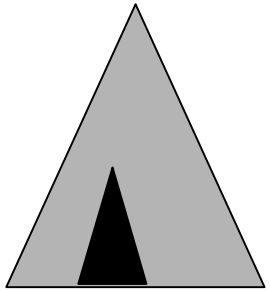
Revocation and Tracing Schemes for Stateless Receivers

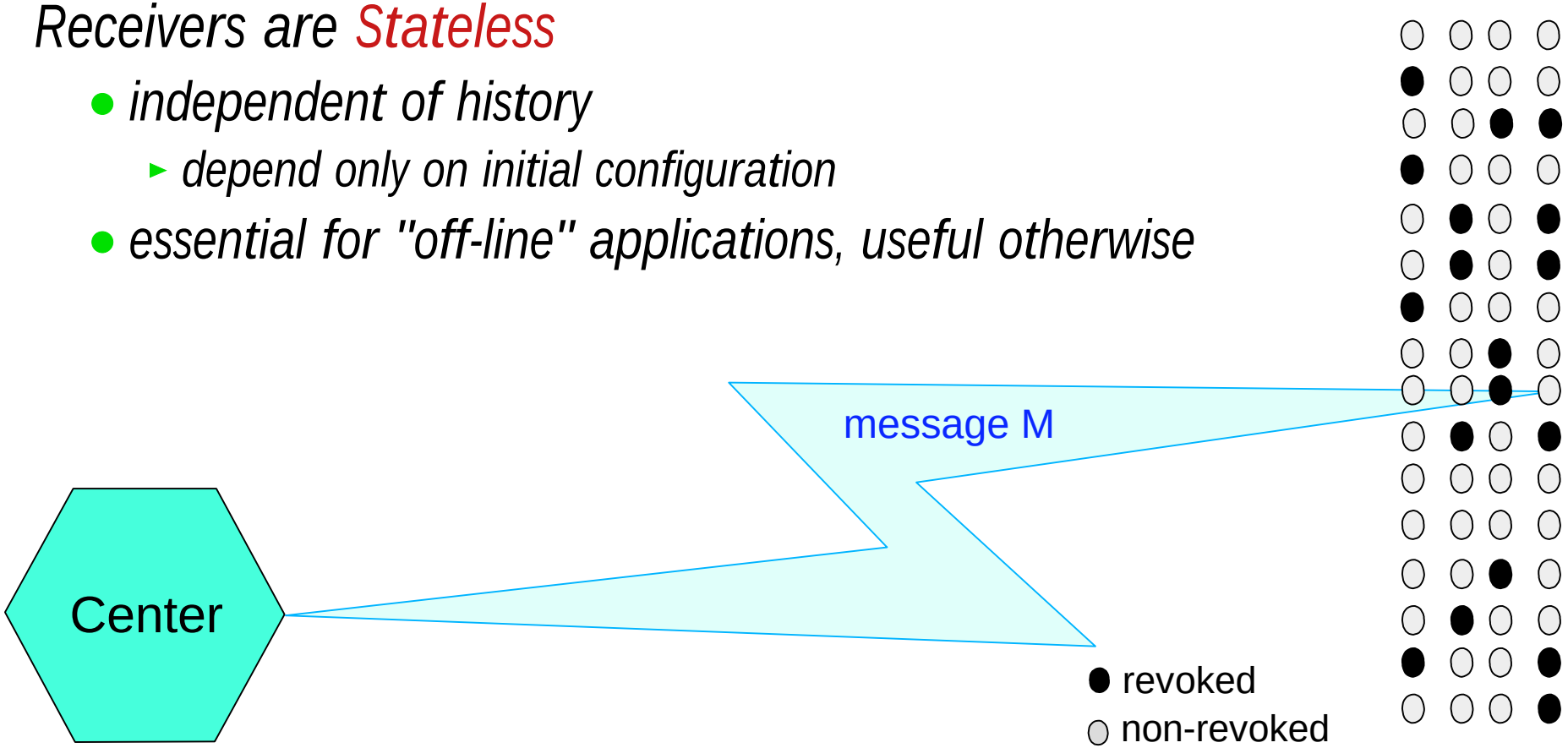
Dalit Naor
Moni Naor
Jeff Lotspiech

IBM Almaden Research Lab
Weizmann Institute, Stanford University



The Broadcast Encryption Problem

- **Center** *transmits a message to a* **large** *group*
- *A* **subset** *of users is revoked and should not decrypt the message*
  - *subset changes* **dynamically**
- *Receivers are Stateless*
  - *independent of history*
    - *depend only on initial configuration*
  - *essential for "off-line" applications, useful otherwise*



