Adapting GDOI for Balanced Batch LKH

Josep Pegueroles

Telematics Engineering Department.
Technical University of Catalonia
Barcelona, Spain
Adapting GDOI for Balanced Batch LKH

Contents

• Introduction
• Benchmark Scenarios
• Performance of Batch Rekeying with benchmark scenarios
• Balanced Batch Rekeying
• Single message rekeying algorithm
• Single message rekeying for batch rekeying
• Adaptation for GDOI
• Work to do
Introduction

- **Goal of the work:**
  - Studying the behavior of batch rekeying in real scenarios and improving Lam-Gouda’s algorithm.

- **Roadmap:**
  - Generating synthetic user behavior patterns (according to benchmark scenarios) in order to simulate performance behavior of batch rekeying algorithms
  - Simulate and detect flaws of proposed batch rekeying algorithm
  - Propose a slight variation in the algorithm
  - Introduce broadcast encryption in multicast key management for single and batch processing
  - Adapt the algorithms to GDOI
  - Implementation of LKH and variation algorithms in GDOI
Benchmark Scenarios


- **2 benchmark scenarios pointed out:**
  - single source broadcast
  - virtual conferences

- **One more scenario included:**
  - netgames

- **All patterns generated in Matlab**
Benchmark Scenarios

- Single source broadcast

- Exponential distribution inter-arrival time
- Based on the classic study of Almeroth and Ammar
  (Multicast Group Behavior in the Internet’s Multicast Backbone (MBone), IEEE Communications, June 1997)
- Mean value of Inter-arrival time can be set by designing
Benchmark Scenarios

- Virtual conferences

- A random number of users establish a virtual conference
- After a random period of time users begin to join/leave the group following an exponential inter-request time statistics
Benchmark Scenarios

- Networked games
  
  - Player interarrival times are highly correlated at short lags which implies that the arrival of some users will lead to others arriving
  - Interarrival times for networked games follow a heavy-tailed distribution
Performance of Batch Rekeying with benchmark scenarios

Why real patterns should be included?


Introduce batch processing for Join/Leave requests but simulation results presented does not fit a real request pattern but a uniformly distributed and only considering the total number of joining and leaving requests but not their position in the tree.

In such cases, it is easy to get an unbalanced tree after a few batches.

An additional balancing algorithm is suggested after some rekeyings to overcome this flaw.
Lam-Gouda Proposal not balanced

New joinings are located substituting leavings

If number of joinings exceeds leavings a subtree is constructed and placed under the shallowest leaf

- Example: Member 4 and Member 8 leave the group
Lam-Gouda Proposal not balanced

- 2 joinings and no leaving
Lam-Gouda Proposal not balanced

- M5 and M6 leave the group
Balanced Batch Rekeying


We have **proposed a batch rekeying method leading to completely balanced trees.**

It is **based on the assumption that:**

- Siblings of departed members are treated as new members with GDOI phase 1 already done
- Members can change their position in the tree from one batch to another, not only in their path to the root but also can jump to another branch.
Balanced Batch Rekeying

“Mark Tree” phase

Keys to be thrown away

GDOI Phase 1 Key

Keys to be reused

M1  M2  M3  M4  M5  M6  M7  M8
Balanced Batch Rekeying

“Prune Tree” phase

GDOI Phase 1 Key

Keys to be reused

M1  M3  M4

M6  M7  M8
Balanced Batch Rekeying

“New Rekey Tree” phase

Position is not taken into account
Members do not know their actual position
The balancing “not much” effort is only done by the GCKS

GDOI Phase 1 Key
Keys to be reused
New KEKs sent by a PUSH message
New KEKs sent by a PULL message

Only the Key-Handle GDOI’s parameter is used
Balanced Batch Rekeying

- Tree depth evolution for Lam-Gouda and balanced batch rekeying algorithm in web tv environment

Effect of the peak arrivals minimized
Balanced Batch Rekeying

- Tree depth evolution for Lam-Gouda and balanced batch rekeying algorithm in virtual conference environment.

If number of members is within a range tree depth remains constant
Balanced Batch Rekeying

- Tree depth evolution for Lam-Gouda and balanced batch rekeying algorithm in networked games environment
Balanced Batch Rekeying

- We have arrived at the same conclusion as Lam and Gouda:
  
  Balancing is better...

  ...but we have introduced balancing in the algorithm itself
Introducing Broadcast Encryption:
Single message rekeying algorithm
Single message rekeying

Assume the following scenario:

We consider the concatenated LKH message

\[ K_{34} \{ K_0' \} K_{34} \{ K_{11}' \} K_{34} \{ K_{22}' \} || K_{21} \{ K_0' \} K_{21} \{ K_{11}' \} || K_{12} \{ K_0' \} \]

as 3 messages cause it is the result of 3 encryption operations. Moreover, parts of this message are not useful for some remaining members.

We want to generate a single message, the same for all the remaining members.

We use number theory and modular reduction.
Single message rekeying

Let every node in the tree be a random number generated as follows:

\[ K_{(i,j)} = F_{r1}(2^i + j) \oplus r \]

When the node wants to be updated, the only necessary information is

\[ P = r \oplus r' \]

and the updated number will be computed as follows:

\[ K'_{(i,j)} = K_{(i,j)} \oplus P = F_{r1}(2^i + j) \oplus r' \]

When updating is needed only \( P \) has to be delivered to the remaining members

At this moment the GCKS only has to store \( r1 \) and \( r' \) and the \( TEK \)

Bin, Jian-Hua. *Optimal Key Storage for Secure Multicast*. Department of Electronic Engineering, Shanghai Jiaotong University
Single message rekeying

But, if the multicast message is constructed as follows:

$$P = r_2 \prod_{i \in S} rnd_i + (r \oplus r')$$

We get only one (and the same) message for all the members.

That only have to divide modulo one of his secret numbers in order to obtain the updating parameter.

Although it can be considered weaker than LKH/OFT/OFC, it provides enough security level for many non critical applications.

Security analysis is included in
http://isg.upc.es/gsec/work.html
Single message rekeying

But...
OK, the GCKS only has to store 3 numbers but we have 1 message of length KN instead of N messages of length K... where’s the improvement?

First, the length of the messages needed in simple LKH is greater than each of the numbers of the modulo product.

Second, If Batch rekeying is considered...
Single message rekeying for batch rekeying

Comparing the total amount of bits used for rekeying in Batch LKH Lam Gouda and batch with single message rekeying...

Adapting GDOI for Balanced Batch LKH

Adaptation for GDOI

- GDOI perfectly fits with all the mentioned algorithms
- Only some considerations of unused parameters in LKH payload message format have to be added
- We have started the task of adding LKH to GDOI Brian Weis’ implementation
Work to do

- Discuss the proposed methods:
  - balanced batch
  - single message rekeying
- Include them in a standard track LKH/OFT/OFC...
- Finishing the implementation of these algorithms in GDOI testbed
Contact Information

Josep Pegueroles
Telematics Engineering Dept.
Technical University of Catalonia
josep.pegueroles@entel.upc.es