On the Security of an Authenticated Group Key Transfer Protocol Based on Secret Sharing

Ruxandra F. Olimid

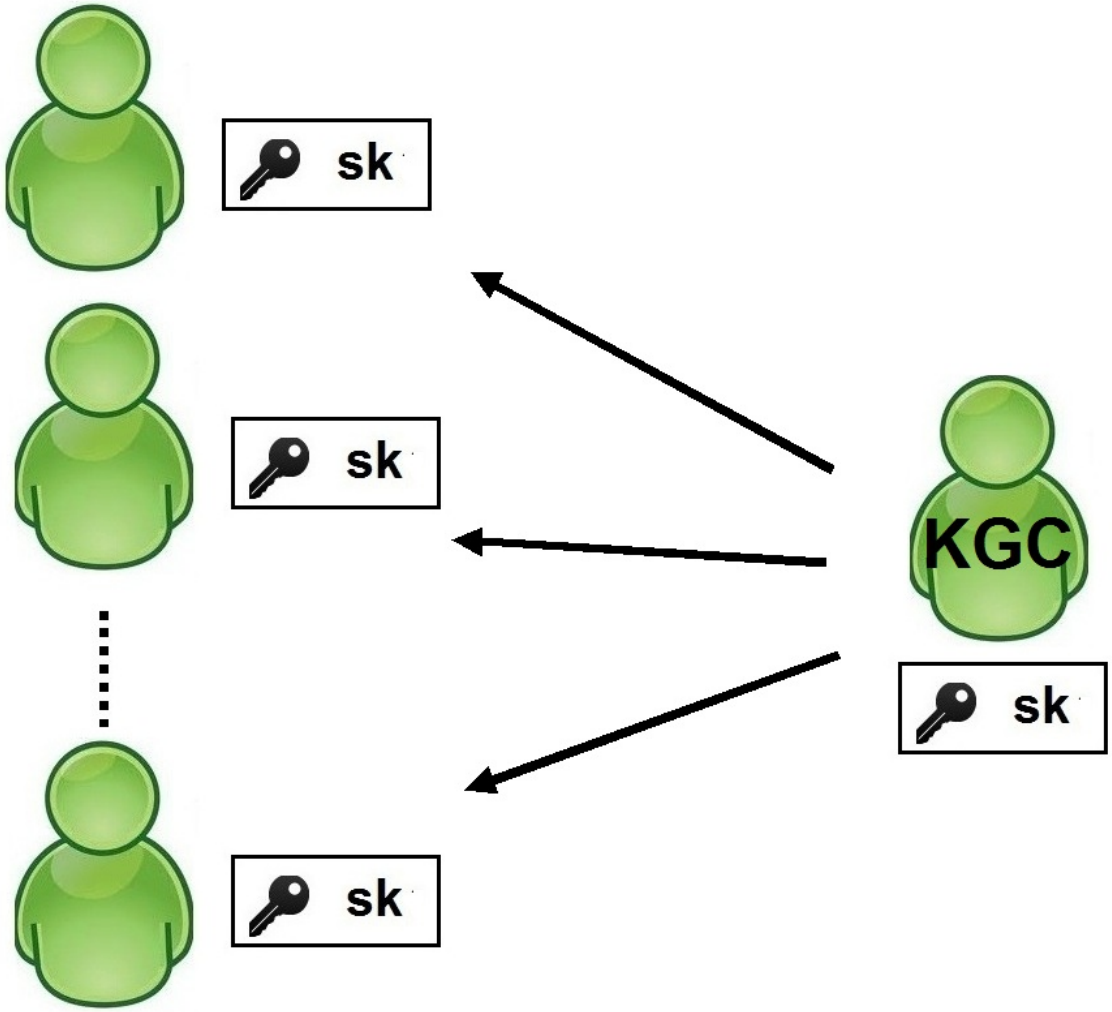
University of Bucharest
ruxandra.olimid@fmi.unibuc.ro

March 2013

Outline

Outline

Group Key Transfer Protocol



Security Goals

- Key Freshness
- Key Confidentiality
- Key Authentication
- Entity Authentication
- Known Key Security
- Forward Secrecy
- ...

Security Goals

- Key Freshness
- **Key Confidentiality**
- Key Authentication
- Entity Authentication
- Known Key Security
- Forward Secrecy
- ...

