A Distribute Key Assignment Protocol for Secure Multicast Based on Proxy Cryptography

Chun-Ying Huang, Yun-Peng Chiu, Kuan-Ta Chen, and Chin-Laung Lei

Distributed Computing and Network Security Lab
Department of Electrical Engineering
National Taiwan University

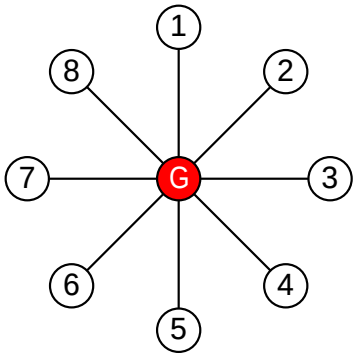
Outline

- Secure Multicast
- Problem Statements and Assumptions
- Our Scheme

Secure Multicast

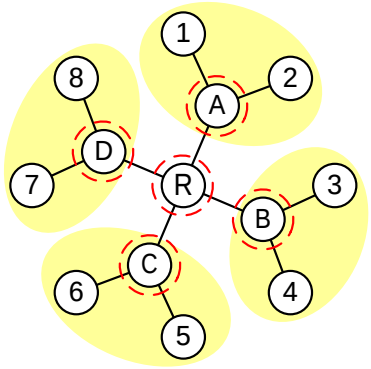
- The Goal
  - Share common secrets between group members.
- Secure Multicast: Classifications and Problems

Centralized



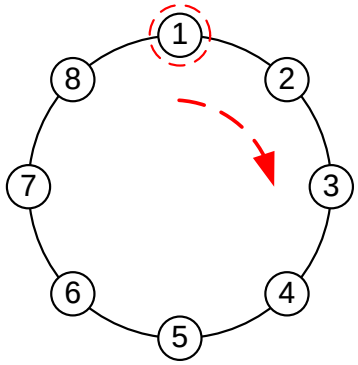
- Need a group controller.
- Bad scalability.

Decentralized



- Scale better, however …
- May still need a group controller.

Distributed (Contributory)



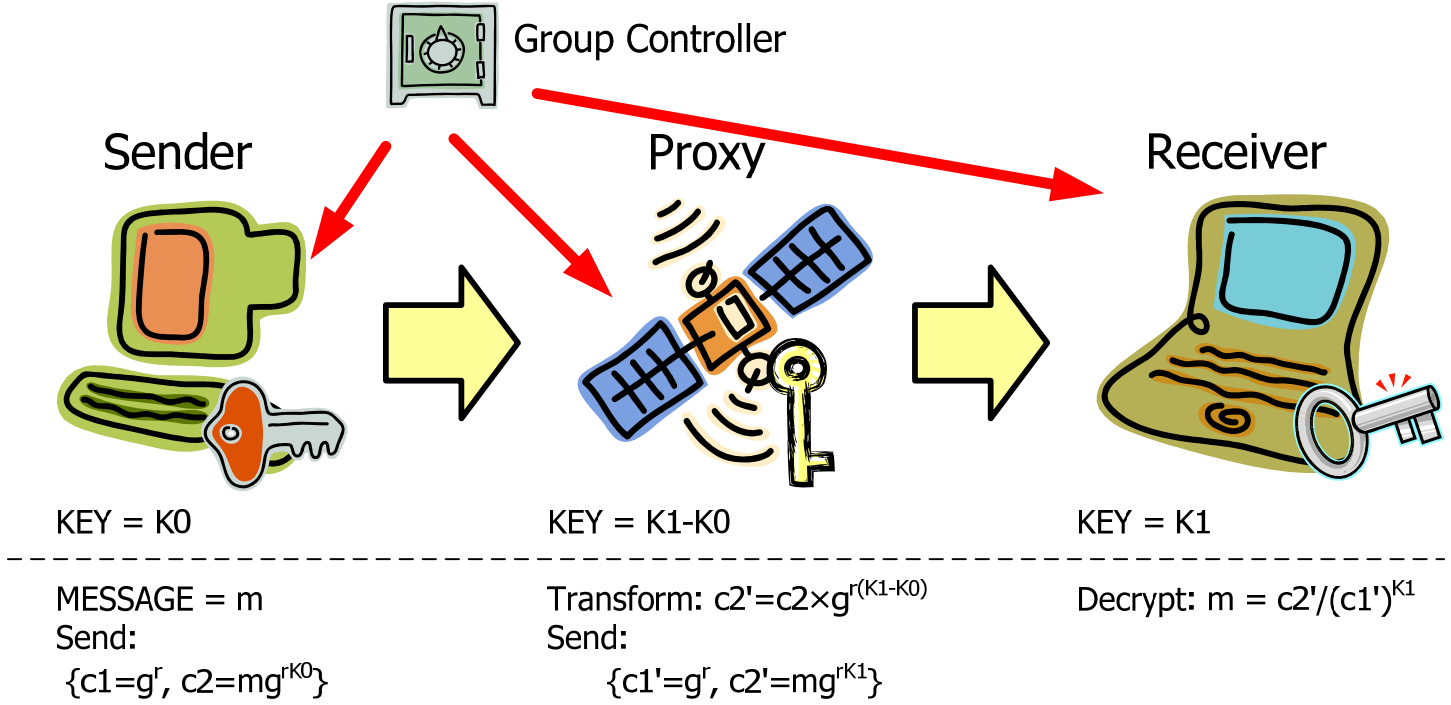
- Need to know group members.
- Bad for a large group.

