9th International Conference on Applied Cryptography and Network Security (ACNS '11)
Nerja (Málaga), Spain, June 7-10, 2011

Fighting Pirates 2.0

By

Paolo D'Arco and Ángel L. Pérez del Pozo

UNIVERSITÀ DI SALERNO

Universidad Rey Juan Carlos

Introduction

- In EUROCRYPT 2009, Billet and Phan presented *Traitors collaborating in public: Pirates 2.0.*
- This was a new attack model against tracing and revoking schemes.
- In this work we present measures to deal with some of these attacks.

1. Background

- Broadcast encryption
- CS and SD
- Traitor tracing

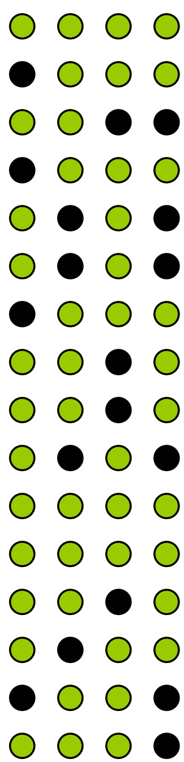
The Broadcast Encryption Problem

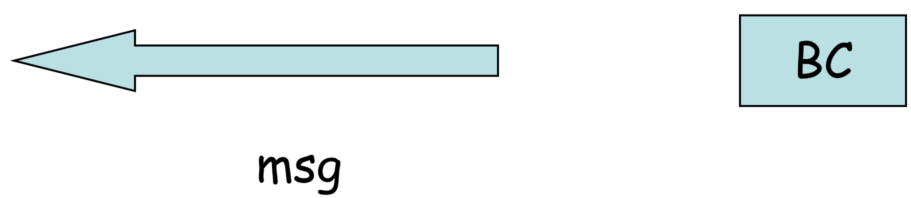
- A center BC broadcast a msg to a set U of N receivers

- A subset R of them are revoked and should not be able to decrypt the msg

    - R changes from time to time

    - We will focus on stateless receivers





● revoked
● non-revoked

Subset Cover Framework
[NNL01]

- Framework encapsulates many previous schemes

- Underlying collection of subsets (of users/devices)

$$S_1, S_2, \ldots, S_W \qquad\qquad S_j \subseteq U$$

- Each subset S_j is associated with a *long-lived* key L_j
  - A user $u \in S_j$ should be able to deduce L_j from its secret information sk_u

