Rethinking the security of some authenticated group key agreement schemes

Qiang Tang and Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
{qiang.tang, c.mitchell}@rhul.ac.uk

16th December 2004

Abstract

In this paper we analyse three improved authenticated group key agreement schemes, all of which are based on the conference key distribution systems proposed by Burmester and Desmedt. We show that all the schemes suffer from a type of impersonation attack, although these schemes are claimed to be secure.

1 Introduction

Burmester and Desmedt [1] have proposed a series of conference key distribution schemes based on public key cryptography. In order to achieve security against active adversaries, Burmester and Desmedt also presented a scheme to guarantee the authenticity of messages exchanged. Choi, Hwang and Lee [2] have proposed two ID-based group key agreement schemes using bilinear pairings. One of their schemes is an unauthenticated variant of one of the Burmester-Desmedt scheme [1] using bilinear maps, and the other is an ID-based authenticated scheme based on the former protocol (referred to here as the CHL scheme). Du et al. [3] have proposed a similar ID-based authenticated group key agreement scheme (referred to here as the DWGW scheme).

Zhang and Chen [4] have pointed out that an impersonation attack can easily be mounted against the CHL and DWGW schemes (The same attack against the CHL scheme also appeared in [5]). They also proposed a modified scheme designed to prevent this impersonation attack. Du et al. [6]

proposed a different modification to their scheme [3] to address the same attack. However, in this paper we show that both the improved schemes in [4, 6] are still vulnerable to impersonation attacks.

2 The schemes, attacks and improvements

In this section, we review the CHL and DWGW schemes as well as the the threats and described improvements in [4, 6]. Due to the similarity of the two schemes, we first describe the DWGW scheme, and then briefly describe how the CHL scheme differs.

2.1 Description of the DWGW scheme

The DWGW scheme is built upon an ID-based public key infrastructure, which consists of a Key Generation Center (KGC) and a number of users. The system parameters are $\{G_1, G_2, e, q, P, H, H_1\}$, where G_1 is a cyclic additive group generated by P whose order is a prime q, G_2 is a cyclic multiplicative group with the same order q, and $e : G_1 \times G_1 \to G_2$ is a bilinear pairing. H and H_1 are two cryptographic hash functions, $H : \{0,1\}^* \to Z_q$, and $H_1 : \{0,1\}^* \to G_1$.

System Setup: The KGC chooses a random number $s \in Z_q^*$, computes its public key $P_{pub} = sP$, and keeps s as its secret master-key. Each user U_i with identity $ID_i \in Z_q^*$ obtains a public and private key pair $< Q_i = H_1(ID_i), S_i = sQ_i >$ from the KGC.

Suppose a collection of n users ($n > 2$) wish to establish a session key; without loss of generality suppose the users are U_1, U_2, $\cdots$, U_n (with identities ID_1, ID_2, $\cdots$, ID_n). They execute the following key agreement protocol. It should be noted that the arithmetic on subscripts is computed modulo n.

- **Round 1**. First, each user U_i chooses and keeps secret a random value $N_i \in Z_q^*$, and then computes and broadcasts $< Y_i = N_iP,\ T_i = H(Y_i)S_i + N_iP_{pub} >$ to all the other participants.
- **Round 2**. Each user U_i verifies that:
$$e(\sum_{j\in\{1,\cdots,n\},j\neq i} T_j, P) = e(\sum_{j\in\{1,\cdots,n\},j\neq i} (H(Y_j)Q_j + Y_j), P_{pub}).$$
Then, U_i computes and broadcasts $X_i = e(P_{pub}, N_i(Y_{i+1} - Y_{i-1}))$.
- **Round 3**. Each user U_i now computes the session key as:
$$\begin{aligned} K &= e(P_{pub}, nN_iY_{i-1})X_i^{n-1}X_{i+1}^{n-2}\cdots X_{i-2} \\ &= e(P,P)^{(N_1N_2+N_2N_3+\cdots+N_nN_1)s} \end{aligned}$$

The CHL scheme has the same system parameters and **System Setup** phase, and its key agreement protocol also works in a similar way except that the computations in **Round 2** and **3** are slightly different:

- In **Round 2**, the verification equation is:

$$e(T_{i-1}+T_{i+1}+T_{i+2},P)=e(\sum_{j=\{i-1,i+1,i+2\}} H(Y_j)Q_j+Y_j,P_{pub})$$

  Then, U_i computes and broadcasts $X_i=e(N_i(Y_{i+2}-Y_{i-1}),Y_{i+1})$.

