n) + P_n] = [T_n + P_n]$ *hold, where* $P_i = [R_i + H_0(ID_i \| R_i) P_{pub}]$ *for* $(1 \leq i \leq n-1)$ *and* $P_n = [R_n + H_0(ID_n \| R_n) P_{pub}]$, *then the integrity of each* U_i*'s message* (ID_i, T_i, S_i, R_i) *for* $(1 \leq i \leq n-1)$ *and* U_n*'s message* $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ *are preserved.*

Proof. Since $T_i = a_iP$, $S_i = (a_i+d_i)(a_iH_1(ID_i\|T_i)+d_i)^{-1}$, $P_i = [R_i+H_0(ID_i\|R_i)P_{pub}]$

$= d_iP$ for $(1\leq i\leq n-1)$ and $P_n = [R_n + H_0(ID_n\|R_n)P_{pub}] = d_nP$.

Then we have,

$\mathrm{S}_i[T_iH_1(ID_i \parallel T_i) + P_i]$

$= S_i[a_iH_1(ID_i \parallel T_i) + d_i]P$

$= (a_i + d_i)(a_iH_1(ID_i \parallel T_i) + d_i)^{-1}(a_iH_1(ID_i \parallel T_i) + d_i)P$

$= (a_i + d_i)P$

$= a_iP + d_iP$

$= T_i + P_i$

Now $T_n = a_nP$ and $S_n = (a_n + d_n)(a_nH_1(ID_n\|\overline{Z}\|Z\|T_n) + d_n)^{-1}$. Therefore,

$\mathrm{S}_n[T_nH_1(ID_n \parallel \overline{Z} \parallel Z \parallel T_n) + P_n]$

$= (a_n + d_n)(a_nH_1(ID_n \parallel \overline{Z} \parallel Z \parallel T_n) + d_n)^{-1}(a_nH_1(ID_n \parallel \overline{Z} \parallel Z \parallel T_n) + d_n)P$

$= (a_n + d_n)P$

$= (a_nP + d_nP)$

$= T_n + P_n$

Thus, we conclude that the integrity of the messages (ID_i, T_i, S_i, R_i) and $(ID_n, \overline{Z}, Z_1,$

$Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ are preserved in the proposed protocol. □

Theorem 5.3.3 (**Passive attack**). *The proposed pairing-free ID-AGKA protocol is secure against the passive attack under the ECDLP and CDHP assumptions. That is, an attacker is unable to obtain the resulting group session key by eavesdropping the messages* (ID_i, T_i, S_i, R_i) $(1 \leq i \leq n-1)$ *and* $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ *transmitted over insecure network.*

Proof. We will show that the proposed ID-AGKA protocol is secure under the ECDLP and CDHP assumptions against the passive adversary who tries to generate the established group session key by listening the communication channel. An AGKA protocol is secure against the passive attack if the protocol is executed in the presence of an adversary but, he cannot get success to obtain the established group session key from the eavesdropped messages exchanged between participants. Assume that an attacker sniffing the communication channel and captures the messages (ID_i, T_i, S_i, R_i) $(1 \leq i \leq n-1)$ and $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ in the current session and tries to generate the group session key $SK = H_1(ID \| Z \| K)$ of that session. From the theorem (5.3.3), we can see that forgeries of these messages are not possible without the private key d_i $(1 \leq i \leq n)$. The attacker can generate the established session key SK if he knows the partial session key $K = (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P$. Due

to difficulties of ECDLP, an attacker cannot derive d_i and a_i, $(1 \leq i \leq n)$ from $P_i = d_i P$ and $T_i = a_i P$ $(1 \leq i \leq n)$, respectively and also the computation of $(a_n + d_n)(a_i + d_i)P$ $(1 \leq i \leq n-1)$ from the pair $\{(T_n + P_n), (T_i + P_i)\} = \{(a_n + d_n)P, (a_i + d_i P)\}$ $(1 \leq i \leq n)$ is infeasible for the CDHP assumption. Therefore, the attacker cannot generate the group session key SK from the public messages (ID_i, T_i, S_i, R_i) $(1 \leq i \leq n-1)$ and $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$. □

Theorem 5.3.4 (**Backward secrecy**). *The proposed ID-AGKA protocol provides backward secrecy for member(s) leaving the group provided the CDHP is intractable in elliptic curve group.*