Preliminaries

- *Notation:*
  - *N - set of **n** users*
  - *R - set of **r** users whose privileges are to be revoked;*
- *Assumption: Stateless devices*
- *Goal: encrypt so that*
  - *a non-revoked user can decrypt correctly*
  - ***No** coalition of revoked users (of an arbitrary size) can decrypt*

New Subset-Cover Revocation and Tracing Algorithms

n - total no. of users

r - no. revocations

t - no. of traitors (illegal users)

Scheme	Message Length	# Keys per Device	Processing Time	# decrypt.	Message Length for t traitors
Complete Subtree	r log n/r	log n	log log n	1	t logn
Subset Difference	2r-1 1.25r (avg.)	$0.5 \log^2 n$	log n applications of a PRSG	1	5t

Components

A system consists of three parts:

1. **Scheme Initiation** - *a method to assign secret information to devices, I_u to user $u \in N$.*
2. **The broadcast algorithm** - *given a message M and a set R of users to be revoked, output a ciphertext message to broadcast to all.*
3. **A decryption algorithm** -
   - *a non-revoked device should produce M from ciphertext.*
   - *Decryption should be based on the current message and the secret information only (i.e. stateless).*
   - *Impossible to produce M from ciphertext even when provided with the secret information of* **all** *revoked users.*

Subset - Cover Framework

An algorithm in the framework is defined by:

- *Underlying collection of subsets (of devices*

  $S_1, S_2, \ldots, S_w$ $\quad S_j \subseteq N$.

- *Each subset* S_j *has a long-lived key* L_j *associated with it;*
  - *A device* $\mathbf{u} \in S_j$ *should be able to deduce* L_j *from its secret information* $\mathbf{I_u}$.
- *Given a revoked set* R*, the non-revoked users* $N \setminus R$ *are partitioned into disjoint subsets*

  $$S_{i_1}, S_{i_2}, \ldots, S_{i_m} \qquad (N \setminus R = \cup_j S_{i_j})$$

  - *a session key* K *is encrypted m times with* $L_{i_1}, L_{i_2}, \ldots, L_{i_m}$

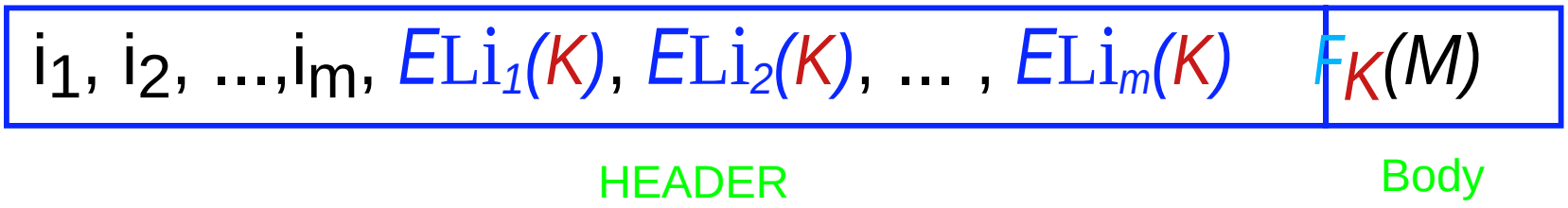
The Broadcast Algorithm

- *Choose a session key K*
- *Given R, find partition of N \ R into disjoint sets*

