Symmetric Key Broadcast Encryption

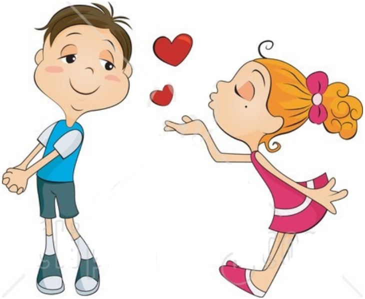
ICISS PhD Symposium

Who? Sanjay Bhattacherjee Palash Sarkar

From? Applied Statistics Unit
Indian Statistical Institute, Kolkata

When? December 17, 2013

ENTITIES



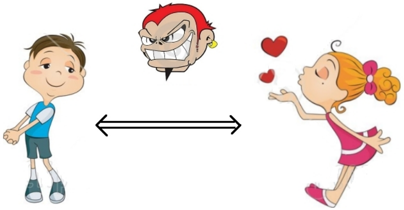
Alice and Bob

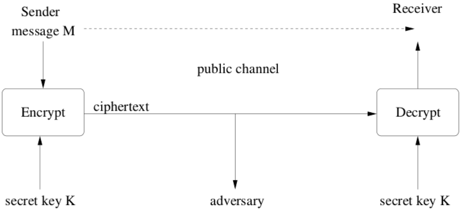
Entities

Oscar

Symmetric Key Cryptography

Framework



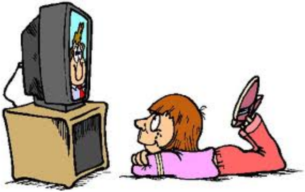


I will be talking about

Trees !!!

I will be talking about

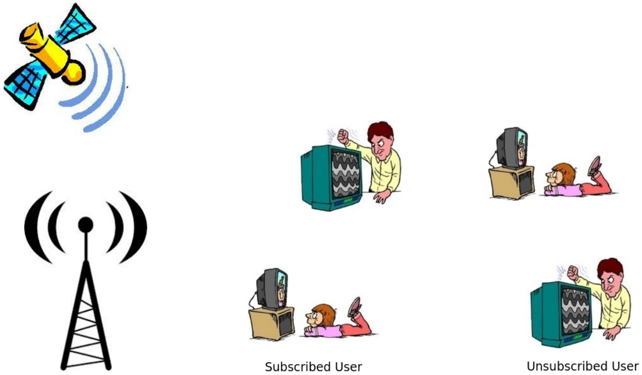
HD-DVD and Blu-ray players !!!



TV !!!

Digital Rights Management

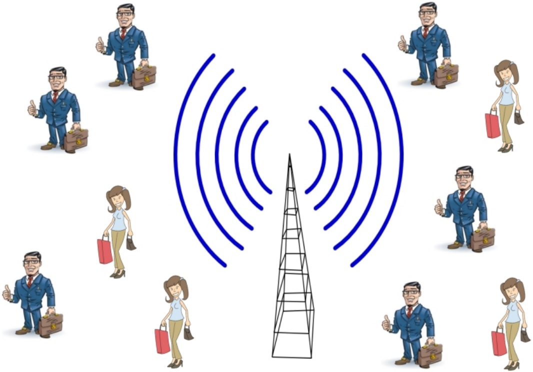
Pay-TV Only a subscribed user is able to decrypt a content.



(Symmetric Key) Broadcast Encryption

The center broadcasts encrypted messages to users

Users may be privileged or revoked.



Basic Solutions

1: Singleton Subset Scheme

Every user gets a unique key.
The message has to be encrypted once for every user.

2: Power Set Scheme

Every subset of users get a unique key.
The user has to store exponential number of keys.

Other Solutions?

Best of both worlds!

Assign keys to only selected subsets.
Call that collection $\mathcal{S}$.

Other Solutions?

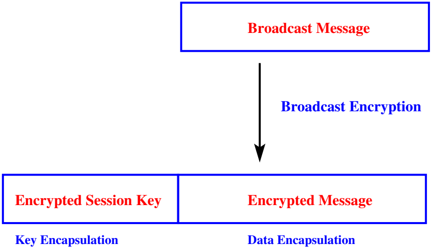
Best of both worlds!

Assign keys to only selected subsets.
Call that collection $\mathcal{S}$.

Divide the message into blocks (1 block per session).

(Header, Body) of a session

Encrypt the message with a session key.
Encrypt the session key for subsets $S_i \in \mathcal{S}$.



Designing BE schemes

Steps (1) Choose subsets for the collection $\mathcal{S}$; and
(2) Design the corresponding cover generation algorithm.

Subset Cover: $\{S_1, S_2, \ldots, S_h\}$, $S_i \in \mathcal{S}$ such that

$$\bigcup_{S_i \in S_c} = \mathcal{N} \setminus \mathcal{R}.$$

Steps

(1) Choose subsets for the collection $\mathcal{S}$; and
(2) Design the corresponding cover generation algorithm.

Subset Cover: $\{S_1, S_2, \ldots, S_h\}$, $S_i \in \mathcal{S}$ such that

$$\bigcup_{S_i \in S_c} = \mathcal{N} \setminus \mathcal{R}.$$

Other factors: full resilience, traitor tracing, etc.

Designing BE schemes

Steps

(1) Choose subsets for the collection $\mathcal{S}$; and
(2) Design the corresponding cover generation algorithm.

Subset Cover: $\{S_1, S_2, \ldots, S_h\}$, $S_i \in \mathcal{S}$ such that

$$\bigcup_{S_i \in S_c} = \mathcal{N} \setminus \mathcal{R}.$$

Other factors: full resilience, traitor tracing, etc.

