PAPER

A One-Round Certificateless Authenticated Group Key Agreement Protocol for Mobile Ad Hoc Networks

Dongxu CHENG[†a)], Jianwei LIU[†], *Nonmembers*, Zhenyu GUAN[†], *Member*, *and* Tao SHANG[†], *Nonmember*

SUMMARY Established in self-organized mode between mobile terminals (MT), mobile Ad Hoc networks are characterized by a fast change of network topology, limited power dissipation of network node, limited network bandwidth and poor security of the network. Therefore, this paper proposes an efficient one round certificateless authenticated group key agreement (OR-CLAGKA) protocol to satisfy the security demand of mobile Ad Hoc networks. Based on elliptic curve public key cryptography (ECC), OR-CLAGKA protocol utilizes the assumption of elliptic curve discrete logarithm problems (ECDLP) to guarantee its security. In contrast with those certificateless authenticated group key agreement (GKA) protocols, OR-CLAGKA protocol can reduce protocol data interaction between group users and it is based on efficient ECC public key infrastructure without calculating bilinear pairings, which involves negligible computational overhead. Thus, it is particularly suitable to deploy OR-CLAGKA protocol on MT devices because of its limited computation capacity and power consumption. Also, under the premise of keeping the forward and backward security, OR-CLAGKA protocol has achieved appropriate optimization to improve the performance of Ad Hoc networks in terms of frequent communication interrupt and reconnection. In addition, it has reduced executive overheads of key agreement protocol to make the protocol more suitable for mobile Ad Hoc network applications.

key words: *certificateless cryptography, group key agreement, mobile Ad Hoc networks, elliptic curve*

1. Introduction

Due to the unique feature of their application, mobile Ad Hoc networks have attracted the attention of many researchers. Thus, the applications of Ad Hoc networks are widely used under special conditions, such as battlefield communication of military units, emergency services of relief work after catastrophes, and sensor arrays based on wireless Ad Hoc network. Under certain special conditions, Ad Hoc networks also need confidential communication. However, since a mobile Ad Hoc network lacks a fixed network base station, it has become common for the network's nodes to access the network from anywhere, and thus the nodes frequently disconnect. Also, limited network bandwidth and node power dissipation greatly challenge the security protocol design based on Ad Hoc Networks. So as to achieve confidential communication between concerned parties on the Ad Hoc networks, it is necessary to reach establishing and distributing secret key agreement without reliable third party, and network communication is vulnerable to being intercepted and analyzed.

GKA protocol for Ad Hoc Networks provides a mechanism for at least two parties to build secret data through Internet, which is known by all parties involved so that members of the group can conduct encryption communication. Under the GKA protocol, every participant can take part in the part of the computing of shared key without any authorization center, through which a shared key can be generated. When group members are changed, the Group Controller (GC) stemming from the topology of mobile Ad Hoc networks can generate a new shared key. This process meets the requirement of forward-security and backward-security.

Adi Shamir [1] put forward the security concept of PKI based on ID for the first time, under which participants can achieve identity-based authentication so long as they know others' public identification. To avoid the Key Escrow problem of PKI based on ID, Al-Riyami and Pa-terson [2] put forward the CL-PKC that authentication is required from both parties, and the system facilitates the appearance of many certificateless key-agreement protocols [3]–[10]. At present, many certificateless CLGKAs [7]–[10] are designed by bilinear pairings, where the computing cost of each node is equivalent to almost 20 times [11] of scalar multiplication in point group of the elliptic curve under the same security level. This leads to a dramatic increase in cryptosystem computing cost. Later on, some CLGKAs [3]–[6], [14] without calculating of bilinear pairings were put forward. Among those GKA protocols, only two of them take rather small computing cost [5], [14], which make them suitable for mobile Ad Hoc networks environment, while the protocol [8] proposed by Heo et al. falls short of perfect forward security.

This paper presents a lightweight and certificateless GKA protocol without computing of bilinear pairings. The main feature of this protocol is that the public key and private key of all participants are only involved in authentication, instead of group key generation. Accordingly, the variation of the public key and private key set caused by the new group member does not affect the calculation of group key during the process of joining group protocol and leaving group protocol. In this design, the amount of calculation of group key generation can be reduced effectively, which makes it more suitable for mobile Ad Hoc network with limited computing resource. The rest of this paper is organized as followed: the second section offers the preliminary knowledge concerning OR-CLAGKA protocol; the third section details this protocol; the analysis of security performance of the protocol is presented in part four, and

Manuscript received May 20, 2016.
Manuscript publicized July 21, 2016.

