http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (1): 215-219, 2008
ISSN 1812-5638

Authenticated Tripartite Key Agreement Protocol

[1,3]Chunbo Ma, [2]Jun Ao and [1]Jianhua Li
[1]School of Information Security Engineering,
Shanghai Jiao Tong University, Shanghai, 200030, People's Republic of China
[2]State Key Laboratory for Radar Signal Processing,
Xidian University, Xi'an, Shanxi, 710071, People's Republic of China
[3]The State Key Laboratory of Information Security,
Institute of Software of Chinese Academy of Sciences, Beijing, 100049, People's Republic of China

Abstract: An authenticated tripartite key agreement mechanism based on Joux's protocol is presented in this paper. The proposed protocol allows the three parties involved in the protocol to agree upon a common session key over an insecure network. The security of the proposed protocol is based on CDH problem and the strong hash function. Its security is improved under the random oracle model.

Key words: Tripartite, key agreement protocol, authenticity, random oracle model

INTRODUCTION

Data exchange over an open channel has become more pervasive as networks have gained in popularity. As one of the fundamental cryptographic primitive to prevent the communication from malicious attacker, key agreement protocols currently have received much attention. Such protocols allow entities to negotiate a common session key over an insecure network. Thereafter, the session key may be sued to implement a desired secure communication.

The first protocol for key agreement was the Diffie and Hellman (1976) protocol. It allows two entities to agree upon a common session key by exchanging messages over an open channel. However, this protocol is unauthenticated and is susceptible to the man-in-the-middle attacks. Subsequently, lots of authenticated two-party key agreement protocols (McCullagh and Barreto, 2005; Jeong *et al*., 2004; Choo, 2004) were presented.

As a natural extend, people is interested in multi-party key agreement protocols (Joux, 2000; Just and Vaudenay, 1996; Lee *et al*., 2002). Among them, Joux's Joux (2000) tripartite one round key agreement protocol using pairings on elliptic curve arrested much attention. To negotiate a common session key, it only requires each entity to transmit only a single broadcast message. Generally speaking, tripartite key agreement protocols have many applications in practice. It provides a range of services for two-party communication, where the third party can be added as a chair or trusted referee. However, just like the Diffie-Hellman protocol, the original Joux's protocol is unauthenticated and vulnerable to man-in-the-middle attacks as well. To provide authenticity, some protocols (Al-Riyami and Paterson, 2003; Nalla and Reddy, 2003; Zhang *et al*., 2002) based on different techniques were proposed in recent years.

In this study, we present a one round authenticated tripartite key agreement protocol using pairings on elliptic curve. It allows three parties to negotiate a common session key over an adversary controlled channel. Moreover, the proposed scheme is proved to be secure against forging attacks and chosen message attacks.

RELATED WORKS

Al-Riyami and Paterson (2003) presented four tripartite authenticated key agreement protocols, which provided authentication using ideas from MTI (Matsumoto *et al*., 1986) and MQV (Law *et al*., 1998). 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 Nalla and Reddy (2003) 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 (2003) and Shim (2003) showed the flaw of the protocol.

Corresponding Author: Chunbo Ma, School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai, 200030, People's Republic of China

Zhang, Liu and Kim Zhang *et al.* (2002) designed an ID-based one round authenticated tripartite key agreement protocol and provided heuristic security analysis. The authenticity is assured by Hess' (2002) ID-based signature mechanism.

BACKGROUND

Preliminaries: 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: $G_1 \times G_2 \rightarrow G_2$ and satisfies the following properties:

- **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.
- **Non-degenerate:** There exists $p \in G_1$, if e(g, p) = 1, then g = O.
- **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.

Complexity Assumptions

Computational Diffie-Hellman Assumption: Given g^a, g^b and g^c for some $a,b,c, \in Z_q^*$, compute $e(g,g)^{abc} \in G_2$. A A (τ, ε)-CDH attacker in G_1 is a probabilistic machine Ω running in time τ such that

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

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_1. The CDH assumption states that it is the case for all polynomial τ and any non-negligible ε.