- In **Round 3**, the session key is computed as:

$$\begin{aligned} K &= e(nN_iY_{i-1},Y_{i+1})X_i^{n-1}X_{i+1}^{n-2}\cdots X_{i-2} \\ &= e(P,P)^{N_1N_2N_3+\cdots+N_nN_1N_2} \end{aligned}$$

2.2 Existing attacks and improvements

Due to the similarity of the impersonation attacks on the CHL and DWGW schemes [4], we only describe the attack on the latter.

For the DWGW scheme, the impersonation attack proposed by Zhang and Chen works as follows. Assume a user U_i has agreed a session key in group G_n, which has n ($n>2$) users. Suppose the authentication data in **Round 1** of the key agreement is (Y_i,T_i), where $Y_i=N_iP$, and $T_i=H(Y_i)S_i+N_iP_{pub}$. Since all the messages are broadcast to the group, U_{i+1} and U_{i-1} can obtain all the messages in the protocol run. With knowledge of (Y_i,T_i) and X_i, they can collude to impersonate U_i to negotiate a new session key in a new group G_m, which contains U_{i+1}, U_{i-1}, and U_i (of course, U_i is impersonated by U_{i+1} and U_{i-1}), and other members. To do so, U_{i+1} and U_{i-1} can just replay all the previous messages of U_{i+1}, U_{i-1} and U_i in **Round 1** of the key agreement protocol, or they can also compute new values for themselves and replay the data of U_i. Success in the latter situation relies on the following fact about the computation of X_i:

$$\begin{aligned} X_i &= e(P_{pub},N_i(Y_{i+1}-Y_{i-1})) \\ &= e(N_iP,s(Y_{i+1}-Y_{i-1})) \\ &= e(Y_i,(N_{i+1}-N_{i-1})P_{pub}) \end{aligned}$$

The only requirement is that the new index of U_i is exactly between the new indices of U_{i+1} and U_{i-1}. After pointing out the impersonation attack, Zhang and Chen proposed the following improved key agreement protocol.

- **Round 1**. First, each user U_i chooses and keeps secret a random value $N_i\in Z_q^*$, and then computes and broadcasts $< Y_i=N_iP$, $T_i=$

$H(Y_i||time||ID_1||\cdots||ID_n)S_i + N_iP_{pub} >$ to all the other participants, where *time* is a time stamp and $||$ represents concatenation.

- **Round 2**. Let $t_{auth} = time||ID_1||\cdots||ID_n$. Each user U_i verifies that:

$$e(\sum_{j\in\{1,\cdots,n\},j\neq i} T_j, P) \quad = \quad e(\sum_{j\in\{1,\cdots,n\},j\neq i} (H(Y_j||t_{auth})Q_j + Y_j), P_{pub}) \quad (1)$$

Then, U_i computes and broadcasts $X_i = e(P_{pub}, N_i(Y_{i+1} - Y_{i-1}))$.

- **Round 3**. Each user U_i now computes the session key as:

$$\begin{aligned} K &= e(P_{pub}, nN_iY_{i-1})X_i^{n-1}X_{i+1}^{n-2}\cdots X_{i-2} \\ &= e(P,P)^{(N_1N_2+N_2N_3+\cdots+N_nN_1)s} \end{aligned} \quad (2)$$

In [6], Du et al. proposed another improved scheme, this time using synchronous counters held by the group members. Each user in the group holds a counter, of which the initial value is 1, and after a successful key agreement, the counter is increased by 1. The improved key agreement protocol works as follows:

- **Round 1**. Suppose c is the current value of the counter. Each user U_i computes and broadcasts $< Y_{i,c} = N_{i,c}P, T_{i,c} = H(Y_{i,c})cS_i + N_{i,c}P_{pub} >$ to all other members of the group and keeps $N_{i,c} \in Z_q^*$ secret. Here, when the counter value is c, counter-specific values $Y_{i,c}$, $N_{i,c}$, and $T_{i,c}$ replace the values of Y_i, N_i, and T_i respectively in the original key agreement protocol of [3].

- **Round 2**. U_i verifies that:

$$e(\sum_{j\in\{1,\cdots,n\},j\neq i} T_{j,c}, P) = e(\sum_{j\in\{1,\cdots,n\},j\neq i} (H(Y_{j,c})cQ_j + Y_{j,c}), P_{pub})$$

Then, U_i computes and broadcasts $X_{i,c} = e(P_{pub}, N_{i,c}(Y_{i+1,c} - Y_{i-1,c}))$.