[†]The authors are with the Department of Electronic and Information Engineering, Beihang University, Beijing 100191, China.
a) E-mail: chengdongxu@buaa.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2016EDP7210

the fifth section is the conclusion.

2. Preliminary

2.1 Security Requirements

There are some security targets and confidentiality requirements in [12] to be met in the design of group key agreement protocol.

Group key secrecy: It is infeasible for a passive adversary to decrypt group key in computation by intercepting data exchanged during protocol implementation.

Weak forward secrecy: One leaving the group will no longer continue to decrypt or to be informed by secret group communication.

Forward secrecy: Passive adversary cannot decrypt subsequent group key even if he has obtained a subset of group key used previously.

Weak backward secrecy: Newly admitted group users cannot decrypt and understand previous secret group communication.

Backward secrecy: Passive adversary cannot decrypt previous group key even if he has obtained a continuous group key subset.

Key independence: When the passive adversary is aware of a certain session key subset, he cannot decrypt or conjecture session key outside of this subset.

Perfect forward secrecy: Active adversary or passive adversary cannot decrypt present group secret communication even though they possess group users' long-term public and private key or previous group key.

2.2 Elliptic Curve Point Group

The Elliptic Curve Cryptosystems (ECC) maintain higher safety performance and computation efficiency with shorter key lengths. Thus, it is quite suitable for ECC to be applied in mobile Ad Hoc networks with low-power dissipation, low bandwidth and limited computing resources [13]. Base field of ECC is defined in finite prime fields or binary finite fields [15], [16], and there are no particular requirements of ECC base field in OR-CLAGKA protocol. This paper chooses a prime finite field F_p (p is a big prime number) as ECC base field in order to describe this protocol clearly.

Let the symbol $E(F_p)$ denote an elliptic curve E in Eq. (1) where $a, b \in F_p$, with the discriminant $4a^3 + 27b^2 \neq 0(modp)$. Supposed $P = (x, y), x, y \in F_p$ is the point satisfying Eq. (1), then the points like P on $E(F_p)$ together with an extra point O at infinity form elliptic curve point group, denoted as G. As for each point (x_p, y_p) on the elliptic curve, x_p is the x-coordinate of G and y_p is the y-coordinate of G.

$$y^2 = x^3 + ax + b(modp) \tag{1}$$

Let n be the order of G that is an additive cyclic group under the point addition '+' [15], [16] defined in elliptic curve point group. The addition of same points in G can be computed in Eq. (2), also defined as scalar multiplication of points on G.

$$kP = P + P + \cdots + P$$
$$(the\ number\ of\ P\ is\ k,\ and\ P \in G) \tag{2}$$

2.3 ECC Security Concepts

The security cornerstone of ECC is to solve Discrete Logarithm Problem based on elliptic curve point group. ECDLP is defined by elliptic curve $E(F_p)$ and point group G as follows: let $Q = kP(P, Q \in G)$ and figure out that positive integer k is less than n under the condition of the known P and Q. It is easier to compute Q through k and P, however, it is infeasible to calculate k through Q and P, which has already been proved.

The security of OR-CLAGKA protocol is not only based on the difficulties of ECDLP but also on the difficulties of CDH problem on ECC, namely, computing abP under given a generator P on G and (aP, bP) with unknown a, b selected randomly and uniformly from Z_n^*. The CDH assumption states that the probability of any polynomial-time algorithm to solve the CDH problem is negligible.

3. Proposed OR-CLAGKA Protocol

3.1 Initial Phase of Protocol

Considering some protocols based on CL-PKC [2]–[6], [8], [9], algorithms named as Setup,Partial-Private-Key-Extract, Set-Secret-Value, Set-Private-Key, and Set-Public-Key are usually defined in the initial phase, which have similar function but different implementation in different protocols. In addition, a Key Agreement Protocol is usually designed as the final part of each protocol. In these CL-PKC protocols, a trusted Key Generation Center (KGC) is needed to generate a public-private key pair, a shared broadcast channel is needed to distribute the public parameters, and a secret channel is needed to distribute partial private key.

OR-CLAGKA protocol adopts authenticated key agreement protocol with two parties involved proposed by He et al [3]. In the initial phase, a slight modification of the second part makes it compliant with group key agreement protocol. Certificateless authenticated group key agreement (CLGKA) protocol requires KGC to generate master key and protocol participants' part-private key. Let m as the number of group users, the users as set $\{M_0, M_1, M_2, \cdots, M_{m-1}\}$, the users corresponding identity ID as set $\{ID_0, ID_1, ID_2, \cdots, ID_{m-1}\}$, the corresponding private and public key pairs as set $\{(sk_0, pk_0), (sk_1, pk_1), \cdots, (sk_{m-1}, pk_{m-1})\}$.

