Stateful Subset Cover

Mattias Johansson[1] Gunnar Kreitz[2] Fredrik Lindholm[1]

[1]Ericsson AB

[2]KTH – Royal Institute of Technology
School of Computer Science and Communication
gkreitz@kth.se

ACNS '06, June 8

Agenda

Soccer World Cup on Cell Phones

- Pay-per-view broadcasting to cell phones
- Only paying customers can watch

KTH Computer Science and Communication

Soccer World Cup on Cell Phones

- Pay-per-view broadcasting to cell phones
- Only paying customers can watch
- Encrypted video

KTH Computer Science and Communication

What is Broadcast Encryption?

- The problem of establishing secure communication with a changing group of receivers
- One key server, multiple receivers
- Network as a broadcast medium

KTH Computer Science and Communication

What is Broadcast Encryption?

- The problem of establishing secure communication with a changing group of receivers
- One key server, multiple receivers
- Network as a broadcast medium
- Berkovits (1991), Fiat and Naor (1994)

KTH Computer Science and Communication

The Basic Principle

- Establish a *Group Key* (sometimes called *Media Key*), K
- Broadcast content encrypted with K

The Basic Principle

- Establish a *Group Key* (sometimes called *Media Key*), K
- Broadcast content encrypted with K
- Updating the group key (depending on application)
    - When some number of members (possibly 1) have left/joined
    - At timed intervals
    - A combination of the above

KTH Computer Science and Communication

Notation and Terminology

- r is the number of revoked users
- m is the number of members
- $u = r + m$ is the number of users

The Naïve Scheme

- One symmetric key for each user

The Naïve Scheme

- One symmetric key for each user
- To establish group key K, broadcast K encrypted with each member's key

The Naïve Scheme

- One symmetric key for each user
- To establish group key K, broadcast K encrypted with each member's key
- Bandwidth is $\Theta(m)$

KTH Computer Science and Communication

Subset Cover-based Broadcast Encryption

- Subset Cover is a principle for constructing Broadcast Encryption schemes

KTH Computer Science and Communication

Subset Cover-based Broadcast Encryption

- Subset Cover is a principle for constructing Broadcast Encryption schemes
- Static family of sets of users
- Each set is associated with a key
- Only users in the subset can compute the key

Subset Cover-based Broadcast Encryption

- Subset Cover is a principle for constructing Broadcast Encryption schemes
- Static family of sets of users
- Each set is associated with a key
- Only users in the subset can compute the key
- Naor, Naor, Lotspiech 2001

Subset Cover (cont'd)

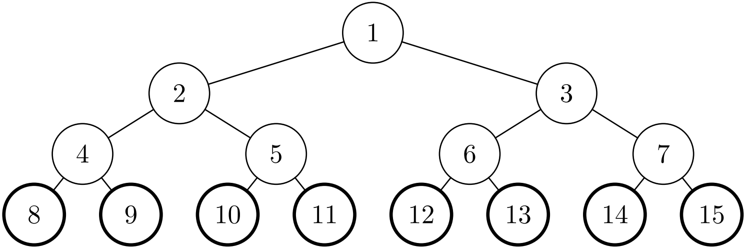
- To distribute a new group key
    1. Compute a cover of the members (avoiding revoked users), using the subsets
    2. Encrypt group key K with subset key for each subset in cover

KTH Computer Science and Communication

Subset Cover (cont'd)

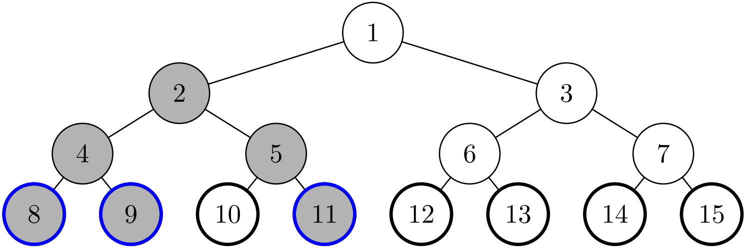
- To distribute a new group key
  1. Compute a cover of the members (avoiding revoked users), using the subsets
  2. Encrypt group key K with subset key for each subset in cover
- Bandwidth is $\Theta(\text{cover size})$

