Provable password-based tripartite key agreement protocol

Chunbo Ma[1,3], Jun Ao[2], and Jianhua Li[1]

[1] School of Information Security Engineering
Shanghai Jiao Tong University, Shanghai, 200030, P. R. China
machunbo@sjtu.edu.cn
[2] State Key Laboratory for Radar Signal Processing,
Xidian University, Xi'an, Shanxi, 710071, P. R. China
Junjunao1@263.net
[3] The State Key Laboratory of Information Security,
Institute of Software of Chinese Academy of Sciences
Beijing, 100049, P. R. China

Abstract. A password-based tripartite key agreement protocol is presented in this paper. The three entities involved in this protocol can negotiate a common session key via a shared password over insecure networks. Proofs are given to show that the proposed protocol is secure against forging and chosen message attacks.
Keywords. Password, Tripartite key agreement, Authentication, Pairing

1. Introduction

Key agreement protocol is receiving more and more attention as the increasing requirement of data exchange over networks. The first protocol for key agreement was presented by Diffie and Hellman [1]. It allows two entities to agree upon a shared session key over an adversary controlled channel. However, the protocol is vulnerable to man-in-middle attack, since it is unauthenticated. To overcome this disadvantage, lots of authenticated two-party key agreement protocols [2][3][4]were presented in recent years.

Multi-party key agreement protocol [5][6][7][11] can be considered as the generalization of two-party protocol. Among them, Joux's tripartite one round key agreement protocol [5] using pairing on elliptic curve arrested much attention. To negotiate a common session key, it only requires each entity to broadcast a single message. However, as the original Diffie-Hellman protocol, it is also authenticated. To provide authenticity, some protocols based on different techniques [8][9][10] were proposed.

As an important authentication means, password-based technique has been studied for a long time. Recently, lots of key agreement protocols [12][13][14] were presented based on password. To the password-based protocols, a human is only required to remember a low entropy password shared between the participants. In fact, password-based schemes are suitable for implementation in many scenarios, especially those where no device is capable of securely storing high-entropy long-term secret key.

In this paper, we present a password-based tripartite key agreement protocol using pairings on elliptic curve. It allows three parties to negotiate a common session key via a shared password over an adversary controlled channel.

The paper is organized as follows. In section 2, we introduce some related works. In section 3, we give the security model and some complexity assumptions. Our protocol is presented in section 4. In section 5, we discuss the security under the random oracle model. Finally we draw conclusions in section 6.

2. Related works

Seo and Sweeney [12] proposed an authenticated Diffie-Hellman key agreement (SAKA) based on password. In contrast to traditional key agreement, the two communicating entities share a common pre-distributed password. Combing password technique and Diffie-Hellman, Yeh and Sun [13] presented another key agreement protocol, which is similar to SAKA.

Kwon et al. [14] proposed a provably secure verifier-based PAKE protocol which is suitable to the Transport Layer Security protocol. They claimed that their protocol withstood Stolen-verifier and know-key attacks. Moreover, it also provides forward secrecy.

Joux [5] presented a three-party key agreement protocol using pairing on elliptic curve. This is the first positive application of pairing in cryptography. Due to lack of authentication, Joux's protocol is susceptible to the man-in-the-middle attacks. Some researchers have further investigated the scheme and proposed group key agreement [15][16] based on ternary tree by extending the basic Joux's protocol.

Al-Riyami and Paterson [17] presented four tripartite authenticated key agreement protocols, which provided authentication using ideas from MTI [18] and MQV [19]. They used certificates of the parties to bind a party's identity with his static keys. The authenticity of the static keys provided by the signature of CA assures that only the parties who possess the static keys are able to obtain the session key. However, since the participants involved in the protocol should verify the certificate of the parties, a huge amount of computing time and storage is needed.

In [20], Nalla and Reddy proposed authenticated tripartite ID-based key agreement protocols. The security of the protocol is discussed under the possible attacks. However, Nall and Reddy's protocol is not secure as they have claimed. Chen [21] and Shim [22] showed the flaw of the protocol.