The Broadcast Algorithm

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

$$S_{i_1}, S_{i_2}, \ldots, S_{i_m}$$

$$U \setminus R = \bigcup S_{i_j}$$

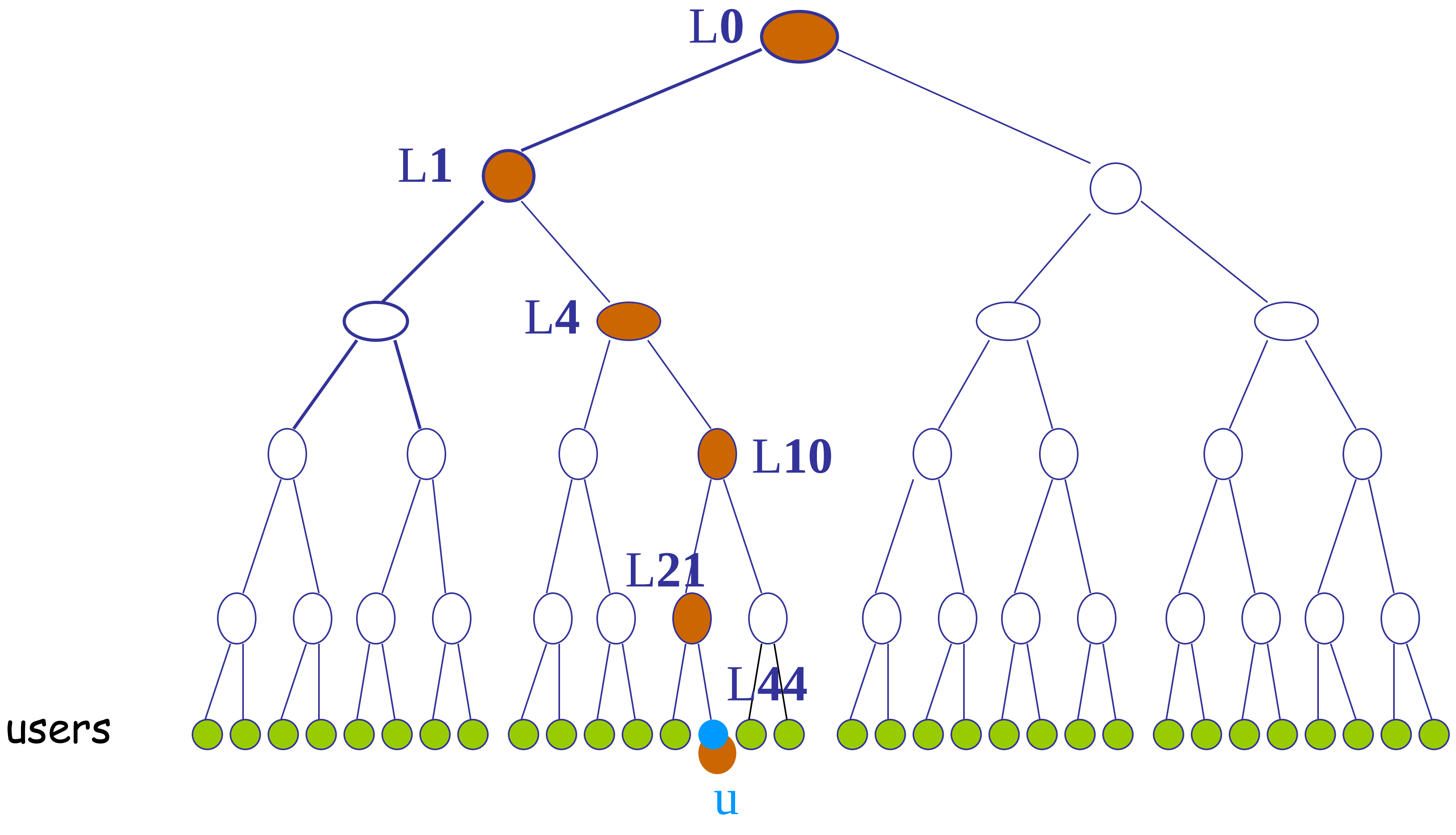
with associated keys $L_{i_1}, L_{i_2}, \ldots, L_{i_m}$

- Encrypt message M

$[i_1, i_2, \ldots, i_m]$, $C_1 = E_{L_{i_1}}(K)$, … , $C_m = E_{L_{i_m}}(K)$	$F_K(M)$

