Establishing a Group Key Using One-Way Accumulators

Teklay Gebremichael

Mid Sweden University

teklay.gebremichael@miun.se

May 11, 2017

Mittuniversitetet
MID SWEDEN UNIVERSITY

About Myself

- BSc in Information Technology, Mekelle Institute of Technolgoy, Ethiopia.
- MSc in Computer Science, University of Trento, Italy.
- Now PhD candidate at Mid Sweden University, Sweden.

Mittuniversitetet
MID SWEDEN UNIVERSITY

Internet of Things

- Internet of Things: Interconnection of users, computing systems, and everyday objects.

Internet of Things

- Internet of Things: Interconnection of users, computing systems, and everyday objects.
- Main research challenges:
  - Scaling and Naming
  - Interoperability (openness)
  - Big Data Analytics
  - Energy
  - **Security and Privacy**

Mittuniversitetet
MID SWEDEN UNIVERSITY

Information Security in Internet of Things

- Cryptography is the main tool for achieving information security in IoT
  1. Confidentiality
  2. Integrity
  3. Authentication

Information Security in Internet of Things

- Cryptography is the main tool for achieving information security in IoT
  1. Confidentiality
  2. Integrity
  3. Authentication
- They all require a **Cryptographic Key**.

Mittuniversitetet
MID SWEDEN UNIVERSITY

Information Security in Internet of Things

- Cryptography is the main tool for achieving information security in IoT
  1. Confidentiality
  2. Integrity
  3. Authentication
- They all require a **Cryptographic Key**.
- Key sharing is usually a challenge.(And specially among a group)

Mittuniversitetet
MID SWEDEN UNIVERSITY

Approaches Today

1. Key sharing schemes based on Symmetric Key Crypto
   - Each device shares a key with every other device (Secure but does not scale well)
   - Single key shared among all devices. (very vulnerable)
   - Key sharing approaches based on observed environment behavior (Limited key size)

Approaches Today

1. Key sharing schemes based on Symmetric Key Crypto
   - Each device shares a key with every other device (Secure but does not scale well)
   - Single key shared among all devices. (very vulnerable)
   - Key sharing approaches based on observed environment behavior (Limited key size)
2. Key sharing schemes based on Public Key Crypto
   - Computationally Expensive (specially for IoT devices)
   - Need a "Trust Anchor" to resolve public keys
   - not suitable for IoT

Mittuniversitetet
MID SWEDEN UNIVERSITY

Research Question

1. How to design distributed key establishment (sharing) schemes ?
2. Schemes where all devices involved do a proportional amount of work in generating the shared key?
3. How about group keys?

Establishing a Group Key Using One Way Accumulators

- (Objective:) Design a scheme that enables devices to form a "secure multicast" group.

Why Group Communication in IoT

- Multicast Applications are very common.
- Example use case:
  1. Smart Home Application : Control of light bulbs
  2. e-health: collection and aggregation of patient data

Mittuniversitetet
MID SWEDEN UNIVERSITY

Establishing a Group Key Using One Way Accumulators

- (Objective:) Design a scheme that enables devices to form a "secure multicast" group.

Basic Assumptions

1. Network consists of n devices $(d_1, d_2, d_3, \cdots, d_n)$ and a "trusted" Gateway (GW).
2. Each device has private/public pairs.
3. A device can request the GW to get a list of the devices in the network.
4. The network is relatively stable (low group join and leave rates)

Mittuniversitetet
MID SWEDEN UNIVERSITY

Leveraging One Way Accumulators

- Establish a scheme that enables devices to form a "secure multicast" group.
- We leverage the concept of one-way accumulators.
- One-Way Accumulator:
  A function $h : \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{X}$ such that:
  1. It is "hard" to invert
  2. $h(h(x, y_1), y_2) = h(h(x, y_2), y_1)$ (Quasi-Commutativity)
  3. Hard to find a collisions.