Zhang, Liu and Kim [23] designed an ID-based one round authenticated tripartite key agreement protocol and provided heuristic security analysis. The authenticity is assured by Hess' [24] ID-based signature mechanism.

3. Background

3.1 Bilinear Maps

Let G_1 be a cyclic multiplicative group generated by g, whose order is a prime q and G_2 be a cyclic multiplicative group of the same order q. Assume that the discrete logarithm in both G_1 and G_2 is intractable. A bilinear pairing is a map $e: G_1 \times G_1 \to G_2$ and satisfies the following properties:

1. *Bilinear:* $e(g^a, p^b) = e(g, p)^{ab}$. For all g, $p \in G_1$ and $a, b \in Z_q$, the equation holds.
2. *Non-degenerate:* There exists $p \in G_1$, if $e(g, p) = 1$, then $g = \mathrm{O}$.
3. *Computable:* For g, $p \in G_1$, there is an efficient algorithm to compute $e(g, p)$.

Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field. Pairings and other parameters should be selected in proactive for efficiency and security.

3.2 Complexity Assumptions

Computational Diffie-Hellman Assumption

Given g^a and g^b for some $a,b,c \in Z_q^*$, compute $e(g,g)^{abc} \in G_2$. A (τ,ε)-CDH attacker in G_2 is a probabilistic machine Ω running in time τ such that

$$Succ_{G_1}^{cdh}(\Omega) = \Pr[\Omega(g, g^a, g^b, g^c) = e(g,g)^{abc}] \geq \varepsilon$$

where the probability is taken over the random values a, b and c. The CDH problem is (τ,ε)-intractable if there is no (τ,ε)-attacker in G_2. The CDH assumption states that it is the case for all polynomial τ and any non-negligible ε.

3.3 Security Notions

The usual security model [25] built on prior work from the two-party setting [26] [27] has been widely used to analyze group key agreement protocol. In this model, several queries are available to the attacker to model his capability. We will use the model to discuss the security of our proposed protocol.

We assume that the users in set $S = \{Alice, Bob, Carol\}$ will negotiate a session key using the key agreement protocol. An attacker can make following three queries.

By accessing to the following oracles, Carol can get, modify and replay the messages transmitted over the Internet.

— $Send(U,m)$ **query.** Carol issues a query on (U,m). Carol is allowed to modify or replay any message he got from the answer of the query in active attack model.

— $\mathrm{Re}veal(i)$ **query.** Carol gets the session key K_i. We suppose that the session key is unique under the given condition.

Above queries can be asked several times. When Carol decides above queries are finished, he issues the query $Test$.

— $Test(j)$ **query.** The oracle chooses a random number $b \in \{0,1\}$. If $b = 0$, the attacker is given the session key K_j, and otherwise given a random number with the same length.

The only restrict to the query is that the query must be fresh, i.e. it has not been asked for a $\mathrm{Re}veal(j)$ query. After receiving the reply of the query $Test$, Carol outputs his guess b'. If $b' = b$, Carol wins the game. We say that if Carol can win the game in a non-negligible probability ε , then Carol has ability to break the protocol by active attack in a non-negligible probability.

4. Our protocol

Let G_1 and G_2 be two groups that support a bilinear map as defined in section 3.1. We assume that there exist three strong one way functions $H_1:\{0,1\}^* \to G_1$, $H_2:\{0,1\}^* \to Z_q^*$ and $H_3:\{0,1\}^* \to \{0,1\}^l$, where l is a secure parameter. Three clients A, B and C who keep a common *password* will agree upon a shared session key over an insecure channel. Let $a \| b$ denote the concatenate of a and b . The clients perform the following steps.

Step1.