Proof. The meaning of backward secrecy of any AGKA protocol is to allow any number of existing members to leave the group and to generate a new group key so that the leaving member(s) cannot compute or trace it. The proposed ID-AGKA protocol provides backward secrecy for the members leaving the group based on the hardness assumption of CDHP in the elliptic curve group. Let $\overline{U} = \{U_{j+1}, U_{j+2}, \cdots, U_{n-1}\}$ be a set of low-power mobile nodes leaving the group after a session. In order to generate a new group key between the remaining group members, U_n selects $a'_n \in_R Z_q^*$, computes $T'_n = a'_n P$, $T' = \sum_{i=1}^{j}(T_i + P_i) + (T'_n + P_n)$, $\overline{Z}' = (a'_n + d_n)T'$, $Z'_i = (a'_n + d_n)^2(T_i + P_i)$

for $(1 \leq i \leq j)$, $S'_n = (a'_n + d_n)(a'_n H_1(ID_n \| \overline{Z}' \| Z' \| T'_n) + d_n)^{-1}$ and then sends $(ID_n, \overline{Z}', Z'_1, Z'_2, \cdots, Z'_j, T'_n, S'_n, R_n)$ to each member $U_i(1 \leq i \leq j)$. Thereafter, each $U_i(1 \leq i \leq j)$ authenticates the message received from U_n and then generates the session key $SK = H_1(ID' \| Z' \| K')$, where $K' = \overline{Z}' - (a_i + d_i)^{-1} Z'_i = (a'_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_j + d_j)\}P$. Any member $U_k(j+1 \leq k \leq n-1)$ of the leaving group $\overline{U}$ can generate the current session key SK' if K' is known. However, none of $U_k(j+1 \leq k \leq n-1)$ can generate K' from the knowledge of previous $K = (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P$, since the new ephemeral secret a'_n is not known and it cannot be derived from $T'_n = a'_n P$ due to difficulties of ECDLP. Furthermore, any member of $\overline{U}$ can compute $\overline{T}' = \sum_{i=1}^{j}(T_i + P_i) = \sum_{i=1}^{j}(a_i + d_i)P$ from $(T_i, P_i), (1 \leq i \leq j)$ but, the new partial session key K' from the pair $(\overline{T}', T'_n) = \{((a_1 + d_1) + (a_2 + d_2) + \cdots + (a_j + d_j))P, (a'_n + d_n)P\}$ cannot be computed due to the hardness of CDHP. Thus, any member from the group $\overline{U}$ cannot generate the subsequent or any future group session key. Hence, the theorem follows. □

Theorem 5.3.5 (**Forward secrecy**). *For member(s) joining in group, the proposed ID-AGKA protocol provides forward secrecy based on hardness assumption of CDHP.*

Proof. The forward secrecy of an AGKA protocol is to allow the new member(s) to join

in a group and develop new group key without proving the scope for generating any previous group key to the new members. In this regard, the proposed ID-AGKA protocol provides forward secrecy based on the infeasibility of breaking the CDHP in the elliptic curve group. Let a set of new low-power mobile nodes $\widehat{U} = \{U_{n+1}, U_{n+2}, \cdots, U_m\}$ joins an existing group. For the generation of new group key among the group members $U_i(1 \leq i \leq m)$, U_n selects a number $a_n'' \in_R Z_q^*$, and then computes $T_n'' = a_n'' P$, $T'' = \sum_{i=1, i \neq n}^{m} (T_i + P_i) + (T_n'' + P_n)$, $\overline{Z}'' = (a_n'' + d_n)T''$, $Z_i'' = (a_n'' + d_n)^2(T_i + P_i)$, $S_n'' = (a_n'' + d_n)(a_n'' H_1(ID_n \| \overline{Z}'' \| Z'' \| T_n'') + d_n)^{-1}$ and broadcast the message $(ID_n, \overline{Z}'', Z_1'', Z_2'', \cdots, Z_{n-1}'', Z_{n+1}'', \cdots, Z_m'', S_n'', T_n'', R_n)$ to each U_i $(1 \leq i \leq m, i \neq n)$. After authentication, each U_i $(1 \leq i \leq m, i \neq n)$ follows the proposed protocol and computes the session key $SK'' = H_1(ID'' \| Z'' \| K'')$. Now, it is to be proved that none of the $U_k(n+1 \leq k \leq m)$ can generate any of previous session key $SK = H_1(ID \| Z \| K)$ (say) computed between $U_i(1 \leq i \leq n)$. Note that a newly joined member U_k can generate the previous SK if and only if the partial session key $K = (a_n + d_n)(T - T_n - P_n) = (a_n + d_n) \sum_{i=1}^{n-1} (T_i + P_i) = (a_n + d_n) \sum_{i=1}^{n-1} (a_i + d_i)P$ is known. Furthermore, any new member of $\widehat{U}$ can compute $\overline{T} = \sum_{i=1}^{n-1} (T_i + P_i)$ from the old messages $(T_i = a_i P, P_i = d_i P)$, $(1 \leq i \leq n-1)$ but, it is not possible to compute SK as no one can generate K from