The Subset Difference Scheme



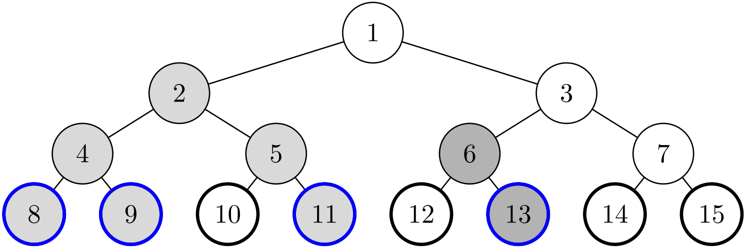
- A scheme based on the Subset Cover principle
- Users are viewed as leafs of a (balanced) binary tree
- Subsets are of the form “all users below node *v* but not below (or in) node *w*”
- Bandwidth is $\min(2r + 1, m)$

The Subset Difference Scheme



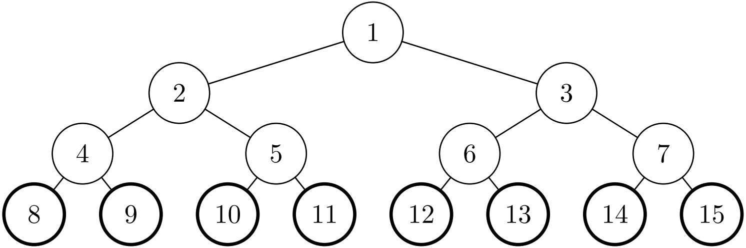
- A scheme based on the Subset Cover principle
- Users are viewed as leafs of a (balanced) binary tree
- Subsets are of the form "all users below node v but not below (or in) node w"
- Bandwidth is $\min(2r + 1, m)$
- Example: $S_{2,10}$

The Subset Difference Scheme



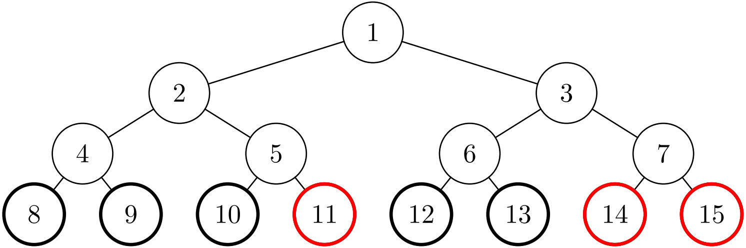
- A scheme based on the Subset Cover principle
- Users are viewed as leafs of a (balanced) binary tree
- Subsets are of the form "all users below node v but not below (or in) node w"
- Bandwidth is $\min(2r+1, m)$
- Examples: $S_{2,10}$ and $S_{6,12}$

Subset Difference Example



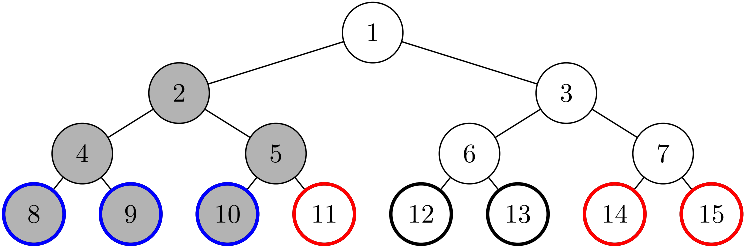
Broadcast:

Subset Difference Example



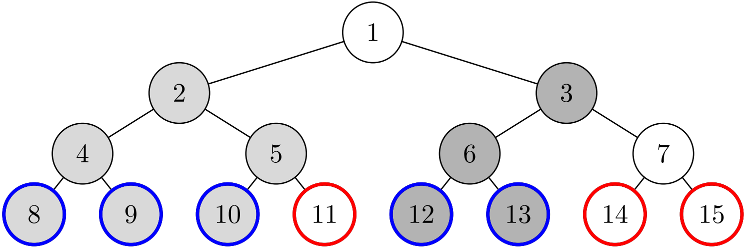
Broadcast:

Subset Difference Example



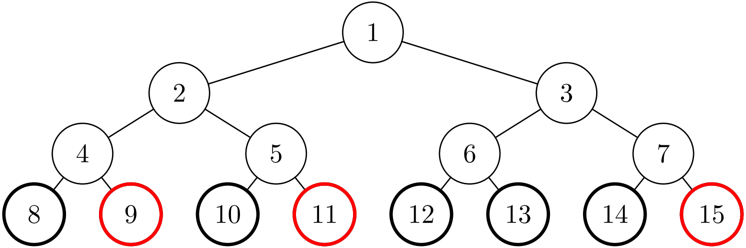
Broadcast: $\mathrm{E}_{K_{2,11}}(K)$

Subset Difference Example



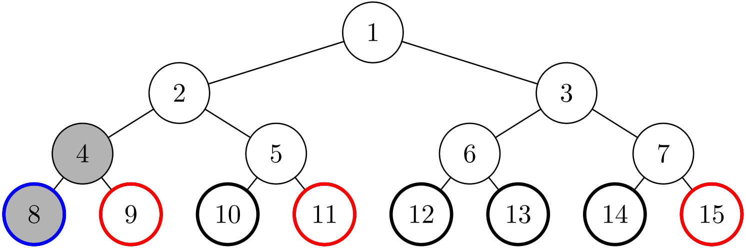
Broadcast: $\mathrm{E}_{K_{2,11}}(K), \mathrm{E}_{K_{3,7}}(K)$