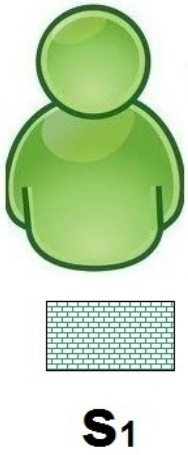
a session key must be available to authorized parties only

Security Goals

- Key Freshness
- Key Confidentiality
- Key Authentication
- Entity Authentication
- **Known Key Security**
- Forward Secrecy
- ...

a compromised session key must have no impact on the confidentiality of other past and future session keys

Secret Sharing



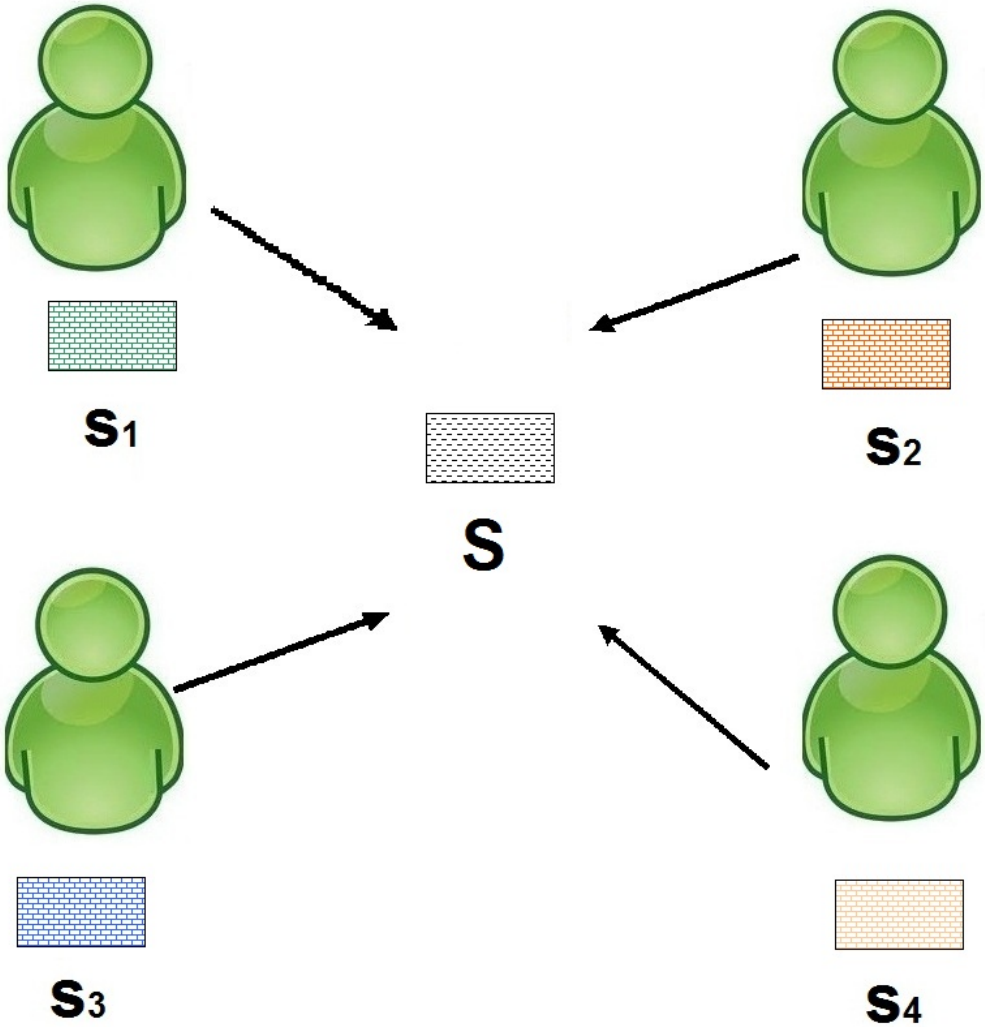
S_1

S_2

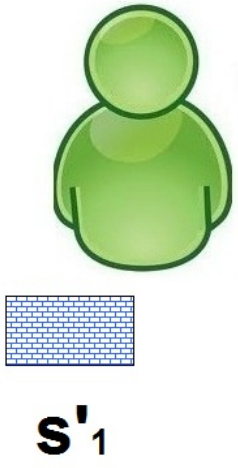
S_3

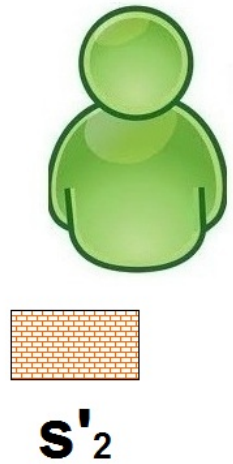
S_4

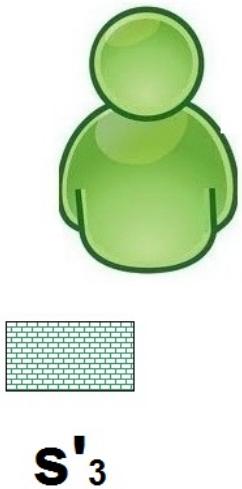
Secret Sharing

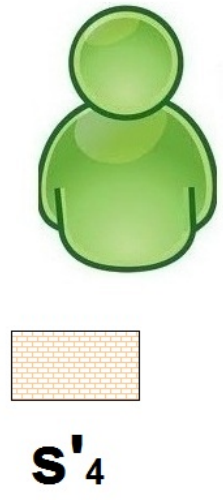


Secret Sharing

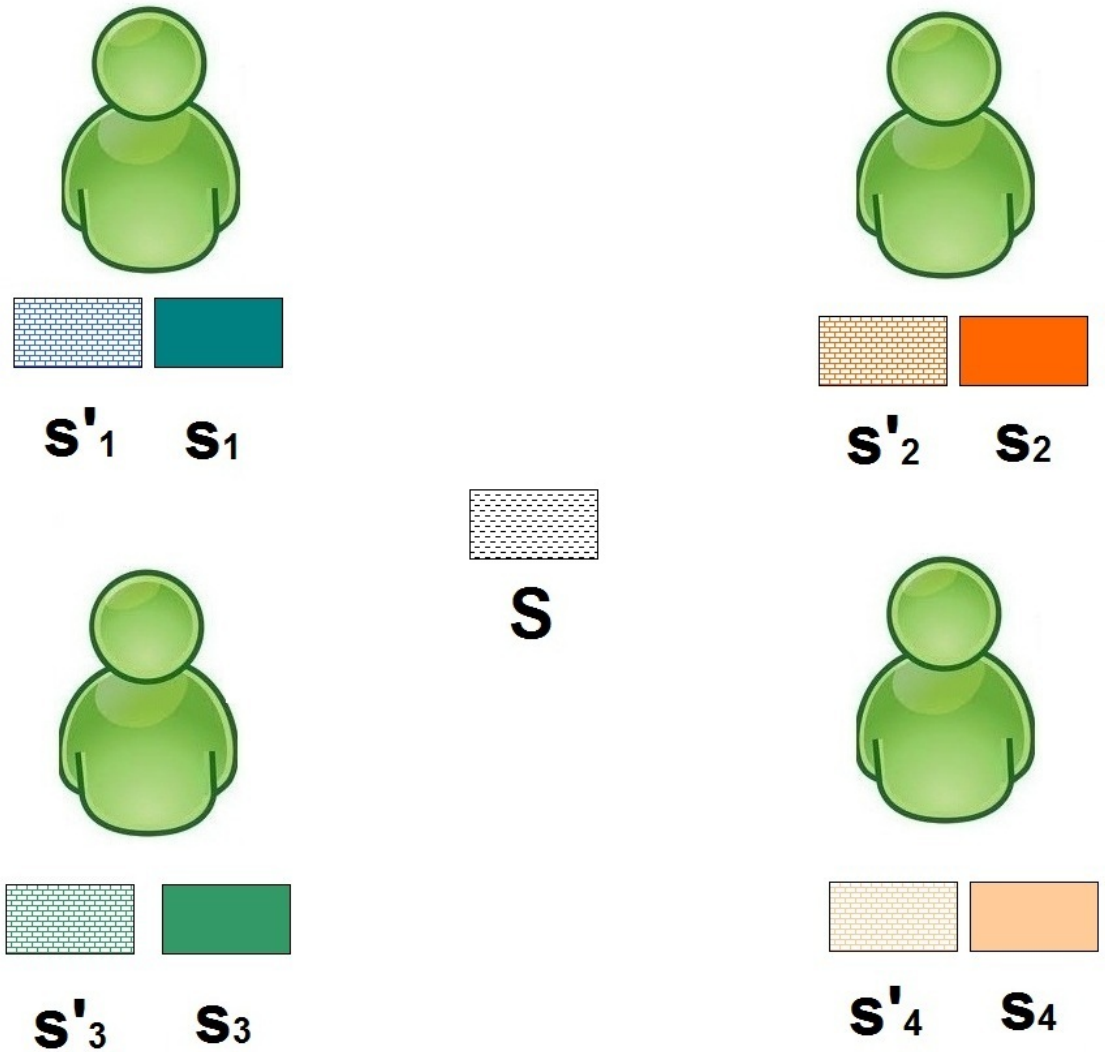








Secret Sharing



Secret Sharing

Derivative Secret Sharing [Sun et. al. 2012]