Efficiency Parameters

(1) User storage,
(2) Header length,
etc.

Outline

Our Results

on the Subset Difference (SD) scheme

on the Layered Subset Difference (LSD) scheme

In Submission

Generalization of the NNL-SD scheme

k-ary tree SD scheme

The Subset Difference (SD) scheme

The SD Scheme

... is the most popular BE scheme

It has been suggested by the Advanced Access Content System (AACS) standard for DRM in optical discs (Blu-ray, HD-DVD)

Legitimate Disc Player

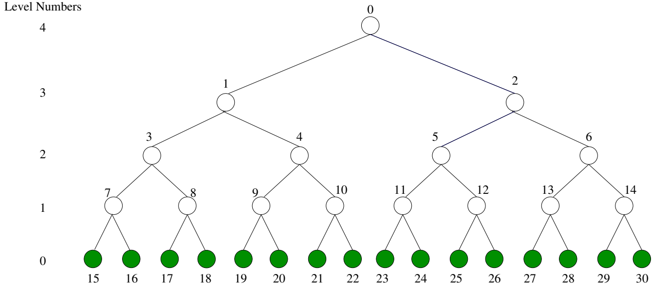
Pirate and Pirated Player

... due to Naor-Naor-Lotspiech (CRYPTO, 2001)

assumes an underlying full binary tree

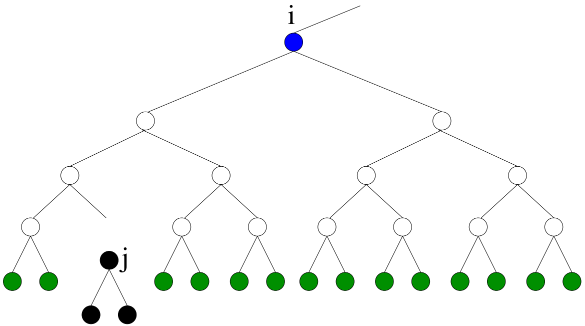
... DUE TO NAOR-NAOR-LOTSPIECH (CRYPTO, 2001)

assumes an underlying full binary tree



Subset Difference (SD) Subset

$S_{i,j} = \mathcal{T}_i \setminus \mathcal{T}_j$: has all users that are in $\mathcal{T}_i$ but not in $\mathcal{T}_j$



Collection $\mathcal{S}$

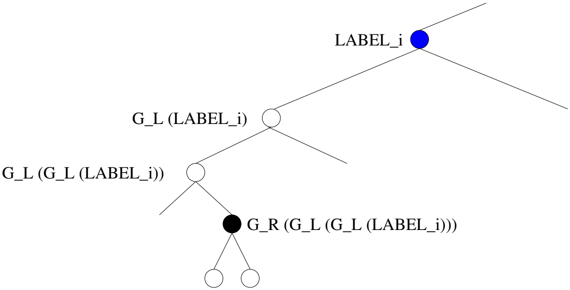
has all subsets $S_{i,j}$ such that $j(\neq i)$ is in the subtree $\mathcal{T}_i$.

Key Assignment

Pseudo-random generator (PRG)

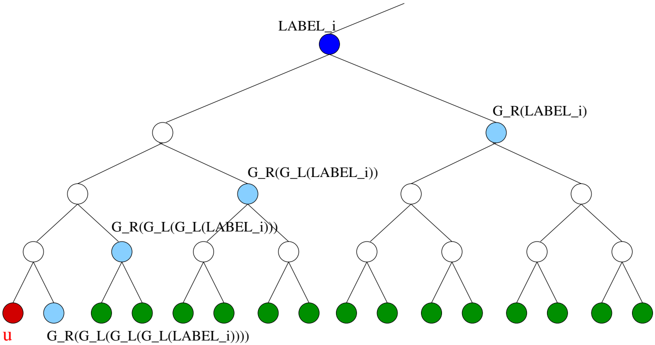
$G : \{0,1\}^k \rightarrow \{0,1\}^{3k}$

$G(seed) = G_L(seed) || G_M(seed) || G_R(seed)$



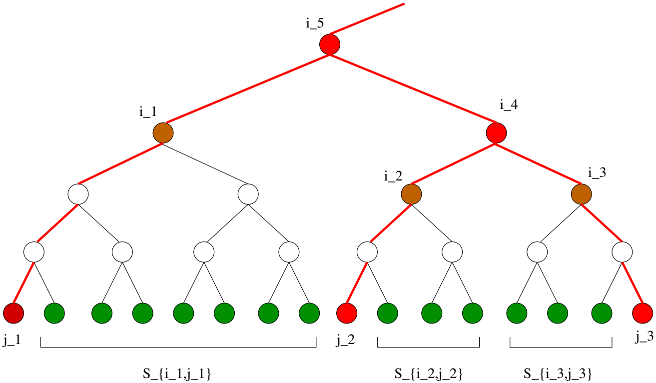
Key of $S_{i,j}$: $L_{i,j} = G_M(G_R(G_L(G_L(LABEL_i))))$

User Storage



User u stores: for every $\mathcal{T}_i$ to which it belongs, the derived labels of nodes "falling-off" from the path between i and u, derived from $LABEL_i$.