Subset Difference Example



Broadcast: $\mathrm{E}_{K_{2,11}}(K), \mathrm{E}_{K_{3,7}}(K)$
User at node 9 leaves, user at node 14 joins

Subset Difference Example

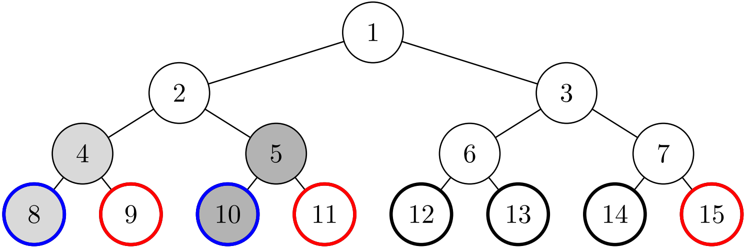


Broadcast: $\mathrm{E}_{K_{2,11}}(K), \mathrm{E}_{K_{3,7}}(K)$

User at node 9 leaves, user at node 14 joins

$\mathrm{E}_{K_{4,9}}(K')$

Subset Difference Example

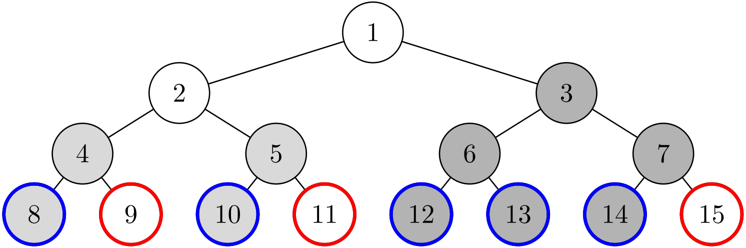


Broadcast: $\mathrm{E}_{K_{2,11}}(K), \mathrm{E}_{K_{3,7}}(K)$

User at node 9 leaves, user at node 14 joins

$\mathrm{E}_{K_{4,9}}(K'), \mathrm{E}_{K_{5,11}}(K')$

Subset Difference Example



Broadcast: $\mathrm{E}_{K_{2,11}}(K)$, $\mathrm{E}_{K_{3,7}}(K)$

User at node 9 leaves, user at node 14 joins

$\mathrm{E}_{K_{4,9}}(K')$, $\mathrm{E}_{K_{5,11}}(K')$, $\mathrm{E}_{K_{3,15}}(K')$

Our Idea

- Adding a *State Key*, K_s, given to all members
- Need to be covered *and* have state key to recover group key
- Only revoked users who have state key need to be avoided in the cover

Our Idea

- Adding a *State Key*, K_s, given to all members
- Need to be covered *and* have state key to recover group key
- Only revoked users who have state key need to be avoided in the cover
- General modification reducing bandwidth for Subset Cover based schemes

Broadcasting a Group Key

1. Calculate cover $\mathcal{C}_j$, covering all joiners, avoiding *all* revoked users
2. Calculate cover $\mathcal{C}$, covering all members not covered in $\mathcal{C}_j$, avoiding revoked users who have state key

Broadcasting a Group Key

1. Calculate cover $\mathcal{C}_j$, covering all joiners, avoiding *all* revoked users
2. Calculate cover $\mathcal{C}$, covering all members not covered in $\mathcal{C}_j$, avoiding revoked users who have state key
3. Select random blinding value R and let $K_e = R \oplus K_s$ (K_s state key)

KTH Computer Science and Communication

Broadcasting a Group Key

1. Calculate cover $\mathcal{C}_j$, covering all joiners, avoiding *all* revoked users
2. Calculate cover $\mathcal{C}$, covering all members not covered in $\mathcal{C}_j$, avoiding revoked users who have state key
3. Select random blinding value R and let $K_e = R \oplus K_s$ (K_s state key)
4. Broadcast
   - $\mathrm{E}_{K_e}(K)$, $\mathrm{E}_{K_e}(K_s')$
   - $\mathrm{E}_{K_c}(K_e)$ for all $c \in \mathcal{C}_j$
   - $\mathrm{E}_{K_c}(R)$ for all $c \in \mathcal{C}$

