Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7519688, 16 pages
https://doi.org/10.1155/2022/7519688

Research Article

Certificateless Cross-Domain Group Authentication Key Agreement Scheme Based on ECC

Liling Cao, Mei Liang, Zheng Zhang, and Shouqi Cao

Department of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China

Correspondence should be addressed to Shouqi Cao; 2212157493@qq.com

Received 10 February 2022; Revised 14 October 2022; Accepted 15 November 2022; Published 5 December 2022

Academic Editor: Jian Su

Focusing on the problem that existing traditional cross-domain group authentication schemes have a high complexity, a certificateless cross-domain group authentication key agreement scheme based on ECC is proposed. The protocol provides scalability and can meet the requirements of cross-domain key negotiation by multiple participants in different domains. Security analysis shows that the proposed scheme is secure in the random oracle security model, it can resist some attacks under the extended Canetti-Krawczyk (eCK) security model. Performance analysis shows that the proposed scheme is of strong practical application value with high efficiency; it costs relatively low amount of calculation and communication.

1. Introduction

The development of technologies such as wireless networks, multicast, and distributed computing has brought new demands for group-oriented network applications, such as multiparty video conferencing, remote video teaching, and online games. As a typical application scenario in the above applications, cross-domain group communication can realize information exchange and transmission between remote cross-domain group members and will provide users with richer services and maximize the use of resources. However, the increase in the size of group members and the heterogeneous network access caused by cross-domains have also brought new security challenges to the design of user identity authentication systems. A secure cross-domain group authentication key agreement protocol will establish a shared key for remote cross-domain group members, establish a secure cross-domain communication channel, ensure the confidentiality and integrity of cross-domain group communication data, and effectively prevents attackers from stealing, tampering, and forging communication data [1–4].

The design of the existing cross-domain group authentication key agreement protocol mainly relies on three types of cryptosystems: public key infrastructure, identity-based cryptosystems, and certificateless public key cryptosystems.

Public key infrastructure (PKI) is an important practical cryptographic technology to ensure network security. In PKI-based identity authentication, the certificate service system binds the digital certificate to the public key of the communicating entity, and the communicating entities verify the authenticity of the certificate to perform identity authentication. Wf et al. [5] adopted elliptic curve cryptosystem based on threshold scheme to construct enterprise cross-domain authentication system with the help of virtual bridge CA model. However, due to the high interaction cost caused by the threshold scheme by splitting key factors, the expansibility of joining and withdrawing members is not strong. Bin [6] improved the existing certificate revocation mechanism, but the certificate verification needs to detect from the book to be verified to the root certificate, so that the verification path is too long and the path verification efficiency is low, which greatly affects the application scope of the cross-domain identity authentication technology. Basin et al. proposed a new PKI architecture, ARPKI [7], which was designed by using formal models to ensure the transparency and reliability of certification-related operations (such as certificate issuance, update, revocation, and verification) and effectively deal with security events such as key loss or disclosure. Zhicheng et al. improved the traditional PKI-based cross-domain authentication by using the alliance chain technology [8] and used the alliance chain to manage the license domain, which simplified the authentication signature steps between entities and had high scalability. Dong et al. proposed a cross-domain authentication scheme that can enhance trust between hospitals [9], which solves the problem of fragmentation and isolation caused by traditional hospitals maintaining their own PKI-based information systems and provides better privacy protection services while realizing the sharing of medical data. Chen et al. used blockchain to replace part of CA functions [10]; the scheme has sufficient scalability and can meet the trust transmission requirements of multiple PKI systems. As the number of legitimate users increases, the calculation and communication overhead of the certificate maintenance process increases. Cross-domain identity authentication has problems such as long trust paths, low certificate verification efficiency, and complex interdomain trust path construction. It is not suitable for cross-domain identity authentication of large-scale group members.

In identity-based cryptosystems, there is no need for PKI to verify users' public keys and identities. Private key generator (PKG) is trusted to generate private keys for users, which can effectively solve the problems such as the overhead of public key certificate management [11]. Changyuan et al. [12] proposed an identity-based signature algorithm to achieve cross-domain authentication by using elliptic curve additive group, avoiding complex bilinear pairings. However, the scheme only analyzes the certification process between the entity and the certification authority and does not consider the extra cost of the certification authority and local resources to verify each other's legitimacy. The identity-based authenticated key agreement protocol without bilinear pairs proposed by Farash and Attari [13] sets multiple independent PKG, and each PKG sets the private key for the user under its jurisdiction. However, this protocol cannot resist temporary key leakage attack and impersonation attack, and it cannot realize implicit key authentication and key confirmation. Cao et al. [14] proposed a key agreement protocol for identity-based authentication with hierarchical PKG, which uses bilinear pair operation in its design, but the operation is not efficient and it is difficult to resist basic impersonation attacks. Kefei et al. [15] proposed an improved scheme on the basis of the literature [14], but this scheme is difficult to resist the temporary key disclosure attacks, and the operation efficiency was low due to the use of bilinear pair operation in the scheme. Since the user's private key is completely determined by the key generation center, PKG can decrypt any user's information and forge any user's signature. There are key escrowing problems, and user identity multiplicity occurs when cross-domain, making cross-domain identity authentication extremely complicated.

Based on the certificateless public key cryptosystem, the user's private key is composed of two parts, that is, the partial private key provided by KGC and the secret value selected by the user. Neither KGC nor the user can generate a complete private key independently, which solves the key escrow problem in the identity-based cryptosystem. Literature [16, 17] proposed a cloud user identity authentication scheme based on certificateless password system, but there was no relevant research on cross-domain authentication. Li et al. [18] proposed a cross-domain authentication key exchange protocol in a wireless grid environment, but the use of too much symmetric encryption causes a large amount of computational overhead. In 2015, Sun et al. [19] proposed a certificateless scheme, but it involves the calculation of linear pairs, so the amount of calculation is huge. In 2016, Cheng et al. [20] proposed a one-round certificateless authenticated group key agreement protocol for mobile ad hoc networks, but Luo et al. [21] indicated that their protocol could not achieve user anonymity that was an important aspect of user privacy protection and could not resist known temporary key attack. In 2018, Yang et al. [22] proposed a cross-domain certificateless key agreement protocol for electronic health systems. Although it achieved the security guarantees of dynamic user management, authentication, and session keys, it did not achieve the real cross-domain and could not resist known temporary key attack indicated by Luo et al. [21]. In 2018, Semal et al. [23] proposed a certificateless group authenticated key agreement protocol for secure communication in untrusted UAV networks, and in 2020, Luo et al. [21] proposed a cross-domain certificateless authenticated group key agreement protocol for 5G network slicings. However, in 2022, Ren et al. [24] indicated that the protocol proposed by Semal et al. could not resist public key replacement attack and protocol proposed by Luo only had a secondary security level; the malicious KGC could collude with some malicious users to attack the protocol.

To sum up, the current cross-domain group communication solutions are mainly focused on the security issues of users' cross-domain communication between two domains. Such solutions not only cannot meet the security requirements of users' cross-domain group communication, but also the communication process is too cumbersome and requires relatively high communication capabilities and computing capabilities, so it cannot provide good guarantees for the security of cross-domain group communication. Therefore, an efficient and scalable key agreement scheme for cross-domain group authentication is urgently needed to solve and realize the communication security problem of efficient cross-domain group authentication. Focusing on the characteristics and requirements of the existing cross-domain group communication, this paper proposes a certificateless cross-domain group key agreement scheme based on ECC.

2. Preliminaries

2.1. Notation. The symbols and their meanings involved in the cross-domain group key negotiation scheme are shown in Table 1.

2.2. Security Model. In our proposed protocol, a novel eCK security model presented by Lippold et al. [25] is adopted. In the subsections, detailed descriptions about the security model including the adversary model, attack game, and security definition are explained.

Table 1: Notation used in this paper.

Notation	Description
U	A collection of all cross-domain participants
Di	The ith domain
RA	The registry
Ppub	Public key of the system
KGCi	Key generation center for domain Di
SK$i = (ki, \mu i)$	The private key of KGCi
PK$i = P$pub$i$	The public key of KGCi
uj^i	The jth participant of the ith domainDi
IDj^i	The identity of uj^i
sk$j^i = (sj^i, xj^i)$	The private key of the user uj^i
pk$j^i = Pj^i$	The public key of the user uj^i
sk$i$	Intragroup key negotiated by the user of the ith domain Di
SK	Cross-domain group key
$\perp$	Represents no message or an unknown value

2.2.1. Adversary Model. There are two types of adversaries. As a dishonest participant, adversary $\mathscr{A}_1$ has no idea of the master key of KGC but has the capability to replace the public key of any participant, which means $\mathscr{A}_1$ can replace the secret value x_i of the participant with a value of his choice. As a malicious KGC, adversary $\mathscr{A}_2$ cannot replace the public key of any participant but can obtain the master key of KGC, which means s_i is easily accessible for $\mathscr{A}_2$.

The security model is defined by an attack game between a challenger $\complement$ and an adversary $\mathscr{A} \in \{\mathscr{A}_1, \mathscr{A}_2\}$. The adversary asks the challenger for a polynomial number of queries, while the challenger issues the replies using simulators and random oracles owned by him. Let $\Pi_{i,j}^m$ represent a simulator of the challenger, which simulates the behavior of participant i in the mth session with intended participant j. Then, simulated as participant i, the simulator $\Pi_{i,j}^m$ executes all the steps that participant i should do in the protocol session. That is, the simulator $\Pi_{i,j}^m$ takes private key of participant i and messages transmitted from participant j (or pseudo-j personated by the adversary) as inputs and sends the corresponding outputs to the adversary.

Definition 1 (accepted session). The session $\Pi_{i,j}^m$ is accepted when it can generate the session key skij^m.

Definition 2 (session identity). The session identity (sID) is denoted as the concatenation of the participants' identities and messages in the session. For instance, sID$\prod_{i,j}^m = \{$IDi, ID$j, M1, M2\}$, where $M1$ is the message transmitted from $\Pi_{i,j}^m$ and $M2$ is the message received by $\Pi_{i,j}^m$

Definition 3 (matched session). Honest participants i and j are involved in the protocol session. The session $\Pi_{i,j}^m$ matches $\Pi_{j,i}^n$ when they have the same session identity.

