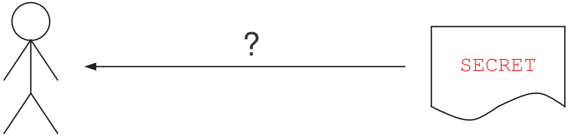
Cryptographic Enforcement of Interval-Based Access Control Policies

Jason Crampton

Information Security Group
Royal Holloway, University of London

University of Luxembourg, 3 July 2012

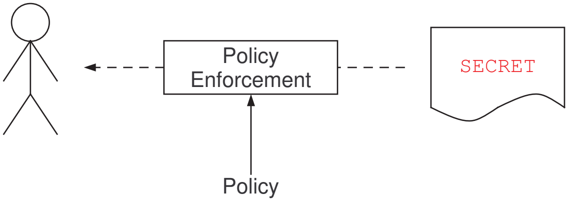
"Traditional" Access Control



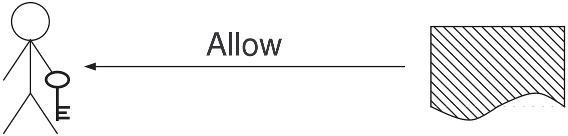
“Traditional” Access Control

“Traditional” Access Control

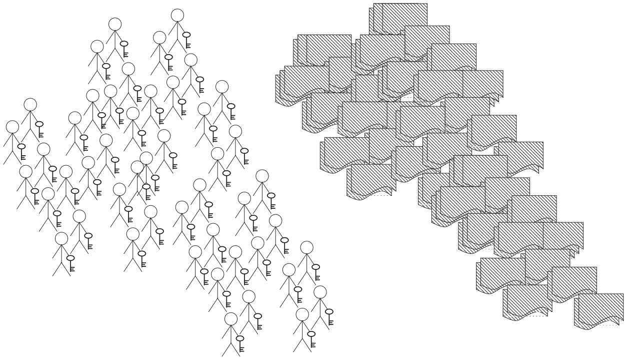
"Traditional" Access Control



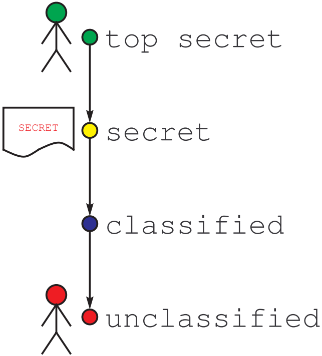
Cryptographically-Enforced Access Control



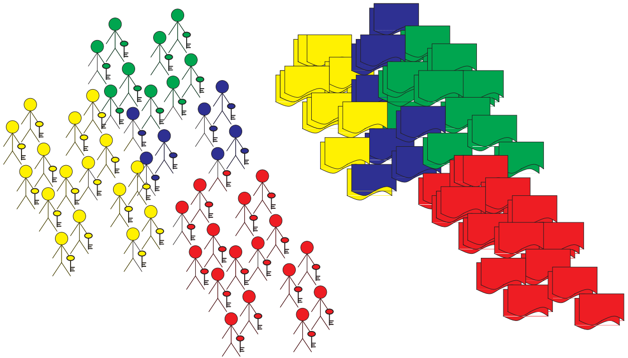
Cryptographically-Enforced Access Control: Scalability



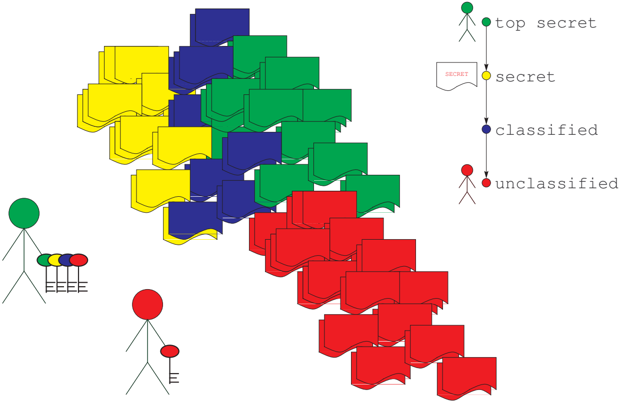
Graph-Based Authorization Policies



Graph-Based Authorization Policies



Graph-Based Authorization Policies

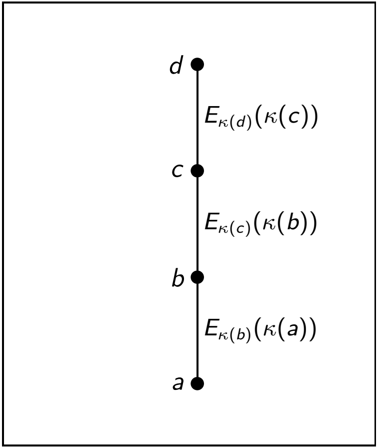


A Generic Single-Key Enforcement Mechanism

- We treat encryption keys like any other protected resource (that is, we encrypt them)