The Problem Statement and Assumptions

- For a large communication group …
  - It would be better to adopt “decentralized” mechanisms.
  - However, we don’t like the group controller.
  - Besides, we should only put limited trust on intermediate branch nodes.
- Our scheme is based on ElGamal proxy encryption
  - It can be easily applied on a source-based multicast tree.
  - It reduces the trust-level on intermediate nodes.
- Assumptions
  - The sender and the receivers are trusted.
  - Proxies are semi-trusted.

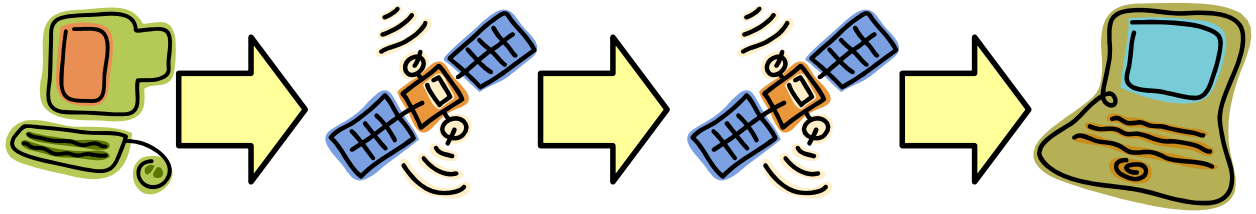
Our Scheme – Adopt Proxy Encryption

- Proxies are only semi-trusted
  - Transform the cipher-text without revealing the message.



Our Scheme – Remove the Group Controller

- Basic Ideas
  - The sender and the proxies generate their own secret keys.
  - The receiver obtain the decryption key using the proposed key-composition protocol.



Sender	Proxy-1	Proxy-2	Receiver
KEY = K0	KEY = K1	KEY = K2	KEY = ?
CoKEY= R+K0	CoKEY= R+K0+K1	CoKEY = R+K0+K1+K2	Obtain R, compute CoKEY-R = K0+K1+K2
MESSAGE = m Send: $\{g^r, mg^{rK0}\}$	Transform with g^{rK1} Send: $\{g^r, mg^{r(K0+K1)}\}$	Transform with g^{rK2} Send: $\{g^r, mg^{r(K0+K1+K2)}\}$	Decrypt using the composed KEY.

Thank You!

Comments or Questions?