Leveraging One Way Accumulators

- Establish a scheme that enables devices to form a "secure multicast" group.
- We leverage the concept of one-way accumulators.
- One-Way Accumulator:
  A function $h : \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{X}$ such that:
  1. It is "hard" to invert
  2. $h(h(x, y_1), y_2) = h(h(x, y_2), y_1)$ (Quasi-Commutativity)
  3. Hard to find a collisions.
- (Example): Modular Exponentiation since
  $exp(exp(x, y_1), y_2) = exp(exp(x, y_2), y_1)$

Mittuniversitetet
MID SWEDEN UNIVERSITY

The proposed Scheme

- Assume d_1 initiates the group creation process(Otherwise, it can do it through the GW).
- "Interested devices" reply "join". (signed with their private keys)
- Assume devices d_2, d_3 and d_4 reply "join".
- Then, d_1 does the following sequence of steps.
  1. compute $z = h(h(h(d_1, d_2), d_3), d_4)$

Mittuniversitetet
MID SWEDEN UNIVERSITY

The proposed Scheme

- Assume d_1 initiates the group creation process(Otherwise, it can do it through the GW).
- "Interested devices" reply "join". (signed with their private keys)
- Assume devices d_2, d_3 and d_4 reply "join".
- Then, d_1 does the following sequence of steps.
  1. compute $z = h(h(h(d_1, d_2), d_3), d_4)$
  2. For each device d_j, compute z_j. (z_j is computed similarly to z with parameter d_j excluded for each z_j)

Mittuniversitetet
MID SWEDEN UNIVERSITY

The proposed Scheme

- Assume d_1 initiates the group creation process(Otherwise, it can do it through the GW).
- "Interested devices" reply "join". (signed with their private keys)
- Assume devices d_2, d_3 and d_4 reply "join".
- Then, d_1 does the following sequence of steps.
  1. compute $z = h(h(h(d_1, d_2), d_3), d_4)$
  2. For each device d_j, compute z_j. (z_j is computed similarly to z with parameter d_j excluded for each z_j)
  3. pick a random $k \in \mathbb{K}$ (This will be the session group key)

Mittuniversitetet
MID SWEDEN UNIVERSITY

The proposed Scheme

- Assume d_1 initiates the group creation process(Otherwise, it can do it through the GW).
- "Interested devices" reply "join". (signed with their private keys)
- Assume devices d_2, d_3 and d_4 reply "join".
- Then, d_1 does the following sequence of steps.
  1. compute $z = h(h(h(d_1, d_2), d_3), d_4)$
  2. For each device d_j, compute z_j. (z_j is computed similarly to z with parameter d_j excluded for each z_j)
  3. pick a random $k \in \mathbb{K}$ (This will be the session group key)
  4. finally, to each device send k, z, and z_j encrypted with their respective public keys.

Mittuniversitetet
MID SWEDEN UNIVERSITY

Continued ...

- any device d_j in the group can send a multicast message by encrypting the message with k.
- To prove its membership to the group it must append to the message the tuple (d_j, z_j).
- Others can verify its membership by computing $h(z_j, d_j)$ and comparing it to z.

Mittuniversitetet
MID SWEDEN UNIVERSITY

Security

- (Threat Model): what can an attacker do?
  1. (Passive): Simply guess the key. Will be able to passively read messages but can only guess the key with probability $\frac{1}{2^n}$, where n is the key size. (We assume this value to be negligible)

Security

- (Threat Model): what can an attacker do?
  1. (Passive): Simply guess the key. Will be able to passively read messages but can only guess the key with probability $\frac{1}{2^n}$, where n is the key size. (We assume this value to be negligible)
  2. (Active): Forge membership. The attacker has to produce a fake $z_j^{'}$ such that $h((z_j)^{'}, d_j) = h(z_j, d_j)$. (Hard by assumption).

Mittuniversitetet
MID SWEDEN UNIVERSITY

Security

- (Threat Model): what can an attacker do?
  1. (Passive): Simply guess the key. Will be able to passively read messages but can only guess the key with probability $\frac{1}{2^n}$, where n is the key size. (We assume this value to be negligible)
  2. (Active): Forge membership. The attacker has to produce a fake $z_j^{'}$ such that $h((z_j)^{'}, d_j) = h(z_j, d_j)$. (Hard by assumption).
  3. Forward Secrecy ?

Mittuniversitetet
MID SWEDEN UNIVERSITY

Security

- (Threat Model): what can an attacker do?
  1. (Passive): Simply guess the key. Will be able to passively read messages but can only guess the key with probability $\frac{1}{2^n}$, where n is the key size. (We assume this value to be negligible)
  2. (Active): Forge membership. The attacker has to produce a fake $z_j^{'}$ such that $h((z_j)^{'}, d_j) = h(z_j, d_j)$. (Hard by assumption).
  3. Forward Secrecy ?
  4. How about group add and leave operations ?

Mittuniversitetet
MID SWEDEN UNIVERSITY

Thank You!