Setup: This algorithm takes a security parameter l as input and returns system parameter and master key as the following:

i) KGC Chooses an l-bit prime p and determines the 5-tuple $\left\{F_p,\ E\left(F_p\right),\ G, P, n\right\}$, where G is the point group of elliptic curve $E\left(F_p\right)$ with n as the order of group and P as base point.

ii) KGC picks $s \in Z_n^*$ as the master private key and computes master public key as $P_{pub} = sP$.

iii) KGC selects a cryptographic secure hash function: $H_1 : \{0,1\} \to Z_n^*$.

iv) KGC publishes $\left\{F_p,\ E\left(F_p\right),\ G, P, P_{pub}, n, H_1\right\}$ as system parameters to all group users and secretly keeps the master private key s.

Set-Secret-Value: The user with identity ID_i picks at random $x_i \in Z_n^*$ to compute $P_i = x_i \cdot P$ and sets x_i as his secret value.

Partial-Private-Key-Extract: The algorithm sets master key, member identity ID_i, P_i on the $E\left(F_p\right)$ and system parameter as input and returns private key based on ID_i for each user.

i) KGC randomly chooses $r_i \in Z_n^*$ and calculates $R_i = r_i \cdot P$, $h_i = H_1\left(ID_i, R_i, P_i\right)$.

ii) KGC computes $s_i = r_i + h_i s \ mod\ n$ and issues $\{s_i, R_i\}$ to the users having identity ID_i through secure channel.

s_i can validate the partial private key by checking whether the equation $s_i \cdot P = R_i + h_i \cdot P_{pub}$ holds.

Set-Private-Key: The user with identity ID_i calculates $sk_i = x_i + s_i$ and takes $\{sk_i, x_i\}$ as its complete private key.

Set-Public-Key: The user with identity ID_i sets $pk_i = P_i + s_i \cdot P = x_i \cdot P + s_i \cdot P = (x_i + s_i) P = sk_i \cdot P$ and takes $\{pk_i, P_i\}$ as his public key and broadcast to all group users.

3.2 Execution Phase of Protocol

Similar to the algorithms described in [4], [7], [8], [12], [14], joining group protocol and leaving group protocol have been technically designed in the proposed algorithm. According to the topological structure feature of mobile Ad Hoc network, the running of OR-CLAGKA protocol needs to appoint a certain user as group controller (GC) which is usually close to the center of network topological structure. Set ID_0 as the identity of GC so as to describe the protocol clearly and without loss of generality.

Phase 1) All non-GC users send protocol data to GC;

Randomly select $a_i, b_i \in Z_n^*$ to compute $U_i = (a_i + b_i + x_i) \cdot pk_0$, $V_i = (b_i + x_i) \cdot P$, $d_i = (a_i + b_i) \cdot sk_i^{-1}$, $C_i = H_1\left(ID_i \parallel a_i \cdot P \parallel V_i \parallel d_i\right)$ with $(1 \le \mathrm{i} \le \mathrm{m} - 1)$ and issue protocol data $\{U_i, V_i, d_i, C_i\}$ to GC.

Phase 2) GC sends protocol data to each non-GC user;

When GC receives the data from M_i, he should verify the validity of the data firstly. Computing $W_i = sk_0^{-1} \cdot U_i - V_i$ and $C_i^{'} = H_1\left(ID_i \parallel W_i \parallel V_i \parallel d_i\right)$, if $C_i \neq C_i^{'}$, protocol operation fails, otherwise computing $V_i^{'} = sk_0^{-1} \cdot U_i - d_i \cdot pk_i$; on the premise of $V_i \neq O$(the infinity point) and $d_i \neq n$, if $V_i^{'} \neq P_i$, protocol operation fails, otherwise continuing protocol execution.

If $V_i^{'} = P_i$ and ID_i is valid, then M_i's identification is confirmed.

When protocol data of all non-GC users remain valid, GC randomly chooses $t_i \in Z_n^* \left(0 \le \mathrm{i} \le \mathrm{m} - 1\right)$ and sends data $\{L_i, T_i, CC_i\}$ to $M_i \left(1 \le \mathrm{i} \le \mathrm{m} - 1\right)$, where $L_i = t_i \cdot P$, $L_i^{'} = t_i \cdot W_i$ are points on the elliptic curve with its coordinate as $(L_{ix}^{'}, L_{iy}^{'})$. Let $L_{ix}^{''} = L_{ix}^{'} \ mod\ n$, if $L_{ix}^{''} = 0$, then we should randomly select t_i again; computing $Z_i = Tmp - W_i$, $Tmp = t_0 \cdot P + (\sum_{j=1}^{m-1} W_j)$ should be calculated only once and computing $T_i = L_{ix}^{''} \cdot Z_i$; computing $CC_i = H_1\left(ID_0 \parallel W_i \parallel L_i \parallel T_i\right)$.