- Client A chooses a random number $x_A \in Z_q^*$ and computes $Q_A = g^{x_A}$. And then he computes $V_A = H_1(ID_A \| password)$, $r_A = H_2(ID_A \| password)$ and $Z_A = (V_A)^{x_A r_A + r_A}$. Thereafter, client A sends (Q_A, Z_A) to the client B and C.
- Client B chooses a random number $x_B \in Z_q^*$ and computes $Q_B = g^{x_B}$. And then he computes $V_B = H_1(ID_B \| password)$, $r_B = H_2(ID_B \| password)$ and $Z_B = (V_B)^{x_B r_B + r_B}$. Thereafter, client B sends (Q_B, Z_B) to the client A and C.
- Client C chooses a random number $x_C \in Z_q^*$ and computes $Q_C = g^{x_C}$. And then he computes $V_C = H_1(ID_C \| password)$, $r_C = H_2(ID_C \| password)$ and $Z_C = (V_C)^{x_C r_C + r_C}$. Thereafter, client C sends (Q_C, Z_C) to the client A and B.

Step2.

- After receiving (Q_B, Z_B) , client A computes r_B and V_B , and then verifies $e(Z_B, g) = e(V_B, Q_B^{r_B} g^{r_B})$. Similarly, he can verify (Q_C, Z_C) via $e(Z_C, g) = e(V_C, Q_C^{r_C} g^{r_C})$. If the results are both **True**, client A computes $U_A = e(Q_B, Q_C)^{x_A}$ and draws the shared session key $K = H_3(U_A \| Q_A \| Q_B \| Q_C)$, otherwise, outputs error message and stops the protocol.
- After receiving (Q_C, Z_C) , client B computes r_C and V_C , and then verifies $e(Z_C, g) = e(V_C, Q_C^{r_C} g^{r_C})$. Similarly, he can verify (Q_A, Z_A) via $e(Z_A, g) = e(V_A, Q_A^{r_A} g^{r_A})$. If the results are both **True**, client B computes $U_B = e(Q_A, Q_C)^{x_B}$ and draws the shared session key $K = H_3(U_B \| Q_A \| Q_B \| Q_C)$, otherwise, outputs error message and stops the protocol.
- After receiving (Q_B, Z_B) , client C computes r_B and V_B , and then verifies $e(Z_B, g) = e(V_B, Q_B^{r_B} g^{r_B})$. Similarly, he can verify (Q_A, Z_A) via $e(Z_A, g) = e(V_A, Q_A^{r_A} g^{r_A})$. If the results are both **True**, client C computes $U_C = e(Q_A, Q_B)^{x_C}$ and draws the shared session key $K = H_3(U_C \| Q_A \| Q_B \| Q_C)$, otherwise, outputs error message and stops the protocol.

The protocol can be illustrated as **Fig. 1**.

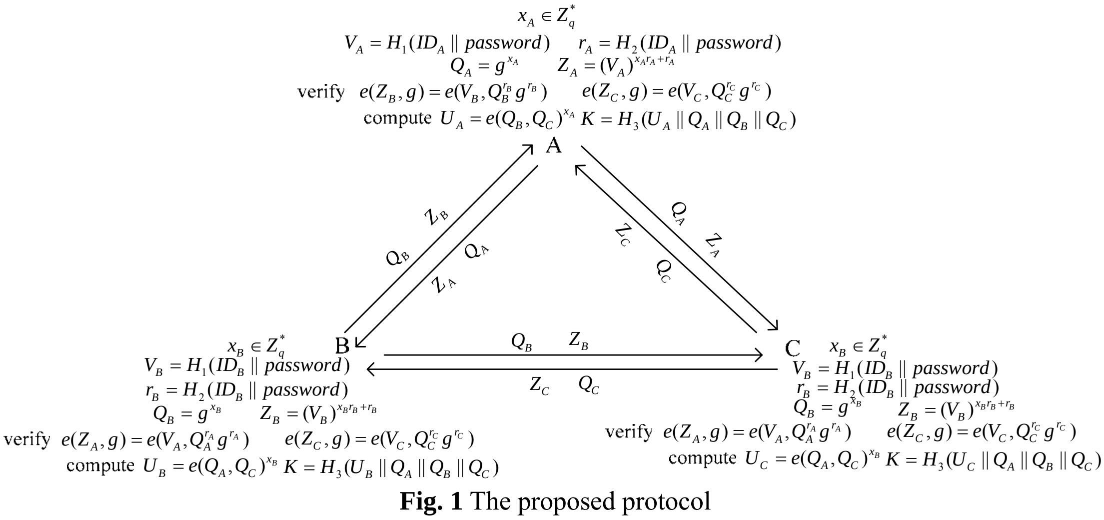

Fig. 1 The proposed protocol

5. Security

The attacker Eve is allowed to invoke the key agreement protocol and obtain, modify and replay any message transmitted over the Internet in our security model. As we have modeled in 3.3, Eve can execute passive attack and active attack to the proposed protocol. Thereby in this section, we will discuss its security under the random oracle model.