the pair $(\overline{T},\ T_n + P_n) = (\sum_{i=1}^{n-1}(a_i + d_i)P,\ (a_n + d_n)P)$ due to the hardness of CDHP. In addition, any of U_k by knowing $K_i'' = \overline{Z}'' - (a_i + d_i)^{-1}Z_i'' = K''$ $(1{\leq}i\ {\leq}m,\ i\ {\neq}n)$ cannot derive K from K'' as the ephemeral secret a_n'' is unknown. Thus, the previous group session key SK cannot be generated by any of the new members of the group $\widehat{U} = \{U_{n+1}, U_{n+2}, \cdots, U_m\}$. Hence, our ID-AGKA protocol provides forward secrecy and the theorem is proved. □

Theorem 5.3.6. *The proposed pairing-free ID-AGKA protocol can resist the impersonation attack. That is, an attacker cannot masquerade either any low-power node or the powerful node to generate the group session key with other legitimate low-power nodes and the powerful node.*

Proof. The messages $(ID_i, T_i, S_i, R_i), (1{\leq}i{\leq}n-1)$ and $(ID_n,\ \overline{Z},\ Z_1,\ Z_2,\ \cdots,\ Z_{n-1},\ T_n,\ S_n,\ R_n)$ are transmitted among all low-power nodes $U_i(1{\leq}i{\leq}n-1)$ and the powerful node U_n over the public networks. It is impossible to forge none of the signatures S_i without the private key d_i of $U_i(1{\leq}i{\leq}n)$. Furthermore, these private keys are protected in the public keys $P_i = d_iP$ $(1{\leq}i{\leq}n)$ under ECDLP. If an adversary $\mathscr{A}$ modifies T_i and/or Z_i $(1{\leq}i\ {\leq}n)$, then the verification equations $S_i[a_iH_1(ID_i\|T_i) + P_i] = (T_i + P_i)$

$(1 \leq i \leq n-1)$ and $S_n[a_n H_1(ID_n \| \overline{Z} \| Z \| T_n) + P_n] = (T_n + P_n)$ does not hold. Therefore, the forgery attack on the message (ID_i, T_i, S_i, R_i), $(1 \leq i \leq n-1)$ and $(ID_n, \overline{Z}, Z_1, Z_2, \cdots, Z_{n-1}, T_n, S_n, R_n)$ are impossible and thus, the proposed protocol is robust against the impersonation attack. □

Theorem 5.3.7. *The proposed pairing-free ID-AGKA protocol withstands the known group session key attack under the difficulties of ECDLP and OWHF assumptions.*

Proof. A group key agreement protocol satisfies the known group session key security if one of the previously established group session key is compromised to an adversary but, the current or future group session keys should not be disclosed from the compromised group session key. Assume that an adversary $\mathscr{A}$ compromised a previous group session key $SK = H_1(ID \| Z \| K)$, where $K = (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P$. The security of the group session key SK mainly depends on the secrecy of K. The adversary $\mathscr{A}$ tries to used the compromised SK to compute other group session key $SK^* = H_1(ID \| Z^* \| K^*)$ (say) where $K^* = (a_n^* + d_n)\{(a_1^* + d_1) + (a_2^* + d_2) + \cdots + (a_{n-1}^* + d_{n-1})\}P$. However, $\mathscr{A}$ cannot launch the known group session key attack successfully, since SK depends on ephemeral secrets a_i $(1 \leq i \leq n)$ and the private keys d_i of U_i $(1 \leq i \leq n)$. The adversary $\mathscr{A}$ knows the information $(T_i, P_i) = (a_i P, d_i P)$ $(1 \leq i \leq n)$

and if (a_n, d_n) is known to him then the partial session key K can be computed and thus the resulting group session key SK will be compromised. However, $\mathscr{A}$ cannot derive (a_n, d_n) from $(T_n, P_n) = (a_nP, d_nP)$ due to difficulties of ECDLP. Furthermore, due to the infeasibility of OWHF, $\mathscr{A}$ cannot extract K from the compromised group session key SK. Therefore, our protocol can resist the known group session key attack. □