2.2.2. Attack Game. Steps of the attack game are described as follows:

(1) The challenger $\complement$ executes the SETUP algorithm:

$$\{s, \text{system params}\} \overset{\text{challenger}}{\longleftarrow} \text{SETUP}(k) \tag{1}$$

For adversary $\mathscr{A}_1$, the challenger $\complement$ sends system params to $\mathscr{A}_1$ but keeps s in secret. For adversary $\mathscr{A}_2$, the challenger $\complement$ sends system params and s to $\mathscr{A}_2$

(2) The adversary asks the challenger $\complement$ for a polynomial number of the following queries:

  (i) Create (ID$_i$): $\complement$ generates private key/public key pair (sk$_i$, pk$_i$) of participant i

  (ii) Rs_i: $\complement$ reveals to $\mathscr{A}$ the partial private key s_i of participant i

  (iii) Rx_i: $\complement$ reveals to $\mathscr{A}$ the secret value x_i of participant i

  (iv) R$t_i(\prod_{i,j}^m)$: $\complement$ reveals to $\mathscr{A}$ the ephemeral private key t_i of participant i in the session $\Pi_{i,j}^m$

  (v) Rs: $\complement$ reveals to $\mathscr{A}$ the master key of KGC. Then, $\mathscr{A}$ can obtain the partial private keys of all participants

  (vi) Rpk$_i$: $\complement$ replaces the public key of participant i with the value chosen by $\mathscr{A}$, which means that the secret values of all participants can be set by $\mathscr{A}$

  (vii) Rsk$_{ij}(\prod_{i,j}^m)$: $\complement$ reveals to $\mathscr{A}$ the accepted session key sk$_{ij}$ if $\Pi_{i,j}^m$ is accepted. Otherwise, $\complement$ returns $\perp$to A

  (viii) Send $(\Pi_{i,j}^m, M)$: $\mathscr{A}$ (pseudoparticipant j) sends the message M to the session $\Pi_{i,j}^m$ (simulated participant i) and gets a reply according to the protocol. If $M = \perp$, simulated participant i of the session $\Pi_{i,j}^m$ is an initiator. Otherwise, it is a responder

(3) When deciding to end aforementioned queries, $\mathscr{A}$ chooses a fresh session (defined later) $\Pi_{i,j}^m$ and asks a test $(\Pi_{i,j}^m)$ query. By tossing a fair coin with $b \in \{0, 1\}$, $\complement$ replies the session key held by $\Pi_{i,j}^m$ if $b = 1$, or a random string if $b = 0$

(4) The adversary asks the challenger $\complement$ for a polynomial number of the above queries about fresh session $\Pi_{i,j}^m$

(5) When terminating the game, $\mathscr{A}$ makes a guess bit b$'$. If b$'$ = b, $\mathscr{A}$ wins the game. The advantage of $\mathscr{A}$ for winning the game is defined as $\text{Adv}_A(k) = |\text{pr}[b = b'] - (1/2)|$

Definition 4 (fresh session against $\mathscr{A}_1$). The accepted session $\Pi^m_{i,j}$ is fresh if none of the following condition holds:

(i) $\mathscr{A}_1$ raises the query $\mathrm{Rsk}_{ij}(\Pi^m_{i,j})$ or $Rsk_{ji}(\Pi^n_{j,i})$ (if the matched session $\Pi^n_{j,i}$ of $\Pi^m_{i,j}$ exists)

(ii) Matching: if honest participant j is engaged in $\Pi^n_{j,i}$ that matches $\Pi^m_{i,j}$, $\mathscr{A}_1$ either inquires both $\mathrm{R}s_i$ (or $\mathrm{R}s$) and $\mathrm{R}t_i(\Pi^m_{i,j})$ or both $\mathrm{R}s_j$ (or $\mathrm{R}s$) and $\mathrm{R}t_j(\Pi^n_{j,i})$

(iii) Not matching: if there is no session matched to $\Pi^m_{i,j}$, $\mathscr{A}_1$ either inquires both $\mathrm{R}s_i$ and $\mathrm{R}t_i(\Pi^m_{i,j})$ or $\mathrm{R}s_j$ (or $\mathrm{R}s$)

Definition 5 (fresh session against $\mathscr{A}_2$).

(i) $\mathscr{A}_2$ raises the query $\mathrm{Rsk}_{ij}(\prod^m_{i,j})$ or $\mathrm{Rsk}_{ji}(\prod^n_{i,j})$ (if the matched session $\Pi^n_{j,i}$ of $\Pi^m_{i,j}$ exists)

(ii) Matching: if honest participant j is engaged in $\Pi^n_{j,i}$ that matches $\Pi^m_{i,j}$, $\mathscr{A}_2$ either inquires both $\mathrm{R}x_i$ (or Rpk_i) and $\mathrm{R}t_i(\prod^m_{i,j})$ or both $\mathrm{R}x_j$ (or Rpk_i) and $\mathrm{R}t_j(\prod^n_{i,j})$

(iii) Not matching: if there is no session matched to $\Pi^m_{i,j}$, $\mathscr{A}_2$ either inquires both $\mathrm{R}x_i$ and $\mathrm{R}t_i(\prod^m_{i,j})$ or $\mathrm{R}x_j$ (or Rpk_i)

According to the adversary model stated in Section 2.2.1, x_i and s_i are deemed to be knowable for $\mathscr{A}_1$ and $\mathscr{A}_2$ by Rpk_i query and $\mathrm{R}s$ query, respectively. Supposing that participant A and B want to establish a session key, the session is not fresh in seven cases for $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively, which is shown in Table 2. Then, the fresh session can appear in seven cases for $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively, as shown in Table 3. As a case of fresh sessions, type FI1$'$ can be regarded as FI1 and type FII1$'$ can be regarded as FII1.

2.2.3. Security Definition. A certificateless cross-domain group key agreement protocol is deemed authenticated key agreement (AKA) secure if no adversary except the protocol participants can get the session key. Detailed and accurate definition is as follows.

Definition 6. A protocol is secure when the following set of conditions is assumed:

(1) In the presence of an adversary $\mathscr{A} \in \{\mathscr{A}_1, \mathscr{A}_2\}$, sessions $\Pi^m_{i,j}$ and $\Pi^n_{j,i}$ always agree on the same session key that distributed uniformly at random

(2) For any adversary $\mathscr{A} \in \{\mathscr{A}_1, \mathscr{A}_2\}$, $\mathrm{Adv}_A(k)$ is negligible

3. Key Agreement Scheme for Cross-Domain Group Authentication

Figure 1 shows the member structure of cross-domain group authentication in the scheme. Suppose there are n domains, each with m members participating in cross-domain authentication, where $uj^i(1 \le i \le n, 1 \le j \le m)$ represents any participant and let $U = \{uj^i\}$ represent the collection of remote participants scattered across domains $Di(1 \le i \le n)$ that intend to negotiate a shared session key. Let the symbol $U^i = \{u1^i, u2^i, \cdots, um^i\}$ represent the set of all participants in the same service domain Di. The protocol consists of two phases: intradomain key negotiation and interdomain key negotiation.

3.1. System Initialization. This algorithm takes a security parameter $k \in Zn^*$ as input, and the RA performs the following steps to generate a master secret key s and a set of public parameter param.

(i) RA generates a group with prime order p and determines a point P of order p as a generator of G

(ii) RA chooses a master secret key $s \in Zn^*$ and computes the corresponding public key $P\mathrm{pub} = sP$

(iii) RA chooses seven cryptographic secure hash functions: $H1(*), H3(*), H4(*), H5(*), H6(*) \in Zn^*$, $H2(*) \in \{0,1\}^k$, and $H7 : \{0,1\}* \times \{0,1\}* \times \cdots \times \{0,1\}* \times Zn \times Zn \longrightarrow Zn^*$

(iv) RA publishes the system generated parameters $\{Fp, E(Fp), G, P, P\mathrm{pub}, n, H1, H2, H3, H4, H5, H6, H7\}$ while keeping the master key (s) secret

3.2. Registration Phase. The registration phase consists of KGCi registration and user registration.

3.2.1. KGCi registration. When KGCi applies to RA as the key generation center of domain Di, it needs to register with RA. The algorithm is as follows:

(1) Set secret value: KGCi randomly chooses $\mu i \in Zq^*$, calculates $P\mathrm{pub}i = \mu i \cdot P$, and sets μi as a secret value

(2) Extract partial private key: key generation center KGCi sends identity information IDi to RA, and RA picks a random number $\lambda i \in Zn^*$ for KGCi; calculates $Wi = \lambda i \cdot P$, $li = H3(\mathrm{ID}i, Wi, P\mathrm{pub}i)$, and $ki = \lambda i + li \cdot s \bmod n$; and issues $\{ki, Wi\}$ to KGCi by secure channel

(3) Set private key: KGCi sets SK$i = (ki, \mu i)$ as his private key

(4) Set public key: KGCi sets PK$i = P\mathrm{pub}i = \mu i \cdot P$ as his public key

3.2.2. User Registration. When user uj^i applies to KGCi for joining the domain as the jth member, the registration needs to be completed at KGCi. The algorithm is as follows:

Table 2: Cases of not fresh session.

Condition		Type	Queries		Known	
Type I adversary A_1	Matching	AI1	RsA (or Rs)	RtA	xA	xB
		AI2	RsB (or Rs)	RtB	xA	xB
		AI3	$RskA, B\left(\prod_{A,B}^{m}\right)$			
		AI4	$RskB, A\left(\prod_{B,A}^{n}\right)$			
	Not matching	AI5	RsA	RtA	xA	xB
		AI6	RsB (or Rs)		xA	xB
		AI7	$RskA, B\left(\prod_{A,B}^{m}\right)$			
Type II adversary A_2	Matching	AI1	RxA (or $Rpki$)	RtA	sA	sB
		AI2	RxB (or $Rpki$)	RtB	sA	sB
		AI3	$RskA, B\left(\prod_{A,B}^{m}\right)$			
		AI4	$RskB, A\left(\prod_{B,A}^{n}\right)$			
	Not matching	AI5	RxA	RtA	sA	sB
		AI6	RxB (or $Rpki$)		sA	sB
		AI7	$RskA, B\left(\prod_{A,B}^{m}\right)$			

Table 3: Cases of fresh session.

Condition	Type I adversary A_1 Type	Queries		Known		Type II adversary A_2 Type	Known		Queries	
Matching	FI1	RsA	RsB	xA	xB	FII1	sA	sB	RxA	RxB
	FI2	RtA	RtB	xA	xB	FII2	sA	sB	RtA	RtB
	FI3	RsA	RtB	xA	xB	FII3	sA	sB	RxA	RtB
	FI4	RtA	RsB	xA	xB	FII4	sA	sB	RtA	RxB
	FI1’	Rs		xA	xB	FII1’	sA	sB	$Rpki$	
Not matching	FI5	RsA		xA	xB	FII5	sA	sB	RxA	
	FI6	RtA		xA	xB	FII6	sA	sB	RtA	

(1) Set secret value: uj^i randomly chooses $xj \in Zn^*$, calculates $Pj^i = xj^i \cdot P$, and sets xj as a secret value

(2) Extract partial private key: user uj^i sends identity information $\mathrm{ID}j^i$ to KGCi, and KGCi picks a random number $rj \in Zn^*$ for uj^i; calculates $Rj = rj \cdot P$, $hj = H3(\mathrm{ID}j^i, Rj, Pj^i)$, and $sj^i = rj + hj \cdot \mu i \bmod n$; and issues $\{sj^i, Rj\}$ to uj^i by secure channel

(3) Set private key: uj^i sets $\mathrm{sk}j^i = (sj^i, xj^i)$ as his private key

(4) Set public key: uj^i sets $\mathrm{pk}j^i = Pj^i = xj^i \cdot P$ as his public key

3.3. Intradomain Key Negotiation. As shown in Figure 2, in the key negotiation phases of intradomain, all participants in set U negotiate the session key of the intradomain group (no negotiation is required if there is only one user in the domain). Pick a participant randomly from all participants in domain Di as the domain head Hi. If user um^i is selected as domain header Hi, users of domain Di are classified into common user $uj^i (1 \le j \le m-1)$ and domain header um^i.