GC calculates shared agreement key: $TK = t_0 \cdot P + \sum_{j=1}^{m-1} W_j$.

Phase 3) Each non-GC user receives protocol data from GC;

When M_i receives protocol data $\{L_i, T_i, CC_i\}$, he can compute $CC_i^{'} = H_1\left(ID_0 \parallel a_i \cdot P \parallel L_i \parallel T_i\right)$. If $CC_i \neq CC_i^{'}$, protocol execution fails, otherwise protocol execution continues.

Non-GC user $M_i \left(1 \le \mathrm{i} \le \mathrm{m} - 1\right)$ computes shared TK_i as follows: calculating $LL_i = a_i \cdot L_i$, a point on elliptic curve with its coordinate (LL_{ix}, LL_{iy}), and let $LL_{ix}^{'} = LL_{ix} \ mod\ n$; computing $T_i^{'} = LL_{ix}^{'\,-1} \cdot T_i$, then $TK_i = T_i^{'} + a_i \cdot P$. Thus $TK = TK_i \left(1 \le \mathrm{i} \le \mathrm{m} - 1\right)$ can be proved.

OR-CLAGKA protocol operation has been completed so far, and every protocol participants have finished one round receiving and issuing of protocol data, that is, one round protocol interaction generates shared group key. Meanwhile, group users should destroy all data randomly generated during protocol execution.

Leaving Group Protocol:

If GC leaves the group, we need to reappoint a new GC based on topology structure of mobile Ad Hoc networks and rerun OR-CLAGKA protocol.

If non-GC users leave the group, we only need to execute adjusted phase 2 and completely consistent phase 3. Supposed user M_j leaves the group, new group key is generated as follows:

GC randomly selects $t_{newi} \in Z_n^* \left(0 \le \mathrm{i} \le \mathrm{m} - 1, \mathrm{i} \neq \mathrm{j}\right)$ and issues $\{L_{newi}, T_{newi}, CC_{newi}\}$ to $M_i \left(1 \le \mathrm{i} \le \mathrm{m} - 1, \mathrm{i} \neq \mathrm{j}\right)$, the difference from original phase 2 is the method to calculate Z_i. Firstly, updating $Tmp_{new} = Tmp + t_{new0} \cdot P - W_j$ and computing $Z_{newi} = Tmp_{new} - W_i$, then the generation of $\{L_{newi}, T_{newi}, CC_{newi}\}$ adopts the same operation steps in original phase 2.

Joining Group Protocol:

If new user M_m with his identity ID_m needs to join group, M_m has to obtain system security parameters and generate public and private key pairing $\{sk_m, pk_m\}$. If the new user joined and left the group once, there is no need to generate new public and private key pairing.

Supposed new user M_m wants to join the group without loss of generality, M_m executes steps in phase 1 and randomly chooses $a_m, b_m \in Z_n^*$ to compute $U_m = (a_m + b_m + x_m) \cdot pk_0$, $V_m = (b_m + x_m) \cdot P$, $d_m = (a_m + b_m) \cdot sk_m^{-1}$, $C_m = H_1\left(ID_m \parallel a_m \cdot P \parallel V_m \parallel d_m\right)$ and send protocol data $\{U_m, V_m, d_m, C_m\}$ to GC.

GC executes the proof procedure in phase 2. If those two verifications pass, we only need to execute similarly adjusted phase 2 and completely consistent phase 3 when non-

GC users leave the group.

3.3 Correctness Proof of Protocol Execution

Assuming the protocol data are transferred correctly in mobile Ad Hoc networks during OR-CLAGKA protocol execution, proof can be conducted as follows:

i) Due to $W_i = sk_0^{-1} \cdot U_i - V_i = (a_i + b_i + x_i) \cdot sk_0^{-1} \cdot sk_0 \cdot P - (b_i + x_i) \cdot P = a_i \cdot P$ and obviously $C_i^{'} = H_1(ID_i \parallel W_i \parallel V_i \parallel d_i) = H_1(ID_i \parallel a_i \cdot P \parallel V_i \parallel d_i) = C_i$, thus $C_i = C_i^{'}$ is proved.