KTH Computer Science and Communication

Advantages

- In traditional subset cover, there are two types of users, “must cover” and “must avoid”
- Now there is a new type, “don’t care”

KTH Computer Science and Communication

Advantages

- In traditional subset cover, there are two types of users, “must cover” and “must avoid”
- Now there is a new type, “don’t care”
- Since most subset cover schemes have bandwidth $\mathcal{O}(r)$, we can always
  - Relabel all “don’t care” as “must cover”
  - Run original cover algorithm

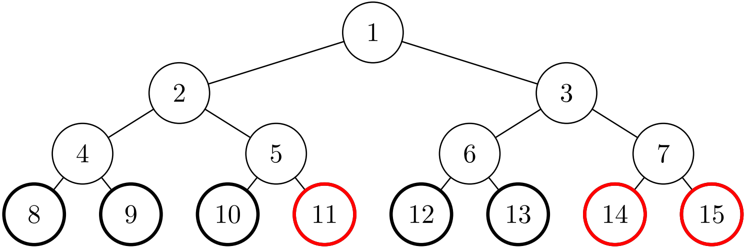
KTH Computer Science and Communication

Advantages

- In traditional subset cover, there are two types of users, “must cover” and “must avoid”
- Now there is a new type, “don’t care”
- Since most subset cover schemes have bandwidth $\mathcal{O}(r)$, we can always
  - Relabel all “don’t care” as “must cover”
  - Run original cover algorithm
- But often we can make better use of the “may cover” nodes by developing a new, scheme-specific, cover algorithm

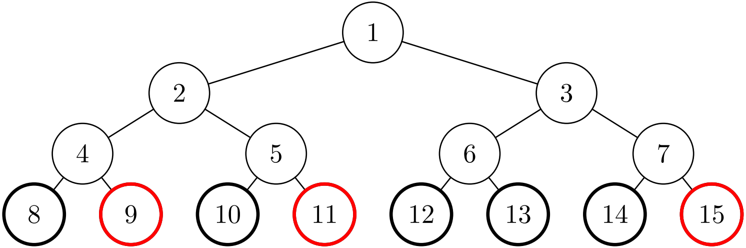
KTH Computer Science and Communication

Stateful Subset Difference Example



Users at nodes 8, 9, 10, 12, and 13 were members and have state key K_s.

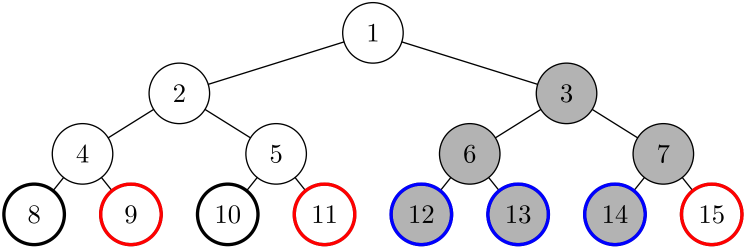
Stateful Subset Difference Example



Users at nodes 8, 9, 10, 12, and 13 were members and have state key K_s. User at node 9 leaves, user at 14 joins.
Broadcast:

KTH Computer Science and Communication

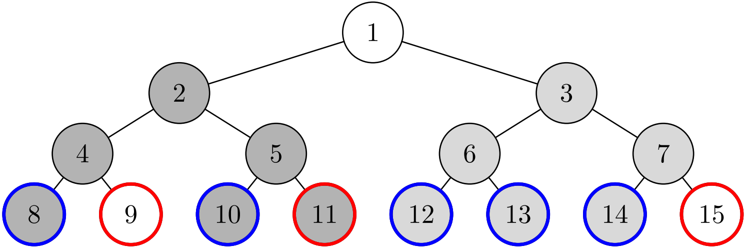
Stateful Subset Difference Example



Users at nodes 8, 9, 10, 12, and 13 were members and have state key K_s. User at node 9 leaves, user at 14 joins.
Broadcast:

$$\mathrm{E}_{K_{3,15}}(K_e), \mathrm{E}_{K_e}(K_g, K_s')$$

Stateful Subset Difference Example



Users at nodes 8, 9, 10, 12, and 13 were members and have state key K_s. User at node 9 leaves, user at 14 joins.
Broadcast:
$\mathrm{E}_{K_{2,9}}(R),\ \mathrm{E}_{K_{3,15}}(K_e), \mathrm{E}_{K_e}(K_g, K_s')$