Subset Cover Finding Algorithm



$$S_{i,j} = \mathcal{T}_i \setminus \mathcal{T}_j$$

Header Length

Build the Header

Encrypt the session key K with long-lived key $L_{i,j}$ of each $S_{i,j} \in \mathcal{C}$

F_{K}(M)	E_{L_1}(K)	···	E_{L_h}(K)

Encrypted Message | Header

Header Length $h = |\mathcal{C}|$

NNL-SD scheme: Summary

Important Parameters:

User storage needed: $O(\log^2(n))$
Header Length in the worst case: $2r - 1$

where...
$|\mathcal{N}| = n$
$|\mathcal{R}| = r$

By [EOPR08], the expected header length is a good estimate of the communication cost.

Problem: For a given n and r, what is the expected header length?

Random Experiment

Choose r users out of n uniformly at random
without replacement
... and revoke them!

This gives a random (n, r)-revocation pattern

Random variable: $X_{n,r} \in \{0, \ldots, 2r-1\}$
(Header length due to the random (n, r)-revocation pattern)

$X^i_{n,r}$: EVENT AT NODE i

$X_{n,r}$ FROM $X^i_{n,r}$

Random variable $X^i_{n,r}$: $= 1$ if some $S_{i,j} \in C$;
$= 0$ otherwise

Since each $X^i_{n,r}$ follow Bernoulli distribution,

$$E[X^i_{n,r}] = \Pr[X^i_{n,r} = 1]$$

$$X_{n,r} = \sum_i X^i_{n,r}$$

$$E[X_{n,r}] = \sum_i \Pr[X^i_{n,r} = 1]$$

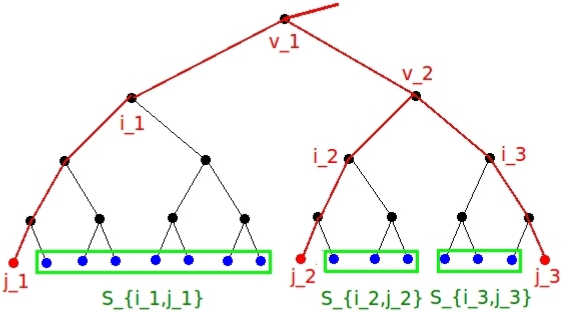
(By linearity of expectation)

EVENT: $X^i_{n,r} = 1$

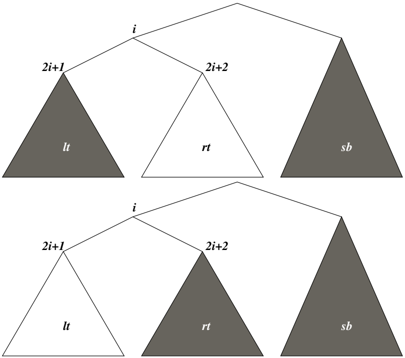
OCCURS WHEN...

There is at least one revoked node in each of the following:

- The sibling subtree of $\mathcal{T}_i$
- Exactly one child subtree of $\mathcal{T}_i$



EVENT: $X^i_{n,r} = 1$



$$\Pr[X^i_{n,r} = 1] \;=\; \Pr[S^i \wedge L^i \wedge \overline{R^i}] \;+\; \Pr[S^i \wedge R^i \wedge \overline{L^i}]$$

S^i is the event that there is at least one revoked user in the sibling subtree of i.

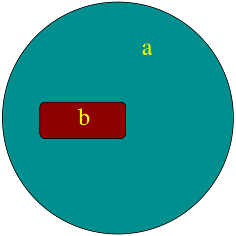
Probability: $\Pr[S^i \wedge R^i \wedge \overline{L^i}]$

$$
\begin{aligned}
\Pr[S^i \wedge R^i \wedge \overline{L^i}] &= \Pr[S^i \wedge R^i | \overline{L^i}] \times \Pr[\overline{L^i}] \\
&= \left(1 - \Pr[\overline{S^i \wedge R^i} | \overline{L^i}]\right) \times \Pr[\overline{L^i}] \\
&= \ldots \\
&= \Pr[\overline{L^i}] - \Pr[\overline{S^i} \wedge \overline{L^i}] \\
&\quad - \Pr[\overline{R^i} \wedge \overline{L^i}] \\
&\quad + \Pr[\overline{S^i} \wedge \overline{R^i} \wedge \overline{L^i}]. \qquad (1)
\end{aligned}
$$

It can be verified that (1) holds even if $\Pr[\overline{L^i}] = 0$.

PROBABILITY: $\eta_r(a, b)$

The probability of choosing r elements from a set of a elements such that b out of these a elements are never chosen:



So, if $b \geq a - r + 1$, then $\eta_r(a, b) = 0$ by definition.
Else, for $0 < b < a - r + 1$,

$$\eta_r(a, b) = \frac{\binom{a-b}{r}}{\binom{a}{r}}.$$

Probability: $\Pr[X^0_{n,r} = 1]$

Let the left and right subtrees of node i have λ_{2i+1} and λ_{2i+2} leaves respectively.
Let the sibling subtree have λ_s leaves.

$$\Pr[X^0_{n,r} = 1] = \eta_r(n, \lambda_1) + \eta_r(n, \lambda_2). \tag{2}$$

$$\begin{aligned}\Pr[X^i_{n,r} = 1] = {} & \eta_r(n, \lambda_{2i+1}) + \eta_r(n, \lambda_{2i+2}) \\ & - \eta_r(n, \lambda_s + \lambda_{2i+1}) \\ & - \eta_r(n, \lambda_s + \lambda_{2i+2}) \\ & - 2\eta_r(n, \lambda_{2i+1} + \lambda_{2i+2}) \\ & + 2\eta_r(n, \lambda_s + \lambda_{2i+1} + \lambda_{2i+2}). \end{aligned} \tag{3}$$