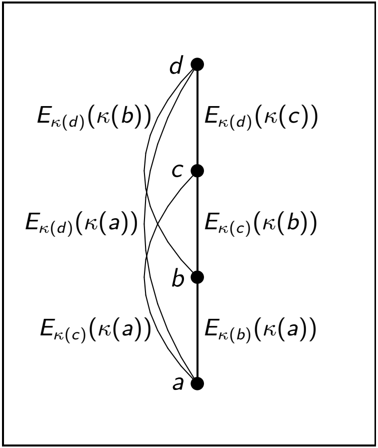
A Generic Single-Key Enforcement Mechanism

- We treat encryption keys like any other protected resource (that is, we encrypt them)
- For every edge (x, y), encrypt $\kappa(y)$ using $\kappa(x)$ (iterative key derivation by the end user)



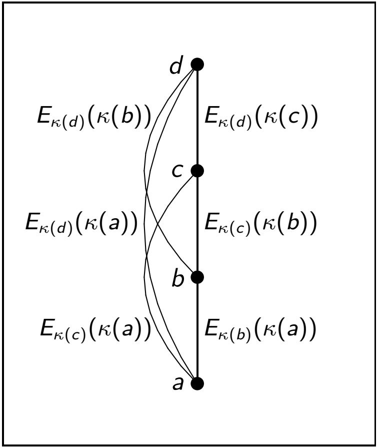
A Generic Single-Key Enforcement Mechanism

- We treat encryption keys like any other protected resource (that is, we encrypt them)
- For every edge (x, y), encrypt $\kappa(y)$ using $\kappa(x)$ (iterative key derivation by the end user)
- Alternatively, for every y that is reachable from x, encrypt $\kappa(y)$ using $\kappa(x)$ (direct key derivation)



A Generic Single-Key Enforcement Mechanism

- We treat encryption keys like any other protected resource (that is, we encrypt them)
- For every edge (x, y), encrypt $\kappa(y)$ using $\kappa(x)$ (iterative key derivation by the end user)
- Alternatively, for every y that is reachable from x, encrypt $\kappa(y)$ using $\kappa(x)$ (direct key derivation)



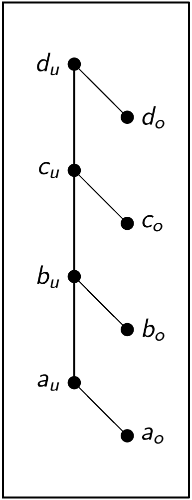
- Clearly, there are trade-offs between
  - the number of keys that need to be encrypted
  - the number of key derivation operations performed by a user

Security Considerations: Key Recovery

- It should be computationally hard for u to derive $\kappa(y)$ unless there is a path from $\lambda(u)$ to y
- More generally, it should be computationally hard for a group of users $U_{\text{Collude}} \subseteq U$ to pool key information and derive $\kappa(y)$ unless there exists $u \in U_{\text{Collude}}$ such that there is a directed path from $\lambda(u)$ to y
- For appropriate choices of encryption function E, edge-based encryption schemes satisfy the above properties

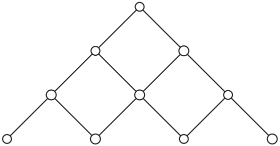
Security Considerations: Key Indistinguishability

- Informally, it should be computationally hard to distinguish between a key $\kappa(y)$ and a random value
- Edge-based encryption schemes do not satisfy this property (since successful key derivation and object decryption provides a means of distinguishing)
- Schemes having key indistinguishability can be constructed (modulo certain assumptions about the attack model) by modifying the graph and the labeling function

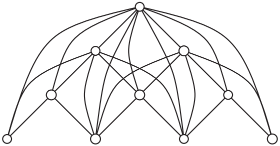


Introduction

- Given an authorization graph $G_{\text{auth}} = (V, E_{\text{auth}})$ and $x, y \in V$, let $(x, y) \in E_{enf}$ if and only if $\kappa(y)$ is encrypted using $\kappa(x)$
- We say $E_{\text{enf}} \subseteq V \times V$ is policy-enforcing if and only if $E^*_{\text{auth}} = E^*_{\text{enf}}$
- The distance between $x, y \in V$ is the number of edges in the shortest path from x to y; the diameter of $G = (V, E)$ is defined to be $\max\{d(x, y) : x, y \in V\}$