1. **Secret Sharing Phase**
   The dealer splits the secret $S \in G$ into 2 parts n times:

$$S = s_1 + s_1' = s_2 + s_2' = ... = s_n + s_n' \quad (1)$$

2. **Shares Distribution Phase**
   The dealer sends the share $s_i' \in G$ to $U_i \in \mathcal{U}$ via a secure channel.
3. **Secret Reconstruction Phase**
   1. The dealer broadcasts the shares $s_1, s_2, ..., s_n$ at once, when the users want to recover the secret S.
   2. Any user $U_i \in \mathcal{U}$ reconstructs the secret as:

$$S = s_i' + s_i \quad (2)$$

Secret Sharing

Derivative Secret Sharing [Sun et. al. 2012]

1. **Secret Sharing Phase**
   The dealer splits the secret $S \in G$ into 2 parts n times:

$$S = s_1 + s_1' = s_2 + s_2' = ... = s_n + s_n' \quad (1)$$

2. **Shares Distribution Phase**
   The dealer sends the share $s_i' \in G$ to $U_i \in \mathcal{U}$ via a secure channel.
3. **Secret Reconstruction Phase**
   1. The dealer broadcasts the shares $s_1, s_2, ..., s_n$ at once, when the users want to recover the secret S.
   2. Any user $U_i \in \mathcal{U}$ reconstructs the secret as:

$$S = s_i' + s_i \quad (2)$$

Secret Sharing

Derivative Secret Sharing [Sun et. al. 2012]

1. **Secret Sharing Phase**
   The dealer splits the secret $S \in G$ into 2 parts n times:

$$S = s_1 + s_1' = s_2 + s_2' = ... = s_n + s_n' \quad (1)$$

2. **Shares Distribution Phase**
   The dealer sends the share $s_i' \in G$ to $U_i \in \mathcal{U}$ via a secure channel.
3. **Secret Reconstruction Phase**
   1. The dealer broadcasts the shares $s_1, s_2, ..., s_n$ at once, when the users want to recover the secret S.
   2. Any user $U_i \in \mathcal{U}$ reconstructs the secret as:

$$S = s_i' + s_i \quad (2)$$

Secret Sharing

Derivative Secret Sharing [Sun et. al. 2012]

1. **Secret Sharing Phase**
   The dealer splits the secret $S \in G$ into 2 parts n times:

$$S = s_1 + s_1' = s_2 + s_2' = ... = s_n + s_n' \tag{1}$$

2. **Shares Distribution Phase**
   The dealer sends the share $s_i' \in G$ to $U_i \in \mathcal{U}$ via a secure channel.
3. **Secret Reconstruction Phase**
   1. The dealer broadcasts the shares $s_1, s_2, ..., s_n$ at once, when the users want to recover the secret S.
   2. Any user $U_i \in \mathcal{U}$ reconstructs the secret as:

$$S = s_i' + s_i \tag{2}$$

Secret Sharing

Derivative Secret Sharing [Sun et. al. 2012]

1. **Secret Sharing Phase**
   The dealer splits the secret $S \in G$ into 2 parts n times:

$$S = s_1 + s_1' = s_2 + s_2' = ... = s_n + s_n' \tag{1}$$

2. **Shares Distribution Phase**
   The dealer sends the share $s_i' \in G$ to $U_i \in \mathcal{U}$ via a secure channel.
3. **Secret Reconstruction Phase**
   1. The dealer broadcasts the shares $s_1, s_2, ..., s_n$ at once, when the users want to recover the secret S.
   2. Any user $U_i \in \mathcal{U}$ reconstructs the secret as:

U_i reconstructs the secret $S_{(1)}$

$$S = s_i' + s_i \tag{2}$$

Secret Sharing

Derivative Secret Sharing [Sun et. al. 2012]

