Contributory Key Agreement in Groups: Quest for Authentication

Jan Hlinovsky
Helsinki University of Technology
jan.hlinovsky@tkk.fi

Abstract

We will look at two popular approaches to contributory group key agreement, Burmester-Desmedt and Group Diffie-Hellman protocols, and attempts to bundle implicit key authentication into the same package with them. Known attacks against some of these authenticated GKA protocols are presented to illustrate the difficult nature of proving the security of a protocol. In these protocols authentication is brought to the protocol with help of pre-shared passwords or authenticated auxiliary channels.

KEYWORDS: contributory group key agreement, authenticated group key agreement

1 Background

Security of group communications typically means using symmetric cryptography with a *group key*, i.e. there is one key that is known to all group members and that key is used for encrypting all traffic. This provides confidentiality of the traffic and also access control at group level.

Appliations where secure group communications is used include many kinds of collaborative applications, distributed computations, multiplayer games, teleconferencing, etc.

There are basically two kinds of approaches to establishing a group key: a key distribution server (or a set of servers) can be used to distribute keys to the group members, or the members themselves can agree on a key. The latter approach can be *contributory*, which means that every participant has an equal contribution to the resulting key. A key distribution server approach is more scalable than the key agreement approach, and is thus being developed as a general solution for IP multicast, for example. The non-centralized approach is applicable for instance in using multicast in ad hoc networks. This study concentrates on the non-centralized key agreement.

As described in [2], an *authenticated group key agreement protocol* is a key agreement protocol that provides *implicit key authentication*, meaning that every protocol party is assured that no outsider can learn the key (unless aided by a dishonest group member).

2 Contributory Group Key Agreement

Contributory group key agreement protocols are typically based on some way of extending the Diffie-Hellman key exchange protocol to n parties. Thus the computational and communication requirements tend to grow linearly to the number of participants, which makes these suitable for relatively small groups.

2.1 Burmester-Desmedt Protocol

In 1994 Burmester and Desmedt proposed[3] the following protocol for group key agreement. The calculations are in a cyclic group $\mathcal{G}$ generated by g. The indices are modulo n (between 1 and n) where n is the size of the group.

1. Each member m_i selects a random exponent r_i and broadcasts $z_i = g^{r_i}$
2. Each member m_i computes and broadcasts $x_i = (z_{i+1}/z_{i-1})^{r_i}$
3. Each member computes the session key $k_i = z_{i-1}^{nr_i} x_i^{n-1} x_{i+1}^{n-2} \cdots x_{i+n-2} = z_{i-1}^{nr_i} \cdot (\frac{z_{i+1}}{z_{i-1}})^{(n-1)r_i} \cdot (\frac{z_{i+2}}{z_i})^{(n-2)r_{i+1}} \ldots = g^{nr_{i-1}r_i} \cdot \frac{g^{(n-1)r_i r_{i+1}}}{g^{(n-1)r_{i-1}r_i}} \cdot \frac{g^{(n-2)r_{i+1}r_{i+2}}}{g^{(n-2)r_i r_{i+1}}} \ldots = g^{r_{i-1}r_i} g^{r_i r_{i+1}} g^{r_{i+1}r_{i+2}} \ldots g^{r_{i+n-2}r_{i+n-1}} = g^{r_1 r_2} g^{r_2 r_3} g^{r_3 r_4} \cdots g^{r_n r_1}$.

While BD is not a direct extension to the DH protocol, it produces a somewhat similar type of key, and the protocol is secure against a passive attacker if the computational Diffie-Hellman problem is hard.

BD protocol is used as a basis for several more evolved protocols, such as the password-based AKE protocols we will study later in this paper.

2.2 Group Diffie-Hellman Key Exchange Protocol

Steiner, Tsudik, and Waidner directly applied Diffie-Hellman protocol to groups of n members, and presented three variations of Group Diffie-Hellman key

Agreement protocols in 1996 [8]. Of the three protocols, the second one, GDH.2, is the one that is used in the initial key agreement in Cliques [9] protocol suite, and it has also been extended to authenticated variations. Again, we operate in a cyclic group $\mathcal{G}$ generated by g.