Algorithm to compute $E[X_{n,r}]$

Computing $\Pr[X^i_{n,r} = 1]$ for each node i in $\mathcal{T}^0$, gives $E[X_{n,r}]$

An $O(r \log n)$ time and $O(1)$ space algorithm

Our Work

Improvement

Our CTSD Scheme allows arbitrary number of users
... hence improved upon the transmission overhead

Analysis

- Detailed combinatorial analysis
  - A dynamic programming algorithm to compute $N(n, r, h)$.
  - Maximum header length for the CTSD scheme: $\min\left(2r-1, \left\lfloor \frac{n}{2} \right\rfloor, n-r\right)$.
  - Given r, find n_r.
  - Generating function for the sequence $N(n, r, h)$ for $n = 2^{\ell_0}$.
- An $O(r \log n)$ algorithm to compute $E[X_{n,r}]$.
  This technique can/has been extended for all tree-based BE schemes we have worked on.
- Theoretical support for the tighter upper bound of 1.25r for $E[X_{n,r}]$

The paper

Sanjay Bhattacherjee and Palash Sarkar. Complete tree subset difference broadcast encryption scheme and its analysis. *Des. Codes Cryptography*, 66(1-3):335-362, 2013.

Outline

Our Results

- on the Subset Difference (SD) scheme
- on the Layered Subset Difference (LSD) scheme

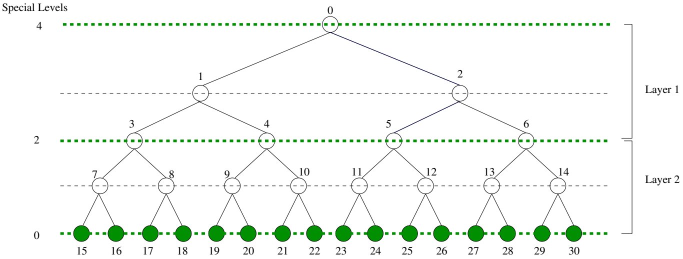
In Submission

- Generalization of the NNL-SD scheme
- k-ary tree SD scheme

Layered Subset Difference Scheme

... due to Halevy-Shamir (CRYPTO, 2002)

Some levels are marked as *"special"*.



A choice of special levels is called a layering strategy.

Important Parameters

NNL-SD scheme

User storage needed: $O(\log^2(n))$
Maximum Header Length: $2r - 1$

HS-LSD scheme

User Storage needed: $O(\log^{3/2} n)$
Maximum header length: $4r - 2$.

Scheme 1: Storage Minimal Layering

General Layering Strategy L

Denoted by the special levels
$\ell_0 > \ell_1 > \ldots > \ell_{e-1} > \ell_e = 0$.
Let $\mathbf{L_e} = (\ell_0, \ldots, \ell_e)$.

Storage Minimal Layering Strategy

SML(ℓ_0): a layering strategy that needs minimum storage among all possible layering strategies for a tree with ℓ_0 levels.

#SML(ℓ_0): storage due to SML(ℓ_0).

MINIMUM FOR A FIXED e

$\mathsf{SML}(e, \ell_0)$: a storage minimal layering *using exactly e layers.* Hence,

$$\#\mathsf{SML}(e, \ell_0) = \min_{\mathbf{L_e}} \mathsf{storage}(\mathbf{L_e}) \tag{4}$$

where the minimum is over all possible layering strategies $\mathbf{L_e}$ with e layers.

OVERALL MINIMUM FOR ALL e

The overall minimum is

$$\#\mathsf{SML}(\ell_0) = \min_{1 \leq e \leq \ell_0} \#\mathsf{SML}(e, \ell_0). \tag{5}$$

Finding the SML

$$\#\mathrm{SML}(e, \ell_0) = \min_{1 \le \ell_1 < \ell_0} \ell_0 + \frac{(\ell_0-\ell_1)(\ell_0-\ell_1-1)}{2} + \#\mathrm{SML}(e-1, \ell_1). \quad (6)$$

Algorithm

A simple $O(\ell_0^3)$ time dynamic programming algorithm computes SMLs.

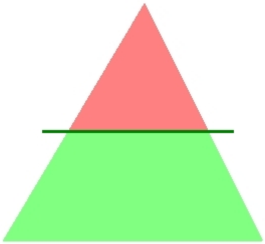
Summary

(1) Storage is reduced.
(2) For practical r, even the expected header length reduces.

Scheme 2: Constrained Minimization of User Storage

Analysis

For a fixed n and r, find the level in the tree whose contribution to the header length is maximum.



Level selection

The maximum occurs for some level $\ell \leq \ell_0 - \lfloor \log_2 r \rfloor$.
For levels $> \ell_0 - \lfloor \log_2 r \rfloor$, the contribution is quite small.

Constrained Minimization of User Storage

The Scheme

(1) Make level $\ell_0 - \lfloor \log_2 r \rfloor$ special. Level 0 is also special.
(2) No level $0 < \ell < \ell_0 - \lfloor \log_2 r \rfloor$ is made special.
(3) The root level is not made special.

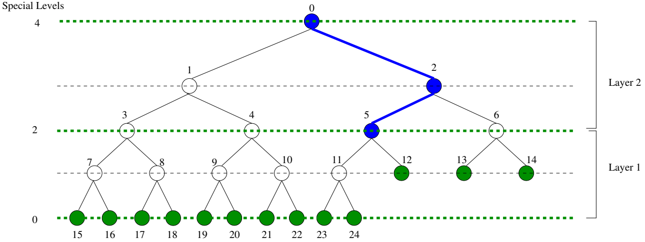
We call this the *constrained minimization layering (CML)* strategy.