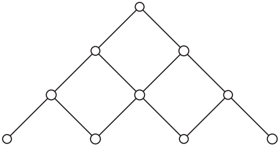
$|E_{auth}| = 12$; $diameter = 3$



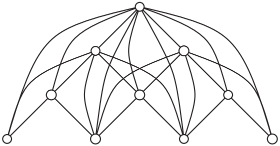
$|E_{enf}| = 25$; $diameter = 1$

Introduction

- Given an authorization graph $G_{\mathrm{auth}} = (V, E_{\mathrm{auth}})$ and $x, y \in V$, let $(x, y) \in E_{enf}$ if and only if $\kappa(y)$ is encrypted using $\kappa(x)$
- We say $E_{\mathrm{enf}} \subseteq V \times V$ is policy-enforcing if and only if $E^*_{\mathrm{auth}} = E^*_{\mathrm{enf}}$
- The distance between $x, y \in V$ is the number of edges in the shortest path from x to y; the diameter of $G = (V, E)$ is defined to be $\max\{d(x, y) : x, y \in V\}$



$|E_{auth}| = 12$; *diameter* $= 3$



$|E_{enf}| = 25$; *diameter* $= 1$

- We are interested in the trade-offs between the cardinality of E_{enf} and the diameter of G_{enf}

Trade-Offs for a Total Order

Let V be a total order on n elements $(V, \leqslant)$; then there exist sets of enforcing edges E_{enf} such that

$\lvert E_{\text{enf}} \rvert$	$d(G_{\text{enf}})$
$\frac{1}{2}n(n-1)$	1
$\Theta(n \log n)$	2
$\Theta(n \log \log n)$	3
$\Theta(n \log^* n)$	4
$n-1$	$n-1$

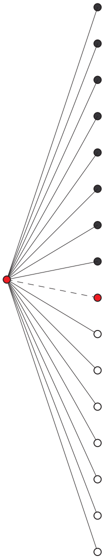
Trade-Offs for a Total Order : An Illustration

Consider a total order of 16 elements, for which we will construct a two-hop scheme

Trade-Offs for a Total Order : An Illustration

Consider a total order of 16 elements, for which we will construct a two-hop scheme

Step 1 Connect the top eight nodes to a "median node" and connect that node to the remaining nodes

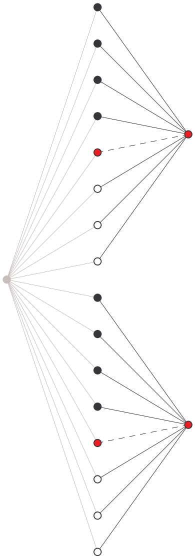


Trade-Offs for a Total Order : An Illustration

Consider a total order of 16 elements, for which we will construct a two-hop scheme

Step 1 Connect the top eight nodes to a "median node" and connect that node to the remaining nodes

Step 2 Repeat for each chain of length 8



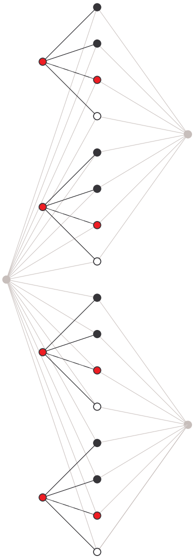
Trade-Offs for a Total Order : An Illustration

Consider a total order of 16 elements, for which we will construct a two-hop scheme

Step 1 Connect the top eight nodes to a "median node" and connect that node to the remaining nodes

Step 2 Repeat for each chain of length 8

Step 3 Repeat for each chain of length 4



Trade-Offs for a Total Order : An Illustration

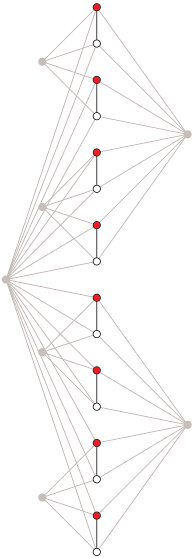
Consider a total order of 16 elements, for which we will construct a two-hop scheme

Step 1 Connect the top eight nodes to a "median node" and connect that node to the remaining nodes

Step 2 Repeat for each chain of length 8

Step 3 Repeat for each chain of length 4

Step 4 Repeat for each chain of length 2



Trade-Offs for a Total Order : An Illustration

Consider a total order of 16 elements, for which we will construct a two-hop scheme