(1) By exchanging messages with domain header um^i, each common user $uj^i (1 \le j \le m-1)$ proves that they are legitimate members of the same domain D i

Step 1: uj^i randomly chooses $tj, aj \in Zn^*$ and calculates $Tj = tjP$ and then sends message $M1 = \{\mathrm{ID}j^i, Tj, Rj\}$ to um^i.

Step2: um^i randomly chooses $tm, am \in Zn^*$ and calculates $Tm = tmP$ and then sends message $M2 = \{\mathrm{ID}m^i, Tm, Rm\}$ to uj^i.

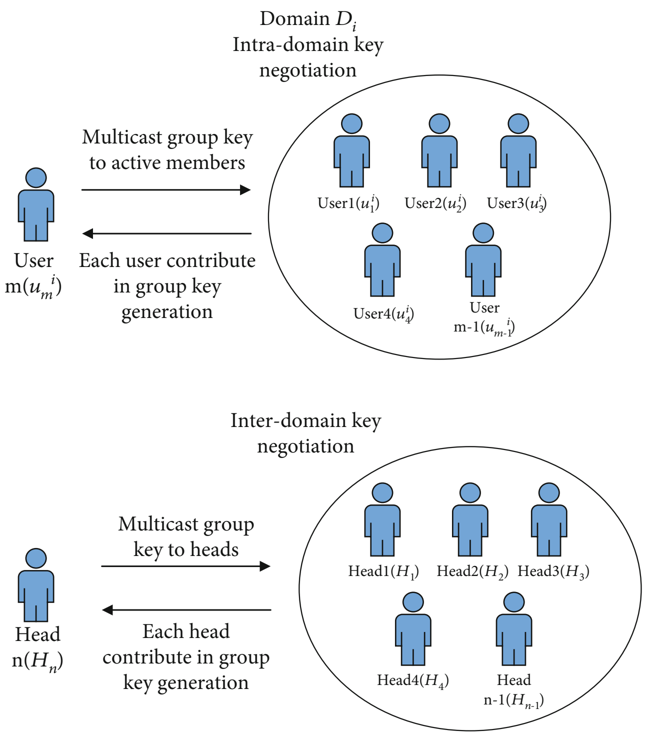

Figure 1: The proposed group member structure of the cross-domain group authentication key agreement scheme.

Step 3: uj^i calculates $kjm = (H3(\mathrm{ID}j^i, Tj, Pj)tj + sj + xj) \cdot (H3(\mathrm{ID}m^i, Tm, Pm)Tm + (Rm + H3(\mathrm{ID}m^i, Tm, Pm)P\mathrm{pub}i + Pm)$ and $kbj = aj \cdot P$ and then sends the signature message $\sigma j = (kbj, kjm)$ to um^i.

Step 4: um^i calculates $kmj = (H3(\mathrm{ID}m^i, Tm, Pm)tm + sm + xm) \cdot (H3(\mathrm{ID}j^i, Tj, Pj)Tj + (Rj + H3(\mathrm{ID}j^i, Tj, Pj)P\mathrm{pub}i + Pj)$. If $kmj = kjm$, uj^i is a legitimate member of domain Di.

(2) Domain header um^i calculates the session key ski of domain Di based on the information received from the negotiation members and sends it to the negotiation members in encrypted form. The legitimate members who in the domain can calculate the session key

Step 1: At first, um^i calculates $kbm = am \cdot P$, $\mathrm{sk}jm = H2(\mathrm{ID}j^i \| \mathrm{ID}m^i \| Tj \| Tm \| kjm)$, $qm = H4(xm^i \| sm^i)$, $bmj = H1(\mathrm{ID}j^i \| \mathrm{ID}m^i \| am \cdot kbj) \oplus qm$, $dm = xm + sm \cdot H1(\mathrm{ID}j^i \| \mathrm{ID}m^i \| qm \cdot P)$, $zam = am + dm \cdot bmj$, $hm = H5(kbm, qm)$, $fm = H6(sk1m + sk2m + \cdots + sk(m-1)m + amxmP)$, and $\mathrm{key}j = H5(am \cdot kbj, qm) \oplus fm$, then calculates the session key $\mathrm{sk}i = H7(\mathrm{ID}1^i \| \mathrm{ID}2^i \| \cdots \| \mathrm{ID}m^i \| fm \| hm)$ of domain Di, and finally sends $\{\mathrm{ID}1^i, \mathrm{ID}2^i, \cdots, \mathrm{ID}m^i, \mathrm{key}j, hm, \sigma m\}$ to uj^i.

Step 2: uj^i receives the message $\{\mathrm{ID}1^i, \mathrm{ID}2^i, \cdots, \mathrm{ID}m^i, \mathrm{key}j, hm, \sigma m\}$ sent by um^i and then calculates $q'm = bmj \oplus H1(\mathrm{ID}j^i \| \mathrm{ID}m^i \| aj \cdot kbm)$ and $k'bm = zam \cdot P - bmj[Pm^i + (Rm + H3(\mathrm{ID}m^i, Rm, Pm^i)P\mathrm{pub}i) \cdot H1(\mathrm{ID}j^i \| \mathrm{ID}m^i \| q'm \cdot P)]$. If $H5(k'bm, q'm) = hm$, uj^i calculates $f'm = \mathrm{key}j \oplus H5(k'bm \cdot aj, q'm)$. Finally, the intradomain session key $\mathrm{sk}i = H7(\mathrm{ID}1^i \| \mathrm{ID}2^i \| \cdots \| \mathrm{ID}m^i \| f'm \| hm) = H7(\mathrm{ID}1^i \| \mathrm{ID}2^i \| \cdots \| \mathrm{ID}m^i \| fm \| hm)$ of the domain Di is calculated.

3.4. Interdomain Key Negotiation. As shown in Figure 3, in the key negotiation phase of interdomain, remote participants scattered in each domain $Di (1 \le i \le n)$ who intend to negotiate the shared session key randomly choose user um^i $(1 \le i \le n)$ as domain head Hi in their respective domains, and the key negotiation between domain heads is carried out. Assume that the domain header is divided into $Hi(um^i, 1 \le i \le n-1)$ and $Hn(um^n)$, and the process of interdomain key negotiation is as follows:

(1) To prove that domain heads $Hi (1 \le i \le n-1)$ and domain head Hn are legitimate members of the same system, they should send messages to each other

Step 1: Hi randomly chooses $ti, bi \in Zn^*$ and calculates $Tj = tjP$ and then sends message $M1 = \{\mathrm{ID}j^i, Tj, Rj\}$ to um^i.

Step2: um^i randomly chooses $tm, am \in Zn^*$ and calculates $Tm = tmP$ and then sends message $M2 = \{\mathrm{ID}m^i, Tm, Rm\}$ to uj^i.

Step 3: Hi calculates $Kin = (H3(\mathrm{ID}m^i, Ti, Pm^i)ti + sm^i + xm^i) \cdot (H3(\mathrm{ID}m^n, Tn, Pm^n)Tn + (Rn + H3(\mathrm{ID}m^n, Tn, Pm^n)P\mathrm{pub} + Pm^n)$ and $Kbi = bi \cdot P$ and then sends the signature message $\sigma i = (Kbi, Kin)$ to Hn.

Step 4: Hn calculates $Kni = (H3(\mathrm{ID}m^n, Ti, Pm^n)ti + sm^n + xm^n) \cdot (H3(\mathrm{ID}m^i, Tn, Pm^i)Tn + (Rn + H3(\mathrm{ID}m^i, Tn, Pm^i)Ppub + Pm^i)$. If $Kni = Kin$, $Hi(um^i, 1 \le i \le n-1)$ and $Hn(um^n)$ are legitimate members of the same system.

(2) Domain head Hn calculates the interdomain session key SK for interdomain negotiation based on the information received from domain head member $Hi(um^i, 1 \le i \le n-1)$ that participates in the negotiation in each domain and sends the key SK to other members in the negotiation in encrypted form. Legitimate members who in the system can calculate the interzone session key

Step 1: At first, Hn calculates $Kbn = bn \cdot P$, $\mathrm{SKin} = H2(\mathrm{ID}m^i \| \mathrm{ID}m^n \| Ti \| Tn \| Kin)$, $Qn = H4(xm^n \| sm^n)$, $Bni = H1(\mathrm{ID}m^i \| \mathrm{ID}m^n \| bn \cdot Kbi) \oplus Qn$, $Dn = xm^n + sm^n \cdot H1(\mathrm{ID}m^i \| \mathrm{ID}m^n \| Qn \cdot P)$, $Zan = bn + Dn \cdot Bni$, $Hn = H5(Kbn, Qn)$, $Fn = H6(\mathrm{SK}1n + \mathrm{SK}2n + \cdots + \mathrm{SK}(n-1)n + bnxm^nP)$, and $\mathrm{KEY}i = H5(bn \cdot Kbi, Qn) \oplus Fn$, then calculates the interdomain session key $\mathrm{SK} = H7(\mathrm{ID}m^1 \| \mathrm{ID}m^2 \| \cdots \| \mathrm{ID}m^n \| Fn \| hn)$, and finally sends $\{\mathrm{ID}m^1, \mathrm{ID}m^2, \cdots, \mathrm{ID}m^n, \mathrm{KEY}i, Hn, \sigma n\}$ to Hi.

Step 2: Hi receives the message $\{\mathrm{ID}m^1, \mathrm{ID}m^2, \cdots, \mathrm{ID}m^n, \mathrm{KEY}i, Hn, \sigma n\}$ sent by Hn and then calculates $Q'n = Bni \oplus H1(\mathrm{ID}m^i \| \mathrm{ID}m^n \| bi \cdot Kbn)$ and $K'bn = Zan \cdot P - Bni[Pm^n + (Rn + H3(\mathrm{ID}m^n, Rm, Pm^n)Ppub) \cdot H1(\mathrm{ID}m^i \| \mathrm{ID}m^n \| Q'n \cdot P)]$. If $H5(K'bn, Q'n) = Hn$, Hi calculates $F'n = \mathrm{KEY}i \oplus H5(K'bn \cdot bi, Q'n)$. Finally, the interdomain session key $\mathrm{SK} = H7(\mathrm{ID}m^1 \| \mathrm{ID}m^2 \| \cdots \| \mathrm{ID}m^n \| F'n \| Hn) = H7(\mathrm{ID}m^1 \| \mathrm{ID}m^2 \| \cdots \| \mathrm{ID}m^n \| Fn \| hn)$ is calculated.