ii) Due to

$$\begin{aligned} V_i^{'} &= sk_0^{-1} \cdot U_i - d_i \cdot pk_i \\ &= (a_i + b_i + x_i) \cdot P - (a_i + b_i) \cdot sk_i^{-1} \cdot sk_i \cdot P \\ &= x_i \cdot P \\ &= P_i \end{aligned}$$

thus $V_i^{'} = P_i$ is proved.

iii) Due to $W_i = a_i \cdot P$ and then $CC_i = CC_i^{'}$.

iv) Due to $LL_i = a_i \cdot L_i = a_i \cdot t_i \cdot P = t_i \cdot W_i = L_i^{'}$, thus $LL_{ix}^{'} = L_{ix}^{''}$ is obtained; then because of $T_i^{'} = LL_{ix}^{'\,-1} \cdot T_i = LL_{ix}^{'\,-1} \cdot L_{ix}^{''} \cdot Z_i = Z_i$, therefore $TK_i = T_i^{'} + a_i \cdot P = Z_i + a_i \cdot P = Z_i + W_i = t_0 \cdot P + \sum_{j=1}^{m-1} W_j \, (j \neq i) + W_i = TK$ is proved.

QED, OR-CLAGKA protocol guarantees every group user can obtain the completely consistent group key if group users execute the protocol in sequence.

4. Protocol Analysis

4.1 Security Analysis

Complete authentication mechanism: When OR-CLAGKA implements phase 2, GC(identity is M_0) will authenticate the protocol data sent by $M_i(1 \leq \mathrm{i} \leq \mathrm{m} - 1)$. So GC can ensure that the protocol data is sent by authentic M_i and has not been tampered. Only GC can complete the identity authentication in this phase.

Firstly, $C_i^{'} = C_i$ in Sect. 3.3 is true. However, it can not prove that U_i, V_i, d_i and C_i are sent by M_i, because it is possible to be forged. There is an attack situation, in which the attacker not know sk_i, private key of M_i, but he also can produce U_i, V_i and C_i. Let $U_i = \alpha \cdot pk_0, V_i = \beta \cdot P, (\alpha, \beta) \in Z_n^*$ as the random number, $C_i = H_1(ID_i \parallel (\alpha - \beta) \cdot P \parallel V_i \parallel d_i)$, obviously $W_i = sk_0^{-1} \cdot U_i - V_i = \alpha \cdot P - \beta \cdot P = (\alpha - \beta) \cdot P$, which makes $C_i = C_i^{'}$ true. However, $P_i = V_i^{'}$ will not be true unless the attacker knows $\{x_i, sk_i\}$ and makes $P_i = U_i \cdot sk_0{}^{-1} - d_i \cdot pk_i$ true through selecting appropriate U_i and d_i. Now, the attacker needs to crack CDH to get x_i and sk_i, secret value and private key of M_i, which is computationally impossible. So, only M_i can generate valid U_i, V_i and d_i, and hash value C_i including the identity information of member M_i. Through verifying $C_i = C_i^{'}$ and $P_i = V_i^{'}$, as well as the data integrity checking provided by C_i, GC can identify the member M_i. Only GC has private key sk_0, so only GC can identify member M_i.

Secondly, GC gets secret data a_iP (which is also used as important authentication data in phase 3) by calculating W_i. If the attacker wants to get a_iP, he needs to get sk_0, the private key of GC, and then he has to crack CDH, which is infeasible in computation.

At last, in phase 3 of the protocol implementation, through verifying $CC_i = CC_i^{'}$, non-GC member M_i verifies that the protocol data sent by GC is integral and the protocol data must be sent by GC, because only GC can calculate the secret data a_iP.

Group key secrecy: According to the analysis above, passive attacker can not crack $a_iP(1 \leq \mathrm{i} \leq \mathrm{m} - 1)$ in the process of OR-CLAGKA protocol implementation. When GC works out all a_iP, he generates $t_0 \in Z_n^*$ randomly and gets $t_0 \cdot P$ by calculating, and finally gets group secret key $TK = t_0 \cdot P + \sum_{j=1}^{m-1} W_j$. So, even passive attacker captures all data sent to GC, he can not crack TK because he lacks the result of $t_0 \cdot P$ and $a_iP(1 \leq \mathrm{i} \leq \mathrm{m} - 1)$.

In addition, when intercepting all protocol data that GC sends to non-GC members, passive attacker can not run cryptanalysis associated with protocol data, because each group of $\{L_i, T_i, CC_i\}$ is related to a random number $t_i \in Z_n^*$, which makes every group of protocol data mutually independent. When one group protocol data $\{L_i, T_i, CC_i\}$ is attacked, if the attacker wants to break Z_i and avoids CDH, he must crack $L_{ix}^{''}$ because of $T_i = L_{ix}^{''} \cdot Z_i$, and then he need to get $t_i \cdot W_i$, however, $t_i \in Z_n^*$ is chosen by GC randomly, when $L_i = t_i \cdot P$ is known, because of CDH problem, the attacker must turn to break W_i. According to the previous analysis, $W_i = a_i \cdot P$ is sent secretly so the attacker can not get it, so he can not work out TK. Therefore, it is infeasible to break group secret key by capturing protocol data in computation.