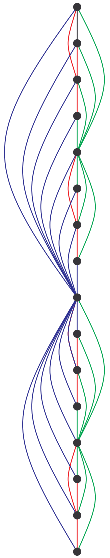
Step 1 Connect the top eight nodes to a "median node" and connect that node to the remaining nodes

Step 2 Repeat for each chain of length 8

Step 3 Repeat for each chain of length 4

Step 4 Repeat for each chain of length 2

For a chain of n elements there are $\log n$ rounds; each round adds fewer than n edges; the diameter of the resulting graph is 2



References

M.J. Atallah, M. Blanton, and K.B. Frikken.
Key management for non-tree access hierarchies.
In *Proceedings of SACMAT 2006*.

H. L. Bodlaender, G. Tel, and N. Santoro.
Trade-offs in non-reversing diameter.
Nordic Journal of Computing, 1994.

J. Crampton, K. M. Martin, and P. Wild.
On key assignment for hierarchical access control.
In *Proceedings of CSFW 2006*.

A. C.-C. Yao.
Space-time tradeoff for answering range queries.
In *Proceedings of STOC 1982*.

Cryptographic Access Control

Space-Time Trade-Offs

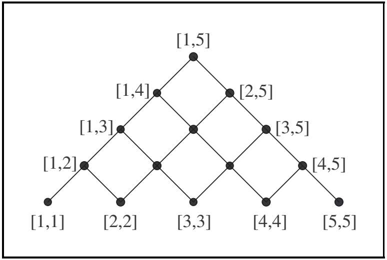
Temporal Access Control

Extensions to Higher Dimensions

Concluding Remarks

Introduction

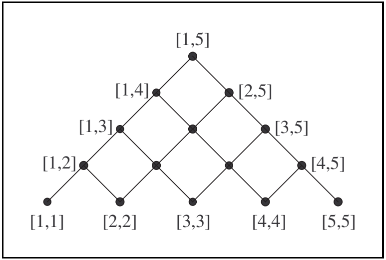
- Protected data is released periodically
- Each release period is regarded as a time point
- An interval is a consecutive sequence of time points: $V = \{[i,j] : 1 \leqslant i \leqslant j \leqslant n\}$
- Each user is authorized for some interval
- The authorization graph resembles a triangular mesh



The Naïve Approach

We could just apply the iterative cryptographic enforcement method to the triangular mesh

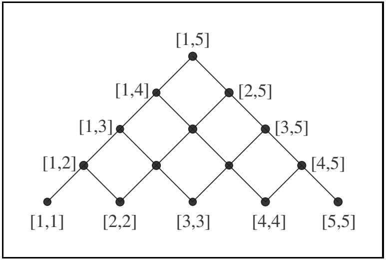
- We require $m(m-1)$ edges
- Key derivation requires no more than $m-1$ hops



The Naïve Approach Or Not?

We could just apply the iterative cryptographic enforcement method to the triangular mesh

- We require $m(m-1)$ edges
- Key derivation requires no more than $m-1$ hops



Alternatively, we could ask what trade-offs are possible for this particular authorization graph and this particular application?

- Solutions to the problem have either adapted methods for total orders or for arbitrary graphs
- We tackle the problem in a more direct way

A Crucial Observation

Protected objects are associated with a particular time point, not an interval

- The key for time point i is assigned label $[i, i]$
- No object is assigned a label $[i, j]$ with $i < j$

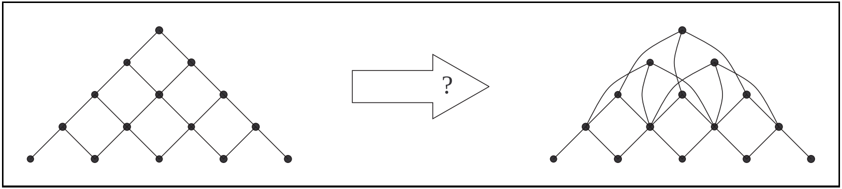
A user only needs to derive keys for labels of the form $[i, i]$

This assertion is not true in general for authorization graphs

Problem Summary

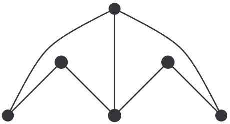
Given $V = \{[i, j] : 1 \leqslant i \leqslant j \leqslant m\}$, find an edge set $E \subseteq V \times V$ such that

1. there exists a path from $[i, j]$ to $[k, k]$ for all $k \in [i, j]$
2. $|E|$ is small
3. the diameter of the graph (V, E) is small