  $Si_1, Si_2, \ldots, Si_m$

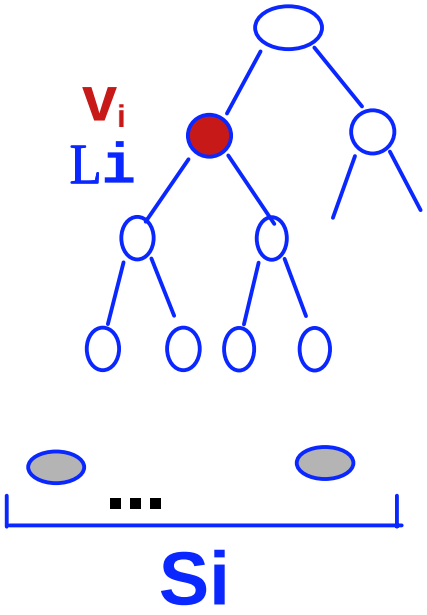
  $N \setminus R = \bigcup_j Si_j$

  with associated keys $Li_1, Li_2, \ldots, Li_m$
- *Encrypt message M*

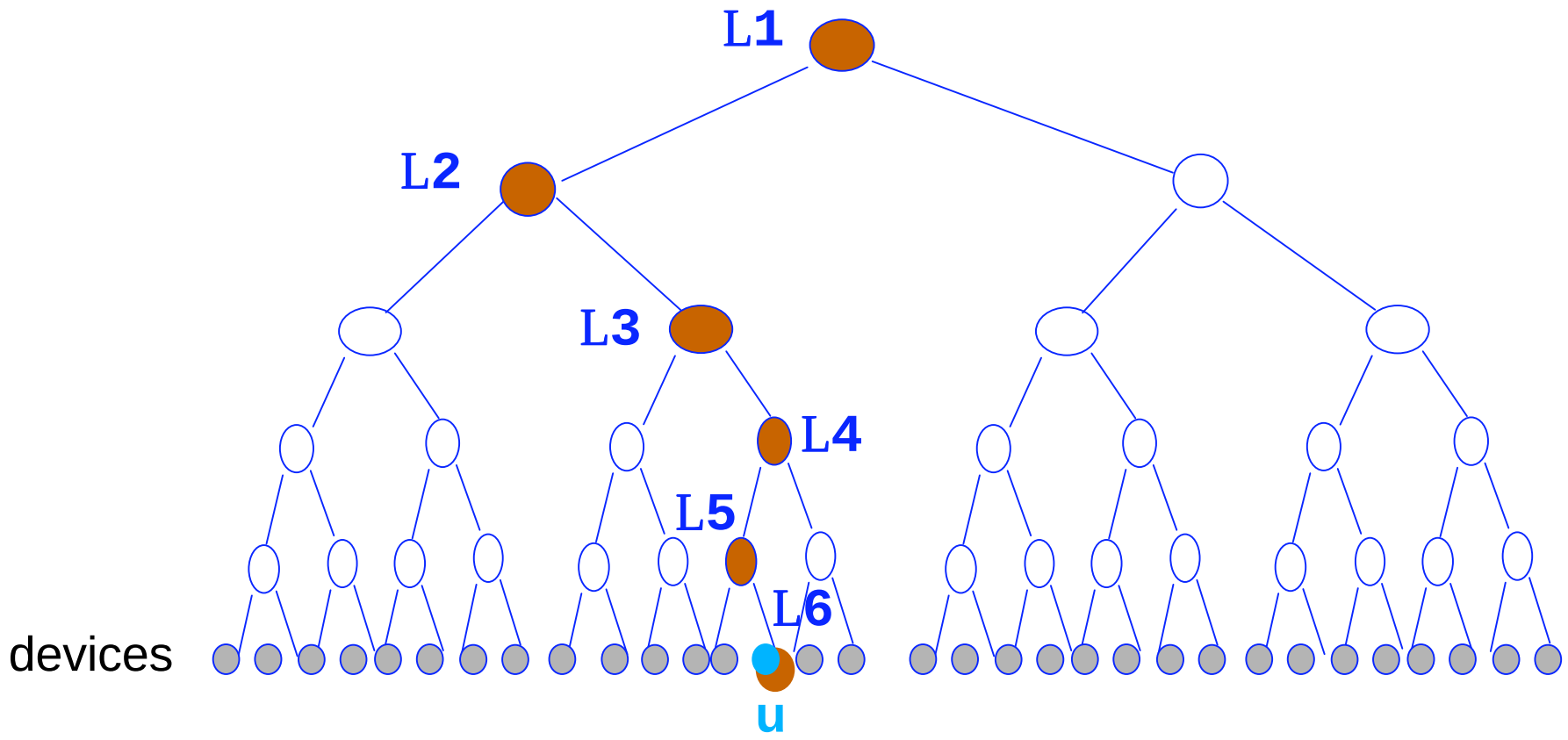


The Complete Subtree Method

- *Imagine a full-binary tree with N leaves*
  - *E.g. if $n=2^{32}$, a 32-levels complete binary tree*
- *Underlying Subsets* $S_1, S_2, \ldots, S_w$
  - *for node v_i in the full tree,*
    - *Si - set of all leaves in the subtree of v_i .*
  - *w = 2n-1*
- *Key assignment:*
  - *assign a key Li to every node v_i in the tree*
- *Device keys:*
  - *store all log n+1 keys along path to the root*
  - *E.g. if $n=2^{32}$, need 33 keys*