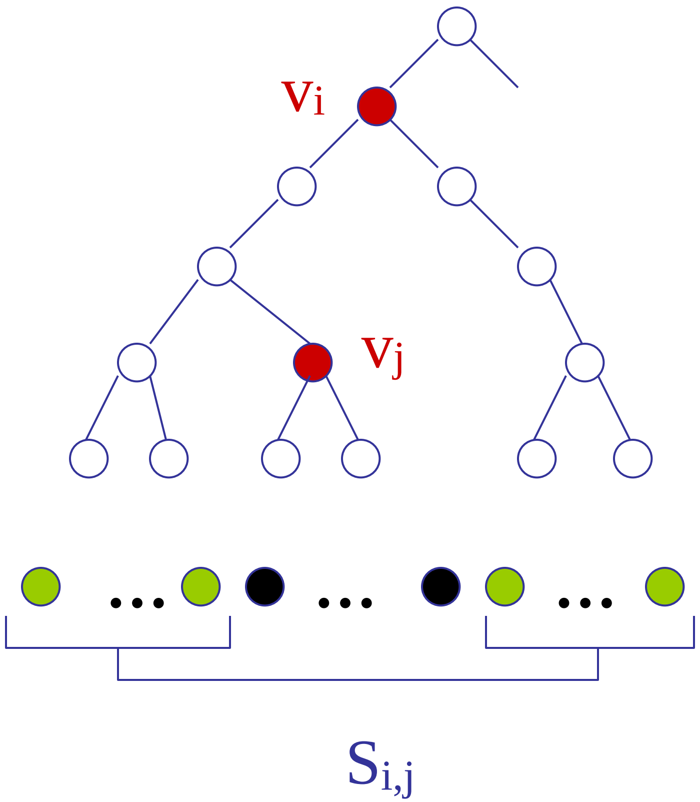
HEADER Body

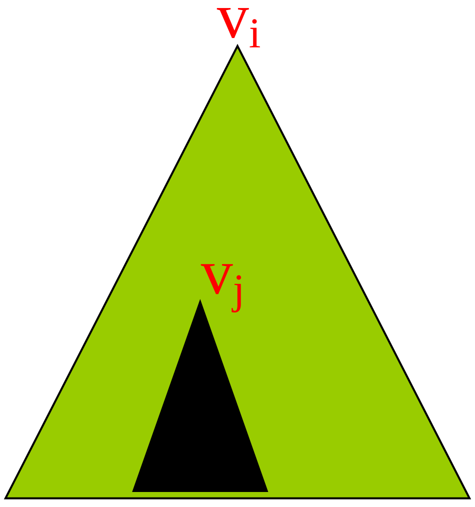
Complete Subtree (CS)



sk_u = { $(0,L_0)$, $(1,L_1)$, $(4,L_4)$, $(10,L_{10})$, $(21,L_{21})$, $(44,L_{44})$}

Subset Difference (SD)





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

Key-assignment for SD

- Naive key-assignment: each user must store too many keys, one for each S_{ij}
- To improve this, a pseudorandom generator is used for key derivation : each user stores only $O((\log N)^2)$ labels
- From labels and PRG, user covered by S_{ij} can derive key L_{ij}

Traitor tracing

- *traitors* : users that collude to produce a *pirate decoder*
- *tracing procedure* : from a pirate decoder the identity of at least one traitor is revealed
- CS and SD feature a tracing procedure:
  - a traitor is identified or
  - a new cover is computed (safe for the pirate decoder)

2. Pirates 2.0 attack

Pirates 2.0: basic features

- Public collusion.
- Partial contribution.
- Anonymity guarantee.
- Large coalitions.
- Imperfect decoders.

Pirates 2.0: the model

- *Contribution* *C* : publicly available set which collects the info traitors give
- *Extraction function* : function of the sk of a traitor which is added to *C*
- *Anonymity level* of a traitor T : # of users which could have contributed to *C* precisely the same info as T

Pirates 2.0: the schemes

Schemes attacked in [BP09]:

- subset cover framework
- analysis for CS and SD
- code based schemes

Our work: countermeasures for CS and SD

Pirates 2.0 attack on CS

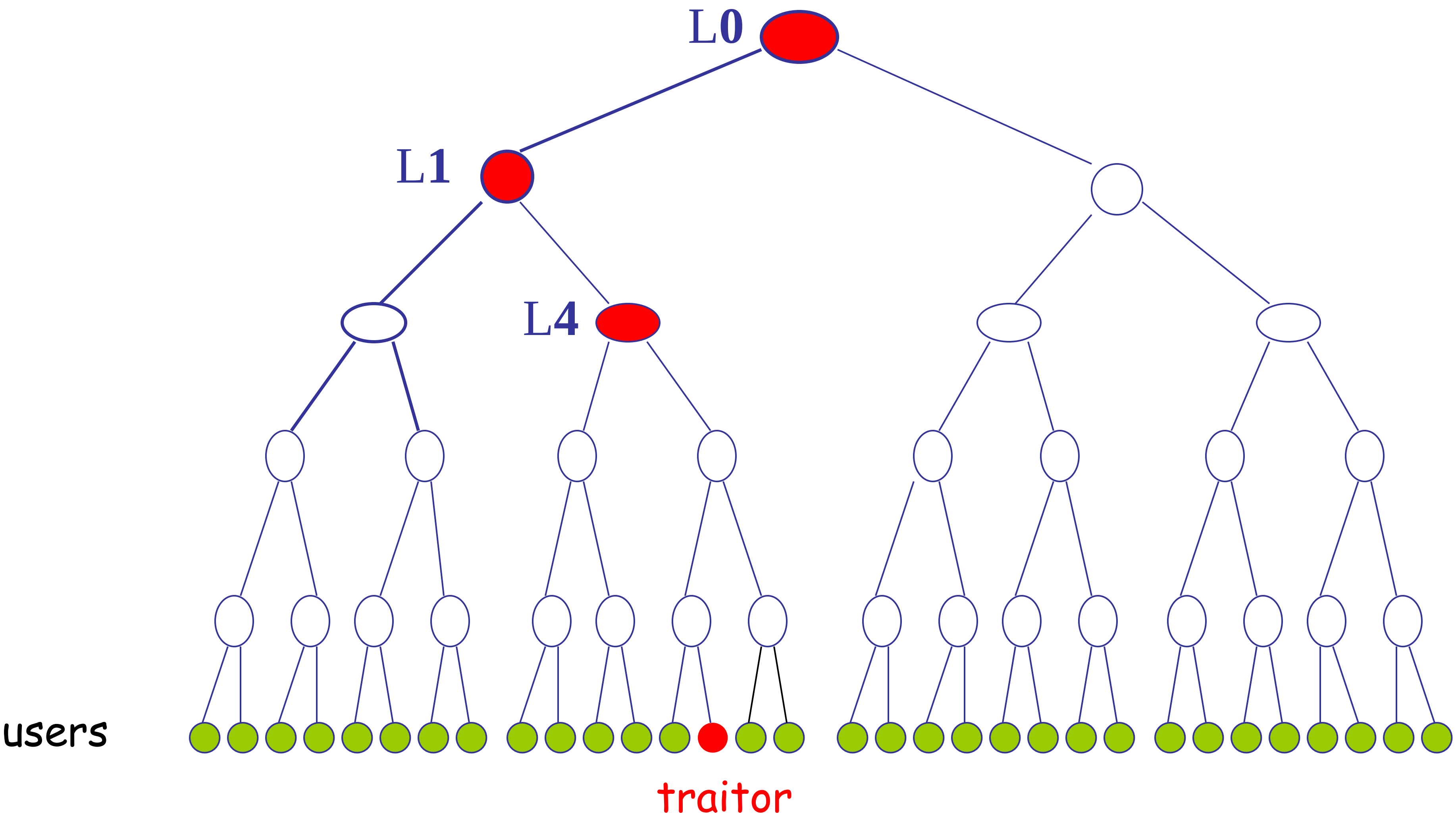
- Extraction functions are projections

  $sk_T=\{(i,L_i)\}_i \Rightarrow f_i(sk)=L_i$

- Traitors contribute with keys corresp. to the upper levels of the tree.