Summary

(1) Storage is reduced ($<$ NNL-SD but $>$ e-HS-LSD).
(2) Expected header length almost same as NNL-SD.

Tackling arbitrary number of users

The Complete Tree LSD (CTLSD) scheme



Header length analysis

Maximum header length: $\min\left(4r-3, \left\lceil \frac{n}{2} \right\rceil, n-r\right)$.
Algorithm to compute the expected header length for a given n, r and $\mathbf{L}$.

THE PAPER

Sanjay Bhattacherjee and Palash Sarkar. Analysis and trade-offs for the (complete tree) layered subset difference broadcast encryption scheme. *IEEE Transactions On Computers*, 99(PrePrints):1, 2013.

Outline

Our Results

on the Subset Difference (SD) scheme

on the Layered Subset Difference (LSD) scheme

In Submission

Generalization of the NNL-SD scheme

k-ary tree SD scheme

Generalization of the NNL-SD scheme

Intuition

Header length and user storage

...depend on the collection $\mathcal{S}$.

Hierarchy of Optimization

Singleton Subset scheme $\rightarrow$ Power Set Scheme

(by varying $\mathcal{S}$)

Outline

Our Results

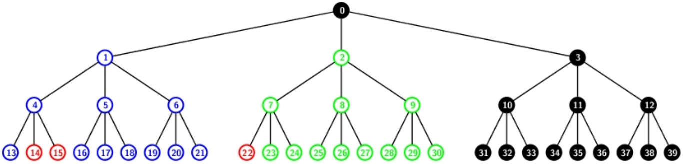
- on the Subset Difference (SD) scheme
- on the Layered Subset Difference (LSD) scheme

In Submission

- Generalization of the NNL-SD scheme
- k-ary tree SD scheme

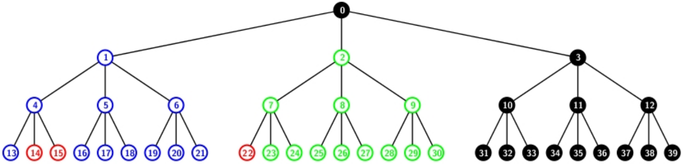
Generalization of the NNL-SD scheme

k-SD scheme assumes a full k-ary tree instead of binary.
Example for $k = 3$, $n = 27$.



Generalization of the NNL-SD scheme

k-SD scheme assumes a full k-ary tree instead of binary. Example for $k = 3$, $n = 27$.



Subsets are of the form $S_{i,\{j_1,\ldots,j_c\}}$ where nodes $j_1, \ldots, j_c$ are siblings in the subtree of i.

k-SD PERFORMANCE

USER STORAGE

$$(\chi_k - 2)\ell_0(\ell_0 + 1)/2$$

$\ell_0 = \lceil \log_k n \rceil$
$\chi_k = $ #cyclotomic cosets modulo $2^k - 1$.

MAXIMUM HEADER LENGTH

is $\min\left(2r - 1, n - r, \lceil n/k \rceil\right)$.

EXPECTED HEADER LENGTH

An $O(r \log n)$ time and $O(1)$ space algorithm computes the expected header length.

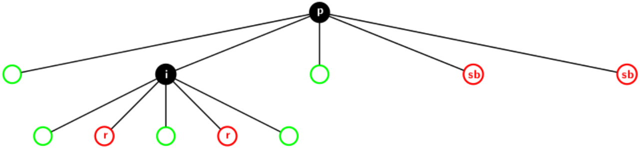


FIGURE: Computing $\Pr[X^i_{n,r} = 1]$

Good time to wake up!

How big is the Pay-TV industry?

BSkyB - Wikipedia, the free encyclopedia - Mozilla Firefox

W BSkyB - Wikipedia, the free en...

en.wikipedia.org/wiki/BSkyB

Finance [edit]

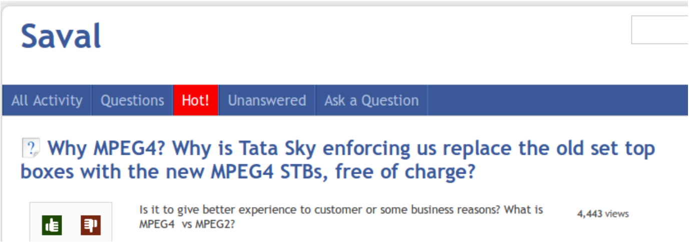
Financial results have been as follows:[1]

Turnover and profit or loss, by fiscal year

Year ended	Turnover (£m)	Profit/(loss) before tax (£m)	Net profit/ (loss)(£m)
30 June 2012	6,791	1,189	906
30 June 2011	6,597	1,014	810
30 June 2010	5,912	1,173	878
30 June 2009	5,359	456	259

BSkyB is the largest pay-TV broadcaster in the UK and Ireland with over 10 million subscribers.

Replacing Set Top Boxes for free?



From saval.in

Bandwidth is costly!

1 Answer

1 like 0 dislike

Initially TataSky had chosen MPEG2 as their technology for broadcasting, eventhough MPEG4 was predominat at tht time (in 2006?). MPEG4 allows for better compression, that means, for a given quality, the size of an MPEG4 video would take less-storage or less bandwidth for broadcasting. MPEG4 supports various encoding standards and H.264 is popular amonst them.

Best answer

Older IRDs (Integrated Receiver Decoders ie. STB) had chipsets that can understand and decode only MPEG2 content. However, HD Plus, HD STBs have always been having MPEG4 support and in fact all the high-definition channels are always transmitted in h.264, which is evident when you carefully observe the static portions of the video such as the channel logos. But, all the SD channels transmitted by Tata Sky till 31st July 2013 were pure MPEG-2 channels.