Weak forward secrecy: Let us analyze in two conditions. When GC leaves the group, OR-CLAGKA protocol completely redoes, and it chooses new GC, every group member chooses a new random number, and the old GC does not know the group secret key newly generated.

When non-GC member leaves the group, GC reselects the random number $t_{newi} \in Z_n^* \, (0 \leq \mathrm{i} \leq \mathrm{m} - 1, \mathrm{i} \neq \mathrm{j})$, and generates new Z_i. $\{L_{newi}, T_{newi}, CC_{newi}\}$ is sent to other group members. As passive attacker, the non-GC member who left can not break new TK_i.

Forward secrecy: When passive attacker gets a subset $\{\mathrm{TK}_{pre0}, \mathrm{TK}_{pre1}, \cdots, \mathrm{TK}_{pre\ n}\}$ of group secret key which is used before, he can not deduce the subsequent group secret key according to this. Because every $\mathrm{TK}_{pre\ j}$ is equivalent to $\sum_{j=1}^{x} t_j \cdot P$, $t_j \in Z_n^*$ as a random number, x as a integer is not less than the number of group members. Every time the group secret key is generated, at least one t_j is updated, and elements in the subset $\{\mathrm{TK}_{pre0}, \mathrm{TK}_{pre1}, \cdots, \mathrm{TK}_{pre\ n}\}$ of group secret key are mutually independent.

Weak backward secrecy: According to the new member **Joining Group Protocol**, new member sends protocol data $\{U_m, V_m, d_m, C_m\}$ to GC, who updates $W_m = a_m \cdot P$. Meanwhile, according to the analysis of **Weak forward secrecy**, GC reselects the random number when implementing phase 2 and phase 3, so as passive attacker, new member can not break former group secret key.

Backward secrecy: Same as the analysis of **Forward secrecy**, the passive attacker can not deduce former group secret key because of the independence of the subset of group secret key.

Key independence: The independence of group secret key has been analyzed in **Forward secrecy**. According to the literature [17], when it proves that the protocol has forward secrecy and backward secrecy, the independence of the group secret key is available.

Perfect forward secrecy: At last, under the circumstance that all members' private keys are revealed for a long time, let us analyze the effect that OR-CLAGKA protocol brings to the security of former group secret key. Because $W_i = sk_0^{-1} \cdot U_i - V_i$, if sk_0 is revealed, the W_i $(0 \leq \mathrm{i} \leq \mathrm{m} - 1)$ will be cracked. However, as to $TK = t_0 \cdot P + \sum_{j=1}^{m-1} W_j$ generated by GC, the attacker will not know TK because $t_0 \in Z_n^*$ is selected randomly.

When GC sends protocol data $\{L_i, T_i, CC_i\}$, it does not involves any group member's private secret key. So, the reveal of all group members' private keys only affects the reveal of W_i. Because $L_i = t_i \cdot P$, $L_i^{'} = t_i \cdot W_i$, $t_i \in Z_n^*$ is generated randomly, attacker can not get $L_i^{'}$ when only knowing W_i and not knowing a_i. Cracking a_i through $W_i = a_i \cdot P$ will face the CDH problem, which is infeasible in calculation. So, even knowing all group members' private keys, the attacker can not get $L_i^{'}$, so he can not get Z_i or $T_i^{'}$ to crack the group secret key.

That is to say, the reveal of group members' private keys at most results in the impracticability of identity authentication, but it will not result in the reveal of any current or former group secret key.

4.2 Performance Comparison Analysis

This section focuses on analyzing complexity and computation overheads of OR-CLAGKA protocol compared with some other relative protocols. Set m as the number of group users so as to conduct performance comparison conveniently.

Seeing that the security hash function used in protocol has received standard algorithm support in cryptology practice, such as SHA-1 Hash Algorithm, the running speed of software and hardware implementation version is rather fast and the computation overheads can be neglected in contrast with scalar multiplications of elliptic curve points. OR-CLAGKA protocol can finish group key agreement in one round, where each non-GC user needs 5 times scalar multiplication and a few point additions and base field multiplications. Compared with scalar multiplication, the point additions and base field multiplications can be neglected, that is, each non-GC user needs to finish $5m - 5$ scalar multiplication in total. In a similar way, GC needs to finish $5m - 4$ scalar multiplication. Therefore, we need $10m - 9$ scalar multiplication to finish OR-CLAGKA protocol.