Security model: The usual security model presented by Bellar and Rogaway (1993) has been widely used to analyze two-party key agreement protocol. Subsequently, McCullagh and Barreto (2005) and some others (Bresson *et al.*, 2004) modified the model to discuss the security of their proposed key agreement protocols. In present our model, we use several queries to define an attacker's capability and use Real-or-Random notion for semantic security.

We assume that there are three clients A, B and C involved in the protocol P. The attacker is allowed to access to all message transmitted over the network and to replay, modify the massage as he wants. Moreover, an attacker's interaction with the clients in the network is modeled by the following oracles.

Send (U, s, M): Attacker makes a query on (U, s, M). Upon receiving the input, the client U outputs some message matching the input. The attacker uses this query to collect the valid output of the client. We denote the s-session among the clients by s.

Reveal (U, s): This query models known key attack in real circumstance. The attacker is allowed to use this query to obtain some old session keys that have been previously accepted.

Corrupt (U): This outputs the long-term secret key held by $U \in \{A,B,C\}$ to the attacker.

Test (U, s): After chosen message attack, the attacker makes a Test-query. The message used to ask Test-query should be fresh, i.e., the message never be used during the entire attack. And the Test-query can only be asked once. When such a query is asked, a bit $b \in \{0, 1\}$ is chosen uniformly at random. If b = 1, the attacker gets back a session key, otherwise a random string with the same length. Therefore, we have.

$$Adv_P^{aka}(Attacker) = |Pr[b' = 1 | b = 1] - Pr[b' = 0 | b = 1]| \\ = 2Pr[b = b'] - 1$$

We say that the Authenticated Key Agreement (AKA) protocol is (t, ε) secure if an attacker allowed to run for time t is successful in breaking the protocol with probability at most ε.

Our protocol: Let G_1 and G_2 be two groups that supports a bilinear map as defined in section 3.1. The entries A, B and C take three random number $a,b,c \in Z_q^*$ as their private key respectively, then their public keys are g^a, g^2 and g^C. Moreover, there exist two strong one way functions H_1: $\{0\,1\}^* \rightarrow G_1$ and H_2: $\{0\,1\}^l \rightarrow G_2$ in the protocol, where l is a security parameter. The three entries perform following steps.

Step 1: A chooses a random number $x_A \in Z^*_q$, $T_A = g^{x_A}$ computes and $W_Z = H_1(T_A, g^b, g^c)$ and then sends $(T_A, (W_A)^a)$ to B and C.

B chooses a random number, $T_A = g^{x_B}$ computes and $W_B = H_1(T_B, g^a, g^c)$ and then sends $(T_B, (W_B)^b)$ to A and C.

C chooses a random number $T_c = g^{x_c}$ computes $T_A = g^{x_C}$ and $W_C = H_1(T_C, g^a, g^b, g^c)$ and then sends $(T_c, (W_c)^b)$ to A and B.

Step 2: A computes W_B and W_C and then verifies $e((W_B)^b, g) = e(W_B, g^b)$ and $e((W_C)^c, g) = e(W_C, g^c)$, respectively. If any one of them is false, entity A feedbacks error information and stops.

B computes W_A and W_C and then verifies $e((W_A)^a, g) = e(W_A, g^a)$ and $e((W_C)^c, g) = e(W_C, g^c)$, respectively. If any one of them is false, entity B feedbacks error information and stops.

C computes W_B and W_C and verifies $e((W_A)^a, g) = e(W_A, g^a)$ and $e((W_B)^b, g) = e(W_B, g^b)$, respectively. If any one of them is false, entity C feedbacks error information and stops.

Step 3: A computes $Q = e(T, T_C)^{x_A}$ and takes $K = H_2(Q, T_A, T_B, T_C)$ as the common session key.

B computes $Q = e(T_A, T_C)^{x_B}$ and takes $K = H_2(Q, T_A, T_B, T_C)$ as the common session key.

C computes $Q = e(T_A, T_B)^{x_C}$ and takes $K = H_2(Q, T_A, T_B, T_C)$ as the common session key.

The proposed one-round authenticated tripartite key agreement protocol can be illustrated as Fig. 1.

Security analysis: The security of our protocol is based on the intractability of CDH assumption and strong one way hash function. We assume that the attacker Eve has advantage $Adv_P^{aka}(Eve)$ in breaking the protocol. Then we have the following theorems.