The Good, the Bad, and the Ugly

- Bandwidth improves considerably

The Good, the Bad, and the Ugly

- Bandwidth improves considerably
- Not collusion-resistant

The Good, the Bad, and the Ugly

- Bandwidth improves considerably
- Not collusion-resistant
- Scheme becomes *stateful*

KTH Computer Science and Communication

The Good, the Bad, and the Ugly

- Bandwidth improves considerably
- Not collusion-resistant
- Scheme becomes *stateful* (not so bad after all)

Notation (again)

- r is the number of revoked users
- m is the number of members
- $u = r + m$ is the number of users

Notation (again)

- r is the number of revoked users
- m is the number of members
- $u = r + m$ is the number of users
- Δr is the number of members removed since last update
- Δm is the number of members added since last update

Performance Comparison

Scheme	Bandwidth
Stateful SD	$\Delta m + 2\Delta r + 1$
LKH	$2(\Delta m + \Delta r)\log m$
SD	$\min(2r + 1, m)$

- Bandwidth becomes linear in $\Delta m + \Delta r$ instead of in r

KTH Computer Science and Communication

Performance Comparison

Scheme	Bandwidth
Stateful SD	$\Delta m + 2\Delta r + 1$
LKH	$2(\Delta m + \Delta r)\log m$
SD	$\min(2r+1, m)$

- Bandwidth becomes linear in $\Delta m + \Delta r$ instead of in r
- We have also adapted the (p;c)-π scheme

KTH Computer Science and Communication

Performance Comparison

Scheme	Bandwidth
Stateful SD	$\Delta m + 2\Delta r + 1$
LKH	$2(\Delta m + \Delta r) \log m$
SD	$\min(2r + 1, m)$

- Bandwidth becomes linear in $\Delta m + \Delta r$ instead of in r
- We have also adapted the (p;c)-π scheme (but the worst-case bandwidth is a bit messy)

KTH Computer Science and Communication

Simulation Data

- Schemes were simulated using artificial data
- Used a *highly* dynamic system where at least 2% of users change state every round

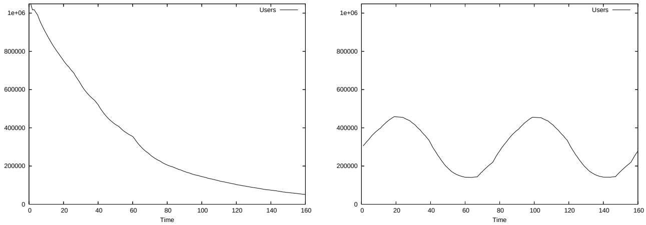


Figure: The Full-range and Sinus-shaped datasets

KTH Computer Science and Communication

Simulation Results

	Sinus		Full-range	
Scheme	Avg.	Max	Avg.	Max
Stateful SD	45	56	43	60
Stateful (1000;1)-π	30	39	28	34
LKH	218	269	241	394
Normal SD	222	296	170	305
Normal (1000;1)-π	154	180	114	180

Table: Keys used (in thousands) in the different schemes

KTH Computer Science and Communication

Old Commercial Threats

- Pirate decoders
  - Based on real user keys
  - Based on weakness in system

Old Commercial Threats

- Pirate decoders
  - Based on real user keys
  - Based on weakness in system
- Legal member redistributing
  - Group key
  - Content

Our Scheme is Not Collusion-resistant

- Assume Alice was a member, and Bob was not
- Alice leaves the group, and shares her state key with Bob

KTH Computer Science and Communication

Our Scheme is Not Collusion-resistant

- Assume Alice was a member, and Bob was not
- Alice leaves the group, and shares her state key with Bob
- If Bob was covered, he can compute the new group key and state key

Our Scheme is Not Collusion-resistant

- Assume Alice was a member, and Bob was not
- Alice leaves the group, and shares her state key with Bob
- If Bob was covered, he can compute the new group key and state key
- Mitigation:
  - Make it hard for users to know their keys
  - Use periodic updates with the underlying scheme, which will revoke all cheaters

KTH Computer Science and Communication

Summary

- A simple, generic modification of subset cover schemes

Summary

- A simple, generic modification of subset cover schemes
- Bandwidth is proportional to *change*, rather than number of revoked users

KTH Computer Science and Communication

Summary

- A simple, generic modification of subset cover schemes
- Bandwidth is proportional to *change*, rather than number of revoked users
- Not collusion-resistant

KTH Computer Science and Communication

Thank you! Questions?

gkreitz@kth.se

KTH Computer Science and Communication