TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 10, October 2014, pp. 7471 ~ 7477
DOI: 10.11591/telkomnika.v12i8.5582

Perfect Forward Secure ID-based Key Agreement Protocol in Group Communication

Pengshuai Qiao
School of Environmental and Municipal Engineering
North China University of Water Resources and Electric Power
email: pengshuaiqiao@163.com

Abstract

Several identity-based key agreement protocols using bilinear pairing have been proposed in recent years and none of them has achieved all required security properties. In this paper, we firstly propose an ID-based one round authenticated group key agreement protocol with bilinear pairings, where all participants can generate the group session key in one round. Based on the intractability of elliptic curve discrete logarithm problem, every user's private key can be proved to be secure. Also an extended version of one round authenticated group key agreement protocol is given, it provide perfect forward secrecy and avoid key escrow by the Key Generation Center. Finally, a comprehensive security analysis and a comprehensive security analysis are provided. By comparing with other protocols, the proposed protocol requires lower computation cost.

Keywords*: key agreement protocol, perfect forward secrecy*

1. Introduction

In recent years, collaborative and group-oriented applications and protocols have gained popularity. These applications typically involve communication over open networks. One of the important requirements is security. A key agreement which provides mutual key authentication between parties is called an authenticated group key agreement (AGKA). Key establishment protocols are one of the most important cryptographic primitives that have been used in our society. In 1976, the first unauthenticated key agreement protocol based on asymmetric cryptographic techniques was proposed by Diffie and Hellman [1]. It can assure the security of communication between the two users. However, it dose not authenticate users, hence suffers the "man-in-the-middle" attack. In 1984, Shamir [2] proposed the idea of ID-based cryptosystem where the identity information of a user functions as his public key. A few key agreement protocols have been developed based on Diffie-Hellman and Shamir's key setup idea. In one of breakthroughs in key agreement, Joux [3] proposed a three party single round key agreement protocol using pairings on elliptic curve. This was the first positive application of bilinear pairings in cryptography. Joux et al. applies the pairing technique and achieves key agreement among three parties in an astonishingly simple way. He names his protocol "tripartite Diffie-Hellman". Again, Joux's original protocol works in the Weil pairing and hence is less convenient for a real application use. Here we introduce a simplified version using the modified Weil pairing.

Since Boneh and Franklin's pioneering work [4] on the ID-based encryption (IBE) system in 2001, several papers have attempted to establish ID-based authenticated key agreement protocol (ID-AGKA). Choi et al. [5] and Du et al. [6] proposed two ID-AGKA protocols from bilinear pairings and BD [7] schemes. However, Zhang and Chen [8] showed an impersonation attack on these two protocols. To prevent such an attack, they suggest adding a time parameter to the message being signed. However, SHIM [9] showed that the protocol is still insecure against insider colluding attacks. In 2006, Lin et al. [10] proposed a multiparty key agreement protocol, but their protocol has disadvantages in number of rounds, pairing-computation and communication bandwidth. Zhou et al. proposed a one-to-many mapping shared key agreement, which is based on one-to-many encryption mechanism model, but the round number of their scheme is two [11].

Received January 5, 2014; Revised July 24, 2014; Accepted August 18, 2014

To realize group key agreement and extend Joux et al.'s protocol, Barua et al. [12]'s first proposed a three-group and a two-group Diffie-Hellman key agreement protocol. After that, many protocols were proposed in [13-17]. Abdel Alim Kamal proposed an attack on Piao et al.'s scheme which describes a polynomial-based key management scheme for secure intra-group and inter-group communication [18]. Marimuthu Rajaram and Thilagavathy Dorairaj Suresh proposed an interval-based key agreement approach which adpots re-keying [19]. To decrease the number of rounds and make AGKA more efficient, Shi et al. proposed one round ID-based AGKA protocol with bilinear parings [20], which can generate the secret session key in one round. Shi et al.'s protocol just requires one round and less transmitted data, so it has a good efficient. However, as illustrated in their literature, in their protocol if two or more than two users' long-term private keys are compromised, the adversary can compute the previous session key, so their protocol cannot provide perfect forward secrecy. Similarly, their protocol also cannot prevent KGC from escrowing the established session keys. Based on Shi et al.'s work, we first propose an ID-based one round authenticated group key agreement protocol which satisfies the required security attributes and provide lower computation cost. The proposed paper's section structure is organized as: **Introduction - Security properties - Proposed ID-AGKA - Security analysis –Efficiency analysis– Conclusion.**

2. Security properties

An authenticated group key agreement protocol is desired to have the following common security properties [21, 22]:

1) Implicit Key Authentication: An n-party key agreement protocol provides implicit key authentication if each member in the set of protocol parties is assured that no party outside the set can learn the group secret key.