Theorem 1: We assume that an attacker Eve1 who can, with success probability ε, forge a valid output of client A to B within a time τ by asking H_1 and Send oracles q_H and q_s queries, respectively, then there exists an attacker Eve2 who running in a time τ can solve the CDH problem with success probability ε, where

$$\varepsilon \geq q_H.\varepsilon,\ \tau \leq \tau+(q_H+q_s+1)t_{pm}$$

Proof: If an attacker Eve1 can forge a valid output of client A to B, then given $g^X, g^Y \in G_1$ there exists an attacker Eve2 can compute $g^{XY} \in G_1$ by running Eve1 as a subroutine. Let the strong one way function H_1 be an oracle. In this game, Eve1 is allowed to access to H_1 and Send oracles and to make chosen message attack. To the queries of Eve1, Eve2 sets $g^X = g^a$ and simulates these oracles to output the matching answers.

H_1 query: Eve1 outputs at most q_H queries on arbitrary message, namely $q_1, q_2 ... q_H$. Eve2 initializes an empty list and

- Chooses a random number $r \in [1, H]$ and defines g^Y as the answer of q^r.
- Chooses a random number $z_i \in Z_q^*$ and defines g^{z_i} as the answer of q_i, where $q_i \neq q_r$.

Eve2 preserves (z_i, g^{z_i}, q_i) in the List.

Send query: Eve1 outputs at most q_s queries on arbitrary message, namely $q_1, q_2 ..., q_s$. To the query q_j, Eve2

- Searches the List, gets $z_j \in Z_q^*$, computes $(g^x)^{z_j}$ and then feedback (g^{z_j}, g^{x,z_j}) as the answer, where $q_j \neq q_r$.
- Outputs $\perp$ and stops, if $q_j = q_r$.

When Eve1 decides above phase is over, he outputs a fresh valid output $((T_A, (W_A)^X)$, i.e., $((T_A, (W_A)^X)$ is not been generated by Send oracle. Since the H is a strong one way function, T_A must have been used to ask oracle H. In other words, $(W_A)^X$ is at least with probability $1/q_H$ equal to $g^{Y,X}$.

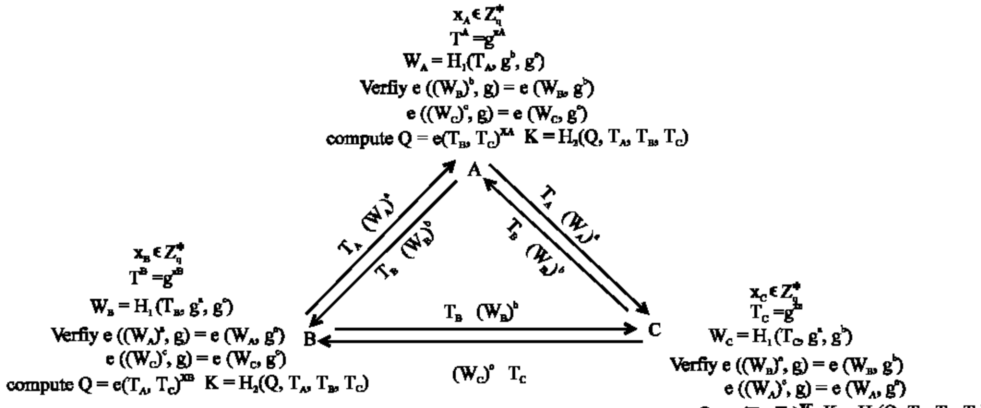

Fig. 1: The proposed protocol

As we have assumed, if the attacker Eve1 can forge the output of client A to B with probability ε via chosen message attack, then Eve2 can solve the CDH problem with probability $\varepsilon \geq q_H.\varepsilon$. Obviously, the running time for Eve2 to solve CDH problem is $\tau \leq \tau + (q_H + q_s + 1)t_{pm}$, where t_{mp} is the time for a point scalar multiplication evaluation in G_1. One can easily get

$$\Pr[\text{Forge}] \leq \text{Succ}_H^{cma}(t, q_H)$$

Note that we can generalize above results to other clients. In other words, Eve1 can forge the output of A to C with the same probability as that of A to B.