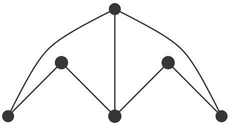
The One-Hop Scheme

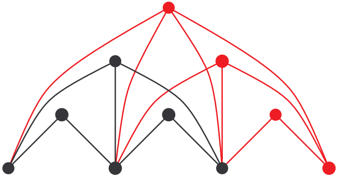
- The one-hop scheme is useful as a base scheme in more complex recursive constructions
  - Every non-"leaf" node is connected to the appropriate "leaf" nodes
  - The diameter of the graph is 1



The One-Hop Scheme

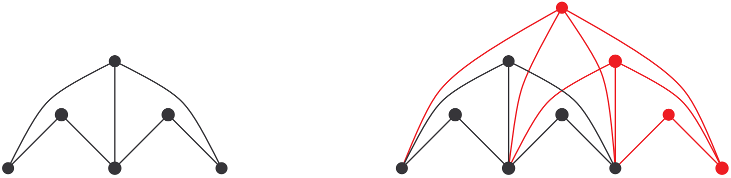
- The one-hop scheme is useful as a base scheme in more complex recursive constructions
  - Every non-"leaf" node is connected to the appropriate "leaf" nodes
  - The diameter of the graph is 1
- $e_m - e_{m-1} = (t_m - 1)$, where $t_m = \frac{1}{2}m(m+1)$





The One-Hop Scheme

- The one-hop scheme is useful as a base scheme in more complex recursive constructions
  - Every non-“leaf” node is connected to the appropriate “leaf” nodes
  - The diameter of the graph is 1
- $e_m - e_{m-1} = (t_m - 1)$, where $t_m = \frac{1}{2}m(m+1)$
  - Whence $e_m = \sum_{i=1}^{m}(t_m - 1) = \frac{1}{6}m(m-1)(m+4)$



Two Results

Let T_m denote the set of intervals $\{[i, j] : 1 \leqslant i \leqslant j \leqslant m\}$

Proposition

Let E be an enforcing set of edges for T_m. Then $|E| \geqslant m(m-1)$.

Two Results

Let T_m denote the set of intervals $\{[i, j] : 1 \leqslant i \leqslant j \leqslant m\}$

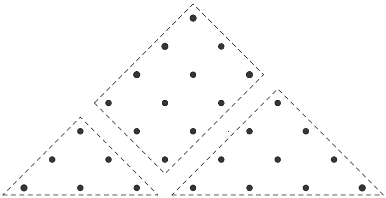
Proposition

Let E be an enforcing set of edges for T_m. Then $|E| \geqslant m(m-1)$.

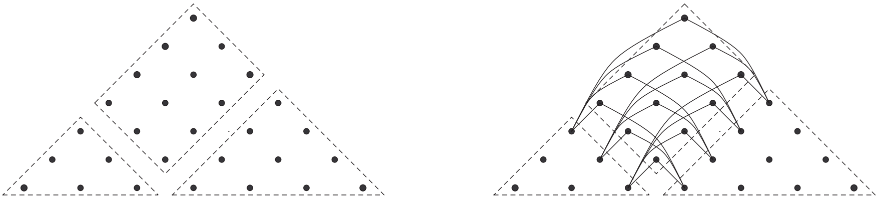
Proposition

There exists an enforcing set of edges E such that $|E| = m(m-1)$ and the diameter of (T_m, E) is $\lceil \log m \rceil$.