Theorem 5.3.8. *The proposed pairing-free ID-AGKA protocol is known session-specific temporary information attack protected under the CDHP assumption, i.e., if the ephemeral secrets* $(a_1, a_2, \cdots, a_n)$ *are disclosed, the established group session key remains secure from this disclosure.*

Proof. The known session-specific temporary information attack is absent in the proposed protocol. In our protocol, all the nodes $U_i (1 \leq i \leq n)$ generates the group session key by computing $SK = H_1(ID \| Z \| K)$, where $K = (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P$ and the security of SK depends on the secrecy of $(a_1, a_2, \cdots, a_n)$ and $(d_1, d_2, \cdots, d_n)$. Suppose that, in the current session, the ephemeral secrets $(a_1, a_2, \cdots, a_n)$ are disclosed to an adversary $\mathscr{A}$. However, $\mathscr{A}$ cannot compute the session key SK, since he/she has no knowledge about $(d_1, d_2, \cdots, d_n)$. The group session key SK can be computed by $\mathscr{A}$ if the partial session key K is known, and it

can be computed if $\mathscr{A}$ knows $(d_1, d_2, \cdots, d_n)$. For this purpose, $\mathscr{A}$ can try to derive $(d_1, d_2, \cdots, d_n)$ directly from $(P_1, P_2, \cdots, P_n)$ but, it is also impossible due to difficulties of ECDLP. Therefore, the proposed protocol protects the known session-specific temporary information attack. □

Theorem 5.3.9. *Under ECDLP and CDHP assumptions, the proposed protocol provides perfect forward security.*

Proof. We can say that an AGKA protocol satisfied the perfect forward secrecy if the private keys of all participating users may be compromised to an adversary without disclosing previously established group session keys. Suppose that the private keys d_i $(1{\leq}i{\leq}n)$ of all the nodes U_i $(1{\leq}i{\leq}n)$ are known to an adversary $\mathscr{A}$. However, $\mathscr{A}$ cannot generate any group key $SK = H_1(ID\|Z\|K)$, where $K = (a_n + d_n)\{(a_1 + d_1) + (a_2 + d_2) + \cdots + (a_{n-1} + d_{n-1})\}P$, since SK depends on both the short-term secrets a_i $(1{\leq}i{\leq}n)$ as well as the private keys d_i $(1{\leq}i{\leq}n)$. With knowing d_i and $(T_i, P_i) = (a_iP, d_iP)$ for $(1{\leq}i{\leq}n)$, $\mathscr{A}$ can compute $d_n(a_i + d_i)P$ but not $a_n(a_i + d_i)P$ without the knowledge of a_n. In addition, $\mathscr{A}$ cannot derive the random number a_n from $T_n = a_nP$, due to ECDLP. Further, $\mathscr{A}$ can try to derive $(a_n + d_n)(a_i + d_i)P$ $(1{\leq}i{\leq}n-1)$ directly from $(T_n + P_n, T_i + P_i) = \{(a_n + d_n)P, (a_i + d_i)P\}$ $(1{\leq}i{\leq}n-1)$

but, it is also not possible due to difficulties of CDHP. Hence, the proposed ID-AGKA protocol achieves perfect forward secrecy property. □

In Table 5.2, we compare the security attributes of the proposed protocol and other existing protocols including Lee et al. [19], Nam et al. [17], Tseng [91], Cheng et al. [20] and Tsai [21]. Note that, Cheng et al.'s protocol and the proposed protocol provides the resilience against all known attacks, whereas others do not.

Table 5.2: Security comparison of the protocols

Protocol	Lee et al. [19]	Nam et al. [17]	Tseng [91]	Cheng et al. [20]	Tsai [21]	Proposed
KSSTIAR	No	No	No	Yes	No	Yes
KRAR	No	No	No	Yes	Yes	Yes
IAR	No	No	Yes	Yes	Yes	Yes
KCIAR	Yes	Yes	Yes	Yes	No	Yes
KGKAR	Yes	Yes	Yes	Yes	No	Yes
PFS	Yes	Yes	Yes	Yes	No	Yes
ACGKA	Yes	No	Yes	Yes	No	Yes
Cryptosystem	PKI/ECC	PKI	PKI	PKI/ECC	PKI/ECC	IBC/ECC
Assumptions	ECDLP/ BCDHP	DDH	DLP	ECDLP/ CDHP	OWHF/ ECDLP/ BCDHP	OWHF/ ECDLP/ CDHP

5.4 Efficiency Analysis

In this section, the efficiency of the proposed protocol is analyzed with respect to communication costs, computation costs and the comparisons of other relevant protocols and our protocol is offered.