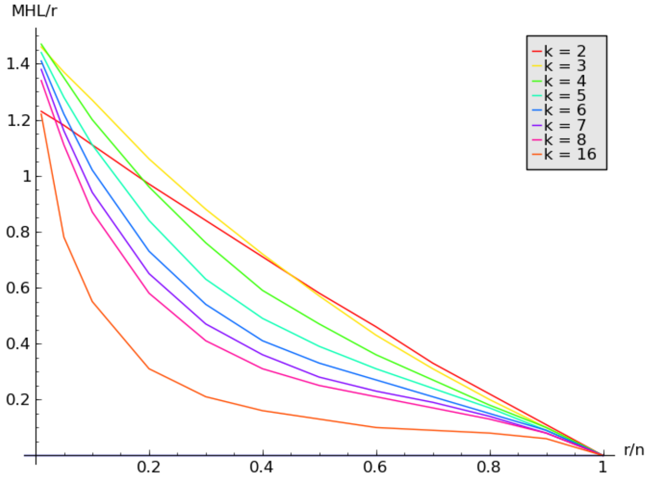
Why is Tata Sky doing this?

The bandwidth that is currently available to Tata Sky is very limited and the transponders Tata Sky supposed to get by 2013 have been delayed. Majority of Tata Sky's competitors,

From saval.in

Impact of k-SD scheme

Plot for *MHL*



Impact of Generalization

The k-ary tree SD scheme improves MHL for $r/n > \delta_k$ (a threshold value for a given k).

In Theory ... we have a hierarchy of optimization between the NNL-SD scheme and the Power Set scheme.

Practically In applications like Pay-TV
... where the sessions change very frequently
... the number of revoked users is moderate
the communication cost can be improved.

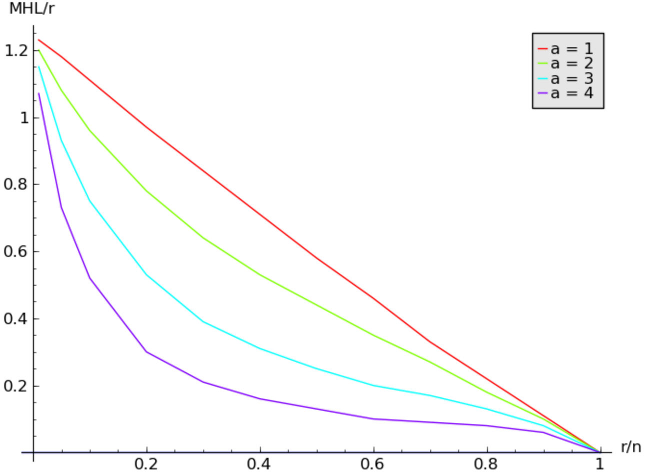
The draft of this work

In submission

...may be found at:
Cryptology ePrint Archive: Report 2013/786

Can we do better?

In preparation



Summary

Work Done

- Accommodating arbitrary number of users
- Tools for understanding the combinatorics of tree-based schemes
- Tools to compute expected header length in tree-based schemes
- Storage Minimal Layering
- Constrained Minimization of User Storage
- Generalization of NNL-SD using k-ary trees

Our results have actual practical and commercial value.

Thank You

My website: www.isical.ac.in/∼sanjayb_r

Some SML examples

SMLs are not unique

ℓ_0	no. of $SML_0(\ell_0)$ layerings	no. of $SML_1(\ell_0)$ layerings
12	10	10
16	6	15
20	6	1
24	35	35
25	35	21
28	1	8

Example SMLs

10 Special levels for $SML_0(12)$	10 Special levels for $SML_1(12)$
12,7,4,2,1,0	8,4,2,1,0
12,8,4,2,1,0	8,5,2,1,0
12,8,5,2,1,0	8,5,3,1,0
12,8,5,3,1,0	9,5,2,1,0
12,7,3,1,0	9,5,3,1,0
12,7,4,1,0	9,6,3,1,0
12,7,4,2,0	8,4,1,0
12,8,4,1,0	8,4,2,0
12,8,4,2,0	8,5,2,0
12,8,5,2,0	9,5,2,0