1. **Secret Sharing Phase**
   The dealer splits the secret $S \in G$ into 2 parts n $\boxed{s'_j = S_{(1)} - s_{j_{(1)}}}$

$$S = s_1 + s'_1 = s_2 + s'_2 = ... = s_n + s'_n \quad (1)$$

2. **Shares Distribution Phase**
   The dealer sends the share $s'_i \in G$ to $U_i \in \mathcal{U}$ via a secure channel.
3. **Secret Reconstruction Phase**
   1. The dealer broadcasts the shares $s_1, s_2, ..., s_n$ at once, when the users want to recover the secret S.
   2. Any user $U_i \in \mathcal{U}$ reconstructs the secret as:

$$S = s'_i + s_i \quad (2)$$

Secret Sharing

Derivative Secret Sharing [Sun et. al. 2012]

1. **Secret Sharing Phase**
   The dealer splits the secret $S \in G$ into 2 parts n $\boxed{s'_j = S_{(1)} - s_{j_{(1)}}}$

$$S = s_1 + s'_1 = s_2 + s'_2 = ... = s_n + s'_n \quad (1)$$

2. **Shares Distribution Phase**
   The dealer sends the share $s'_i \in G$ to $U_i \in \mathcal{U}$ via a secure channel.
3. **Secret Reconstruction Phase**
   1. The dealer broadcasts the shares $s_1, s_2, ..., s_n$ at once, when the users want to recover the secret S.
   2. Any user $U_i \in \mathcal{U}$ reconstructs the secret as:

$$S = s'_i + s_i \qquad \boxed{S_{(2)} = s'_j + s_{j_{(2)}}} \quad 2)$$

Outline

The Group Key Transfer Protocol [Sun et. al. 2012]

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, \ldots, U_t\}$
   2. $KGC \rightarrow^* : \{U_1, U_2, \ldots, U_t\}$
   3. $U_i, i = 1, \ldots, t, r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G, S = s_1 + s_1' = \ldots = s_t + s_t', K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s_i', r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, \ldots, g^{s_t + r_t}, U_1, \ldots, U_t, r_1, \ldots, r_t)$
      $KGC \rightarrow^* : (M_1, \ldots, M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i}, s_i', r_i)$,
      computes $K' = g^{s_i'} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, \ldots, g^{s_t + r_t}, U_1, \ldots, U_t, r_1, \ldots, r_t)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, \ldots, U_t, r_1, \ldots r_t)$
   6. KGC computes $h_i' = H(s_i', K, U_1, \ldots, U_t, r_1, \ldots r_t)$
      checks that $h_i' = h_i$

The Group Key Transfer Protocol [Sun et. al. 2012]

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^* : \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t,\ r_i \leftarrow^R \mathbb{Z}_p^*;\ U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G,\ S = s_1 + s_1' = ... = s_t + s_t',\ K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s_i', r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^* : (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i}, s_i', r_i)$,
      computes $K' = g^{s_i'} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h_i' = H(s_i', K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h_i' = h_i$

The Group Key Transfer Protocol [Sun et. al. 2012]

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s_1' = ... = s_t + s_t'$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s_i', r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i}, s_i', r_i)$,
      computes $K' = g^{s_i'} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h_i' = H(s_i', K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h_i' = h_i$

The Group Key Transfer Protocol [Sun et. al. 2012]

1. **Phase 1: User Registration**

   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$

2. **Phase 2: Group Key Generation and Distribution**

   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t,\ r_i \leftarrow^R \mathbb{Z}_p^*;\ U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G,\ S = s_1 + s_1' = ... = s_t + s_t',\ K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s_i', r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i},\ s_i', r_i)$,
      computes $K' = g^{s_i'} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1, ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h_i' = H(s_i', K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h_i' = h_i$

The Group Key Transfer Protocol [Sun et. al. 2012]

1. **Phase 1: User Registration**

   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$

2. **Phase 2: Group Key Generation and Distribution**

   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s_1' = ... = s_t + s_t'$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s_i', r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i}, s_i', r_i)$,
      computes $K' = g^{s_i'} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h_i' = H(s_i', K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h_i' = h_i$

The Group Key Transfer Protocol [Sun et. al. 2012]

1. **Phase 1: User Registration**

   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$

2. **Phase 2: Group Key Generation and Distribution**

   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t,\ r_i \leftarrow^R \mathbb{Z}_p^*;\ U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G,\ S = s_1 + s_1' = ... = s_t + s_t',\ K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s_i', r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1,\ ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i},\ s_i', r_i)$,
      computes $K' = g^{s_i'} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1,\ ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h_i' = H(s_i', K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h_i' = h_i$