$u_j^i (1 \le j \le m-1)$ $\qquad\qquad$ u_m^i

Choose t_j , $a_j \in Z_n^*$

$T_j = t_j P$

$\xrightarrow{M_1 = \{ID_j^i, T_j, R_j\}}$

choose $t_m, a_m \in Z_p^*$

$T_m = t_m P$

$\xleftarrow{M_2 = \{ID_m^i, T_m, R_m\}}$

$$k_{jm} = (H_3(ID_j^i, T_j, P_j^i) t_j + s_j^i + x_j^i) \cdot$$
$$(H_3(ID_m^i, T_m, P_m^i) T_m + (R_m + H_3(ID_m^i, T_m, P_m^i) P_{pubi} + P_m^i)$$

$$k_{bj} = a_j \cdot P$$

$$\sigma_j = (k_{bj}, k_{jm}) \quad \xrightarrow{\{\sigma_j\}}$$

$$k_{mj} = (H_3(ID_m^i, T_m, P_m^i) t_m + s_m^i + x_m^i) \cdot$$
$$(H_3(ID_j^i, T_j, P_j^i) T_j + (R_j + H_3(ID_j^i, T_j, P_j^i) P_{pubi} + P_j^i)$$

if $k_{mj} = k_{jm}$,continue

$$k_{bm} = a_m \cdot P$$

$$sk_{jm} = H_2(ID_j^i \| ID_m^i \| T_j \| T_m \| k_{jm})$$

$$q_m = H_4(x_m^i, s_m^i)$$

$$b_{mj} = H_1(ID_j^i \| ID_m^i \| a_m \cdot k_{bj}) \oplus q_m$$

$$d_m = x_m^i + s_m^i \cdot H_1(ID_j^i \| ID_m^i \| q_m \cdot P)$$

$$z_{am} = a_m + d_m \cdot b_{mj}$$

$$h_m = H_5(k_{bm}, q_m)$$

$$f_m = H_6(sk_{1m} + sk_{2m} + \cdots + sk_{(m-1)m} + a_m x_m^i P)$$

$$sk_i = H_7(ID_1^i \| ID_2^i \| \cdots \ ID_m^i \| f_m \| h_m)$$

$$key_j = H_5(a_m \cdot k_{bj}, q_m) \oplus f_m$$

$$\sigma_m = (k_{bm}, b_{mj}, z_{am})$$

$\xleftarrow{\{ID_1^i, ID_2^i, \cdots, ID_m^i, key_j, h_m, \sigma_m\}}$

$$q'_m = b_{mj} \oplus H_1(ID_j^i \| ID_m^i \| a_j \cdot k_{bm})$$

$$k'_{bm} = z_{am} \cdot P - b_{mj}\left[P_m^i + (R_m + H_3(ID_m^i, R_m, P_m^i) P_{pubi}) \cdot H_1(ID_j^i \| ID_m^i \| q'_m \cdot P)\right]$$

If $H_5(k'_{bm}, q'_m) = h_m$,continue

$$f'_m = key_j \oplus H_5(k'_{bm} \cdot a_j, q'_m)$$

$$sk_i = H_7(ID_1^i \| ID_2^i \| \cdots \| ID_m^i \| f'_m \| h_m)$$

Figure 2: Key negotiation phases of intradomain.

3.5. Join Phase. Assume that user $um+1$ wants to join the protocol, obtains the system parameters of KGCi, generates its own public key $\mathrm{pk}j^i = Pj^i = xj^i \cdot P$ and private key $\mathrm{sk}j^i = (sj^i, xj^i)$, and performs the key negotiation again. The procedure is the same as steps (1) to (2) in intradomain key negotiation.

$H_i(u_m{}^i, 1 \le i \le n-1)$ $\qquad\qquad$ $H_n(u_m{}^n)$

Choose t_i, $b_i \in Z_n{}^*$

$T_i = t_i P$

$\xrightarrow{m1=\{ID_m{}^i, T_i, R_i\}}$

choose $t_n, b_n \in Z_n{}^*$

$T_n = t_n P$

$\xleftarrow{m2=\{ID_m{}^n, T_n, R_n\}}$

$$K_{in} = (H_3(ID_m{}^i, T_i, P_m{}^i)t_i + s_m{}^i + x_m{}^i)\cdot$$
$$(H_3(ID_m{}^n, T_n, P_m{}^n)T_n + (R_n + H_3(ID_m{}^n, T_n, P_m{}^n)P_{pub} + P_m{}^n)$$

$$K_{bi} = b_i \cdot P$$

$$\sigma_i = (K_{bi}, K_{in})$$

$\xrightarrow{\{\sigma_i\}}$

$$K_{ni} = (H_3(ID_m{}^n, T_n, P_m{}^n)t_n + s_m{}^n + x_m{}^n)\cdot$$
$$(H_3(ID_m{}^i, T_i, P_m{}^i)T_i + (R_i + H_3(ID_m{}^i, T_i, P_m{}^i)P_{pub} + P_m{}^i)$$

if $K_{ni} = K_{in}$,continue

$$K_{bn} = b_n \cdot P$$

$$SK_{in} = H_2(ID_m{}^i \,||\, ID_m{}^n \,||\, T_i \,||\, T_n \,||\, K_{in})$$

$$Q_n = H_4(x_m{}^n, s_m{}^n)$$

$$B_{ni} = H_1(ID_m{}^i \,||\, ID_m{}^n \,||\, b_n \cdot K_{bi}) \oplus Q_n$$

$$D_n = x_m{}^n + s_m{}^n \cdot H_1(ID_m{}^i \,||\, ID_m{}^n \,||\, Q_n \cdot P)$$

$$Z_{an} = b_n + D_n \cdot B_{ni}$$

$$H_n = H_5(K_{bn}, Q_n)$$

$$F_n = H_6(SK_{1n} + SK_{2n} + \cdots + SK_{(n-1)n} + b_n x_m{}^n P)$$

$$SK = H_7(ID_m{}^1 \,||\, ID_m{}^2 \,||\cdots||\, ID_m{}^n \,||\, F_n \,||\, h_n)$$

$$KEY_i = H_5(b_n \cdot K_{bi}, Q_n) \oplus F_n$$

$$\sigma_n = (K_{bn}, B_{ni}, Z_{an})$$

$\xleftarrow{\{ID_m{}^1, ID_m{}^2, \cdots, ID_m{}^n, KEY_i, H_n, \sigma_n\}}$

$$Q'_n = B_{ni} \oplus H_1(ID_m{}^i \,||\, ID_m{}^n \,||\, b_i \cdot K_{bn})$$

$$K'_{bn} = Z_{an} \cdot P - B_{ni}\left[P_m{}^n + (R_n + H_3(ID_m{}^n, R_m, P_m{}^n)P_{pub}) \cdot H_1(ID_m{}^i \,||\, ID_m{}^n \,||\, Q'_n \cdot P)\right]$$

If $H_5(K'_{bn}, Q'_n) = H_n$,continue

$$F'_n = KEY_i \oplus H_5(K'_{bn} \cdot b_i, Q'_n)$$

$$SK = H_7(ID_m{}^1 \,||\, ID_m{}^2 \,||\cdots||\, ID_m{}^n \,||\, F'_n \,||\, H_n)$$

Figure 3: Key negotiation phase of interdomain.

3.6. Removal Phase

(1) If the domain head leave, the remaining members will reselect a computationally powerful member as the domain head, and intradomain key negotiation and interdomain key negotiation are restarted. Perform steps (1) to (2) in intradomain key negotiation, and perform steps (1) to (2) in interdomain key negotiation

(2) If it is a common intradomain member that leaves, the intradomain member initiates key negotiation

again. The procedure is the same as steps (1) to (2) in intradomain key negotiation

4. Security Analysis

Take intradomain key negotiation as an example. The intradomain session key is $\mathrm{sk}i = H7(\mathrm{ID}1^i \| \mathrm{ID}2^i \| \cdots \| \mathrm{ID}m^i \| fm \| hm)$, where $fm = H6(\mathrm{sk}1m + \mathrm{sk}2m + \cdots + \mathrm{sk}(m-1)m + amxm^i P)$. Hence, we can prove the security of the protocol by proving the security of $\mathrm{sk}jm (1 \le j \le m-1)$.

For ease of understanding, the negotiation process of $\mathrm{sk}jm$ in the protocol is assumed to be the negotiation process of A and B in domain Di to generate $\mathrm{sk}AB$.

Suppose $H_1(*)$, $H_2(*)$, and $H_3(*)$ are treated as random oracles owned by $\mathfrak{C}$. $\mathrm{RDDH}(aP, bP, cP)$ is a DDH oracle, which outputs 1 if $abP = cP$, otherwise 0. Assume that $\mathscr{A}$ makes at most q_i queries to $H_i (2 \le i \le 3)$, q_c queries to $\mathrm{Create}(\mathrm{ID}_i)$, q_{si} queries to $\mathrm{R}s_i(\mathrm{ID}_i)$, q_x queries to $\mathrm{R}x_i(\mathrm{ID}_i)$, q_t queries to $\mathrm{R}t_i(\Pi^f_{i,j})$, q_{sk} queries to $\mathrm{Rsk}_{ij}(\Pi^f_{i,j})$, q_{sd} queries to $\mathrm{Send}(\Pi^f_{i,j}, M)$, q_s queries to $\mathrm{R}s$, $q_{\mathrm{pk}i}$ queries to $\mathrm{Rpk}_i(x'_i)$, and q_{rddh} queries to $\mathrm{RDDH}(aP, bP, cP)$. Assume also that bounded running time of query H_i $(0 \le i \le 3)$ is t_i, $\mathrm{Create}(\mathrm{ID}_i)$ is t_c, $\mathrm{R}s_i(\mathrm{ID}_i)$ is t_{si}, $\mathrm{R}x_i(\mathrm{ID}_i)$ is t_x, $\mathrm{R}t_i(\Pi^f_{i,j})$ is t_t, $\mathrm{Rsk}_{ij}(\Pi^f_{i,j})$ is t_{sk}, $\mathrm{Send}(\Pi^f_{i,j}, M)$ is t_{sd}, $\mathrm{R}s$ is t_s, $\mathrm{Rpk}_i(x'_i)$ is $t_{\mathrm{pk}i}$, and RDDH is t_{rddh}.

The challenger $\mathfrak{C}$ maintains the query lists as follows:

L_{RH3-1}: a tuple of $(\mathrm{ID}_i, R_i, P_i, h_i)$

L_{RH2}: a tuple of $(\mathrm{ID}_i, \mathrm{ID}_j, (T_i)\Pi^m_{i,j}, (T_j)\Pi^n_{j,i}, K_{ij}, h_{2ij})$

L_{RH3-2}: a tuple of $(\mathrm{ID}_i, T_i, P_i, h_{TPi})$

L_C: a tuple of $(\mathrm{ID}_i, s_i, x_i, P_i, R_i)$

$L_t(\Pi^f_{i,j})$: a tuple of $(\Pi^f_{i,j}, t_i, T_i)$

Given an instance of the GDH problem, for unknown $A, B \in Z^*_n$, by giving $P, AP, BP, P \in E/Fq$ and an oracle DDH, compute $\mathrm{CP} = \mathrm{ABP}$.