2) Perfect Forward Secrecy: We say that a protocol has partial forward secrecy if one or more but not all the entities' long-term keys can be corrupted without compromising previously established session keys, and we say that a protocol has perfect forward secrecy if the long-term keys of all the entities involved may be corrupted without compromising any session key previously established by these entities.

3) Known Session Key Security: Resistance to known session key security is the property that each run produces a different session key and compromise of past session keys does not allow compromise of future session keys.

4) Key-Compromise Impersonation: When A's private key is compromised, it may be desirable that this event does not enable an adversary to impersonate other entities to A.

5) Unknown Key-Share: In an unknown key-share attack, an adversary convinces a group of entities that they share a key with the adversary, whereas in fact the key is shared between the group and another party.

6) No Key Control: It should not be possible for any of the participants or an adversary to force the session key to a pre-selected value or predict the value of the session key.

3. Proposed ID-AGKA

3.1. System Setup

We take G_1 to be a cyclic elliptic curve group with large prime order q and the bilinear map $e : G_1 \times G_1 \rightarrow G_2$. The key generation center (KGC) generates the system parameters $\{q, G_1, G_2, e, P, H_1, H_2\}$. s is randomly chosen from Z_q^* as the KGC's private key. P_{pub} ($=sP$) is the KGC's public key. Each user U_i has an identity $ID_i \in \{0, 1\}^*$ and long-term public key $Q_i = H_1(ID_i)P + P_{pub}$. U_i submits his ID_i to the KGC and KGC sends back the long-term private key $S_i = (H_1(ID_i)+s)^{-1}P$ to user U_i securely.

3.2. The ID-AGKA

The protocol one round authenticated group key agreement includes three phases: data transmission phase, verification phase and key computation phase. It is illustrated as follows:

1) To two users U_i, U_j ($1 \le i, j \le m, i \ne j$), U_i picks a random integer $r_i \in Z_q^*$ as his ephemeral private key. Then he computes $T_i = r_i^{-1}Q_i$ and sends T_i to U_j. Upon the receipt of T_i, U_j also picks a random integer $r_j \in Z_q^*$ as his ephemeral private key, computes $T_{j,i} = r_j r_i^{-1}Q_i$ and $T_j = r_j^{-1}Q_j$

respectively, and then sends the data $\{T_{j,i}, T_j\}$ to U_i. Finally, U_i computes $T_{i,j}= r_i r_j^{-1} Q_j$ and returns it to U_j.

2) To verify the validity of received data, U_i computes $e(T_{j,i}, r_i T_j)$ and compares it with $e(Q_i, Q_j)$. If they are not the same value, U_i stops the session. Otherwise, U_i is sure that the received messages are from U_j. Similarly, U_j computes $e(T_{i,j}, r_j T_i)$ and compares with $e(Q_i, Q_j)$.If they are not same, U_j stops the session. Otherwise, U_j is sure that the received messages are valid.

3) Upon the receipt of $T_{1,i}, T_{2,i}, \ldots, T_{i-1,i}, T_{i+1,i}, \ldots, T_{m,i}$ from other users, user U_i computes the secret session key:

$$K_i= H_2 (e(Q_i + \sum_{j=1, j\neq i}^{m} T_{j,i}, r_i S_i))= H_2 (e_{(P,P)}^{(r_1+r_2+\ldots+r_m)}) \quad (1)$$

Each user performs the procedure above, thus all users in the group can get the same session key as follows:

$$K = H_2 (e_{(P,P)}^{(r_1+r_2+\ldots+r_m)}) \quad (2)$$

At round 1, we assume that n users $U_1^{(1)}, U_2^{(1)}, \ldots, U_n^{(1)}$ (n≥2) want to share a common session secret key. Each $U_i^{(1)}$ chooses a random number $r_i^{(1)}$ as his ephemeral private key. We take an integer number m as the based number for groups division and partition the n users into $\left\lceil \frac{n}{m} \right\rceil$ subgroups, for n≥m≥2. The subgroup j, for j=1, 2, …, $\left\lceil \frac{n}{m} \right\rceil - 1$, has m users and computes the common session sub-key $K_{1j}= H_2(e_{(P,P)}^{\sum_{i=1}^{m} r_{ji}^{(1)}})$ by the proposed protocol. The last subgroup $\left\lceil \frac{n}{m} \right\rceil$ has n (mod m) users. If the value of n (mod m) is not equal to one, users in the last subgroup also use protocol to generate the common session sub-key K_{1j}, where $j=\left\lceil \frac{n}{m} \right\rceil$. If the value of n (mod m) is equal to one, it means the last subgroup only contains one user, and we take this user's ephemeral private key as the last sub-key.