The Group Key Transfer Protocol [Sun et. al. 2012]

1. **Phase 1: User Registration**

   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$

2. **Phase 2: Group Key Generation and Distribution**

   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s_1' = ... = s_t + s_t'$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s_i', r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i}, s_i', r_i)$,
      computes $K' = g^{s_i'} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h_i' = H(s_i', K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h_i' = h_i$

Outline

The Attacks

1. Insider attack
2. Known key attack

Insider Attack

Let:

- $(k_1) \neq (k_2)$ be 2 sessions of the protocol;
- $U_a \in \mathcal{U}_{(k_1)} \setminus \mathcal{U}_{(k_2)}$;
- $U_b \in \mathcal{U}_{(k_1)} \bigcap \mathcal{U}_{(k_2)}$

Insider Attack

Let:

- $(k_1) \neq (k_2)$ be 2 sessions of the protocol;
- $U_a \in \mathcal{U}_{(k_1)} \setminus \mathcal{U}_{(k_2)}$;
- $U_b \in \mathcal{U}_{(k_1)} \bigcap \mathcal{U}_{(k_2)}$

Insider Attack

Let:

- $(k_1) \neq (k_2)$ be 2 sessions of the protocol;
- $U_a \in \mathcal{U}_{(k_1)} \setminus \mathcal{U}_{(k_2)}$;
- $U_b \in \mathcal{U}_{(k_1)} \bigcap \mathcal{U}_{(k_2)}$

Insider Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s'_1 = ... = s_t + s'_t$, $K = g^S$
      $M_i = (g^{s_i+r_i}, U_i, H(U_i, g^{s_i+r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1+r_1}, ..., g^{s_t+r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i+r_i}, s'_i, r_i)$,
      computes $K' = g^{s'_i} \cdot g^{s_i+r_i}/g^{r_i}$
      checks that $Auth = H(K', g^{s_1+r_1}, ..., g^{s_t+r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

Insider Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t,\ r_i \leftarrow^R \mathbb{Z}_p^*;\ U_i$
   4. $S \leftarrow^R G,\ S = s_1 + s'_1 = ... = s_t + s'_t,\ K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1,\ ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i},\ s'_i, r_i)$,
      computes $K' = g^{s'_i} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1,\ ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

U_a is qualified to determine $K_{(k_1)}$

Insider Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s'_1 = ... = s_t + s'_t$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H$
      computes $K' = g^{s'_i} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..$ $, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

$$g^{s'_i} = \frac{K_{(k_1)} \cdot g^{r_{i(k_1)}}}{g^{s_{i(k_1)} + r_{i(k_1)}}}$$

Insider Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s'_1 = ... = s_t + s'_t$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H$
      computes $K' = g^{s'_i} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..$ $, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

$$g^{s_{j_{(k_2)}}} = \frac{g^{s_{j_{(k_2)}} + r_{j_{(k_2)}}}}{g^{r_{j_{(k_2)}}}}$$

$$g^{s'_i} = \frac{K_{(k_1)} \cdot g^{r_{i_{(k_1)}}}}{g^{s_{i_{(k_1)}} + r_{i_{(k_1)}}}}$$

Insider Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s'_1 = ... = s_t + s'_t$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H$
      computes $K' = g^{s'_i} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..$ $, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

$$g^{s_{j_{(k_2)}}} = \frac{g^{s_{j_{(k_2)}} + r_{j_{(k_2)}}}}{g^{r_{j_{(k_2)}}}}$$

$$K_{(k_2)} = g^{s'_b} \cdot g^{s_{b_{(k_2)}}}$$

$$g^{s'_i} = \frac{K_{(k_1)} \cdot g^{r_{i_{(k_1)}}}}{g^{s_{i_{(k_1)}} + r_{i_{(k_1)}}}}$$

Known Key Attack

- The attacker discloses a key $K_{(k_1)}$;
- $U_b \in \mathcal{U}_{(k_1)} \bigcap \mathcal{U}_{(k_2)}$

Known Key Attack

- The attacker discloses a key $K_{(k_1)}$;
- $U_b \in \mathcal{U}_{(k_1)} \bigcap \mathcal{U}_{(k_2)}$

Known Key Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s'_1 = ... = s_t + s'_t$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i}, s'_i, r_i)$,
      computes $K' = g^{s'_i} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