Lemma 7. *Suppose $\mathscr{A}_1$ win the game in case FI1 with advantage ε and running time t, with the help of $\mathscr{A}_1$; an algorithm Γ can be constructed to solve the above instance of the GDH problem with advantage ε′ and running time τ by interacting with $\mathscr{A}_1$.*

$$\varepsilon' = \frac{4}{q_c q_{sd} q_t (q_c - 1)(q_t - 1)} \bullet \varepsilon,$$
$$\tau \le \sum_{i=2}^{3} q_i t_i + q_c t_c + q_s t_s + q_t t_t + q_{sk} t_{sk} + q_{sd} t_{sd} + q_s t_s + q_{pki} t_{pki} + q_{rddh} t_{rddh} + t + t_{CP}. \tag{2}$$

Proof. To interact with $\mathscr{A}_1$, a GDH slover Γ simulates as $\mathfrak{C}$ and runs the following steps to solve the above instance of the GDH problem with the help of $\mathscr{A}_1$:

(C1) Γ executes the SETUP algorithm and sends system params to $\mathscr{A}_1$.

(C2) Suppose that $\mathscr{A}_1$ will choose $\Pi^m_{A,B}$ for challenge in the next step. $\mathscr{A}_1$ asks the Γ for a polynomial number of the queries.

$\mathrm{Create}(\mathrm{ID}_i)$: on receiving (ID_i), Γ performs as follows:

(1) If L_C contains a tuple of $(\mathrm{ID}_i, s_i, x_i, P_i, R_i)$, Γ returns all the elements of the tuple to $\mathscr{A}_1$

(2) Otherwise, Γ randomly chooses x_i, s_i, h_i and then computes $P_i = x_i P$ and $R_i = s_i P - h_i P_{\mathrm{pub}}$; Γ inserts $(\mathrm{ID}_i, s_i, x_i, P_i, R_i)$ to L_C and $(\mathrm{ID}_i, R_i, P_i, h_i)$ to L_{RH3-1} and returns $(\mathrm{ID}_i, s_i, x_i, P_i, R_i)$ to $\mathscr{A}_1$

All the following queries should be asked after $\mathrm{Create}(\mathrm{ID}_i)$.

H_{3-1}query: on receiving $(\mathrm{ID}_i, R_i, P_i)$, after query $\mathrm{Create}(\mathrm{ID}_i)$, there must be a tuple of $(\mathrm{ID}_i, R_i, P_i, h_i)$ in L_{RH3-1}; Γ returns h_i to $\mathscr{A}_1$.

H_{3-2}query: on receiving $(\mathrm{ID}_i, T_i, P_i)$, if L_{RH3-2} contains a tuple of $(\mathrm{ID}_i, T_i, P_i, h_{TPi})$, Γ returns h_{TPi} to $\mathscr{A}_1$. Otherwise, Γ randomly chooses h_{TPi} that has not been chosen by Γ and inserts $(\mathrm{ID}_i, T_i, P_i, h_{TPi})$ to L_{RH3-2} and returns h_{TPi} to $\mathscr{A}_1$.

$\mathrm{R}s_i(\mathrm{ID}_i)$: on receiving ID_i, Γ returns s_i to $\mathscr{A}_1$ from L_C.

$\mathrm{R}x_i(\mathrm{ID}_i)$: on receiving ID_i, Γ returns x_i to $\mathscr{A}_1$ from L_C.

$\mathrm{R}s$: Γ returns $\perp$ to $\mathscr{A}_1$.

$\mathrm{Rpk}_i(x'_i)$: on receiving x'_i, Γ computes $P'_i = x'_i P$ and updates all the tuples with $x_i = x'_i$, $P_i = P'_i$

$\mathrm{R}t_i(\Pi^f_{i,j})$:

(1) If $i = A$, $j = B$, $f = m$ or $i = B$, $j = A$, and $f = n$

(a) If $i = A$, $j = B$, and $f = m$, Γ sets $T_A = AP$, inserts $(\Pi^m_{A,B}, \perp, AP)$ to $L_t(\Pi^f_{i,j})$, and returns $T_A = AP$ to $\mathscr{A}_1$

(b) If $i = B$, $j = A$, and $f = n$, Γ sets $T_B = BP$, inserts $(\Pi^n_{B,A}, \perp, BP)$ to $L_t(\Pi^f_{i,j})$, and returns $T_B = BP$ to $\mathscr{A}_1$

(2) Otherwise,

(a) if $L_t(\Pi^f_{i,j})$ contains a tuple of $(\Pi^f_{i,j}, t_i, T_i)$, Γ returns t_i, T_i to $\mathscr{A}_1$

(b) Γ randomly chooses t_i, computes $T_i = t_i P$, inserts $(\Pi^f_{i,j}, t_i, T_i)$ to $L_t(\Pi^f_{i,j})$, and returns t_i, T_i to $\mathscr{A}_1$

$\mathrm{Send}(\Pi^f_{i,j}, M)$: If the matched session $\Pi^n_{j,i}$ of $\Pi^m_{i,j}$ exists, this query should be asked after $\mathrm{Create}(\mathrm{ID}_i)$ and $\mathrm{R}t_i(\Pi^f_{i,j})$ when $\mathscr{A}_1$ gets T_i and R_i.

(1) $\mathscr{A}_1$ gets $\{\mathrm{ID}_i, T_i, R_i\}$ from L_C and $L_t(\Pi^f_{i,j})$ and sets $M = \{\mathrm{ID}_i, T_i, R_i\}$, and Γ returns $\{\mathrm{ID}_j, T_j, R_j\}$ from L_C and $L_t(\Pi^f_{i,j})$. That is, $\mathscr{A}_1$ initiates the session

$\Pi_{i,j}^m$ with message $\{\mathrm{ID}_i, T_i, R_i\}$ and gets respond with $\{\mathrm{ID}_j, T_j, R_j\}$ of the matched session $\Pi_{j,i}^n$

(2) If $M = \perp$, Γ gets T_i, T_j, R_i, and R_j from L_C and $L_t(\Pi_{i,j}^f)$ and initiates the session $\Pi_{i,j}^m$ with $\{\mathrm{ID}_i, T_i, R_i\}$. Then, $\mathscr{A}_1$ gets $\{\mathrm{ID}_j, T_j, R_j\}$ from Γ as the message of the matched session $\Pi_{j,i}^n$

H_2query: on receiving $(\mathrm{ID}_i, \mathrm{ID}_j, (T_i)\Pi_{i,j}^m, (T_j)\Pi_{j,i}^n, K_{ij}, \perp)$

(1) If L_{RH2} contains a tuple of $(\mathrm{ID}_i, \mathrm{ID}_j, (T_i)\Pi_{i,j}^m, (T_j)\Pi_{j,i}^n, K_{ij}, h_{2ij})$, Γ returns h_{2ij} to $\mathscr{A}_1$

(2) Otherwise, Γ randomly chooses h_{2ij} that has not been chosen by Γ and inserts $(\mathrm{ID}_i, \mathrm{ID}_j, (T_i)\Pi_{i,j}^m, (T_j)\Pi_{j,i}^n, K_{ij}, h_{2ij})$ to L_{RH2} and returns h_{2ij} to $\mathscr{A}_1$

$\mathrm{Rsk}_{ij}(\Pi_{i,j}^m)/\mathrm{Rsk}_{ji}(\Pi_{j,i}^n)$:

(1) If $i \neq A$ and $j \neq B$, Γ gets K_{ij} from L_{RH2}; gets s_i, s_j, x_i, x_j, t_i, t_j, h_{TPi}, and h_{TPj} from L_C, $L_t(\Pi_{i,j}^f)$, and L_{RH3-2}; computes $K_{ij}^* = h_{TPi}h_{TPj}t_it_jP + h_{TPi}t_is_jP + h_{TPi}t_ix_jP + h_{TPj}s_it_jP + s_is_jP + s_ix_jP + h_{TPj}x_it_jP + x_is_jP + x_ix_jP$; and checks whether $K_{ij} = K_{ij}^*$ or not. If they are equal, Γ returns h_{2ij} from L_{RH2}. Otherwise, Γ sets $K_{ij} = K_{ij}^*$, randomly chooses h_{2ij} that has not been chosen by Γ and inserts $(\mathrm{ID}_i, \mathrm{ID}_j, (T_i)\Pi_{i,j}^m, (T_j)\Pi_{j,i}^n, K_{ij}, h_{2ij})$ to L_{RH2}, and returns h_{2ij} to $\mathscr{A}_1$

(2) Otherwise, Γ returns $\perp$ to $\mathscr{A}_1$

(C3) $\mathscr{A}_1$ asks a test $(\Pi_{A,B}^m)$ query; the challenger $\complement$ performs as follows:

(1) $\complement$gets s_A, s_B, x_A, x_B, T_A, T_B, h_{TPA}, and h_{TPB} from L_C, L_{RH3-2}, and L_t $(\Pi_{i,j}^f)$ where $T_A = AP$, $T_B = BP$

(2) $\complement$ computes K_{AB}^* with the candidate solution of ABP and gets h_{2AB}^* from L_{RH2} with $(\mathrm{ID}_A, \mathrm{ID}_B, (T_A)\Pi_{A,B}^m, (T_B)\Pi_{B,A}^n, K_{AB}^*)$

(3) $\complement$ picks randomly $b \in \{0,1\}$. If $b = 1$, $\complement$ replies $h_{2AB} = h_{2AB}^*$ to $\mathscr{A}_1$; otherwise, $\complement$ replies h_{2AB} as a random string to $\mathscr{A}_1$

(C4) $\mathscr{A}_1$ asks the Γ for a polynomial number of the queries about fresh session $\Pi_{i,j}^m$.

(C5) $\mathscr{A}_1$ makes a guess bit b$'$.

$\mathscr{A}_1$ wins the game in case FI1 by guessing $b' = b$. After the test $(\Pi_{A,B}^m)$ query, $\mathscr{A}_1$ asks H_2 query with $(\mathrm{ID}_A, \mathrm{ID}_B, (T_A)\Pi_{A,B}^m, (T_B)\Pi_{B,A}^n, \perp, h_{2AB})$. Γ gets K_{AB} in L_{RH2} and asks RDDH oracle with $\mathrm{RDDH}(H_3(\mathrm{ID}_A, T_A, P_A)T_A + (R_A + H_3(\mathrm{ID}_A, R_A, P_A)P_{\mathrm{pub}}) + P_A), (H_3(\mathrm{ID}_B, T_B, P_B)\ T_B + (R_B + H_3(\mathrm{ID}_B, R_B, P_B)P_{pub}) + P_B), K_{AB})$. If $b' = b = 1$ with the probability of 1/2, the above RDDH oracle will return 1; then, Γ can get $K_{AB} = K_{AB}^*$ and compute $C \bullet P$ with

$$C \bullet P = t_At_B \bullet P = (h_{\mathrm{TPA}}h_{\mathrm{TPB}})^{-1}(K_{AB}^* - s_Bh_{\mathrm{TPA}}T_A - x_Bh_{\mathrm{TPA}}T_A - s_Ah_{\mathrm{TPB}}T_B - s_As_BP - s_Ax_BP - x_Ah_{\mathrm{TPB}}T_B - x_As_BP - x_Ax_BP). \tag{3}$$

Let T_{mul} be the time for one scalar multiplication operation and T_{add} be the point addition operation over elliptic curve. The time to compute CP is $t_{\mathrm{CP}} = 8T_{\mathrm{mul}} + 8T_{\mathrm{add}}$.