At the next round, each subgroup $U_j^{(2)}$, for j=1, 2, …, $\left\lceil \frac{n}{m} \right\rceil$, takes K_{1j} as his ephemeral private keys $r_j^{(2)}$ respectively and broadcasts $U_a^{(1)}$'s public key as the subgroup public value, here $U_a^{(1)}$ is a member of subgroup $U_j^{(2)}$ and a≡1(mod m). We partition these $\left\lceil \frac{n}{m} \right\rceil$ subgroups into $\left\lceil \frac{n}{m^2} \right\rceil$ subgroups and use the same procedure as the round 1. The following rounds work as above. And the protocol does not stop until the number of subgroups is one.

4. Security Analysis

4.1. Implicit Key Authentication

Implicit key authentication to a user A implies that only the users with whom A wants to agree upon a common key may be able to compute a particular key. This is an ideal property for secure group communication since it gets rid of the need for a separate authentication mechanism key sharing and can withstand the man-in-the-middle attacks. In our protocol the secret session key is computed by each user's long-term private key and ephemeral private key. Therefore, in any run of the protocol, each user assures that no other partner except for the intended one who has his own long-term private key and the value of ephemeral private key can learn the secure session key. If an attacker wants to impersonate all other users to the user U_i (1≤i≤m), the attacker just selects (m-1) ephemeral private keys r_j' (1≤j≤m, j≠i) and sends $T_{j,i}'= r_i^{-1} r_j' Q_i$ (1≤j≤m, j≠i) to the user U_i. Figure 1 shows the impersonation attack.

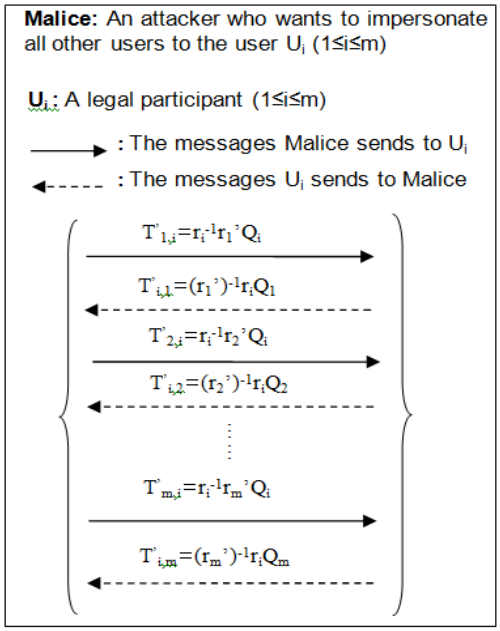

Figure 1. Impersonation Attack on Protocol OR-AGKA

However, in Figure 1, the attacker Malice cannot compute the final secret session key without the knowledge of other users' long-term private keys.

4.2. Key-Compromise Impersonation

Key-compromise impersonation states that the attacker who has compromised the long-term private key of user A can not only impersonate A but also impersonate the other users to A. In our protocol OR-AGKA, suppose that an adversary has got the long-term private key of a certain user U_i ($1 \le i \le m$), he can impersonate U_i. However, if he wants to masquerade as user U_j ($1 \le j \le m$), he can choose an ephemeral private key r_j' and send $T_{j',i}$ to U_i. This attack is shown in Figure 2.

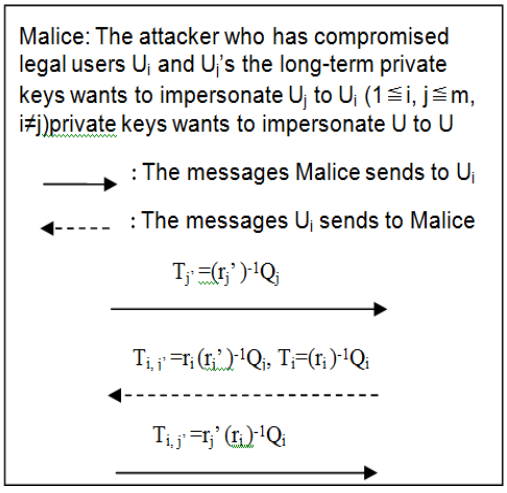

Figure 2. Key-compromise Impersonation Attack on OR-AGKA

But without U_i's ephemeral private key r_i, the adversary cannot computer the K_i. Meanwhile, upon receiving $T_{t,\ j}$ ($1 \le t \le m$, $t \neq j$) from other partners, the adversary still cannot

computer K_j without user U_j's long-term private key S_j and ephemeral private key r_j. Therefore even the adversary has got the long-term private key of a certain user, he still cannot impersonate as other users.