5.4.1 Communication Efficiency

In this section, we compare the efficiency of the proposed protocol and other existing protocols [17, 19, 20, 21, 91] in terms of communication loads. We show that the proposed protocol is well suited for the imbalanced mobile networks.We consider the communication efficiency in terms of number of rounds, message size sent by each low-power node, message size sent by the powerful node and the contributory property. In our protocol, the size of the message sent by each low-power node U_i ($1 \leq i \leq n-1$) is $|ID|+3|G_q|$ and the size of the message sent by the powerful node U_n is $|ID|+(n+3)|G_q|$ where as Lee et al. [19], Cheng et al. [20] and Tsai [21] need $|ID|+2|G_q|$ and $|ID|+n|G_q|$, respectively. The communication efficiency comparison of the proposed protocol and other protocols are given in Table 5.3.

Table 5.3: Comparison in terms of communication load of the protocols

Protocol	Lee et al. [19]	Nam et al. [17]	Tseng [91]	Cheng et al. [20]	Tsai [21]	Proposed
NR	2	2	2	2	2	2
MSLNP	$\|ID\|+2\|G_q\|$	$\|ID\|+\|q\|$	$\|ID\|+\|q\|$	$\|ID\|+2\|G_q\|$	$\|ID\|+2\|G_q\|$	$\|ID\|+3\|G_q\|$
MSPN	$\|ID\|+n\|G_q\|$	$\|ID\|+n\|q\|$	$\|ID\|+(n-1)\|q\|$	$\|ID\|+n\|G_q\|$	$\|ID\|+n\|G_q\|$	$\|ID\|+(n+3)\|G_q\|$
ACGKA	Yes	No	Yes	Yes	Yes	Yes
R/J	No	No	No	No	No	Yes

ACGKA: Authenticated contributory group key agreement;
MSPN: Message size sent by the powerful node; **NR:** Number of rounds;
$|$ **T** $|$**:** Bit length of a message T; **R/J:** Removal/Joining phase

5.4.2 Computation Efficiency

In this section, we examined the computation costs required by each low-power node U_i ($1 \leq i \leq n-1$) and the powerful node U_n of the proposed protocol and other relevant

protocols [17, 19, 20, 21, 91]. In Table 4.2 (Chapter 4), we define various computational complexities and their conversions in terms of T_{ML}. Let us consider the computational cost of U_n which has stronger computing capability with fewer restrictions. In Step 2 of the proposed protocol, U_n verifies the signatures S_i $(1 \leq i \leq n-1)$ by performing $S_i[T_i H_1(ID_i \| T_i) + P_i] = [T_i + P_i], (1 \leq i \leq n-1)$. If the verifications are correct, U_n computes $T_n = a_n P$, $T = \sum_{i=1}^{n}(T_i + P_i)$, $\overline{Z} = (a_n + d_n)T$, $Z_i = (a_n + d_n)^2(T_i + P_i)$, $(1 \leq i \leq n-1)$ and $K = (a_n + d_n)(T - T_n - P_n)$. Finally, U_n computes the session key SK. Thus, the computation required by the powerful node is $(3n-1)T_{EM} + (4n-3)T_{EA} \approx (87.48n - 29.36)T_{ML}$.

Now, we consider the computational complexity of each low-power mobile node U_i $(1 \leq i \leq n-1)$. In general, the mobile devices have limited resource and low computing ability. In Step 1 of the proposed protocol, node U_i $(1 \leq i \leq n-1)$ selects a number $a_i \in_R Z_q^*$ and then computes $T_i = a_i P$. In Step 3, each U_i checks U_n's signature S_n by performing $S_n[T_n(H_1(ID_n \| \overline{Z} \| Z \| T_n) + P_n] = [T_n + P_n]$. If it holds, each U_i computes $K_i = \overline{Z} - (a_i + d_i)^{-1} Z_i$ and the session key as $SK = H_1(ID \| Z \| K)$. It requires $4T_{EM} + 4T_{EA} + T_{IN} \approx 128.08 T_{ML}$. Therefore, the overall computation costs of the proposed protocol is $(87.48n + 88.72)T_{ML} \approx 87(n+1)T_{ML}$. We summarized the computation efficiency of the proposed protocol and other relevant protocols [17, 19, 20, 21, 91] in Table 5.4.