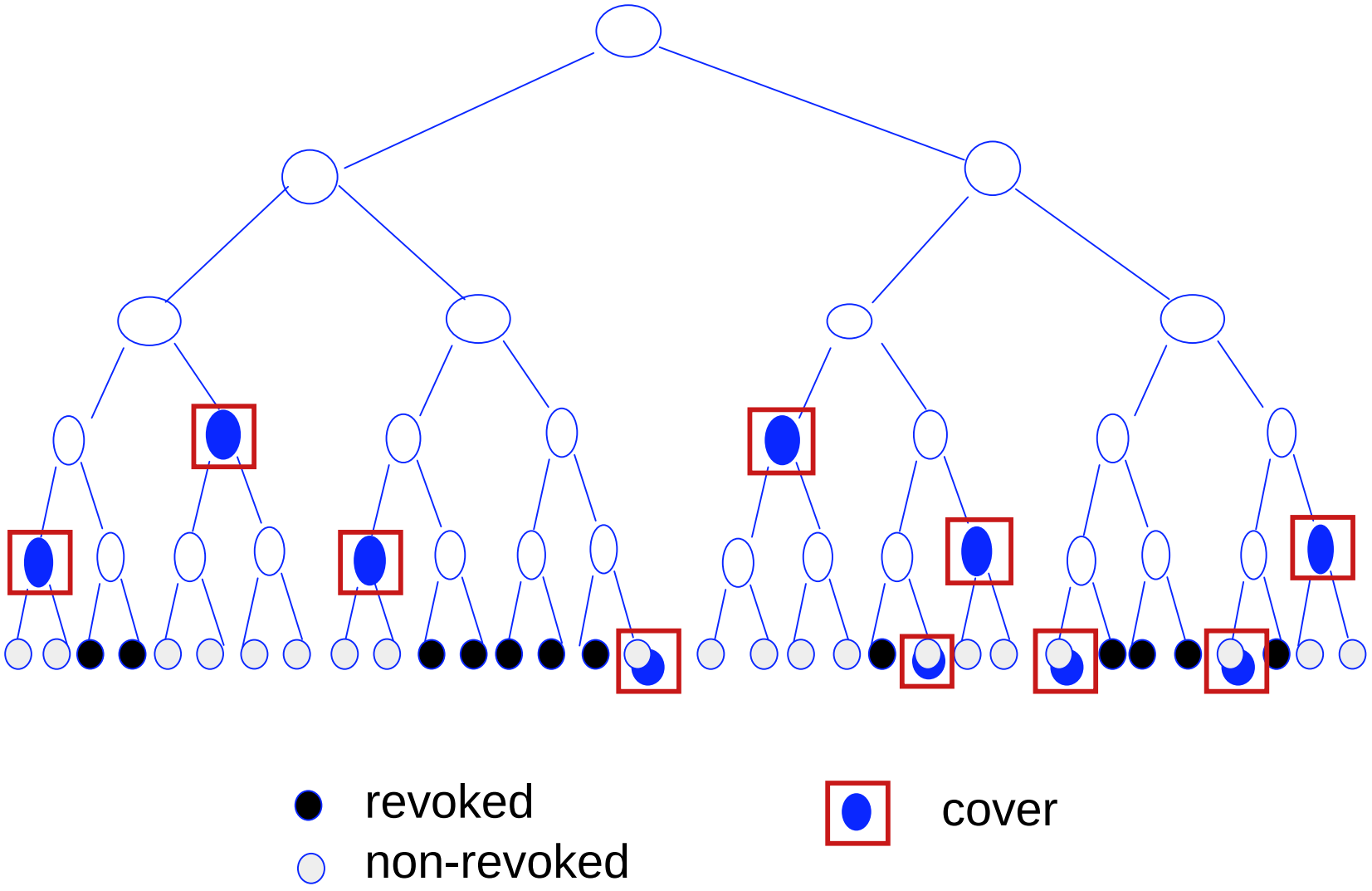
Complete Subtree: Key Assignment



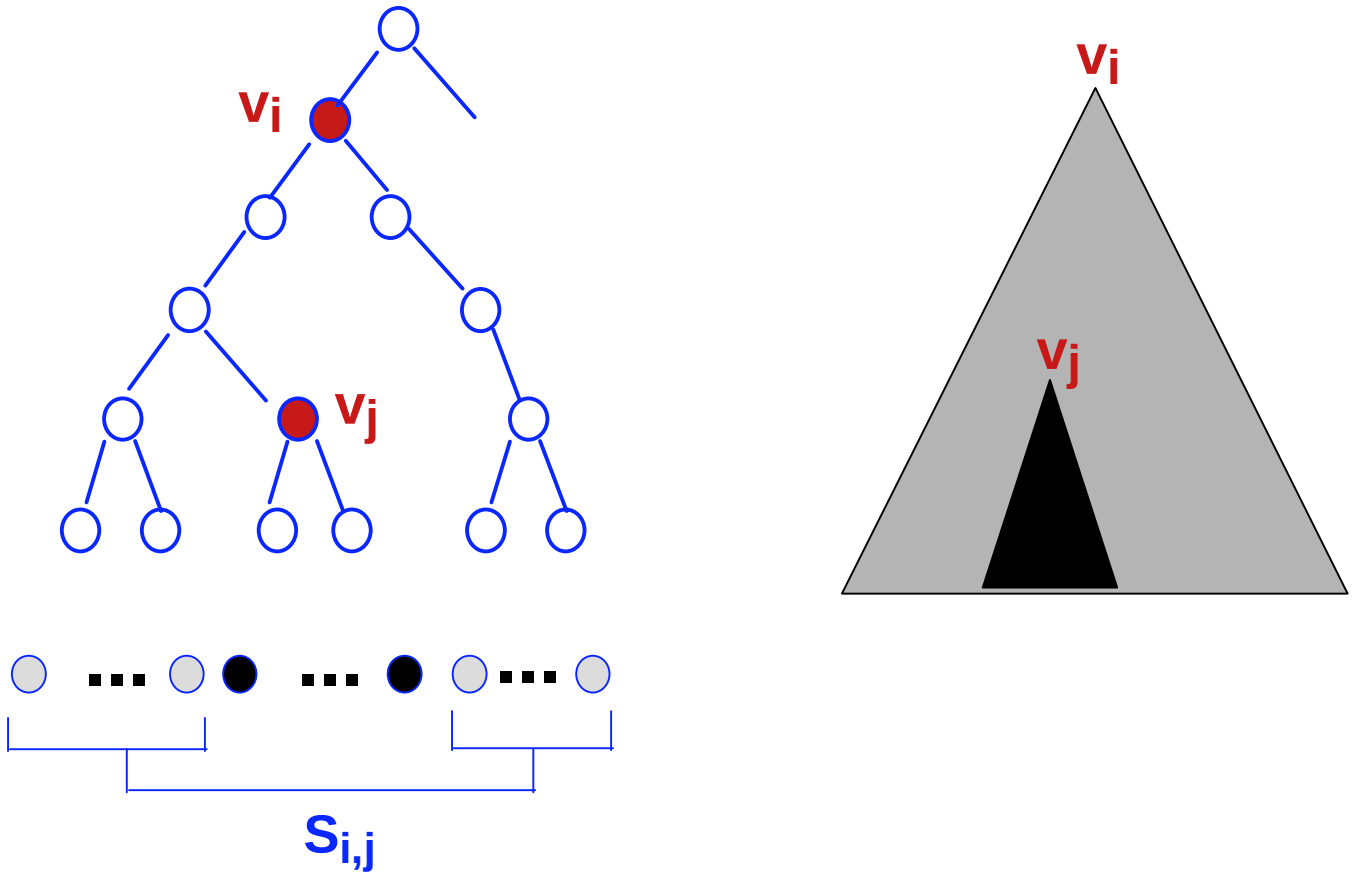
$$I_u = \{ L1, L2, L3, L4, L5, L6 \}$$