4.3. Perfect Forward Secrecy

If the long-term private keys of some participants are compromised, the secrecy of previous session keys should not be affected. And we say that a protocol has partial forward secrecy if compromise of the long-term keys of one or more but not all the participants does not compromise previously established session keys, and we say that a protocol has perfect forward secrecy if compromise of the long-term private keys of all the participant does not compromise any session key previously established by these participants. And KGC forward secrecy is another security issue. If at any run the KGC's private key is compromised, it does not compromise the previously established session keys [13]. In our protocol, the compromise of the entire partners' long-term private key or KGC's private key gives no help about the session key, since the session key is computed not only from long-term key but also from users' ephemeral private keys. By this feature, our protocol can provide perfect forward secrecy and KGC forward secrecy.

4.4. Known Session Key Security

Each run of the protocol should result in a unique secret session key. The compromise of one session key should not compromise other session keys and the knowledge of previous session keys do not allow deduction of future session keys. Because in our protocol, the final session key comprises every partner's ephemeral private key r_i ($1 \le i \le m$), it is unique. It's impossible for adversary to compute the current session key from the compromised session keys.

4.5. Unknown Key-shared Resistance

Unknown key-shared means user A shares a key with a different party user C than intended user B and A does not know it. To our protocol, at least it is required to know two users' long-term private keys to initial unknown key-shared attack. However, it's difficult for the adversary except for KGC to have more than two users' long-term private keys at the same time.

4.6. No Key Control

No key control means no any participant in the group can influence and control the outcome of the secret session key. Because every run of our protocol, the secret session key is determined by all users in the group, and no one can control or pre-determine the session key.

5. Efficiency Analysis

As illustrated in Barua et al.'s literature [6], the efficiency of AGKA protocols mainly involves the communication and computation costs. In each round, a user may have to transmit data to some or all the other users. Additionally, each user has to perform some operations like scalar multiplications, pairing computations. Communication overhead is affected by the number of rounds, total group element sent, total messages exchanged. And computation costs include total of pairing computation, total of scalar multiplications.

In this section, we use notations as follows:

- R(n): The total number of rounds for n users
- S_i: The number of scalar multiplications in round i
- P_i: The number of pairing-computations in round i
- B_i: The number of messages transmitted in round i

In protocol OR-AGKA, m users can finish key agreement in one round, and the efficiency is: $S_1 = m^2$, $P_1 = 3m$, $B_1 = 2C_m^1 C_{m-1}^1$. For n users to generate a common session key, in round i, if we take an integer number m as the based number for groups division, N_i subgroups will be divided into $\left\lceil \frac{N_i}{m} \right\rceil$ new subgroups, R(n) will be $\lceil \log_m n \rceil$ ($2 \le m \le n$).

In the cases which have the same total number of rounds, when every round $N_i \pmod m = 0$, the cost for generating the common session key will be maximum. The computational overhead of proposed protocol is summarized and compared with other protocols in Table 1. As shown in Table 1, even if our protocol needs more rounds, it is possible to provide lower computation cost if we choose an appropriate m.

Table 1. Comparison with other Protocols (2≤m≤n)

Schemes	R	S	P	B
[12]	$\lceil \log_3 n \rceil$	$< 5(n-1)$	$\leq 9(n-1)$	$\leq 5n\lceil \log_3 n \rceil + 3$
[20]	1	n^2	n	$n(n-1)$
[6]	2	n(n+5	4n	3(n-1
[10]	2	n	2n	2n
[19]	1	n	0	2n
EOR-AGKA	$\lceil \log_m n \rceil$	$\leq (n-1)m^2/(m-1)$	$\leq 3(n-1)m./(m-1)$	$\leq 2m(n-1)$

R: Total No. of rounds.
S: Total No. of scalar multiplications.
P: Total No. of pair-computations.
B: Total No. of transmitted messages.