TABLE OF COMPARISON

ℓ_0	$r_{\min}$	$r_{\max}$	scheme	special levels	storage	normalized header lengths for $(r_{\min}, \ldots, r_{\max})$
12	2^2	2^6	SD	12,0	78	(1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00)
			e-HS	12,8,4,0	**42**	(**1.69, 1.59, 1.56, 1.56, 1.57, 1.57, 1.57, 1.56, 1.55**,1.53,1.52)
			SML_0	12,8,5,3,1,0	40	(**1.68, 1.57, 1.54, 1.54, 1.54, 1.55, 1.55, 1.54, 1.54**,1.53,1.52)
			SML_1	8,5,3,1,0	**32**	(**1.68, 1.57, 1.54, 1.54, 1.54, 1.55, 1.55, 1.54, 1.54**,1.53,1.52)
16	2^3	2^8	SD	16,0	136	(1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00)
			HS	16,12,8,4,0	**64**	(1.63,**1.65, 1.66, 1.64, 1.62, 1.60, 1.58, 1.57, 1.57, 1.56**)
			SML_0	16,11,7,4,2,1,0	61	(1.69,**1.60, 1.63**,1.65,1.65,1.64,1.63,1.62,1.60,1.59)
			SML_1	12,8,5,3,1,0	**50**	(1.63,**1.64, 1.65, 1.63, 1.60, 1.58, 1.57, 1.56, 1.55, 1.54**)
20	2^4	2^{10}	SD	20,0	210	(1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00)
			e-HS	20,15,10,5,0	**90**	(1.64,1.72,1.69,1.66,**1.64, 1.62, 1.61, 1.61, 1.60, 1.60**)
			SML_0	20,15,10,6,3,1,0	85	(1.64,1.72,1.69,1.66,**1.63, 1.62, 1.61, 1.60, 1.60, 1.60**)
			SML_1	15,10,6,3,1,0	**70**	((1.64,1.72,1.69,1.66,**1.63, 1.62, 1.61, 1.60, 1.60, 1.60**)
24	2^5	2^{12}	SD	24,0	300	(1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00)
			e-HS	24,19,14,9,4,0	**116**	(1.62,1.64,1.62,1.64,**1.67, 1.69, 1.71, 1.71, 1.72, 1.72**)
			SML_0	24,18,12,7,3,1,0	112	(1.65,1.74,1.70,1.67,**1.65, 1.63, 1.63, 1.62, 1.62, 1.63**)
			SML_1	18,12,8,5,3,1,0	**94**	(1.65,1.74,1.69,1.66,**1.63, 1.62, 1.61, 1.60, 1.60, 1.60**)
25	2^5	2^{12}	SD	24,0	325	(1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00)
			HS	25,20,15,10,5,0	**125**	(1.62,1.64,1.62,1.64,**1.67, 1.69, 1.71, 1.71, 1.72, 1.72**)
			SML_0	25,19,13,9,6,3,1,0	119	(1.65,1.74,1.69,1.66,**1.63, 1.62, 1.61, 1.60, 1.60, 1.60**)
			SML_1	19,13,9,6,3,1,0	**100**	(1.65,1.74,1.69,1.66,**1.63, 1.62, 1.61, 1.60, 1.60, 1.60**)
28	2^6	2^{14}	SD	28,0	406	(1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00)
			e-HS	28,22,16,10,5,0	**146**	(1.64,1.65,**1.63, 1.66, 1.69, 1.71, 1.73, 1.74, 1.75, 1.75**)
			SML_0	28,21,15,10,6,3,1,0	140	(1.65,1.70,1.65,**1.63, 1.62, 1.63, 1.64, 1.66, 1.67, 1.68**)
			SML_1	22,16,11,7,4,2,0	**119**	(1.64,1.65,**1.62, 1.64, 1.67, 1.69, 1.70, 1.71, 1.72, 1.72**)

Comparison: NNL-SD vs e-HS-LSD vs CML

Table of Comparison

ℓ_0	$r_{\min}$	$r_{\max}$	scheme	special levels	storage	normalized header lengths for $(r_{\min},\ldots,r_{\max})$
12	2^2	2^6	SD	$12,0$	78	$(1,\ldots,1)$
			e-HS	$12,8,4,0$	42	$(1.69,1.59,1.56,1.56,1.57,1.57,1.57,1.56,1.55,1.53,1.52)$
			CML	$10,0$	58	$(1.15,1.01,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00)$
16	2^6	2^8	SD	$16,0$	136	$(1,\ldots,1)$
			HS	$16,12,8,4,0$	64	$(1.66,1.64,1.62,1.61,1.59,1.58,1.58,1.57,1.57,1.56)$
			CML	$10,0$	76	$(1.14,1.08,1.05,1.03,1.01,1.01,1.00,1.00,1.00,1.00)$
20	2^8	2^{10}	SD	$20,0$	210	$(1,\ldots,1)$
			e-HS	$20,15,10,5,0$	90	$(1.68,1.66,1.64,1.63,1.62,1.61,1.61,1.60,1.60,1.60)$
			CML	$16,12,0$	110	$(1.14,1.08,1.04,1.03,1.01,1.01,1.00,1.00,1.00,1.00)$
24	2^{10}	2^{12}	SD	$24,0$	300	$(1,\ldots,1)$
			e-HS	$24,19,14,9,4,0$	116	$(1.63,1.64,1.66,1.68,1.69,1.71,1.71,1.72,1.72,1.72)$
			CML	$19,14,0$	149	$(1.14,1.08,1.04,1.03,1.01,1.01,1.00,1.00,1.00,1.00)$
25	2^{10}	2^{12}	SD	$25,0$	325	$(1,\ldots,1)$
			e-HS	$25,20,15,10,5,0$	125	$(1.63,1.64,1.66,1.68,1.69,1.71,1.71,1.72,1.72,1.72)$
			CML	$20,15,0$	165	$(1.14,1.08,1.04,1.03,1.01,1.01,1.00,1.00,1.00,1.00)$
28	2^{10}	2^{14}	SD	$28,0$	406	$(1,\ldots,1)$
			e-HS	$28,22,16,10,5,0$	146	$(1.69,1.63,1.64,1.67,1.69,1.72,1.73,1.74,1.75,1.75)$
			CML	$23,18,0$	219	$(1.14,1.08,1.04,1.03,1.01,1.01,1.00,1.00,1.00,1.00)$