Known Key Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t,\ r_i \leftarrow^R \mathbb{Z}_p^*;\ U_i$
   4. $S \leftarrow^R G,\ S = s_1 + s'_1 = ... = s_t + s'_t,\ K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1,\ ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, g^{s_i + r_i},\ s'_i, r_i)$,
      computes $K' = g^{s'_i} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1,\ ..., U_t, r_1, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

The attacker knows $K_{(k_1)}$

Known Key Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation and Distribution**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t,\ r_i \leftarrow^R \mathbb{Z}_p^*;\ U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G,\ S = s_1 + s_1' = ... = s_t + s_t',\ K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s_i', r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t},\ U_1,\ ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H$
      computes $K' = g^{s_i'} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..$ $, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h_i' = H(s_i', K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h_i' = h_i$

$$g^{s_i'} = \frac{K_{(k_1)} \cdot g^{r_{i(k_1)}}}{g^{s_{i(k_1)} + r_{i(k_1)}}}$$

Known Key Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s'_1 = ... = s_t + s'_t$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H$
      computes $K' = g^{s'_i} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..$ $, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

$$g^{s_{j_{(k_2)}}} = \frac{g^{s_{j_{(k_2)}} + r_{j_{(k_2)}}}}{g^{r_{j_{(k_2)}}}}$$

$$g^{s'_i} = \frac{K_{(k_1)} \cdot g^{r_{i_{(k_1)}}}}{g^{s_{i_{(k_1)}} + r_{i_{(k_1)}}}}$$

Known Key Attack

1. **Phase 1: User Registration**
   KGC shares a long-term secret $s'_i \in G$ with each user $U_i \in \mathcal{U}$
2. **Phase 2: Group Key Generation**
   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t$, $r_i \leftarrow^R \mathbb{Z}_p^*$; $U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G$, $S = s_1 + s'_1 = ... = s_t + s'_t$, $K = g^S$
      $M_i = (g^{s_i + r_i}, U_i, H(U_i, g^{s_i + r_i}, s'_i, r_i)), i = 1, \ldots, t$
      $Auth = H(K, g^{s_1 + r_1}, ..., g^{s_t + r_t}, U_1, ..., U_t, r_1, ..., r_t)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H$
      computes $K' = g^{s'_i} \cdot g^{s_i + r_i} / g^{r_i}$
      checks that $Auth = H(K', g^{s_1 + r_1}, ..$ $, ..., r_t)$
      $U_i \rightarrow KGC : h_i = H(s'_i, K', U_1, ..., U_t, r_1, ...r_t)$
   6. KGC computes $h'_i = H(s'_i, K, U_1, ..., U_t, r_1, ...r_t)$
      checks that $h'_i = h_i$

$$g^{s_{j_{(k_2)}}} = \frac{g^{s_{j_{(k_2)}} + r_{j_{(k_2)}}}}{g^{r_{j_{(k_2)}}}}$$

$$K_{(k_2)} = g^{s'_b} \cdot g^{s_{b_{(k_2)}}}$$

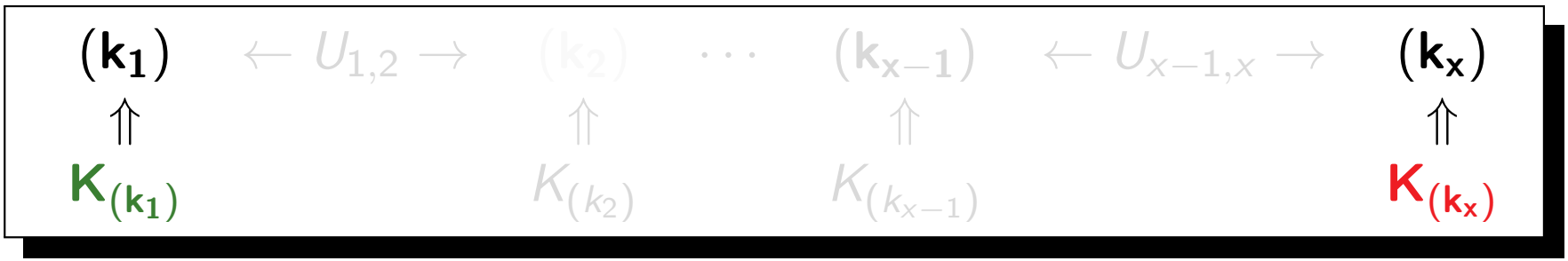
$$g^{s'_i} = \frac{K_{(k_1)} \cdot g^{r_{i_{(k_1)}}}}{g^{s_{i_{(k_1)}} + r_{i_{(k_1)}}}}$$