6. Conclusion

In this paper, a secure, efficient and flexible ID-based one round authenticated group key agreement protocol using bilinear pairings is proposed. The proposed protocol focuses on round, mutual authentication, bandwidth efficiency and provides perfect forward secrecy. After security analysis and performance analysis, it shows that the proposed scheme provides strong security and lower computation cost than previously known AGKA protocols. In the future scope, the comparison of schemes was given by two questions: which scheme is suitable for different scenario and to which degree these schemes will impact the systems' performance consumption.

References

[1] W Diffie, M Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*. 1976; 22(6): 644-645.
[2] A Shamir. Identity-based cryptosystem and signature schemes. *Lecture Notes in Computer Science*. 1984; 196: 47-53.
[3] A Joux. A one round protocol for tripartite Diffie-Hellman. *Lecture Notes in Computer Science*. 2000; 1838: 385-394.
[4] D Boneh, M Franklin. Identity-based encryption from the Weil paring. *Lecture Notes in Computer Science.* 2001; 2139: 213~229.
[5] K Choi, J Hwang, D Lee. Efficient ID-based group key agreement with bilinear maps. *Lecture Notes in Computer Science*. 2004; 2947: 130-144.
[6] X Du, Y Wang, J Ge, Y Wang. ID-based Authenticated Two Round Multi-Party Key Agreement. *Cryptology ePrint Archive*. 2003; Report 2003/247.
[7] M Burmester, Y Desmedt. A secure and efficient conference key distribution system. *Lecture Notes in Computer Science*. 1995; 950: 275-286.
[8] FG Zhang, XF Chen. Attack on Two ID-based Authenticated Group Key Agreement Schemes. *Cryptology ePrint Archive*. 2003; Report 2003/259.
[9] Kyung-Ah SHIM. Further Analysis of ID-Based Authenticated Group Key Agreement Protocol from Bilinear Maps. *IEICE TRANSACTIONS.* 2007; E90-A(1); 295-298.
[10] CH Lin, HH Lin, JH Chang. Multiparty Key Agreement for Secure Teleconferencing. *Systems. Man and Cybernetic (SMC)*. Taipei. 2006; 5: 3702-3707.
[11] Jian Zhou and Xianwei Zhou. Key Agreement Procotol in DSN. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(2): 809-818.
[12] R Barua, R Dutta, P Sarkar. Extending Joux's protocol to multi party key agreement. *Cryptography ePrint Archive*. 2003; Reprot 2003/062.
[13] L Chen, C Kudla. *Identity based authenticated key agreement protocols from pairing*. Proceeding of 16th IEEE Security Foundations Workshop. California. 2003: 219-233.

[14] Li Xiehua, Wang Yongjun. Security Enhanced Authentication and Key Agreement Protocol in Next Generation Mobile Network. *International Journal of Advancements in Computing Technology*. 2012; 4(3): 215-222.
[15] Liping Zhang, Guiling Li, Cong Xiong, Shao-Hui Zhu. A Pairing-free Identity-based Authenticated Key Agreement Protocol for Wireless and Mobile Networks. *International Journal of Advancements in Computing Technology*. 2012; 4(5): 287-294.
[16] Bin Hao, Yu Yang, Shoushan Luo, Yixian Yang, Fuqiang Liu. An Authenticated Clustering-based Group Key Agreement for Large Ad Hoc Networks. *Advances in Information Sciences and Service Sciences*. 2012; 4(7): 281-291.
[17] Ziyi You and Xiaoyao Xie. A Novel Group Key Agreement Protocol for Wireless Mesh Network. *Journal of Convergence Information Technology*. 2011; 6(2): 86-101.
[18] Abdel Alim Kamal. Cryptanalysis of a Polynomial-based Key Management Scheme for Secure Group Communication. *International Journal of Network Security*. 2013; 15(1): 68-70.
[19] Marimuthu Rajaram, Thilagavathy Dorairaj Suresh. An Interval-based Contributory Key Agreement. *International Journal of Network Security*, 2011; 13(2): 92-97.
[20] Yijuan Shi, Gongliang Chen, Jianhua Li. ID-based one round authenticated group key agreement protocol with bilinear pairings. Information Technology: Coding and Computing (ITCC). 2005; 1: 757 - 761.
[21] S Blake-Wilson, D Johnson, A Menezes. Key Agreement Protocols and their Security Analysis. *Lecture Notes in Computer Science.* 1997; 1355: 30-45.
[22] A Menezes, P van Oorschot, S Vanstone. Handbook of Applied Cryptography. Fifth Edition. Florida: CRC Press. 2001.