Comparison: CTSD vs CTLSD

Table of Comparison

n	scheme	special layers	storage	$r_{\min}$	$r_{\max}$	header length normalized by CTSD
10^3	CTSD	10,0	55	2^2	2^5	$(1,\ldots,1)$
	CTLSD	8,0	**39**	2^2	2^5	$(1.09, 1.02, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00)$
10^4	CTSD	14,0	105	2^4	2^7	$(1,\ldots,1)$
	CTLSD	10,0	**65**	2^4	2^7	$(1.04, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00)$
10^5	CTSD	17,0	153	2^6	2^8	$(1,\ldots,1)$
	CTLSD	11,0	**87**	2^6	2^8	$(1.08, 1.04, 1.02, 1.01, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00)$
10^6	CTSD	20,0	210	2^8	2^{10}	$(1,\ldots,1)$
	CTLSD	16,12,0	**110**	2^8	2^{10}	$(1.13, 1.07, 1.04, 1.02, 1.01, 1.01, 1.00, 1.00, 1.00, 1.00)$
10^7	CTSD	24,0	300	2^{10}	2^{12}	$(1,\ldots,1)$
	CTLSD	19,14,0	**149**	2^{10}	2^{12}	$(1.04, 1.02, 1.01, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00)$
10^8	CTSD	27,0	378	2^{10}	2^{13}	$(1,\ldots,1)$
	CTLSD	22,17,0	**200**	2^{10}	2^{13}	$(1.08, 1.04, 1.02, 1.01, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00)$
10^9	CTSD	30,0	465	2^{10}	2^{15}	$(1,\ldots,1)$
	CTLSD	25,20,0	**260**	2^{10}	2^{15}	$(1.12, 1.07, 1.04, 1.02, 1.01, 1.01, 1.00, 1.00, 1.00, 1.00)$

Comparison of the kSD scheme

Table of Comparison

n	k	us_k	MHL_k/r	n	k	us_k	MHL_k/r
	2	55	$(1.10, 0.98, 0.72)$		2	105	$(1.11, 0.97, 0.71)$
	3	56	$(1.27, 1.06, 0.72)$		3	90	$(1.26, 1.07, 0.72)$
	4	60	$(1.21, 0.96, 0.59)$		4	112	$(1.20, 0.96, 0.59)$
10^3	5	90	$(1.11, 0.84, 0.50)$	10^4	5	126	$(1.11, 0.84, 0.49)$
	6	120	$(1.03, 0.73, 0.42)$		6	252	$(1.02, 0.73, 0.41)$
	7	180	$(0.95, 0.65, 0.36)$		7	270	$(0.94, 0.65, 0.36)$
	8	340	$(0.86, 0.58, 0.32)$		8	510	$(0.86, 0.58, 0.31)$
	2	153	$(1.11, 0.97, 0.71)$		2	210	$(1.11, 0.97, 0.71)$
	3	132	$(1.27, 1.06, 0.72)$		3	182	$(1.27, 1.07, 0.72)$
	4	180	$(1.20, 0.96, 0.59)$		4	220	$(1.20, 0.96, 0.59)$
10^5	5	216	$(1.11, 0.84, 0.49)$	10^6	5	270	$(1.11, 0.84, 0.49)$
	6	336	$(1.02, 0.73, 0.41)$		6	432	$(1.02, 0.73, 0.41)$
	7	378	$(0.94, 0.65, 0.36)$		7	648	$(0.94, 0.65, 0.36)$
	8	714	$(0.87, 0.58, 0.31)$		8	952	$(0.87, 0.58, 0.31)$
	2	300	$(1.11, 0.97, 0.71)$		2	378	$(1.11, 0.97, 0.71)$
	3	240	$(1.27, 1.06, 0.72)$		3	306	$(1.27, 1.06, 0.72)$
	4	312	$(1.20, 0.96, 0.59)$		4	420	$(1.20, 0.96, 0.59)$
10^7	5	396	$(1.11, 0.84, 0.49)$	10^8	5	468	$(1.11, 0.84, 0.49)$
	6	540	$(1.02, 0.73, 0.41)$		6	792	$(1.02, 0.73, 0.41)$
	7	810	$(0.94, 0.65, 0.36)$		7	990	$(0.94, 0.65, 0.36)$
	8	1224	$(0.87, 0.58, 0.31)$		8	1530	$(0.87, 0.58, 0.31)$

Comparison of the kSD scheme

Table of Comparison

k \ r/n	(0.01,	0.05,	0.10,	0.20,	0.30,	0.40,	0.50,	0.60,	0.70,	0.80,	0.90,	1.00)
2	(1.23,	1.18,	1.11,	0.97,	0.84,	0.71,	0.58,	0.46,	0.33,	0.22,	0.11,	0.00)
3	(1.46,	1.37,	1.27,	1.06,	0.88,	0.72,	**0.57**,	0.43,	0.31,	0.20,	0.10,	0.00)
4	(1.47,	1.35,	1.20,	**0.96**,	0.76,	0.59,	0.47,	0.36,	0.27,	0.18,	0.10,	0.00)
5	(1.44,	1.28,	1.11,	**0.84**,	0.63,	0.49,	0.39,	0.31,	0.24,	0.17,	0.09,	0.00)
6	(1.41,	1.22,	**1.02**,	0.73,	0.54,	0.41,	0.33,	0.27,	0.21,	0.15,	0.09,	0.00)
7	(1.38,	**1.16**,	0.94,	0.65,	0.47,	0.36,	0.28,	0.23,	0.19,	0.14,	0.08,	0.00)
8	(1.34,	**1.11**,	0.87,	0.58,	0.41,	0.31,	0.25,	0.21,	0.17,	0.13,	0.08,	0.00)
16	(**1.22**,	0.78,	0.55,	0.31,	0.21,	0.16,	0.13,	0.10,	0.09,	0.08,	0.06,	0.00)

k	3	4	5	6	7	8	16
δ_k	0.44	0.19	0.11	0.07	0.05	0.04	< 0.01