Theorem 2: Assume that the CDH assumption holds and then we say that our protocol is secure against chosen message attack.

Proof: We assume that the attacker Eve1 can break the protocol via chosen message attack, then given g^X, g^Y, $g^Z \in G_1$, there exists an attacker Eve2 can compute e $(g,g)^{XYZ} \in G_2$ by running Eve1 as a subroutine. In this game, Eve1 is allowed to access to H_1, H_2, Send, Reveal, Corrupt and Test oracles and to make chosen message attack. To the queries of Eve1, Eve2 sets $g^X = g^A$ and simulates these oracles to output the matching answers.

H_1 query: Eve2 initializes an empty List1. When Eve1 asks oracle H_1 on arbitrary message m_i, Eve2 searches the matching records in List1. If there is no matching records in the List1, then Eve2 chooses a random number $z_i \in Z_q^*$ and outputs g^{zi} as the answer and then preserves (m_i, z_i, g^{zi}) in List1.

H_2 query: Eve2 initializes an empty List2. When Eve1 asks oracle H_2 on arbitrary message m_i, Eve2 searches the matching records in List2. If there is no matching records in the List2, then Eve2 chooses a random string $\lambda_i \in \{0,1\}^1$ as the answer and then preserves (m_i, λ_i) in List2.

Send query: Here we define three kinds of queries. Eve1 asks at most q_s Send queries for client B to A, namely q_1, q_2,..., q_s. Eve2 chooses a random number $r \in [1,s]$. To the query $q_i \neq q_r$, Eve2

- Chooses $v_i \in Z^*_q$, computes g^{vi}. Thereafter, Eve1 chooses a random number, outputs $z_i \in Z^*_q$ and then preserves (g^{vi}, z_i, g^{zi}) in List1. Finally, he outputs $(g^{vi}, g^{vi,zi})$ as the answer.
- In the case of $q_i = q_r$, Eve1 chooses a random number $z_i \in Z^*_q$, computes g^{zi} and then preserves (g^Y, z_i, g^{zi}) in List1. Subsequently, Eve1 outputs error message and halts.

Eve1 asks at most q_s Send queries for client C to A, namely $q_1, q_2, \ldots q_s$. To the query $q_1 \neq q_r$, Eve2

- Chooses $w_i \in Z^*_q$, computes g^{wi}. Thereafter, Eve1 chooses a random number $z_i \in Z^*_q$, outputs g^{zi} and then preserves (g^{wi}, z_i, g^{zi}) in List1. Finally, he outputs $(g^{wi}, g^{wi,zi})$ as the answer.
- In the case of $q_i = q_r$, Eve1 chooses a random number $z_i \in Z^*_q$, computes g^{zi} and then preserves (g^Z, z_i, g^{zi}) in List1. Subsequently, Eve1 outputs error message and halts.

Eve1 asks client A at most q_s Send queries, namely q_1, $q_2, \ldots, q_s$. To the query $q_1 \neq q_2$, Eve2

- Chooses $u_i \in Z^*_q$, computes g^{ui}. Thereafter, Eve1 chooses a random number $z_i \in Z^*_q$, outputs g^{zi} and then preserves (g^{ui}, z_i, g^{zi}) in List1. Finally, he outputs $(gui, g^{ui, zi})$ as the answer.
- In the case of $q_1 = q_r$, Eve1 chooses a random number $z_i \in Z^*_q$, computes g^{zi} and then preserves (g^X, z_i, g^{zi}) in List1. Subsequently, Eve1 outputs error message and halts.

Reveal query: To the query on (U, s), if the s-session key is accepted, Eve2 outputs the session key as the answer. However, ask r-session key is not permitted.

Corrupt query: To the query on (U), Eve2 outputs the private key of $U \in \{A, B, C\}$ as the answer.

The above oracles can be asked several times. When Eve1 decides it is over, he can ask test oracle. The test oracle can be asked only once.