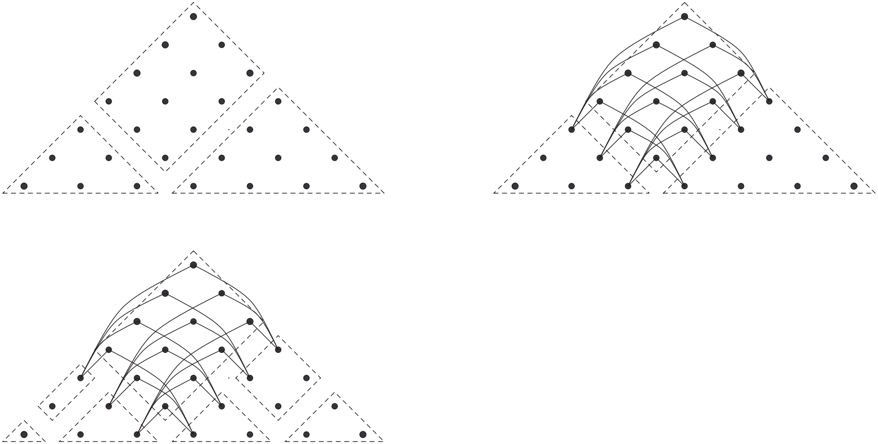
An Explicit Construction for T_7



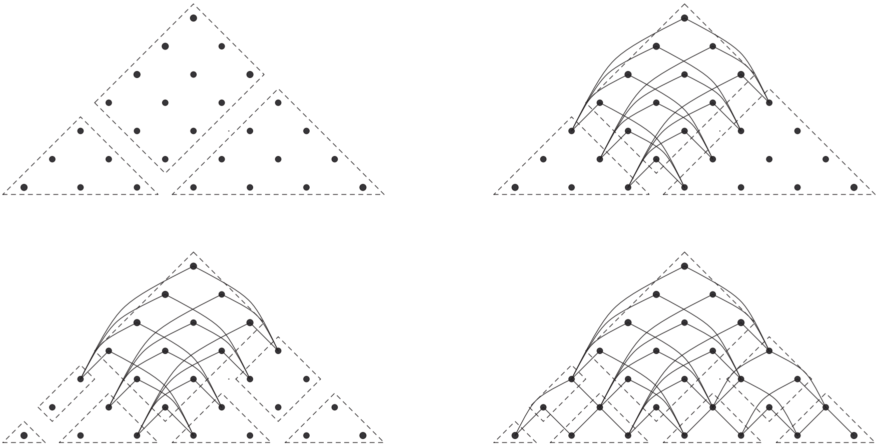
An Explicit Construction for T_7



An Explicit Construction for T_7

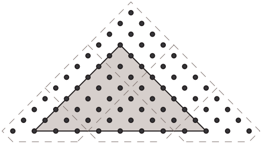


An Explicit Construction for T_7



Nodes and Supernodes

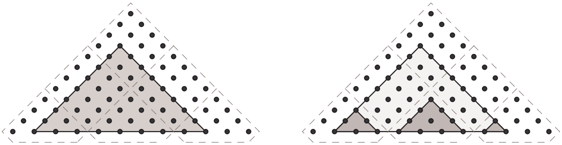
If $m = ab$, then T_m can be regarded as a copy of T_b in which the "supernodes" are copies of T_a and D_a



Nodes and Supernodes

If $m = ab$, then T_m can be regarded as a copy of T_b in which the "supernodes" are copies of T_a and D_a

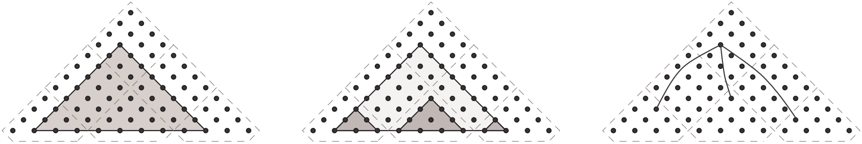
- Each interval in D_a is the disjoint union of no more than b intervals in copies of T_a



Nodes and Supernodes

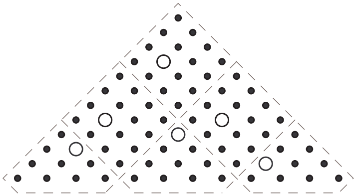
If $m = ab$, then T_m can be regarded as a copy of T_b in which the "supernodes" are copies of T_a and D_a

- Each interval in D_a is the disjoint union of no more than b intervals in copies of T_a
- Given an interval in D_a add edges to appropriate nodes in copies of T_a



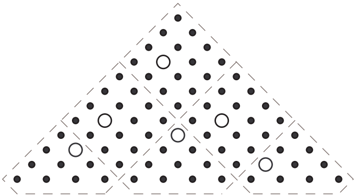
A Two-Hop Scheme

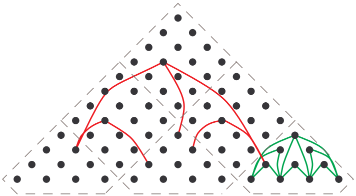
- Divide T_m into a^2 blocks so that each block contains a single node from each D_a
- Each node in a block occupies the same relative position within its respective copy of D_a



A Two-Hop Scheme

- Divide T_m into a^2 blocks so that each block contains a single node from each D_a
- Each node in a block occupies the same relative position within its respective copy of D_a