Then, if $\mathscr{A}_1$ win the game in case FI1 with advantage ε and running time t, then an algorithm Γ can be constructed to solve the GDH problem with advantage ε' and running time τ by interacting with $\mathscr{A}_1$ only if the game is completed, where events E_1 and E_2 occur.

E_1: $\mathscr{A}_1$ chooses $\Pi_{A,B}^m$ for challenge.

E_2: $b' = b = 1$.

Meanwhile, event E_1 occurs which means all of the events E_{1-1}, E_{1-2}, and E_{1-3} occur.

E_{1-1} : $\mathscr{A}_1$ chooses participants A and B for the challenge with $\mathrm{Create}(\mathrm{ID}_A)$ and $\mathrm{Create}(\mathrm{ID}_B)$ query.

E_{1-2}: $\mathscr{A}_1$ makes query $\mathrm{Send}(\Pi_{A,B}^m, \{\mathrm{ID}_A, T_A, R_A\})$ or $\mathrm{Send}(\Pi_{A,B}^m, \perp)$.

E_{1-3}: $\mathscr{A}_1$ makes $Rt_A(\Pi_{A,B}^m)$ query and $Rt_B(\Pi_{B,A}^n)$ query.

$$\varepsilon' = \Pr[E_{1-1}] \bullet \Pr[E_{1-2}] \bullet \Pr[E_{1-3}] \bullet \Pr[E_2] \bullet \varepsilon = \frac{1}{C_{q_c}^2}\frac{C_2^1}{C_{q_{\mathrm{sd}}}^1}\frac{1}{C_{q_t}^2} \bullet \frac{1}{2} \bullet \varepsilon = \frac{4}{q_cq_{\mathrm{sd}}q_t(q_c-1)(q_t-1)} \bullet \varepsilon,$$

$$\tau \leq \sum_{i=2}^{3} q_it_i + q_ct_c + q_st_s + q_tt_t + q_{\mathrm{sk}}t_{\mathrm{sk}} + q_{\mathrm{sd}}t_{\mathrm{sd}} + q_st_s + q_{\mathrm{pki}}t_{\mathrm{pki}} + q_{\mathrm{rddh}}t_{\mathrm{rddh}} + t + t_{\mathrm{CP}}. \tag{4}$$

□

Lemma 8. *The same as Lemma 7 but in case FI2.*

Proof. To interact with $\mathscr{A}_1$, a GDH slover Γ runs the same steps as that in Lemma 7 to solve the instance of the GDH problem. $\mathscr{A}_1$ asks the Γ for a polynomial number of the queries as shown in Lemma 7; Γ answers the following queries differently:

$\mathrm{Create}(\mathrm{ID}_i)$: on receiving (ID_i), Γ performs as follows:

(1) If L_C contains a tuple of $(\mathrm{ID}_i, s_i, x_i, P_i, R_i)$

(a) If $i \neq A, B$, Γ returns all the elements of the tuple to $\mathscr{A}_1$

(b) Otherwise, Γ returns $(\mathrm{ID}_i, \perp, x_i, P_i, R_i)$ to $\mathscr{A}_1$

(2) Otherwise,

(a) if $i \neq A, B$, Γ randomly chooses x_i, s_i, h_i and then computes $P_i = x_i P$ and $R_i = s_i P - h_i P_{\text{pub}}$; Γ inserts ($\text{ID}_i, s_i, x_i, P_i, R_i$) to L_C and $(\text{ID}_i, R_i, P_i, h_i)$ to L_{RH3-1} and returns $(\text{ID}_i, s_i, x_i, P_i, R_i)$ to $\mathscr{A}_1$

(b) Γ randomly chooses x_i, h_i and computes $P_i = x_i P$ and $R_i = IP - h_i P_{\text{pub}}$ ($I = A$ when $i = A$; I=B when $i = B$); Γ inserts $(\text{ID}_i,\perp,x_i, P_i, R_i)$ to L_C and $(\text{ID}_i, R_i, P_i, h_i)$ to L_{RH3-1} and returns $(\text{ID}_i,\perp,x_i, P_i, R_i)$ to $\mathscr{A}_1$

$\text{R}s_i(\text{ID}_i)$: on receiving ID_i, Γ performs as follows:

(1) If $i \neq A, B$, Γ returns s_i from L_C to $\mathscr{A}_1$

(2) Otherwise, Γ returns $\perp$ to $\mathscr{A}_1$

$\text{R}t_i(\Pi^f_{i,j})$:

(1) If $L_t(\Pi^f_{i,j})$ contains a tuple of $(\Pi^f_{i,j}, t_i, T_i)$, Γ returns t_i, T_i to $\mathscr{A}_1$

(2) Otherwise, Γ randomly chooses t_i, computes $T_i = t_i P$, inserts $(\Pi^f_{i,j}, t_i, T_i)$ to L_t $(\Pi^f_{i,j})$, and returns t_i, T_i to $\mathscr{A}_1$

Moreover, the first step of (C3) is also different with that of Lemma 7.

(C3) $A1$ asks a test $(\Pi^m_{A,B})$ query; the challenger $\mathbb{C}$ performs as follows:

(1) $\mathbb{C}$ gets x_A, x_B, t_A, t_B, R_A, R_B, P_A, P_B, h_A, h_B, T_A, T_B, h_{TPA}, and h_{TPB} from L_C, L_{RH3-1}, L_{RH3-2}, and L_t $(\Pi^f_{i,j})$ where $R_A + h_A P_{\text{pub}} = AP$, $R_B + h_B P_{\text{pub}} = BP$

Γ gets $C \bullet P$ with

$$\begin{aligned} C\bullet P = s_A s_B \bullet P = \big(&K^*_{AB} - h_{\text{TPA}} h_{\text{TPB}} t_A t_B \bullet P - h_{\text{TPA}} t_A \left(R_B + h_B P_{\text{pub}}\right) \\ &- x_B h_{\text{TPA}} T_A - h_{\text{TPB}} t_B \left(R_A + h_A P_{\text{pub}}\right) - x_B \left(R_A + h_A P_{\text{pub}}\right) \\ &- x_A h_{\text{TPB}} T_B - x_A \left(R_B + h_B P_{\text{pub}}\right) - x_A x_B P\big). \end{aligned} \tag{5}$$

□

Lemma 9. *The same as Lemma 7 but in case FI3.*

Proof. To interact with $\mathscr{A}_1$, a GDH slover Γ runs the same steps as that in Lemma 7 to solve the instance of the GDH problem. $\mathscr{A}_1$ asks the Γ for a polynomial number of the queries as shown in Lemma 7; Γ answers the following queries differently:

Create(ID_i): on receiving (ID_i), Γ performs as follows:

(1) If L_C contains a tuple of $(\text{ID}_i, s_i, x_i, P_i, R_i)$

(a) If $i \neq B$, Γ returns all the elements of the tuple to $\mathscr{A}_1$

(b) Otherwise, Γ returns $(\text{ID}_i,\perp,x_i, P_i, R_i)$ to $\mathscr{A}_1$

(1) Otherwise,

(a) if $i \neq B$, Γ randomly chooses x_i, s_i, h_i and then computes $P_i = x_i P$ and $R_i = s_i P - h_i P_{\text{pub}}$; Γ inserts $(\text{ID}_i, s_i, x_i, P_i, R_i)$ to L_C and $(\text{ID}_i, R_i, P_i, h_i)$ to L_{RH3-1} and returns $(\text{ID}_i, s_i, x_i, P_i, R_i)$ to $\mathscr{A}_1$

(b) Γ randomly chooses x_i, h_i and computes $P_i = x_i P$ and $R_i = BP - h_i P_{pub}$; Γ inserts $(\text{ID}_i,\perp,x_i, P_i, R_i)$ to L_C and $(\text{ID}_i, R_i, P_i, h_i)$ to L_{RH3-1} and returns $(\text{ID}_i,\perp,x_i, P_i, R_i)$ to $\mathscr{A}_1$

$\text{R}s_i(\text{ID}_i)$: on receiving ID_i, Γ performs as follows:

(1) If $i \neq B$, Γ returns s_i from L_C to $\mathscr{A}_1$

(2) Otherwise, Γ returns $\perp$ to $\mathscr{A}_1$

$\text{R}t_i(\Pi^f_{i,j})$:

(1) If $i = A$, $j = B$, and $f = m$, Γ sets $T_A = AP$, inserts ($\Pi^m_{A,B}$,$\perp$,AP) to $L_t(\Pi^f_{i,j})$, and returns $T_A = \text{AP}$ to $\mathscr{A}_1$

(2) Otherwise,

(a) if $L_t(\Pi^f_{i,j})$ contains a tuple of $(\Pi^f_{i,j}, t_i, T_i)$, Γ returns t_i, T_i to $\mathscr{A}_1$

(b) Γ randomly chooses t_i, computes $T_i = t_i P$, inserts ($\Pi^f_{i,j}, t_i, T_i$) to $L_t(\Pi^f_{i,j})$, and returns t_i, T_i to $\mathscr{A}_1$

Moreover, the first step of (C3) is also different with that of Lemma 7.

(C3) $A1$ asks a test $(\Pi^m_{A,B})$ query; the challenger $\mathbb{C}$ performs as follows:

(1) $\mathbb{C}$ gets s_A, x_A, x_B, t_B, R_A, R_B, P_A, P_B, T_A, T_B, h_A, h_B, h_{TPA}, and h_{TPB} from L_C, L_{RH3-1}, L_{RH3-2}, and $L_t(\Pi^f_{i,j}$) where $T_A = AP$, $R_B + h_B P_{\text{pub}} = BP$

Γ gets $C \bullet P$ with

$$\begin{aligned} C\bullet P = t_A s_B \bullet P = (h_{\text{TPA}})^{-1} (&K^*_{AB} - h_{\text{TPA}} h_{\text{TPB}} t_B T_A - x_B h_{\text{TPA}} T_A \\ &- h_{\text{TPB}} t_B \left(R_A + h_A P_{\text{pub}}\right) - s_A \left(R_B + h_B P_{\text{pub}}\right) \\ &- x_B \left(R_A + h_A P_{\text{pub}}\right) - x_A h_{\text{TPB}} T_B - x_A \left(R_B + h_B P_{\text{pub}}\right) - x_A x_B P). \end{aligned} \tag{6}$$

□

Lemma 10. *The same as Lemma 7 but in case FI4.*

Proof. Case FI4 has no essential difference with case FI3; the proof can be omitted.

Table 4: Calculation time of password operation.