Subset Cover of non-revoked devices

Complete Subtree Method



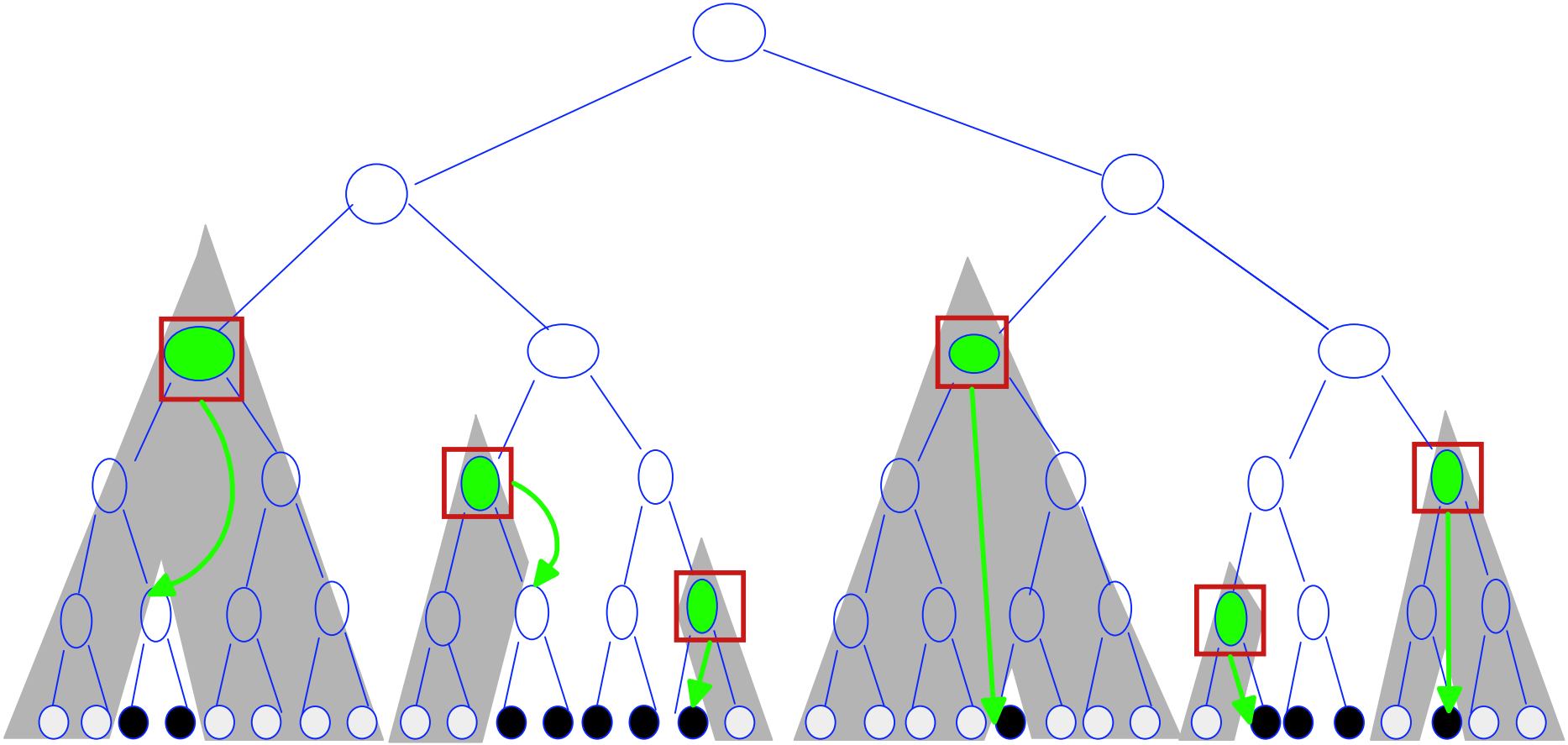
Subset Difference Definition



$S_{i,j}$ = Set of all leaves in the subtree of