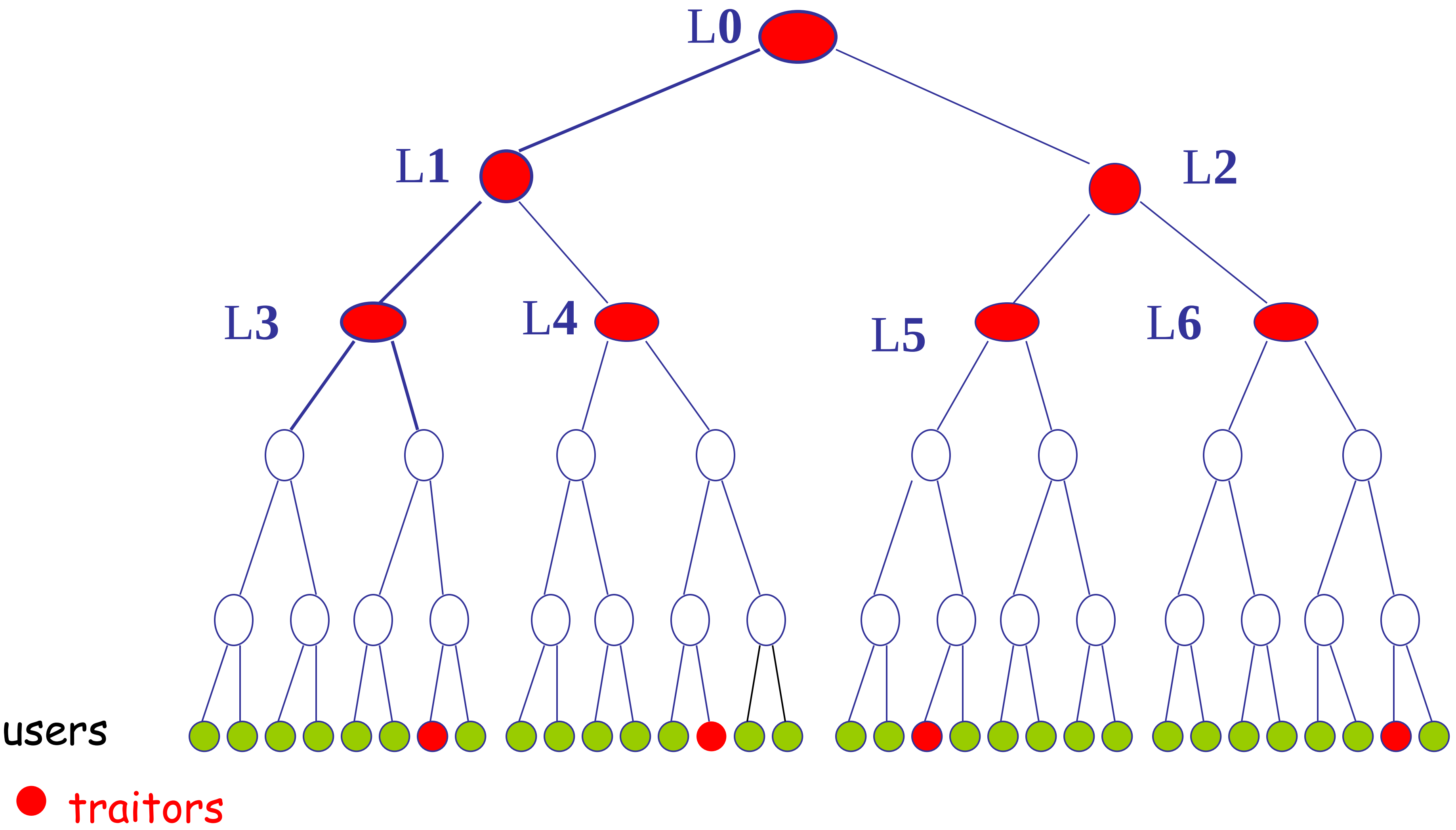
- These subtrees cover a large # of users $\Rightarrow$ high anonymity level

Contributed info (1 traitor)



contribution = $\{ L_0 , L_1 , L_4 \}$

Contributed info (>1 traitor)



contribution = { L_0 , … , L_6}

Pirates 2.0 attack on CS

Theorem [BP09] :

- system with N users
- r revoked users
- $d \log d$ randomly selected traitors
- length of ciphertext header < $d(N-r) / N$

Then:

- successful pirate decoder (high prob.)
- anonymity level for traitors: N / d

Analog result for SD

3. Partial measures

Partial measure for CS : hiding labels

- Attack is successful because users know the level of their keys.
- Idea: hide the level
- BC sends to user u covered by subtree S_i

  $(\pi(i), L_i)$ instead of (i, L_i)

where π is a secret permutation of labels

- Broadcast $(\pi(i), E_{L_i}(K))$

Partial measure for CS : hiding labels

Cons:

- By public collaboration, traitors can estimate the level of their keys.

Pros:

- A traitor must trust the others.
- Traitors lose the anonimity guarantee.
- "Cheap" to implement.

Partial measure for CS : or-based construction

- Idea: use the OR-protocols from [GSY99] to reduce anonimity level
- For each subtree S_i, BC fixes set of keys

  $K_i = \{L_{i1},\ldots,L_{im}\}$

  and a prob. dist. D_i over K_i
- User u covered by S_i receives a single key L_{ij} according to D_i
- All keys in K_i are used to broadcast

Partial measure for CS : or-based construction

Cons:

- Total # of gen. keys grows by m factor
- Ciphertext length grows by m factor

Pros:

- # keys per user remains the same
- anon. level is reduced
- anon. guarantee is lost (only probabilistic)

4. Hybrid CS and SD

Hybrid CS scheme: Idea

Combine two constructions:

- CS scheme from [NNL01].
- Polynomial-based scheme from [NP00].