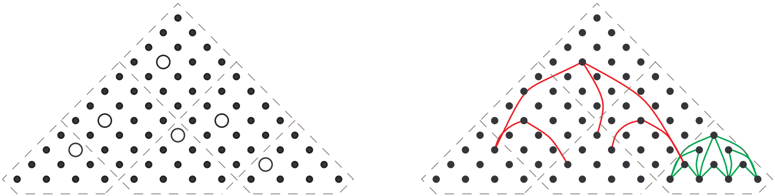




- Construct a^2 copies of a 1-hop scheme for T_b and a 1-hop scheme for each copy of T_a

A Two-Hop Scheme

- Divide T_m into a^2 blocks so that each block contains a single node from each D_a
- Each node in a block occupies the same relative position within its respective copy of D_a

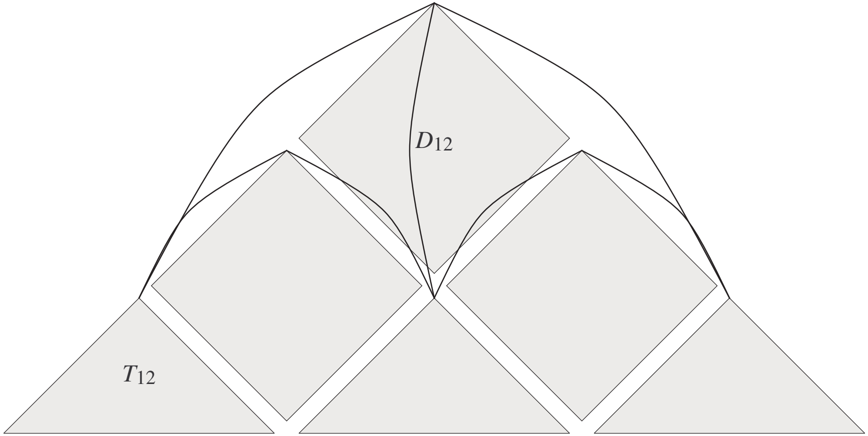


- Construct a^2 copies of a 1-hop scheme for T_b and a 1-hop scheme for each copy of T_a
- In total, the number of edges required is