Vi but <u>not</u> in Vj

Subset Cover of Non-Revoked Devices

Subset-Difference Method



Cover is Very Small !!

Fundamental property:

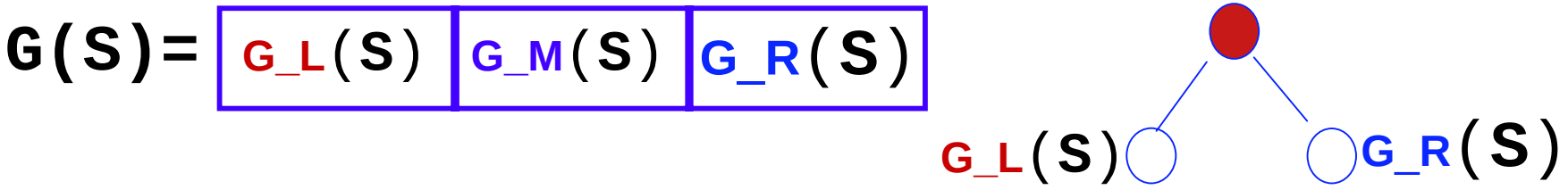
Size of the subset cover in the difference-subset method is

At most 2r-1 in the worst case

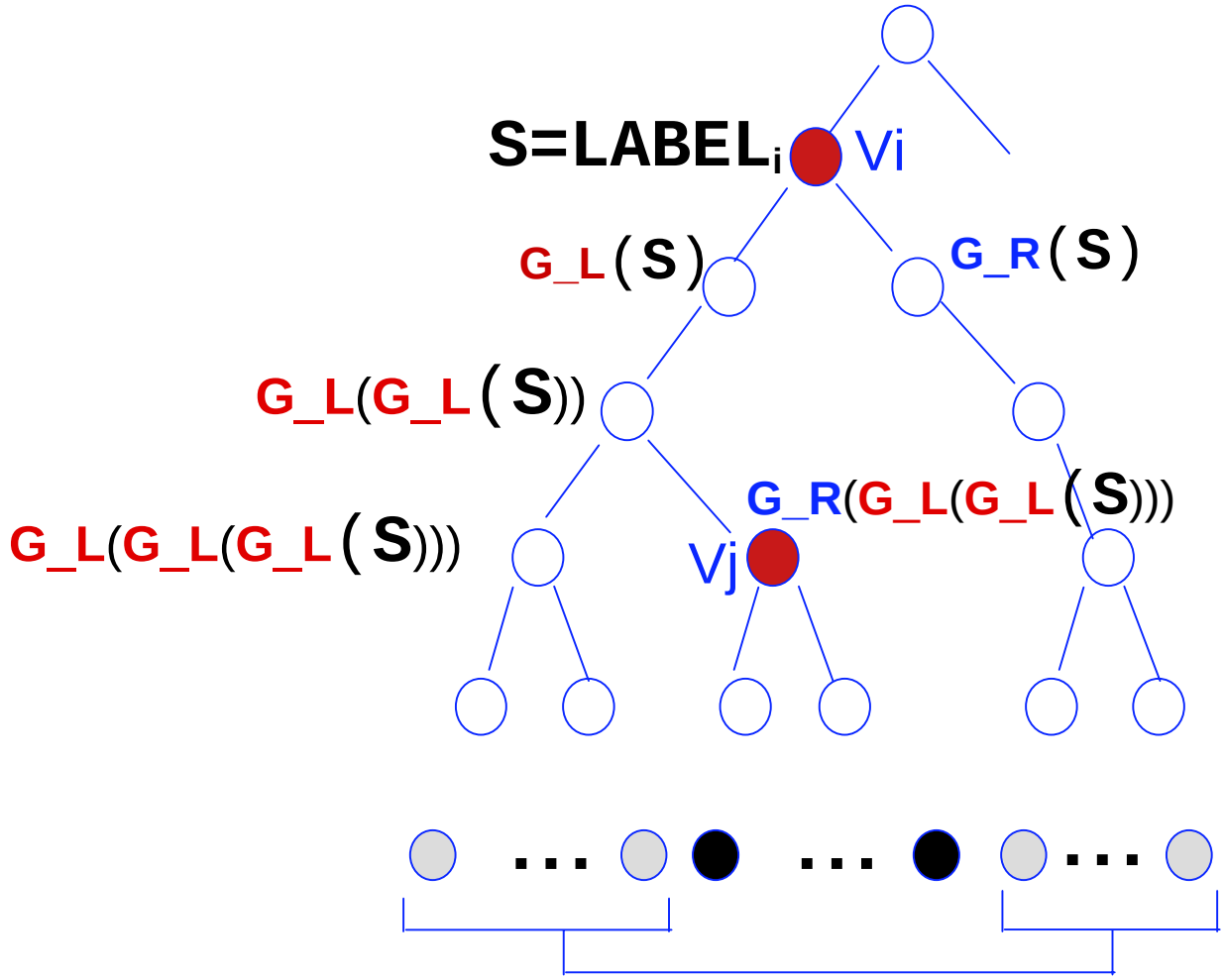
1.25r in the average case !