Theorem 1. We assume that an attacker Eve1 has ability to forge a valid output of client A to B with non-negligible probability ε, and then there exists another attacker Eve2 can solve the **CDH** problem with the same probability.

Proof. We assume Eve1 can forge a valid output of client A to B with non-negligible probability ε by choosing a random number to generate Q, and then given g^m and g^n, the attacker Eve2 can compute $g^{m \cdot n}$ by running Eve1 as a subroutine.

Eve2 initializes the system, and sets $g^m = g^{r_A}$ and $g^n = V_A$. As we have assumed, Eve1 chooses a random number $x_A \in Z_q^*$ and computes $Q = g^{x_A}$, and then outputs a valid (Q, Z) which will be transmitted from A to B. Thereby, Eve2 gets

$$Z = (V_A)^{x_A r_A + r_A} = (V_A)^{r_A \cdot (x_A + 1)}$$
$$= (g^{n \cdot m})^{(x_A+1)}$$

Since Eve2 implements Eve1 as a subroutine, he can obtain the random number x_A chosen by Eve1 and computes

$$g^{n \cdot m} = Z^{(x_A+1)^{-1}}$$

with a non-negligible probability ε.

□

Theorem 2. We assume that an attacker Eve who can, with success probability ε, break the protocol within a time τ by asking **H₃** and **Send** oracles at most q_H and q_s queries respectively, then there exists an attacker Carol who running in a time τ' can solve the **CDH** problem with success probability ε', where

$$\varepsilon' \geq q_H \cdot \varepsilon, \qquad \tau' \leq \tau + 3(2q_s + 1)t_{pm}.$$

Here t_{pm} is the time for a point scalar multiplication evaluation in G_1

Proof. If an attacker Eve can break the protocol via chosen message attack, then there exists an attacker Carol can solve **CDH** problem by running Eve as a subroutine, i.e. given $g^m, g^n, g^w \in G_1$, Carol can decide whether $T = e(g, g)^{mnw}$. Eve is allowed to query oracles **H_3** and **Send**. To Eve's queries, Carol gives simulative answers. In our protocol, H_1 and H_2 are just used to generate more secure values based on password, so we don't give more consideration about them. Carol chooses a random number $\lambda_i \in Z_q^*$, and sets $V_{i \in \{A,B,C\}} = g^{\lambda_i}$. Moreover, since Carol runs Eve as a subroutine, we assume that Carol knows the *password*, and can obtain $r_{i \in \{A,B,C\}}$.

H_3 queries. Carol initializes an empty $List1$. To the query on message m_i, Carol checks the records in $List1$. If there exists matching record, Carol outputs it as the answer, otherwise chooses a random string $Str_i \in \{0,1\}^l$ as the answer, and then preserves (m_i, Str_i) in $List1$.

Send queries. Attacker Eve can issue following queries.

— Eve issues at most q_s queries to client A, i.e. $q_1, q_2, \cdots, q_s$. Carol initializes an empty $List2$ and chooses a random number $r \in [1, s]$.
  - To the query $q_{i \neq r}$, Carol checks the records in $List2$. If there exists matching record, Carol outputs it as the answer, otherwise, chooses a random number $x_A \in Z_q^*$, computes $Q_A = g^{x_A}$ and $Z_A = (V_A)^{x_A r_A + r_A}$, and then feedbacks to Eve. Finally, Carol preserves (q_i, x_A, Q_A, Z_A) in $List2$.
  - To the query q_r, Carol sets $Q_A = g^m$, computes $Z_A = (g^m)^{\lambda_A r_A} g^{\lambda_A r_A}$, and then feedbacks to Eve. Finally, Carol preserves (q_r, Q_A, Z_A) in $List2$.