- **Round 3**. Each user U_i now computes the session key as:

$$\begin{aligned} K &= e(P_{pub}, nN_{i,c}Y_{i-1,c})X_{i,c}^{n-1}X_{i+1,c}^{n-2}\cdots X_{i-2,c} \\ &= e(P,P)^{(N_{1,c}N_{2,c}+N_{2,c}N_{3,c}+\cdots+N_{n,c}N_{1,c})s} \end{aligned}$$

and updates the value of the counter to $c+1$.

3 Further security vulnerabilities

Although the authors of [4, 6] have improved the DWDW scheme to prevent the impersonation attack, we now show that three of the improved schemes are still vulnerable to an impersonation attack.

1. Firstly we show that the improved scheme by Du et al. [6] suffers from an impersonation attack. Suppose that, when the counter c equals 1, the transcript pair $< Y_{i,1}, T_{i,1} >$ is computed as follows.

$$Y_{i,1} = N_{i,1}P,\ T_{i,1} = H(Y_{i,1})S_i + N_{i,1}P_{pub}$$

Then, given $< Y_{i,1}, T_{i,1} >$, an attacker can compute a valid transcript pair $< Y_{i,c^*}, T_{i,c^*} >$, where $c^* = xH(Y_{i,1})(H(xN_{i,1}P))^{-1} \bmod q$, $N_{i,c^*} = xN_{i,1}$, and x is any number in Z_q^*, as follows:

$$\begin{aligned} Y_{i,c^*} &= xY_{i,1} \\ &= xN_{i,1}P \\ &= N_{i,c^*}P \end{aligned}$$

and

$$\begin{aligned} T_{i,c^*} &= xT_{i,1} \\ &= x(H(Y_{i,1})S_i + N_{i,1}P_{pub}) \\ &= H(Y_{i,c^*})(xH(Y_{i,1})H(Y_{i,c^*})^{-1})S_i + xN_{i,1}P_{pub} \\ &= H(Y_{i,c^*})(xH(Y_{i,1})H(xN_{i,1}P)^{-1})S_i + xN_{i,1}P_{pub} \\ &= H(Y_{i,c^*})(xH(Y_{i,1})H(xN_{i,1}P)^{-1})S_i + N_{i,c^*}P_{pub} \end{aligned}$$

The validity of $< Y_{i,c^*}, T_{i,c^*} >$ can easily be verified against the definitions in **Round 1** of the improved scheme. So, as in the impersonation attack given in [4], any two users U_{i+1} and U_{i-1} who have obtained $< Y_{i,1}, T_{i,1} >$, can still impersonate U_i in the c^*-th run of the key agreement protocol, where $c^* = xH(Y_{i,1})(H(xN_{i,1}P))^{-1} \bmod q$, and x is any number in Z_q^*. In order to deploy the impersonation attack, the only additional work needed for U_{i+1} and U_{i-1} is to try different x to get an acceptable counter value c^*. However we point out that, although our attack is theoretically operable, in practice the complexity for the attacker(s) to compute an x for a specific c^* is $O(z^{|q|})$. The precise practicality of the attack depends on how counter values are managed and used (see also below).

In order to deploy the attack, the malicious users only need to obtain an valid transcript pair $< Y_{i,r}, T_{i,r} >$, where r is any number in Z_q^*; the attack implementation is similar.

In fact, the authors of [6] have not specified that the counter value should be synchronised among all the possible users, which include those users that are not in a specific communication group. Without this requirement, U_{i+1} and U_{i-1} can easily impersonate U_i to those who do not maintain the latest counter value.

2. Secondly we show that in the improved DWGW scheme of Zhang and Chen [4], three or more users can collude to impersonate another user in the key agreement protocol. Suppose a collection of n users ($n > 4$) wish to establish a session key. Suppose the users are U_1, U_2, $\cdots$, U_n. Without loss of generality, we show that U_{i-1}, U_i, and U_{i+1} are able to impersonate another user U_j ($j \neq i-1, j \neq i, j \neq i+1$) in the key agreement process. We assume that all the users except U_{i-1}, U_i, U_{i+1}, and U_j act honestly in the key agreement process. It should be clear that, assuming that all values are exchanged successfully amongst the members of the group, then equation (1) will hold for all members other than U_i and U_j if and only if the following equation holds.