From the efficiency analysis, it is seen that the proposed protocol reduces the computation cost enormously; however, it slightly increases the transmission overheads in comparison with the other methods. Actually this overhead provides additional security features in the proposed protocol and helps in removing the bilinear pairing and MTP hash function for the sufficient reduction of computation cost. In summary, the proposed protocol has the following main features: (1) Avoidance of implementation difficulties for bilinear pairings and MTP hash function, (2) Resilient for all known attacks, (3) Flexible for member joining/leaving operations, (4) High computation efficiency, and (5) Easy realizable in imbalanced wireless mobile networks.

Table 5.4: Comparison in terms of computation cost of the protocols

Protocol	CCLPN	CCPN	TCC
Lee et al. [19]	$3T_{BP} + 3T_{EM} + T_{IN} + T_{MTP} + (n-1)T_{EA} = (388.48 + 0.12n)T_{ML}$	$(2n-1)T_{BP} + 3T_{EM} + T_{MTP} + (n-1)T_{EA} = (174.12n + 28.88)T_{ML}$	$(174.24n + 417.36)T_{ML} \approx 174(n+2)T_{ML}$
Nam et al. [17]	$T_{EX} + T_{ML} = 241T_{ML}$	$(n+1)T_{EX} + nT_{IN} + (2n-2)T_{ML} = (233.6n + 238)T_{ML}$	$(233.6n+479)T_{ML} \approx 233(n+2)T_{ML}$
Tseng [91]	$T_{EX} + (n+1)T_{ML} = (n + 241)T_{ML}$	$(2n+1)T_{EX} + nT_{ML} = (281n + 240)T_{ML}$	$(482n + 481)T_{ML} \approx 482(n+2)T_{ML}$
Cheng et al. [20]	$2T_{BP} + 3T_{EM} + T_{IN} + 3T_{MTP} + (n-1)T_{EA} = (359.48 + 0.12n)T_{ML}$	$(2n-1)T_{BP} + T_{EM} + nT_{MTP} + (n-1)T_{EA} = (203.12n - 58.12)T_{ML}$	$(203.24n + 301.36)T_{ML} \approx 203(n+1)T_{ML}$
Tsai [21]	$T_{BP}+3T_{EM}+2T_{IN}+3T_{MTP}+T_{EA} = 284.32T_{ML}$	$(n-1)T_{BP} + (2n-1)T_{EM} + T_{IN} + (n-1)T_{EA} = (145.12n - 104.52)T_{ML}$	$(145.12n - 179.08)T_{ML} \approx 145(n-1)T_{ML}$
Proposed	$4T_{EM} + 4T_{EA} + T_{IN} \approx 128.08T_{ML}$	$(3n-1)T_{EM} + (4n-3)T_{EA} \approx (87.48n - 29.36)T_{ML}$	$(87.48n+88.72)T_{ML} \approx 87(n+1)T_{ML}$
CCLPN: Computation cost of each low-power node; **CCPN:** Computation cost of powerful node; **TCC:** Total computation cost			

5.5 Chapter Summary

In this chapter, an ID-AGKA protocol using ECC with bilinear pairing-free operation is proposed. It, instead of probabilistic MTP hash function as used in earlier pairing-based technique, uses a general cryptographic hash function. As a result, our protocol, although it is IBC- and ECC-based, eliminates the public key certificates with enhanced security based on ECDLP, CDHP and OWHF, and reduced computation cost. In addition, the proposed protocol is a contributory group key agreement protocol in which the common secret session key is derived as a function of the contributions provided by all mobile nodes. The proposed protocol, which is secure, efficient, and contributory-based, is suitable in many insecure imbalanced mobile network applications such as internet stock quotes, audio and music delivery, pay-per-view TV, etc.

Two secure authenticated session key agreement protocols for two-party and multi-party (group) are presented in Chapters 4 and 5, respectively for confidential and efficient communication over insecure networks. In the next two chapters, i.e., Chapters 6 and 7, we present ECC-based two digital signature schemes using CL-PKC and IBC for message integrity, authentication, non-repudiation and designing online e-cash system, respectively.