1. **Rounds 1 to n-1:** Member m_i selects a random exponent r_i and sends $\{g^{(r_1 \cdots r_i)/r_j} | j \in [1, i]\}, g^{r_1 \cdots r_i} \equiv C_i$ to m_{i+1}.
2. **Round n:** m_n selects a random r_n and broadcasts $\{g^{(r_1 \cdots r_n)/r_i} | i \in [1, n[\} \equiv C_n$

The GDH protocols are secure against a passive eavesdropper (assuming the CDH problem is hard). An active attacker can still masquerade as some member m_i and get access to the group key. For this reason there have been several attempts to create an authenticated variant.

2.3 Authenticated GDH

One attempt to modify the GDH.2 protocol to provide implicit key authentication is to assume that one group member m_n shares a secret K_{in} with each m_i separately, and then modify the last message in GDH.2 so that the message parts are blinded using the keys shared with the recipients. The message that m_n broadcasts becomes $\{g^{\frac{r_1 \cdots r_n}{r_i} \cdot K_{in}} | i \in [1, n[\}$ in A-GDH.2 protocol. This protocol is presented for instance in [2] and there are Cliques variants that use this version instead of the unauthenticated GDH.2 for initial key agreement.

Unfortunately, the A-GDH.2 protocol has been shown flawed by Pereira and Quisquater in 2001[7]. In a later paper[6] the same authors even prove that "it is in fact impossible to design a scalable authenticated group key agreement protocol based on the same building blocks as the A-GDH ones". Pereira and Quisquater describe attacks against implicit key authentication (IKA) property, perfect forward secrecy, and resistance to known-key attack. Let us look closer to the attack against IKA property.

2.3.1 Pereira's and Quisquater's attack against IKA property

In Cliques, the exponentiation of a value by r_i is called r_i-service, which is what a member m_i does in A-GDH.2 when $i < n$. In a group of size 3, m_1 provides r_1-service, m_2 provides r_2-service, and m_3 provides r_3K_{13}-service and r_3K_{23}-service. Let's say there is a protocol run going on between m_1, m_2, and m_3, and the intruder m_I wants to fool m_2. Suppose there is a second protocol run between m_I, m_2, and m_3, where services r'_2, r'_3K_{I3}, and r'_3K_{23} are available. The intruder takes a random value g^y and uses the services provided by m_3 to get back values $g^{yr'_3K_{I3}}$ and $g^{yr'_3K_{23}}$. The intruder will then use the r_2-service in the *first* protocol run to get $g^{yr'_3K_{I3}}$ exponentiated to $g^{yr'_3K_{I3}r_2}$ which the intruder can further exponentiate with K_{I3}^{-1} to get the value $g^{yr'_3r_2}$. The intruder then uses the value $g^{yr'_3K_{23}}$ to replace the value sent by m_3 to m_2 in the first protocol run. Member m_2 will now exponentiate this to $K_{23}^{-1}r_2$, believing this is the group key. As a result, m_2 and m_I share a key $g^{yr'_3r_2}$ that m_2 believes to be the group key, and thus the attack has succeeded and the IKA property is broken.

2.4 Dutta & Barua

In 2004, Kim, Lee and Lee presented an authenticated group key agreement protocol that used a signature scheme to achieve authentication [5]. Dutta and Barua took the Kim-Lee-Lee protocol as a starting point with an aim to replace the signature scheme with password-based symmetric encryption, and made some modifications to avoid dictionary attack. The protocol presented by Dutta and Barua in [4] is as follows:

1. Each member m_i selects a random exponent r_i and a random key k_i, calculates $z_i = g^{r_i}$ and sends $z_i^* = \mathcal{E}_{pw}(z_i)$ to neighbors m_{i-1} and m_{i+1}.
2. Each member m_i decrypts z_{i-1} and z_{i+1} and computes $K_i^L = \mathcal{H}(z_{i-1}^{r_i}) = \mathcal{H}(g^{r_i r_{i-1}})$ and $K_i^R = \mathcal{H}(z_{i+1}^{r_i}) = \mathcal{H}(g^{r_i r_{i+1}})$. Then for $i \in [1, n[$ the member m_i broadcasts $\mathcal{E}'_{pw}(k_i || K_i^L \oplus K_i^R)$, and m_n broadcasts $\mathcal{E}''_{pw}(k_n \oplus K_n^R)$. (Again all indices are modulo n, between 1 and n.)
3. Each member decrypts the messages and computes the session key $sk = \mathcal{H}(k_1 || \ldots || k_n)$. Note that $K_i^L = K_{i-1}^R$, which enables the group members to work through the chain of XORs to get k_n.

