Contributory Key Agreement in Groups: Quest for Authentication

T-110.7290 Research Seminar on Network Security
10 Nov 2006

Jan Hlinovsky

Helsinki University of Technology

`jan.hlinovsky@tkk.fi`

Introduction

- Secure group communications: establish a symmetric *group key*
- *Contributory*: every participant has an equal contribution to the resulting key
- *Implicit key authentication*: every protocol party is assured that no outsider can learn the key
- Typical approach: agree on a generator g, every participant chooses a random exponent as their contribution to the key (à la Diffie-Hellman)

Burmester and Desmedt 1994

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

Group Diffie-Hellman Key Exchange Protocol

- Steiner, Tsudik, and Waidner 1996
- Three protocols, of which GDH.2 is used e.g. in Cliques
- **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}.
- **Round n:** m_n selects a random r_n and broadcasts $\{g^{(r_1 \cdots r_n)/r_i} | i \in [1, n[\} \equiv C_n$

Authenticated Group Diffie-Hellman Key Exchange

- All members need to share a separate key with m_n
- **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}.
- m_n selects a random r_n and broadcasts $\{g^{\frac{r_1 \cdots r_n}{r_i} \cdot K_{in}} | i \in [1, n[\}$

Pereira's and Quisquater's Attack, part 1

- We call exponentiation of a value by r_i r_i-service
- 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.
- Suppose there is a protocol run going on between m_1, m_2, and m_3, and a second protocol run between the intruder m_I, m_2, and m_3

Pereira's and Quisquater's Attack, part 2

- intruder takes a random value g^y and uses the services provided by m_3 to get back values $g^{yr_3'K_{I3}}$ and $g^{yr_3'K_{23}}$
- intruder will then use the r_2-service in the *first* protocol run to get $g^{yr_3'K_{I3}}$ exponentiated to $g^{yr_3'K_{I3}r_2}$ which the intruder can further exponentiate with K_{I3}^{-1} to get the value $g^{yr_3'r_2}$
- intruder then uses the value $g^{yr_3'K_{23}}$ to replace the value sent by m_3 to m_2 in the first protocol run.
- m_2 will now exponentiate this to $K_{23}^{-1}r_2$, believing this is the group key

Dutta & Barua

- Each member m_i selects a random exponent r_i and a random key k_i, calculates $z_i = g^{r_i}$ and broadcasts $z_i^* = \mathcal{E}_{pw}(z_i)$
- 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 $i \in [1, n[$ m_i broadcasts $\mathcal{E}'_{pw}(k_i || K_i^L \oplus K_i^R)$, and m_n broadcasts $\mathcal{E}''_{pw}(k_n \oplus K_n^R)$.
- Each member decrypts the messages and computes the session key $sk = \mathcal{H}(k_1 || \ldots || k_n)$.

Attack against Dutta & 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 $K_2^L = \mathcal{H}(g^{x_1 x_2})$ and $K_2^R = \mathcal{H}(g^{x_2 x_3}) = \mathcal{H}(g^{x_1 x_2})$ and broadcasts $\mathcal{E}'_{pw}(k_2 || K_2^L \oplus K_i^R) = \mathcal{E}'_{pw}(k_2 || 0^k)$
- attacker can now do an offline dictionary attack to find a password that will decrypt the message to a nonce and k zeroes

Abdalla et al 1

- Each member m_i selects a random nonce N_i and broadcasts (m_i, N_i). The session is defined as $S = m_1||N_1|| \ldots ||m_i||N_i|| \ldots ||m_n||N_n$. Each member has a symmetric key $k_i = H(S, i, pw)$, selects a random exponent r_i, calculates $z_i = g^{r_i}$ and broadcasts $z_i^* = \mathcal{E}_{k_i}(z_i)$.
- Each member m_i decrypts z_{i-1} and z_{i+1} and computes and broadcasts $x_i = (z_{i+1}/z_{i-1})^{r_i}$

Abdalla et al 2

- Each member computes the secret $K_i = z_{i-1}^{nr_i} x_i^{n-1} \cdots x_{i+n-2}$ and broadcasts his key confirmation $Auth_i = Auth(S, \{z_j^*, x_j\}_j, K_i, i)$.
- After receiving and checking each key confirmation, each player computes the session key $sk_i = G(S, \{z_j^*, x_j, Auth_j\}_j, K_i)$

Authentication with Auxiliary Channels

- Wong and Stajano 2006
- Modified version of the Cliques Initial Key Agreement protocol (GDH.2)
- Assumes auxiliary channels that have the property of *data-origin authenticity*

GDH.2 with Auxiliary Channels, rounds 1 to $n-1$ (part1)

- 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
- m_{i+1} responds with an ack message using pushbutton channel
- m_i sends R_i to m_{i+1} using visual channel
- m_i sends K_i to m_{i+1} using open channel
- m_{i+1} verfies MAC and sends the outcome over the pushbutton channel

GDH.2 with Auxiliary Channels, rounds 1 to $n-1$ (part2)

- m_n sends $C_n|MAC_n$ to all m_is using the open channel
- all m_is respond with an ack message using pushbutton channel
- m_n sends R_n to all m_is using visual channel
- m_n sends K_n to all m_is using open channel
- all m_is verify the MAC and send the outcome over the pushbutton channel

Summary

- Burmester-Desmedt and GDH popular starting points for authenticated extended versions
- Several approaches are based on pre-shared keys or passwords, some of them have been proved broken
- Auxiliary channels can make authenticated key agreement simpler