Table 1 Performance comparison of related protocols

Protocol	Number of rounds	Pairings	Scalar Multiplications
our proposed protocol	1	0	$10m - 9$
Geng et al.'s protocol [6][1]	2	$4m$	0
Kumar A. et al.'s protocol [5][2]	2	0	$10m$
Kumar A. et al.'s protocol [4]	3	0	$14m$

Notes: 1. One-time bilinear pairings computation equals to 20 times scalar multiplications, thus, Geng et al.'s protocol [6] equals to 80*m times scalar multiplications.*
2. Kumar A. et al.'s protocol [5] needs to conduct a high-cost signature operation to protocol data.

When new users join or non-GC users leave the group, the computing times of scalar multiplication to generate new group key decreases to $5m + 6$ and $5m - 9$ respectively.

We compare those certificateless authenticated group key agreement protocols regarding the number of rounds, bilinear pairings, and scalar multiplications as shown in Table 1.

From Table 1, the proposed protocol is the best regarding to computation cost and execution efficiency.

Finally, we analyze the demand of the storage space needed by a single Ad Hoc user. According to the security strength of commercial ECC algorithm [18], a 32-bytes space is needed to store an element of base field, and a 64-bytes space is needed to store the value of ECC curve coordinates. Since GC and non-GC users each only need to store the value of m times of $\{pk_i, P_i\}$ and less than 10 intermediate variables (when variables $\{a_i, b_i, U_i, L_i, T_i, \cdots\}$ are all considered as intermediate variables, the memory used by old variables can be recycled by new variables), the total storage space required is no more than $128m + 64 \times 10$ bytes for one user, those are 125.6 Kbytes for 1000 users. Twenty years ago, 32 Mbytes SDRAM memory has been made [19]. It is common that Hand held smart devices is equipped with more than 32Mbytes SDRAM, which can be used as Ad hoc terminals. Therefore, the maximum number of group users in this protocol mainly depends on processing ability of smart devices.

5. Conclusions

Based on the hardness of ECDLP and the related CDH problem, OR-CLAGKA protocol stemming from CL-PKC structure achieve effective implementation performance. The protocol itself has high execution efficiency and smaller protocol data size in addition to one-round group key agreement. Protocol data increases in a linear way along with the increase of group users, which is suitable for low bandwidth applications in Ad Hoc networks. Meanwhile, the protocol has no particular additional requirements for network topology and fit for hierarchical or divisional deployment. Furthermore, hierarchical or divisional deployment can reduce GC's computation burden remarkably when the number of group users is very big.

In addition, when users join or leave the group frequently, OR-CLAGKA protocol effectively reduces the number of protocol execution steps and protocol data throughput without vitiating secrecy, which makes it more suitable for limited computing resources and bandwidth applications in mobile Ad Hoc networks.

Acknowledgments

This work is supported by the National Key Basic Research Program (973 program) through project 2012CB315905, by the Fundamental Research Funds for the Central Universities through project No. YWF-15-GJSYS-059.

References

[1] A. Shamir, Identity-Based Cryptosystems and Signature Schemes, Advances in Cryptology, pp.47–53, Springer Berlin Heidelberg, 1984.

[2] S.S. Al-Riyami and K.G. Paterson, "Certificateless public key cryptography," Proceedings of ASIACRYPT 2003, LNCS 2894, Springer-Verlag, pp.452–473, 2003.

[3] D. He and Y. Chen, "An efficient certificateless authenticated key agreement protocol without bilinear pairings," Mathematical & Computer Modelling, vol.54, pp.3143–3152, 2011.

[4] A. Kumar and S. Tripathi, "A Pairing Free Anonymous Certificateless Group Key Agreement Protocol for Dynamic Group," Wireless Personal Communications, vol.82, no.2, pp.1027–1045, 2015.

[5] A. Kumar, S. Tripathi, and P. Jaiswal, "A Pairing Free Certificateless Group Key Agreement Protocol with Constant Round," Smart Innovation Systems & Technologies, vol.28, pp.341–349, 2014.

[6] M. Geng, F. Zhang, and M. Gao, "A Secure Certificateless Authenticated Group Key Agreement Protocol," 2009 International Conference on Multimedia Information Networking and Security, MINES '09, pp.342–346, 2009.

[7] E.-J. Lee, S.-E. Lee, and K.-Y. Yoo, "A Certificateless Authenticated Group Key Agreement Protocol Providing Forward Secrecy," Proc. 2008 International Symposium on Ubiquitous Multimedia Computing, IEEE Computer Society, pp.124–129, 2008.