— Eve issues at most q_s queries to client B, i.e. $q_1, q_2, \cdots, q_s$. Carol initializes an empty $List3$.

- To the query $q_{i \neq r}$, Carol checks the records in $List3$. If there exists matching record, Carol outputs it as the answer, otherwise, chooses a random number $x_B \in Z_q^*$, computes $Q_B = g^{x_B}$ and $Z_B = (V_B)^{x_B r_B + r_B}$, and then feedbacks to Eve. Finally, Carol preserves (q_i, x_B, Q_B, Z_B) in $List3$.
- To the query q_r, Carol sets $Q_B = g^n$, computes $Z_B = (g^n)^{\lambda_B r_B} g^{\lambda_B r_B}$, and then feedbacks to Eve. Finally, Carol preserves (q_r, Q_B, Z_B) in $List3$.

— Eve issues at most q_s queries to client C, i.e. $q_1, q_2, \cdots, q_s$. Carol initializes an empty $List4$.

- To the query $q_{i \neq r}$, Carol checks the records in $List4$. If there exists matching record, Carol outputs it as the answer, otherwise, chooses a random number $x_C \in Z_q^*$, computes $Q_C = g^{x_C}$ and $Z_C = (V_C)^{x_C r_C + r_C}$, and then feedbacks to Eve. Finally, Carol preserves (q_i, x_C, Q_C, Z_C) in $List4$.
- To the query q_r, Carol sets $Q_C = g^w$, computes $Z_C = (g^w)^{\lambda_C r_C} g^{\lambda_C r_C}$, and then feedbacks to Eve. Finally, Carol preserves (q_r, Q_C, Z_C) in $List4$.

Reveal queries. When Eve queries on $i \neq r$, Carol outputs the matching i-session key. Of course, if there is not matching key, Carol outputs error message.

Since above simulation is perfect, the attacker Eve can't distinguish the simulated outputs from the actual results. Eve is allowed to ask above two oracles several times. When he decides this phase is over, he outputs **Test** query.

Test query. Carol chooses a random number $b \in \{0,1\}$. If $b = 1$, Carol outputs r-th session key, otherwise, outputs a random string with the same length as the answer. Note that the **Test** query can be asked only once. After receiving the answer of Test query, Eve outputs a guess bit b'.

We assume that the attacker Eve running in time τ can break the protocol with probability ε and asks $\mathbf{H_3}$ at most $q_H \in Z^*$ queries. If Eve can guess $b' = b$ with an non-negligible probability, then he must have queried H_3 on $m = e(g,g)^{mnw}$ with probability $\varepsilon' \geq q_H \cdot \varepsilon$. Thereby, Eve2 can solve **CDH** problem by finding the matching value in $List1$.

□

6. Conclusions

The password-based authenticated technique has been studied for a few years. Recently, two-party key agreement protocols based on password have received much attention. In this paper, we design a password-based tripartite key agreement protocol that is suitable for the user who has no place to store the high-entropy long-term secret key or has not support from public key infrastructure.

References

1 W. Diffie and M. Hellman. New directions in cryptography. IEEE Transactions on Information Theory. Vol. 22, No. 6, pp. 644-654, 1976.

2 N. McCullagh and P. S. L. M. Barreto. A new two-party identity-based authenticated key agreement. In proceedings of CT-RSA 2005, LNCS 3376, pp. 262-274.

3 I. R. Jeong, J. Katz and D. H. Lee. One-round protocols for two-party authenticated key exchange. In proceedings of ACNS 2004, LNCS 3089, pp. 220-232, Springer-Verlag, 2004.

4 K. K. R. Choo. Revisit of McCullagh-Barreto two-party ID-based authenticated key agreement protocols. Http://eprint.iacr.org/2004/343.

5 A. Joux. A on round protocol for tripartite Diffie-Hellman. In proceedings of ANTS 4, LNCS 1838, pp. 385-394, Springer-Verlag, 2000.

6 M. Just and S. Vaudenay. Authenticated multi-parts key agreement. In proceedings of Asiacrypt 1996, LNCS 1163, pp. 36-49, Springer-Verlag, 1996.