   $$e(T_i + T_j, P) = e(H(Y_i||t_{auth})Q_i + Y_i + H(Y_j||t_{auth})Q_j + Y_j, P_{pub}) \quad (3)$$

   To mount the attack, in **Round 1** U_i impersonates U_j to compute and broadcast $< Y_j = N_jP, T_j = R >$, where N_j is any number in Z_q^*, R is any element in G_1. Then, U_i computes and broadcasts his own message $< Y_i = -H(Y_j||t_{auth})Q_j, T_i = H(Y_i||t_{auth})S_i + N_jP_{pub} - R >$. It is straightforward to verify that $T_i + T_j = s(H(Y_i||t_{auth})Q_i + Y_i + H(Y_j||t_{auth})Q_j + Y_j)$, and hence (3) holds. Hence in **Round 2** the verification by U_k ($k \neq i, k \neq j$) will succeed.

   In **Round 2**, U_i first impersonates U_j to compute and broadcast $X_j = e(P_{pub}, N_j(Y_{j+1} - Y_{j-1}))$, and then he computes and broadcasts his own message $X_i = e(P_{pub}, N_i^*(Y_{i+1} - Y_{i-1}))$, where N_i^* is any number in Z_q^*. U_{i-1} computes and broadcasts $X_{i-1} = e(P_{pub}, N_{i-1}(N_i^*P - Y_{i-2}))$. U_{i+1} computes and broadcasts $X_{i+1} = e(P_{pub}, N_{i+1}(Y_{i+2} - N_i^*P))$.

   In **Round 3**, any U_m ($m \neq i$) can compute the common session key as:

   $$\begin{aligned} K &= e(P_{pub}, nN_mY_{m-1})X_m^{n-1}X_{m+1}^{n-2}\cdots X_{m-2} \\ &= e(P,P)^{(N_1N_2+N_2N_3+\cdots+N_{i-1}N_i^*+N_i^*N_{i+1}+\cdots+N_nN_1)s} \end{aligned}$$

3. The above attack can also be mounted on the improved CHL scheme of Zhang and Chen [4] in a similar way. In brief, suppose a collection of n users ($n > 5$) wish to establish a session key (suppose the users are U_1, U_2, $\cdots$, U_n); then users U_{i-2}, U_{i-1}, U_{i+1}, and U_{i+2} can collude to impersonate U_i.

In fact, our attacks and the attacks proposed in [4] are all due to the lack of direct authentication of the key materials (X_i) that are used to generate the session key. So in all the schemes, even if an key is successfully generated, impersonation attacks might have occurred during the key agreement process.

4 Conclusion

In this paper we have analysed three improved ID-based authenticated group key agreement schemes, and shown that all of them are still vulnerable to impersonation attacks. In order to prevent the attack against the improved scheme of Du et al. [6], each user should be required to authenticate each of the messages received from the other users in **Round 2** of the key agreement protocol, and to synchronise the counter value before executing the protocol. One possible way to prevent the impersonation attack against the improved schemes of Zhang and Chen [4] is again to require each user to authenticate every message received from another user in **Round 2** of the key agreement protocol.

5 Acknowledgement

We would like to thank Kenny Paterson for many useful comments.

References

[1] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In A. D. Santis, editor, *Advances in Cryptology—EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 275–286. Springer-Verlag, 1994.

[2] K. Y. Choi, J. Y. Hwang, and D. H. Lee. Efficient ID-based group key agreement with bilinear maps. In F. Bao, R. Deng, and J. Y. Zhou, editors, *Proceedings of the 2004 International Workshop on Practice and Theory in Public Key Cryptography (PKC '04)*, volume 2947 of *Lecture Notes in Computer Science*, pages 130–144. Springer-Verlag, 2004.

[3] X. J. Du, Y. Wang, J. H. Ge, and Y. M. Wang. ID-based authenticated two round multiparty key agreement. Cryptology ePrint Archive: Report 2003/247, 2003.

[4] F. G. Zhang and X. F. Chen. Attacks on two ID-based authenticated group key agreement schemes. Cryptology ePrint Archive, Report 2003/259, 2003.

[5] F. G. Zhang and X. F. Chen. Attack on an ID-based authenticated group key agreement scheme from PKC 2004. *Information Processing Letters*, 91(4):191–193, 2004.

[6] X. J. Du, Y. Wang, J. H. Ge, and Y. M. Wang. An improved ID-based authenticated group key agreement scheme. Cryptology ePrint Archive, Report 2003/260, 2003.