Chain Attack

- Apparent limitation: a common qualified user must exist for both sessions;
- Chain extension:

Chain Attack

- Apparent limitation: a common qualified user must exist for both sessions;
- Chain extension:



Chain Attack

- Apparent limitation: a common qualified user must exist for both sessions;
- Chain extension:

$$(\mathbf{k_1}) \quad \leftarrow U_{1,2} \rightarrow \quad (\mathbf{k_2}) \quad \cdots \quad (\mathbf{k_{x-1}}) \quad \leftarrow U_{x-1,x} \rightarrow \quad (\mathbf{k_x})$$
$$\Uparrow \qquad\qquad \Uparrow \qquad\qquad \Uparrow \qquad\qquad \Uparrow$$
$$\mathbf{K_{(k_1)}} \qquad K_{(k_2)} \qquad K_{(k_{x-1})} \qquad \mathbf{K_{(k_x)}}$$

Chain Attack

- Apparent limitation: a common qualified user must exist for both sessions;
- Chain extension:

$$\begin{array}{ccccccc} (\mathbf{k_1}) & \leftarrow U_{1,2} \rightarrow & (\mathbf{k_2}) & \cdots & (\mathbf{k_{x-1}}) & \leftarrow U_{x-1,x} \rightarrow & (\mathbf{k_x}) \\ \Uparrow & & \Uparrow & & \Uparrow & & \Uparrow \\ \mathbf{K_{(k_1)}} & & K_{(k_2)} & & K_{(k_{x-1})} & & \mathbf{K_{(k_x)}} \end{array}$$

Chain Attack

- Apparent limitation: a common qualified user must exist for both sessions;
- Chain extension:

$$\begin{array}{ccccccc} (\mathbf{k_1}) & \leftarrow U_{1,2} \rightarrow & (\mathbf{k_2}) & \cdots & (\mathbf{k_{x-1}}) & \leftarrow U_{x-1,x} \rightarrow & (\mathbf{k_x}) \\ \Uparrow & & \Uparrow & & \Uparrow & & \Uparrow \\ \mathbf{K_{(k_1)}} & & K_{(k_2)} & & K_{(k_{x-1})} & & \mathbf{K_{(k_x)}} \end{array}$$

Outline

Countermeasure

1. **Phase 1: User Registration**

   KGC shares a long-term secret $s_i' \in G$ with each user $U_i \in \mathcal{U}$

2. **Phase 2: Group Key Generation and Distribution**

   1. $U_1 \rightarrow KGC : \{U_1, U_2, ..., U_t\}$
   2. $KGC \rightarrow^*: \{U_1, U_2, ..., U_t\}$
   3. $U_i, i = 1, \ldots, t,\ r_i \leftarrow^R \mathbb{Z}_p^*;\ U_i \rightarrow KGC : r_i$
   4. $S \leftarrow^R G,\ S = s_1 + s_1' = ... = s_t + s_t',\ \alpha \leftarrow^R G, K = \alpha^S$
      $M_i = (\alpha^{s_i + r_i}, U_i, H(U_i, \alpha^{s_i + r_i}, s_i', r_i, \alpha)), i = 1, \ldots, t$
      $Auth = H(K, \alpha^{s_1 + r_1}, ..., \alpha^{s_t + r_t},\ U_1, ..., U_t, r_1, ..., r_t, \alpha)$
      $KGC \rightarrow^*: (M_1, ..., M_t, Auth, \alpha)$
   5. $U_i, i = 1, \ldots, t$ verifies that $h = H(U_i, \alpha^{s_i + r_i},\ s_i', r_i, \alpha)$,
      computes $K' = \alpha^{s_i'} \cdot \alpha^{s_i + r_i} / \alpha^{r_i}$
      checks that $Auth = H(K', \alpha^{s_1 + r_1}, ..., \alpha^{s_t + r_t},\ U_1, ..., U_t, r_1, ..., r_t, \alpha)$
      $U_i \rightarrow KGC : h_i = H(s_i', K', U_1, ..., U_t, r_1, ...r_t, \alpha)$
   6. KGC computes $h_i' = H(s_i', K, U_1, ..., U_t, r_1, ...r_t, \alpha)$
      checks that $h_i' = h_i$

Thank you!

Questions