Hybrid CS: Parameters

- $G = \langle g \rangle$: group of order q with hard DDH.
- threshold value $t > 0$
- (public) reconstruction values $\{I_1,\ldots,I_t\}$ in $Z_q \setminus \{0\}$
- User u gets I_u in $Z_q \setminus \{0, I_1,\ldots,I_t\}$

Hybrid CS: Setup

For each subtree S_i, BC

- chooses (secret) t-degree polymial

  $$P_i(x) \leftarrow_{\$} Z_q[x]$$

- sends to each user u covered by S_i

  $$(i, P_i(I_u))$$

Hybrid CS: Broadcast

For new session, BC

- chooses session key K
- computes a cover $S=\{S_i\}$ for leg. users
- for each subtree S_i in S:
  1. $r_i \leftarrow_\$ Z_q$
  2. $\forall\ j = 1,\ldots,t \quad d_{ij} := g^{r_i P_i(I_j)}$
  3. $K_i := g^{r_i P_i(0)}$
  4. broadcasts $(i, g^{r_i}, \{d_{ij}\}_j, E_{K_i}(K))$
- broadcasts $F_K(M)$

Hybrid CS: Decryption

Leg. user u, from

broadcast: $(i, g^{r_i}, \{d_{ij} := g^{r_i P_i(I_j)}\}, E_{K_i}(K))$

u info: $(i, P_i(I_u)), I_u$

(public) values: $\{I_1, \ldots, I_t\}$

computes the subtree key $K_i := g^{r_i P_i(0)}$ by "polynomial interpolation in the exponent".

Then recovers session key K

Hybrid SD scheme: Idea

Also combine the 2 constructions:

- SD scheme from [NNL01].
- Polynomial-based scheme from [NP00].

Not an immediate generalization of previous construction:

- We preserve the pseudorandom key generation which allows each user to store only $O((\log N)^2)$ labels.

Hybrid SD: Parameters

- $G = \langle g \rangle$: group of order q with hard DDH.
- threshold value $t > 0$
- (public) reconstruction values
  $\{I_1,\ldots,I_t\}$ in $Z_q \setminus \{0\}$
- User u gets I_u in $Z_q \setminus \{0, I_1,\ldots,I_t\}$

Hybrid SD: Setup

BC generates an instance of SD with Z_q as set for keys L_{ij}

Then, for each subtree S_i, BC

- chooses (secret) t-degree polymial
  $P_i(x) \leftarrow_{\$} Z_q[x]$
- sends to each user u covered by $S_{i,*}$
  $(i, P_i(I_u))$ and
  labels that SD assigns to him

Hybrid SD: Broadcast

For new session, BC

- chooses session key K
- computes a cover $S=\{S_{ij}\}$ for leg. users
- for each subtree S_{ij} in S:
  1. $r_i \leftarrow_{\$} Z_q$
  2. $\forall\ k = 1,\ldots,t \quad d_{ijk}:=g^{ri\ Pi(Ik)\ Lij}$
  3. $K_{ij}:=g^{ri\ Pi(0)\ Lij}$
  4. broadcasts $(ij,\ g^{ri},\ \{d_{ijk}\}_k,\ E_{Kij}(K))$
- broadcasts $F_K\ (M)$

Hybrid SD: Decryption

- Again, leg. user u recovers subtree key K_{ij} by "polynomial interpolation in the exponent".
- Then u recovers session key K

Hybrid CS and SD: Analysis

- Each pair (i, $P_i(I_u)$) determines univocally user u
- Therefore the Pirates 2.0 strategy that uses projection functions does not work anymore, as anonymity level drops to 1 (traitor can be traced)

Hybrid CS and SD: Analysis

- We also prove that our schemes satisfy the key-ind property in the Subset-Cover framework.
- This implies that they are secure against arbitrary coalitions of revoked users.
- They are also as efficient as CS and SD, in terms of key storage and bandwidth (with a t factor growth)

Hybrid CS and SD: Analysis

Price to pay:

- Broadcast and decryption computations are more expensive than ones in CS and SD (exponentiations)
- t+1 users covered by subtree S_i can compute and distribute $P_i(0)$, which allows to decrypt if S_i is used

Hybrid CS and SD: Analysis

Advantages:

- Pirates 2.0 with proj. func. are traced
- Secure against arb. coa. of rev. users
- Efficient as CS and SD both in:
  - Key storage
  - Bandwith (asymptotically)

Open problems

- It is of interest to formally define a security model which covers all possible Pirates 2.0 attacks
- and find and prove schemes (existing or new) to be secure in this extended model.

Thank you!

Questions?