7 H. K. Lee, H. S. Lee and Y. R. Lee. Multi-party authenticated key agreement protocols from multi linear forms. Http://eprint.iacr.org/2002/166.

8 S. Al-Riyami and K. G. Paterson. Tripartite authenticated key agreement protocols from pairings. In proceedings of IMA Conference of Cryprography and Coding, LNCS 2898, pp. 332-359.

9 D. Nalla and K. C. Reddy. ID-based tripartite authenticated key agreement protocols from pairings. Http://eprint.iacr.org/2003/004.

10 F. Zhang, S. Liu and K. Kim. ID-based one round authenticated tripartite key agreement protocol with pairings. Http://eprint.iacr.org/2002/122.

11 Dutta, R. and Barua, R.: Constant round dynamic group key agreement. *Lecture Notes in Computer Science* 3650 (2005) 74-88.

12 Seo, D. H., Sweeney, P.: Simple Authenticated Key Agreement Algorithm. *Electronics Letters*. 35(13) (1999) 1073-1074.

13 Yeh, H. T., Sun, H. M.: Simple Authenticated Key Agreement Protocol resistant to Password Guessing Attacks. *ACM SIGOPS Operation Systems Review*. 36(4) (2002) 14-22.

14 Kwon, J. O., Sakurai, K., and Lee, D. H.: one-round protocol for two-party verifier-based password-authenticated key exchange. *The 10th IFIP Open Conference on Communications and Multimedia Security (CMS 2006), Lecture Notes in Computer Science* 4237 (2006) 87-96.

15 R. Barua, R. Dutta, P. Sarkar. Extending Joux Protocol to Multi Party Key Agreement. In Proceedings of Indocrypt 2003, LNCS 2940, pp. 205-217, Springer-Verlag, 2003.

16 R. Barua, R. Dutta, P. Sarkar. Provably Secure Authencitcated Tree Based Group Key Agreement. In Proceedings of ICICS 2004, LNCS 3269, pp. 92-104. Springer-Verlag, 2004.

17 S. Al-Riyami and K. G. Paterson. Tripartite authenticated key agreement protocols from pairings. In proceedings of IMA Conference of Cryprography and Coding, LNCS 2898, pp. 332-359.

18 T. Matsumoto, Y. Takashima and H. Imai. On seeking smart public-key distribution systems. In transactions of IEICE of Japan. E69, pp. 99-106, 1986.

19 L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. An efficient protocol for authenticated key agreement. Http://citeseer.nj.nec.com/law98efficient.

20 D. Nalla and K. C. Reddy. ID-based tripartite authenticated key agreement protocols from pairings. Http://eprint.iacr.org/2003/004.

21 Z. Chen. Security analysis on Nalla-Reddy's ID-based tripartite authenticated key agreement protocol. Http://eprint.iacr.org/2003/103.

22 K. Shim. Cryptanalysis of ID-based tripartite authenticated key agreement protocol. Http://eprint.iacr.org/2003/115.

23 F. Zhang, S. Liu and K. Kim. ID-based one round authenticated tripartite key agreement protocol with pairings. Http://eprint.iacr.org/2002/122.

24 F. Hess. Efficient identity based signature schemes based on pairings. In proceedings of SAC 2002, LNCS 2595, pp. 310-324, Springer-Verlag, 2002.

25 E. Bresson, O. Chevassut, A. Essiari and D. Pointcheval. Mutual Authentication and Group Key Agreement for Low-power Mobile Devices. Computer Communication, 27(17), pp. 1730-1737, 2004. A preliminary version appeared in proceedings of the 5th IFIP-TC6/IEEE ,MWCN 2003, pp. 59-62, 2003.

26 M. Bellare, R. Canetti, and H. Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols. In Proceedings of the 30th Annual Symposium on the Theory of Computing, pp. 419-428. ACM, 1998.

27 M. Burmester and Y. Desmedt. A Secure and Efficient Conference Key Distribution System. In Proceedings of Eurocrypt 1994, LNCS 950, pp. 275-286, Springer-Verlag, 1995.