When proving security properties of the protocol, the authors assume for instance that "adversary never participates as a user in the protocol" and "an instance of a user participates in at most one session". In a more realistic model, the protocol is not secure because the same password is used by all users as an encryption key. Abdalla et al[1] present the following simple attack against the protocol of Dutta and Barua: An attacker plays the role of U_3 with honest users U_1 and U_2. He receives $z_1^* = \mathcal{E}_{pw}(z_1)$ and $z_2^* = \mathcal{E}_{pw}(z_2)$ and resends the first of these as his own contribution to the key, i.e. $z_3^* = z_1^*$. Now m_2 is computing the values $K_2^L = \mathcal{H}(g^{r_1 r_2})$ and $K_2^R = \mathcal{H}(g^{r_2 r_3}) = \mathcal{H}(g^{r_1 r_2})$. Then m_2 broadcasts $\mathcal{E}'_{pw}(k_2 || K_2^L \oplus K_i^R) = \mathcal{E}'_{pw}(k_2 || 0^k)$. The attacker knows part of the plaintext and can now do an offline dictionary attack to find a password that will decrypt the message to a nonce and k zeroes.

2.5 Abdalla et al

After breaking the protocol of Dutta and Barua (and also another one, a yet another password-based version of the Burmester-Desmedt protocol, this time by Lee, Hwang, and Lee), Abdalla et al go on to propose a protocol of their own[1]. It is also a password-based version of the Burmester-Desmedt protocol, but this time the passwords are used together with a session identifier and member's index to create individual encryption keys. The session identifier is created in a preliminary round. H, G, and $Auth$ are hash functions.

1. Each member m_i selects a random nonce N_i and broadcasts (m_i, N_i).
2. The session is defined as $S = m_1||N_1||\dots||m_i||N_i||\dots||m_n||N_n$. Each member has a specific index i and a specific symmetric key $k_i = H(S, i, pw)$. Each member m_i selects a random exponent r_i, calculates $z_i = g^{r_i}$ and broadcasts $z_i^* = \mathcal{E}_{k_i}(z_i)$. This is the only part of the protocol that is sent encrypted.
3. Each member m_i decrypts z_{i-1} and z_{i+1} and computes and broadcasts $x_i = (z_{i+1}/z_{i-1})^{r_i}$
4. Each member computes the secret $K_i = z_{i-1}^{nr_i} x_i^{n-1} x_{i+1}^{n-2} \cdots x_{i+n-2}$ and broadcasts his key confirmation $Auth_i = Auth(S, \{z_j^*, x_j\}_j, K_i, i)$.
5. After receiving and checking each key confirmation, each player computes the session key $sk_i = G(S, \{z_j^*, x_j, Auth_j\}_j, K_i)$

The price for the added security is increased complexity: compared to Dutta-Barua (and Burmester-Desmedt), the number of broadcasts is doubled, and pre-shared common password is still required. Next we will have a look at how the complexity can be reduced if there are auxiliary channels available.

3 Authenticated Key Agreement with Help of Auxiliary Channels

Wong and Stajano [10] present a protocol for using auxiliary channels to achieve authenticated key agreement. They take the Cliques protocol as a starting point, but avoid the problems presented by Pereira and Quisquater by sending the message C_i together with a MAC that includes also a random nonce as a commitment. The nonce is revealed through an auxiliary channel only after the recipient has acknowledged the C_i and MAC message. The acknowledgement is also done through an auxiliary channel. The security is based on the assumption that the auxiliary channels have the property of *data-origin authenticity*. Thus the protocol is not secure in the Dolev-Yao model, and tries not to be. Quite contrary, it explicitely assumes that the attacker does not have control over the auxiliary channels. The protocol is based on (A)GDH.2, and the rounds 1 to $n-1$ are augmented as follows:

1. m_i chooses a random nonce R_i and one-time key K_i, computes a $MAC_i = MAC_{K_i}(I_i|I_{i+1}|C_i|R_i)$ where I_i and I_{i+1} are identifiers and C_i is the same value as in GDH.2, and sends $C_i|MAC_i$ to m_{i+1} using "normal" open channel
2. m_{i+1} responds with an ack message using push-button channel
3. m_i sends R_i to m_{i+1} using visual channel
4. m_i sends K_i to m_{i+1} using open channel
5. m_{i+1} verfies MAC and sends the outcome over the pushbutton channel

Wong and Stajano also present a variation where two different hash functions are used instead of MAC and nonce:

1. m_i sends $H_1(C_i)$ to m_{i+1} using open channel
2. m_{i+1} responds with an ack message using push-button channel
3. m_i sends $H_2(C_i)$ to m_{i+1} using visual channel
4. m_i sends C_i to m_{i+1} using open channel
5. m_{i+1} verfies $H_1(C_i)$ and $H_2(C_i)$ and sends the outcome over the pushbutton channel

The final round of GDH.2 is also augmented using MAC value as a commitment:

1. m_n sends $C_n|MAC_n$ to all m_is using the open channel
2. all m_is respond with an ack message using push-button channel
3. m_n sends R_n to all m_is using visual channel
4. m_n sends K_n to all m_is using open channel
5. all m_is verify the MAC and send the outcome over the pushbutton channel

To sum up, the main point of Wong and Stajano is not so much to modify the protocol but to modify the attacker model. The assumption that there are integrity preserving auxiliary channels between all group members is crucial. They say that such channels are often present in communication situations using ad-hoc networks, but they are often neglected. They use unidirectional "Visual" and "Push-button" channels in their protocols as examples of auxiliary channels.

The auxiliary channels can be applied to authenticated key agreement protocols where there are pairwise symmetric keys between participants, or to unauthenticated key agreement protocols where such keys are not needed. The auxiliary channels make the key agreement authenticated in both situations.

Authenticated key agreement protocols typically require the participants to share pairwise keys, or common password, or both. Managing these keys and passwords is ususally left out from the scheme, but sharing the keys or passwords requires some authenticated auxiliary channel that has to preserve both confidentiality and integrity. In this context, the Wong and Stajano's scheme is useful, as the need of auxiliary channels is well defined, and confidentiality is not required.

4 Conclusions

We have looked at two contributory group key agreement protocols, Burmester-Desmedt and GDH.2, and several extensions that try to turn them into authenticated protocols. Several approaches are based on preshared keys or passwords, some of them have been proved broken. Abdalla et al present a password-based protocol that seems secure, but is noticably heavier than previous protocols. Wong and Stajano assume authenticated auxiliary channels that enable implicit key authentication with relatively simple changes to the basic GDH.2 protocol.

References

[1] Michel Abdalla, Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Password-based group key exchange in a constant number of rounds. In *Proceedings of the 2006 International Workshop on Practice and Theory in Public Key Cryptography (PKC 2006) (April 23-26, 2006, New York, USA) M. Yung Ed., Pages 427-442, LNCS 3958*. Springer-Verlag, 2006.

[2] Giuseppe Ateniese, Michael Steiner, and Gene Tsudik. Authenticated group key agreement and friends. In *ACM Conference on Computer and Communications Security*, pages 17–26, 1998.

[3] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In *In Advances in Cryptology – EUROCRYPT '94, volume 950 of Lecture Notes in Computer Science. Springer-Verlag, 1995.*, pages 275–286.

[4] Ratna Dutta and Rana Barua. Password-based encrypted group key agreement. *International Journal of Network Security, Vol.3, No.1, PP.23-34, July 2006.*, 2006.

[5] Hyun-Jeong Kim, Su-Mi Lee, and Dong Hoon Lee. Constant-round authenticated group key exchange for dynamic groups. In *ASIACRYPT, volume 3329 of Lecture Notes in Computer Science*, pages 245–259, 2004.

[6] O. Pereira and J. Quisquater. Generic insecurity of cliques-type authenticated group key agreement protocols. In *Proceedings of the 17th IEEE Computer Security Foundations Workshop*, pages 16–29. IEEE Computer Society Press, 2004.

[7] Olivier Pereira and Jean-Jacques Quisquater. Security analysis of the Cliques protocols suites. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop - CSFW'01*, pages 73–81. IEEE Computer Society Press, 2001.

[8] Michael Steiner, Gene Tsudik, and Michael Waidner. Diffie-hellman key distribution extended to group communication. In *Proceedings of the 3rd ACM conference on Computer and Communications Security*, pages 31 – 37, 1996.

[9] Michael Steiner, Gene Tsudik, and Michael Waidner. CLIQUES: A new approach to group key agreement. In *Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS'98)*, pages 380–387, Amsterdam, 1998. IEEE Computer Society Press.

[10] Ford-Long Wong and Frank Stajano. Multichannel protocols for group key agreement in arbitrary topologies. In *Proceedings of 3rd IEEE Workshop on Pervasive Computing and Communications Security*, 2006.