Symbol	Meaning	Time
T_{mul}	A dot product operation time on an elliptic curve	7.3529 ms
T_{h}	Hash function operation time	0.0004 ms
T_{m}	Modular multiplication operation time	0.0147 ms
$T_{\exp}$	Modular exponential operation time	18.38225 ms
T_{add}	One point plus operation time on an elliptic curve	0.0613 ms

Γ gets $C{\bullet}P$ with

$$C{\bullet}P = s_A t_B{\bullet}P = (h_{\text{TPB}})^{-1}(K^*_{AB} - h_{\text{TPA}}h_{\text{TPB}}t_A T_B - h_{\text{TPA}}t_A \cdot (R_B + h_B P_{\text{pub}}) - x_B h_{\text{TPA}} T_A - s_B(R_A + h_A P_{\text{pub}}) - x_B(R_A + h_A P_{\text{pub}}) - x_A h_{\text{TPB}} T_B - x_A(R_B + h_B P_{\text{pub}}) - x_A x_B P). \tag{7}$$

□

Lemma 11. *The same as Lemma 7 but in case FI5 and with different ε'. $\varepsilon' = (1/q_c q_{sd} q_t (q_c - 1)){\bullet}\varepsilon$*

Proof. To interact with $\mathscr{A}_1$, a GDH slover Γ runs the same steps as that in Lemma 7 to solve the instance of the GDH problem. $\mathscr{A}_1$ asks the Γ for a polynomial number of the queries as shown in Lemma 9; Γ answers the following queries differently:

$\mathrm{R}t_i(\Pi^f_{i,j})$:

(1) If $i = A$, $j = B$, $f = m$ or $j = A$, $i = B$, and $f = n$

(a) If $i = A$, $j = B$, and $f = m$, Γ sets $T_A = AP$, inserts $(\Pi^m_{A,B},\perp,AP)$ to $L_t(\Pi^f_{i,j})$, and returns $T_A = AP$ to $\mathscr{A}_1$

(b) Otherwise, Γ returns $\perp$ to $\mathscr{A}_1$

(2) Otherwise,

(a) if $L_t(\Pi^f_{i,j})$ contains a tuple of $(\Pi^f_{i,j}, t_i, T_i)$, Γ returns t_i, T_i to $\mathscr{A}_1$

(b) Γ randomly chooses t_i, computes $T_i = t_i P$, inserts $(\Pi^f_{i,j}, t_i, T_i)$ to $L_t(\Pi^f_{i,j})$, and returns t_i, T_i to $\mathscr{A}_1$

$\mathrm{Send}(\Pi^f_{i,j}, M)$:
The matched session $\Pi^n_{j,i}$ of $\Pi^m_{i,j}$ does not exist.

(1) This query should be asked after $\mathrm{Create}(\mathrm{ID}_i)$ and $\mathrm{R}t_i(\Pi^f_{i,j})$ when $\mathscr{A}_1$ gets T_i and R_i. $\mathscr{A}_1$ gets $\{\mathrm{ID}_i, T_i, R_i\}$ from L_C and $L_t(\Pi^f_{i,j})$ and sets $M = \{\mathrm{ID}_i, T_i, R_i\}$; Γ returns $\{\mathrm{ID}_j,\perp,R_j\}$ to $\mathscr{A}_1$ as response

(2) If $M = \perp$, Γ initiates the session $\Pi^m_{i,j}$ with $\{\mathrm{ID}_i, T_i, R_i\}$

(a) If $i = A$, $j = B$, and $f = m$, this query should be asked after $\mathrm{Create}(\mathrm{ID}_B)$. $\mathscr{A}_1$ gets x_B, P_B from L_C, sets $t'_B = x_B$, and sets $T'_B = P_B$, which responds with $\{\mathrm{ID}_B, T'_B, R_B\}$

(b) Otherwise, $\mathscr{A}_1$ responds with $\{\mathrm{ID}_j,\perp,R_j\}$

Moreover, the first two steps of (C3) is also different with that of Lemma 9.

(C3) $\mathscr{A}_1$ asks a test $(\Pi^m_{A,B})$ query; the challenger $\mathbb{C}$ performs as follows:

(1) $\mathbb{C}$ gets $s_A, x_A, x_B, R_A, R_B, P_A, P_B, T_A, h_A, h_B, h_{TPA}$, and h_{TPB} from L_C, L_{RH3-1}, L_{RH3-2}, and L_t $(\Pi^f_{i,j})$ where $T_A = AP, R_B + h_B P_{\text{pub}} = BP$. $\mathbb{C}$ gets T'_B from the response message from $\mathscr{A}_1$

(2) Let $T'_B = B'P$; $\mathbb{C}$ computes K^*_{AB} with the candidate solution of ABP and $AB'P$ and gets h^*_{2AB} from L_{RH2} with $(\mathrm{ID}_A, \mathrm{ID}_B, (T_A)\Pi^m_{A,B}, (T_B)\Pi^n_{B,A}, K^*_{AB})$

As the proof in Lemma 7, if $\mathscr{A}_1$ win the game in case FI5, an algorithm Γ can be constructed to solve the GDH problem only if the game is completed, where events E_1 and E_2 occur. In case FI5, event E_1 occurs which means all of the events E_{1-1}, E_{1-4}, and E_{1-5} occur.

E_{1-1} : $\mathscr{A}_1$ chooses participant A and B for challenge with $\mathrm{Create}(\mathrm{ID}_A)$ and $\mathrm{Create}(\mathrm{ID}_B)$ query

E_{1-4}: $\mathscr{A}_1$ makes query $\mathrm{Send}(\Pi^m_{A,B},\perp)$

E_{1-5}: $\mathscr{A}_1$ makes $\mathrm{R}t_A(\Pi^m_{A,B})$ query

$$\begin{aligned}\varepsilon' &= \Pr[E_{1-1}]{\bullet}\Pr[E_{1-4}]{\bullet}\Pr[E_{1-5}]{\bullet}\Pr[E_2]\varepsilon \\ &= \frac{1}{C^2_{q_c}}\frac{1}{C^1_{q_{sd}}}\frac{1}{C^1_{q_t}}{\bullet}\frac{1}{2}{\bullet}\varepsilon = \frac{1}{q_c q_{sd} q_t (q_c - 1)}{\bullet}\varepsilon.\end{aligned} \tag{8}$$

As Γ gets the response with $\{\mathrm{ID}_B, T'_B, R_B\}$ in query $\mathrm{Send}(\Pi^m_{A,B},\perp)$, in which $t'_B = x_B$ and $T'_B = P_B$; then, Γ

Table 5: Computation of overhead comparison.

Protocols	Number of rounds	Low-power mobile node	Powerful node	Join phase	Removal phase	Total computation cost
Chuang and Tseng [28]	3	$2T\exp + 3Th$	$(m+1)T\exp + (2m+1)Th$	$2T\exp + (2m+3)Th$	$(2m+2)Th$	$(m+5)T\exp + (6m+9)Th$
Wu et al. [29]	2	$2T\exp + 2Th$	$(m+1)T\exp + (m+1)Th$	$2T\exp + (m+4)Th$	$(m+4)Th$	$(m+5)T\exp + (3m+11)Th$
Islam et al. [30]	2	$4T\text{mul} + 4Th$	$(3m+2)T\text{mul} + (2m+2)Th$	$(2m+5)T\text{mul} + 6Th$	$(2m+4)T\text{mul} + 5Th$	$(7m+15)T\text{mul} + (2m+17)Th$
Mandal et al. [31]	2	$5T\text{mul} + 9Th$	$(4m+2)T\text{mul} + (5m+7)Th$	$(3m+4)T\text{mul} + (2m+11)Th$	$(3m+4)T\text{mul} + (2m+11)Th$	$(10m+15)T\text{mul} + (9m+38)Th$
Luo et al. [21]	1	$5T\text{mul} + T\text{add} + 4Th$	$(5m+1)T\text{mul} + 3mT\text{add} + 4mTh$	$(3m+10)T\text{mul} + (m+5)T\text{add} + (m+8)Th$	$(3m-1)T\text{mul} + (m+2)T\text{add} + mTh$	$(11m+15)T\text{mul} + (5m+8)T\text{add} + (6m+12)Th$
Proposed scheme	2	$5T\text{mul} + 3Th$	$4(m-1)T\text{mul} + 6(m-1)Th$	$(4m+5)T\text{mul} + (6m+3)Th$	$(4m-3)T\text{mul} + (6m-9)Th$	$(12m+3)T\text{mul} + (18m-9)Th$

computes $C \bullet P$ with

$$C \bullet P = t_A s_B \bullet P = (h_{\mathrm{TPA}})^{-1} \Big(K^*_{AB} - h_{\mathrm{TPA}} h_{\mathrm{TPB}} t'_B T_A - x_B h_{\mathrm{TPA}} T_A - h_{\mathrm{TPB}} t'_B (R_A + h_A P_{\mathrm{pub}}) - s_A (R_B + h_B P_{\mathrm{pub}}) - x_B (R_A + h_A P_{\mathrm{pub}}) - x_A h_{\mathrm{TPB}} T'_B - x_A (R_B + h_B P_{\mathrm{pub}}) - x_A x_B P \Big) = (h_{\mathrm{TPA}})^{-1} (K^*_{AB} - h_{\mathrm{TPA}} h_{\mathrm{TPB}} x_B T_A - x_B h_{\mathrm{TPA}} T_A - h_{\mathrm{TPB}} x_B (R_A + h_A P_{\mathrm{pub}}) - s_A (R_B + h_B P_{\mathrm{pub}}) - x_B (R_A + h_A P_{\mathrm{pub}}) - x_A h_{\mathrm{TPB}} P_B - x_A (R_B + h_B P_{\mathrm{pub}}) - x_A x_B P). \tag{9}$$

□

Lemma 12. *The same as Lemma 11 but in case FI5.*

Proof. To interact with $\mathscr{A}_1$, a GDH slover Γ runs the same steps as that in Lemma 7 to solve the instance of the GDH problem. $\mathscr{A}_1$ asks the Γ for a polynomial number of the queries as shown in Lemma 8; Γ answers the following queries differently:

$\mathrm{R}t_i(\Pi^f_{i,j})$:

(1) If $j = A$, $i = B$, and $f = n$, Γ returns $\perp$ to $\mathscr{A}_1$

(2) Otherwise,

(a) if L_t $(\Pi^f_{i,j})$ contains a tuple of $(\Pi^f_{i,j}, t_i, T_i)$, Γ returns t_i, T_i to $\mathscr{A}_1$

(b) Γ randomly chooses t_i, computes $T_i = t_i P$, inserts ($\Pi^f_{i,j}, t_i, T_i$) to $L_t(\Pi^f_{i,j})$, and returns t_i, T_i to $\mathscr{A}_1$

$\mathrm{Send}(\Pi^f_{i,j}, M)$: the same as that of Lemma 11.

Moreover, the first two steps of (C3) is also different with that of Lemma 8.