Key-Assignment
Subset-Difference Method

- *Naive approach to the key assignment:*
  - *assign a key* $L_{i,j}$ *to every pair* $[v_i, v_j]$ *in the tree*
  - *impractical: each device must store O(n) keys...*
- *Use G, a pseudo-random sequence generator that* **triples** *the input length (k → 3k) à la GGM*
- *Use G to derive a labeling process*
  - *S - label @ node,*
  - $G_L(S)$ *- label @ left child,* $G_R(S)$ *- label @ right child*
  - $G_M(S)$ *- key @ node.*



Key-Assignment
Subset-Difference Method

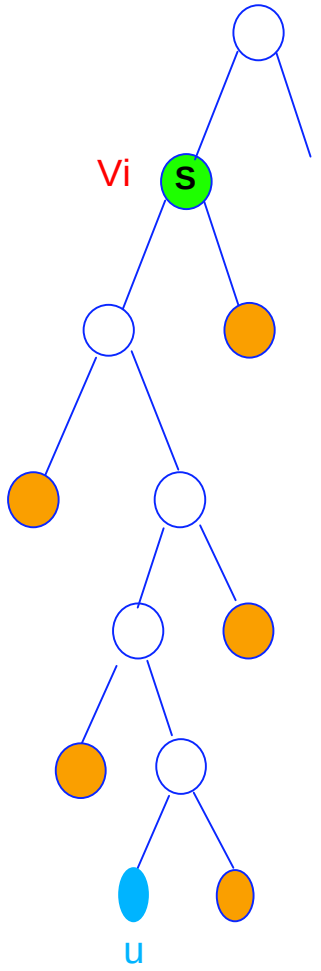


$LABEL_{i,j} = G\_R(G\_L(G\_L(L_i)))$

$L_{i,j} = G\_M(LABEL_{i,j})$

Providing Keys to Devices

- *A device corresponds to a leaf u in the tree*
- *For every Vi ancestor of u whose label is S*
  - *u receives all labels@nodes that are* **hanging off the path** *from Vi to u. These labels are all derived from S.*
- *u can compute all keys of the sets it belongs to rooted at Vi, and only them.*



Only 13 bytes per Single Revocation

- *For N= 2^{32} and 7-bytes session-key*
  - *total of 1.25 * 7 + 4 < 13 bytes/revocations*
  - *530 labels/device*