[8] S. Heo, Z. Kim, and K. Kim, "Certificateless Authenticated Group Key Agreement Protocol for Dynamic Groups," IEEE Global Telecommunications Conference, GLOBECOM '07, pp.464–468, 2007.

[9] J. Teng and C. Wu, "A provable authenticated certificateless group key agreement with constant rounds," Journal of Communications & Networks, vol.14, no.1, pp.104–110, 2012.

[10] S. Wang, Z. Cao, and L. Wang, "Efficient certificateless authenticated key agreement protocol from pairings," Wuhan University Journal of Natural Sciences, vol.11, no.5, pp.1278–1282, 2006.

[11] L. Chen, Z. Cheng, and N.P. Smart, "Identity-based Key Agreement Protocols from Pairings," Int. J. Inf. Secur., vol.6, no.4, pp.213–241, 2007.

[12] S. Zheng, D. Manz, and J. Alves-Foss, "A communication-computation efficient group key algorithm for large and dynamic groups," Computer Networks the International Journal of Computer & Telecommunications Networking, vol.51, no.1, pp.69–93, 2007.

[13] Y. Wang, B. Ramamurthy, and X. Zou, "The Performance of Elliptic Curve Based Group Diffie-Hellman Protocols for Secure Group Communication over Ad Hoc Networks," 2006 IEEE International Conference on Communications, ICC '06, pp.2243–2248, 2006.

[14] G. Xiaozhuo, X. Taizhong, Z. Weihua, and W. Yongming, "A Pairing-Free Certificateless Authenticated Group Key Agreement Protocol," High PERFORMANCE Computing and Communications, 2014 IEEE Intl Symp on Cyberspace Safety and Security, 2014 IEEE Intl Conf on Embedded Software and Syst. IEEE, pp.510–513, 2014.

[15] Elliptic Curve Cryptography, SECG Std. SEC1, 2009, available from http://www.secg.org/sec1-v2.pdf

[16] Recommended Elliptic Curve Domain Parameters, SECG Std. SEC2, 2010, available from http://www.secg.org/sec2-v2.pdf

[17] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," ACM Trans. Inf. Syst. Secur., vol.7, no.1, pp.60–96, 2004.

[18] State Cryptography Administration. Public key cryptographic algorithm SM2 based on elliptic curve[EB/OL]. [2010-12-10]. http://www.oscca.gov.cn/UpFile/2010122214822692.pdf

[19] T. Saeki, Y. Nakaoka, M. Fujita, A. Tanaka, K. Nagata, K. Sakakibara, T. Matano, Y. Hoshino, K. Miyano, S. Isa, S. Nakazawa, E. Kakehashi, J.M. Drynan, M. Komuro, T. Fukase, H. Iwasaki, M. Takenaka, J. Sekine, M. Igeta, N. Nakanishi, T. Itani, I. Yoshida, K. Yoshino, S. Hashimoto, T. Yoshii, M. Ichinose, T. Imura, M. Uziie, S. Kikuchi, K. Koyama, Y. Fukuzo, and T. Okuda, "A 2.5-ns clock access, 250-MHz, 256-Mb SDRAM with synchronous mirror delay," IEEE J. Solid-State Circuits, vol.31, no.11, pp.1656–1668, 1996.

Dongxu Cheng received M.S. degree in Communication and information system from PLA University of Science and Technology, China, 2004. He is currently a Ph.D. candidate in department of Electronic and Information Engineering in Beihang University, Beijing, China. His major research interests include security of Communication networks and trusted computing.

Jianwei Liu is currently a full professor and party secretary in the Department of Electronic and Information Engineering, Beihang University. He received Ph.D., Communication & Electronic System Department, Xidian University, 1998. He is a senior member of Chinese Institute of Electronics, a director of Chinese Association for Cryptologic Research. Currently his more than 100 papers, 1 translation work, and 5 monographs and textbooks have been published. His research interests include wireless communication network, cryptography, information security, communication network security, channel coding, and modulation technology.

Zhenyu Guan received Ph.D. degree in Electrical and Electronic Engineering department from Imperial College London, UK, 2013. He is now a lecture in department of Electronic Information Engineering in Beihang University, Beijing, China. His research interests include cryptography, security of cyber space and security of information network.

Tao Shang is an Associate Professor of College of Electronic and Information Engineering at Beihang University, Beijing, China. He received his PHD in Computer Science from Kochi University of Technology, Japan, in 2006. His current research interests include network security, network coding, and quantum cryptography.