Test query: When Eve1 makes a Test query, Eve2 chooses a random number $b \in \{0, 1\}$. If b =1, Eve2 queries Reveal on (U, r) and outputs r-th session key K_r as the answer, where $U \in \{A, B, C\}$, otherwise outputs an arbitrary string RK of same length. Upon receiving the feedback from Eve2, Eve1 outputs his guess b.

We have assumed that the attacker Eve1 running in time τ can break the protocol with probability ε. If Eve1 can guess b' = b with an non-negligible probability, then he must have queried H_2 on $Q = e(g,g)^{XYZ}$ with advantage $\frac{1}{2}\text{Adv}_P^{aka}(\text{Eve1})$, since $\frac{1}{2}\text{Adv}_P^{aka}(\text{Eve1}) = \Pr[b = b'] - \frac{1}{2}$. Thereby, Eve2 can solve CDH problem by finding the matching value in List2. One can easily have

$$\text{Adv}_P^{aka}(\text{Eve1}) \leq 2q_s \text{Succ}_{G_1}^{cdh}(t)$$

Theorem 3: Let Eve be an attacker allowed to make at most q_H queries to the hash oracles and q_s queries to Send oracle. Then Eve can break the protocol with following advantage.

$$Adv_P^{aka}(Eve) \leq 6 \cdot Succ_H^{cma}(t, q_H) + 2q_s Succ_{G_1}^{cdh}(t)$$

The security of our protocol is based on the intractability of CDH assumption and the difficulty of forging a valid output of the client $U \in \{A, B, C\}$. Then we can easily get the conclusion of Theorem 3 by Theorem 1 and Theorem 2.

CONCLUSIONS

Secure data exchange is a basic requirement in networks. Key agreement as one of fundamental primitive is playing an important role in secrecy communication. To date, lots of key agreement scheme have been presented, but some of them have been broken. How to design secure key agreement protocols to withstand malicious attackers hidden in the networks has become an important issue. In this study, we present a one round authenticated tripartite key agreement mechanism based on Joux's protocol. It can be used in some scenarios, where three parties need to negotiate a common session key over an adversary controlled channel. We discuss the proposed protocol's security under the random oracle model and show that it can withstand chosen message attacks and forging attacks.

REFERENCES

Al-Riyami, S. and K.G. Paterson, 2003. Tripartite authenticated key agreement protocols from pairings. In proceedings of IMA Conference of Cryprography and Coding, LNCS, 2898: 332-359.

Bellar, M. and P. Rogaway, 1993. Entity authentication and key distribution. In: Advanced in Cryptography-Crypto, Springer-Verlag, LNCS, 773: 110-125.

Bresson, E., O. Chevassut, A. Essiari and D. Pointcheval, 2004. Mutual authentication and group key agreement for low-power mobile devices. Comp. Commun., 27: 1730-1737.

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

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

Diffie, W. and M. Hellman, 1976. New directions in cryptography. IEEE Tran. Inform. Theory, 22: 644-654.

Hess, F., 2002. Efficient Identity Based Signature Schemes Based on Pairings. In: Proceedings of SAC, Springer-Verlag, LNCS, 2595: 310-324.

Jeong, I.R., J. Katz and D.H. Lee, 2004. One-Round Protocols for Two-Party Authenticated Key Exchange. In: Proceedings of ACNS, Springer-Verlag, LNCS, 3089: 220-232.

Joux, A., 2000. An on Round Protocol for Tripartite Diffie-Hellman. In: Proceedings of ANTS 4, Springer-Verlag, LNCS, 1838: 385-394.

Just, M. and S. Vaudenay, 1996. Authenticated Multi-Parts Key Agreement. In: Proceedings of Asiacrypt, Springer-Verlag, LNCS, 1163: 36-49.

Law, L., A. Menezes, M. Qu, J. Solinas and S. Vanstone, 1998. An efficient protocol for authenticated key agreement. http://citeseer.nj.nec.com/law98efficient.

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

Matsumoto, T., Y. Takashima and H. Imai, 1986. On seeking smart public-key distribution systems. Tran. IEICE Jap., E69: 99-106.

McCullagh, N. and P.S.L.M. Barreto, 2005. A new two-party identity-based authenticated key agreement. Proce. CT-RSA, LNCS, 3376: 262-274.

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

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

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