(C3) $\mathscr{A}_1$ asks a test $(\Pi^m_{A,B})$ query; the challenger $\mathfrak{C}$ performs as follows:

(1) $\mathfrak{C}$ gets $x_A, x_B, t_A, R_A, R_B, P_A, P_B, h_A, h_B, T_A, h_{TPA}$, and h_{TPB} from L_C, L_{RH3-1}, L_{RH3-2}, and $L_t(\Pi^f_{i,j})$ where $R_A + h_A P_{\mathrm{pub}} = AP, R_B + h_B P_{\mathrm{pub}} = BP$. $\mathfrak{C}$ gets T'_B from the response message from $\mathscr{A}_1$

(2) Let $T'_B = B'P$; $\mathfrak{C}$ computes K^*_{AB} with the candidate solution of ABP and $AB'P$ and gets h^*_{2AB} from L_{RH2} with $(\mathrm{ID}_A, \mathrm{ID}_B, (T_A)\Pi^m_{A,B}, (T_B)\Pi^n_{B,A}, K^*_{AB})$

Γ gets $C \bullet P$ with

$$C \bullet P = s_A s_B \bullet P = \Big(K^*_{AB} - h_{\mathrm{TPA}} h_{\mathrm{TPB}} t_A t'_B \bullet P - h_{\mathrm{TPA}} t_A (R_B + h_B P_{\mathrm{pub}}) - x_B h_{\mathrm{TPA}} T_A - h_{\mathrm{TPB}} t'_B (R_A + h_A P_{\mathrm{pub}}) - x_B (R_A + h_A P_{\mathrm{pub}}) - x_A h_{\mathrm{TPB}} T'_B - x_A (R_B + h_B P_{\mathrm{pub}}) - x_A x_B P \Big) = s_A s_B \bullet P = (K^*_{AB} - h_{\mathrm{TPA}} h_{\mathrm{TPB}} t_A x_B \bullet P - h_{\mathrm{TPA}} t_A (R_B + h_B P_{\mathrm{pub}}) - x_B h_{\mathrm{TPA}} T_A - h_{\mathrm{TPB}} x_B (R_A + h_A P_{\mathrm{pub}}) - x_B (R_A + h_A P_{\mathrm{pub}}) - x_A h_{\mathrm{TPB}} P_B - x_A (R_B + h_B P_{\mathrm{pub}}) - x_A x_B P). \tag{10}$$

With the similar proof, Lemma 13 to 18 can be derived for case FII1-FII6, which need not be given in this paper. Then, according to Lemma 7 to 18, if $\mathscr{A}$ win the game in polynomial time, Γ can solve the GDH problem, which is contradictory with the security assumption of GDH problem. Then, we conclude that $\mathscr{A}$ cannot win the Game and $\mathrm{Adv}_A(k)$ is negligible. Therefore, our protocol is secure under the eCK security model with the GDH assumption. □

5. Performance Analysis

According to the research work in literatures [26, 27], the calculation time of relevant operation in protocol execution is shown in Table 4. The cross-domain authentication scheme proposed in this paper is compared and analyzed with existing schemes of the same type, and the cost comparison of scheme calculation is shown in Table 5. The parameters used in the table are as follows:

m: number of participants in each domain *Di*

n: the number of domains for cross-domain key negotiation

N: the number of participants in a negotiation

In this scheme, we use the dot product of elliptic curve operation and hash function to encryption scheme. Chuang and Tseng and Wu et al.'s scheme [28, 29] used modular exponential operation. Table 4 shows that the operation time of modular finger is longer than a dot product operation time on elliptic curve. Compared with that, our calculation cost is smaller and the number of rounds negotiated is also less. Though the computation cost of [30] is lower than our proposed model, however Tan's scheme lacks perfect forward secrecy [32], and the schemes do not have a signature mechanism that supports confidentiality, integrity, authentication, and nonrepudiation in a logical step. Mandal et al.'s scheme [31] used signature verification with a signature verification time of 2.005 s, but they did not include the signature verification time into the calculation cost. So compared with Mandal et al.'s scheme, our computational cost is still lower. Although in Luo et al.'s scheme [21], the group session key is calculated in one-round communication and the total computation cost is lower than our proposed protocol when *m* is large, our proposed protocol has lower computation cost for low-power mobile node and powerful node when there is no member joining or removing from the group. In addition, Luo et al.'s scheme only had a secondary security level, the malicious KGC could collude with some malicious users to attack the protocol indicated by Ren et al. [24].

6. The Conclusion

In view of the shortage and complexity of cross-domain group authentication communication schemes, this paper proposes a certificateless cross-domain group key management scheme based on ECC. In the proposed scheme, key negotiation is divided into two parts: intradomain key negotiation and interdomain key negotiation. On the basis of ensuring security, group cross-domain communication is realized. It avoids the complex certification path construction and verification process and reduces the length of the trust path. The scheme is proved to be secure in random oracle model with low computational cost and is suitable for users' group communication requirements across multiple domains.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Key R&D Program of China on Blue Granary Technology Innovation (2019YFD0900805), the Program for Shanghai Collaborative Innovation Center for Cultivating Elite Breeds and Green-Culture of Aquaculture Animals (Project No. 2021-KJ-02-12), and the Science and Technology on Near-Surface Detection Laboratory (Project No. 6142414200410). The authors would also like to acknowledge the support of Shang Hai Ocean University throughout the research process.

References

[1] C. Guo, *Research on Cross-Domain Group Authentication Key Exchange Protocol and Its Applications*, BEIJING INSTITUTE OF TECHNOLOGY, 2015.

[2] C. Liling and G. Wancheng, "Analysis of certificateless signcryption schemes and construction of a secure and efficient pairing-free one based on ECC," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 9, 2018.

[3] Y. Zhou, X. Chen, and M. Kolberg, "An anonymous and efficient ECC-based authentication scheme for SIP," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8886585, 11 pages, 2020.

[4] K. A. Alezabi, F. Hashim, S. J. Hashim, B. M. Ali, and A. Jamalipour, "Efficient authentication and re-authentication protocols for 4G/5G heterogeneous networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020.

[5] Z. Wf, W. Xm, W. Guo, and H. Dk, "An efficient inter-enterprise authentication scheme for VE based on the elliptic curve cryptosystem," *Acta Electronica Sinica*, vol. 42, no. 6, p. 1095, 2014.

[6] H. Bin, *Improvement and Research on Mechanism of Certificate Revocation Based on PKI*, SHANGHAI JIAO TONG UNIVERSITY, 2015.

[7] D. Basin, C. Cremers, T. H.-J. Kim, A. Perrig, R. Sasse, and P. Szalachowski, "Design, analysis, and implementation of ARPKI an attack-resilient public-key infrastructure," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 3, pp. 393–408, 2018.

[8] Z. Zhicheng, L. Lixin, and L. Zuohui, "Efficient cross-domain authentication scheme based on blockchain technology," *Journal of Computer Applications*, vol. 38, no. 2, p. 316, 2018.

[9] G. Dong, Y. Chen, J. Fan, J. Bai, P. Zhang, and F. Li, "Anonymous cross-domain authentication scheme for medical PKI system," in *Proceedings of the ACM Turing Celebration Conference*, China, 2019.

[10] Y. Chen, G. Dong, J. Bai, Y. Hao, F. Li, and H. Peng, "Trust enhancement scheme for cross domain authentication of PKI system," in *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 103–110, Guilin, China, 2019.

[11] Q. Chen, T. Wu, C. Hu, A. Chen, and Q. Zheng, "An identity-based cross-domain authenticated asymmetric group key agreement," *Information*, vol. 12, no. 3, p. 112, 2021.

[12] L. Changyuan, H. Shiwei, and X. Hongzhi, "Identity-based cross-domain authentication scheme in pervasive computing environments," *Journal on Communications*, vol. 32, no. 9, pp. 111–115+122, 2011.

[13] M. S. Farash and M. A. Attari, "An enhanced authenticated key agreement for session initiation protocol," *Information Technology and Control*, vol. 42, no. 4, pp. 333–342, 2013.

[14] C. L. Cao, M. Q. Liu, R. Zhang, and Y. X. Yang, "Provably secure authenticated key agreement protocol based on hierarchical identity," *Journal of Electronics & Information Technology*, vol. 36, no. 12, pp. 2848–2854, 2014.

[15] M. Kefei, C. Jie, and L. Jianwei, "Security analysis and improvements of hierarchical identity based authenticated key agreement scheme," *Journal of Electronics & Information Technology*, vol. 38, no. 10, pp. 2619–2626, 2016.

[16] M. Zhang and Y. Zhang, "Certificateless anonymous user authentication protocol for cloud computing," in *2015 International Conference on Intelligent Transportation, Big Data and Smart City*, pp. 200–203, Halong Bay, Vietnam, 2015.

[17] Z. Dong, L. Zhang, and J. Li, "Security enhanced anonymous remote user authentication and key agreement for cloud computing," in *2014 IEEE 17th International Conference on Computational Science and Engineering*, pp. 1746–1751, Chengdu, China, 2014.

[18] Y. Li, W. Chen, Z. Cai, and Y. Fang, "CAKA: a novel certificateless-based cross-domain authenticated key agreement protocol for wireless mesh networks," *Wireless Networks*, vol. 22, no. 8, pp. 2523–2535, 2016.

[19] H. Sun, B. He, C. Chen, T. Y. Wu, C. H. Lin, and H. Wang, "A provable authenticated group key agreement protocol for mobile environment," *Information Sciences*, vol. 321, pp. 224–237, 2015.

[20] D. Cheng, J. Liu, Z. Guan, and T. Shang, "A one-round certificateless authenticated group key agreement protocol for mobile ad hoc networks," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 11, pp. 2716–2722, 2016.

[21] M. Luo, J. Wu, and X. Li, "Cross-domain certificateless authenticated group key agreement protocol for 5G network slicings," *Telecommunication Systems*, vol. 74, no. 4, pp. 437–449, 2020.

[22] Y. Yang, X. Zheng, X. Liu, S. Zhong, and V. Chang, "Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system," *Future Generation Computer Systems*, vol. 84, pp. 160–176, 2018.

[23] B. Semal, K. Markantonakis, and R. N. Akram, "A certificateless group authenticated key agreement protocol for secure communication in untrusted UAV networks," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pp. 233–240, London, UK, 2018.

[24] H. Ren, S. Kim, D. Seo, and I. Lee, "A certificateless-based one-round authenticated group key agreement protocol to prevent

impersonation attacks," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 5, pp. 1687–1707, 2022.

[25] G. Lippold, C. Boyd, and J. G. Nieto, "Strongly secure certificateless key agreement," in *Pairing-Based Cryptography – Pairing 2009*, pp. 206–230, Springer, Berlin, Heidelberg, 2009.

[26] L. Xu and F. Wu, "Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care," *Journal of Medical Systems*, vol. 39, no. 2, p. 10, 2015.

[27] F. Wu, L. Xu, S. Kumari, and X. Li, "A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client-server networks," *Computers & Electrical Engineering*, vol. 45, pp. 274–285, 2015.

[28] Y.-H. Chuang and Y.-M. Tseng, "An efficient dynamic group key agreement protocol for imbalanced wireless networks," *International Journal of Network Management*, vol. 20, no. 4, pp. 167–180, 2010.

[29] T.-Y. Wu, Y.-M. Tseng, and C.-W. Yu, "Two-round contributory group key exchange protocol for wireless network environments," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, 2011.

[30] S. K. H. Islam and G. P. Biswas, "A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *Annals of Telecommunications*, vol. 67, no. 11-12, pp. 547–558, 2012.

[31] S. Mandal, S. Mohanty, and B. Majhi, "CL-AGKA: certificateless authenticated group key agreement protocol for mobile networks," *Wireless Networks*, vol. 26, no. 4, pp. 3011–3031, 2020.

[32] Z. Tan, "An efficient pairing-free identity-based authenticated group key agreement protocol," *International Journal of Communication Systems*, vol. 28, no. 3, pp. 534–545, 2015.