$$\frac{1}{6}ab(a(b-1)(b+4)+(a-1)(a+4))$$

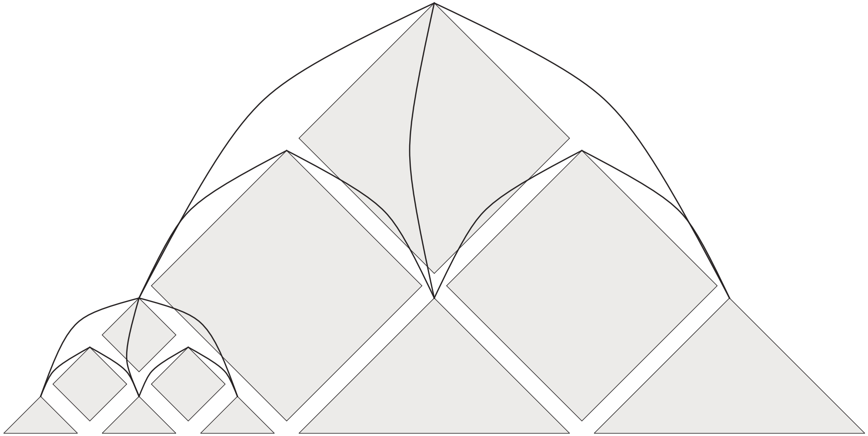
Generalizing the Two-Hop Construction

Writing $36 = 3.3.4$ we obtain the following decomposition of T_{36}



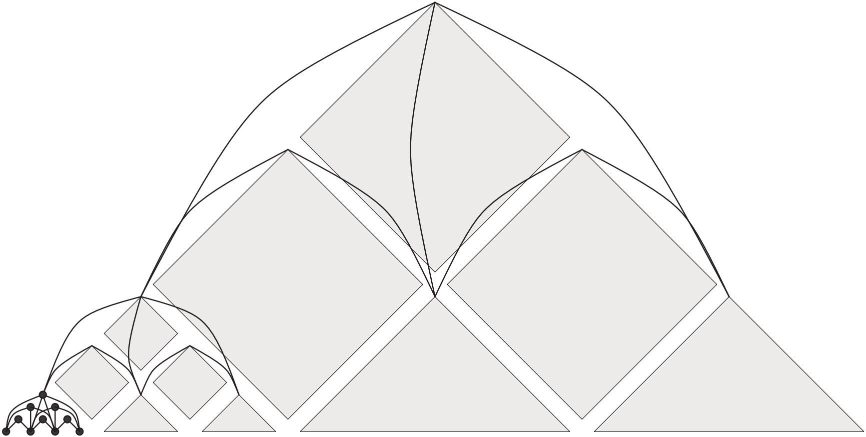
Generalizing the Two-Hop Construction

Writing $36 = 3.3.4$ we obtain the following decomposition of T_{36}



Generalizing the Two-Hop Construction

Writing $36 = 3.3.4$ we obtain the following decomposition of T_{36}



Generalizing the Two-Hop Construction

Theorem

Let $m = \prod_{i=1}^{d} a_i$*, where* a_i *is an integer and* $2 \leqslant a_i \leqslant a_{i+1}$ *for all* i*. Then there exists an enforcing set of edges* E *such that the diameter of* (T_m, E) *is* d *and*

$$|E| = \frac{m^2}{6} \sum_{i=1}^{d} \frac{(a_i - 1)(a_i + 4)}{\pi_i},$$

where $\pi_i = a_1 \ldots a_i$*.*

Some Remarks about the Term $\frac{(a_i - 1)(a_i + 4)}{\pi_i}$

- Successive terms in the summation are approximately equal when $a_{i+1} \approx a_i^2$ (minimize d)
- The ith term in the summation is minimized when $a_i = 2$ (minimize $|E|$)
- Consider $m = 36$

Factors	$\|E\|$	d
6.6	$36^2 . \frac{175}{108}$	2
4.9	$36^2 . \frac{153}{108}$	2
3.3.4	$36^2 . \frac{124}{108}$	3
2.2.3.3	$36^2 . \frac{109}{108}$	4

Corollary 1

Theorem

...there exists an enforcing set of edges E such that the diameter of (T_m, E) is d and

$$|E| = \frac{m^2}{6} \sum_{i=1}^{d} \frac{(a_i - 1)(a_i + 4)}{\pi_i}$$

Corollary

If $m = a^d$, then there exists an enforcing edge set E such that $|E| = \frac{1}{6} m(m-1)(a+4)$ and the diameter of (T_m, E) is $d = \log_a m$.

Corollary 2

Theorem

...there exists an enforcing set of edges E such that the diameter of (T_m, E) is d and

$$|E| = \frac{m^2}{6} \sum_{i=1}^{d} \frac{(a_i - 1)(a_i + 4)}{\pi_i}$$

Corollary

Let $m = 2^{2^d}$ for some integer $d \geqslant 2$. Then there exists an enforcing edge set E such that

$$|E| < m^2 \left(1 + \frac{1}{6} \log \log m\right)$$

and the diameter of (T_m, E) is $\log \log m$.

Related Work

M.J. Atallah, M. Blanton, and K.B. Frikken.
Incorporating temporal capabilities in existing key management schemes.
In *Proceedings of ESORICS 2007*.

- M.J. Atallah, M. Blanton, and K.B. Frikken.
  Key management for non-tree access hierarchies.
  In *Proceedings of SACMAT 2006*.

A. De Santis, A.L. Ferrara, and B. Masucci.
New constructions for provably-secure time-bound hierarchical key assignment schemes.
Theoretical Computer Science, 2008.

- B. Dushnik and E.W. Miller.
  Partially ordered sets.
  American Journal of Mathematics, 1941.
- M. Thorup.
  Shortcutting planar digraphs.
  Combinatorics, Probability & Computing, 1995.

Comparison

	Public Storage	**Derivation**
Atallah *et al.* 2007	$\mathcal{O}\left(m^2 \log m\right)$	4
	$\mathcal{O}\left(m^2\right)$	$\mathcal{O}\left(\log^* m\right)$
De Santis *et al.*, 2008	$\mathcal{O}\left(m^2\right)$	$\mathcal{O}\left(\log m \log^* m\right)$
	$\mathcal{O}\left(m^2 \log m\right)$	$\mathcal{O}\left(\log^* m\right)$
	$\mathcal{O}\left(m^2 \log m \log\log m\right)$	3
Crampton, 2009	$m(m-1)$	$\lceil \log m \rceil$
	$\frac{1}{6}m(m-1)(\sqrt{m}+4)$	2
Crampton, 2010	$m^2\left(1+\frac{1}{6}\lceil \log\log m \rceil\right)$	$\lceil \log\log m \rceil$

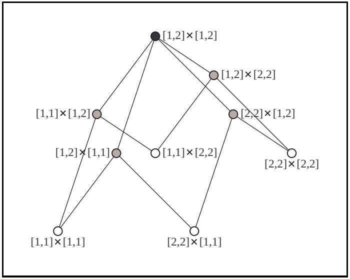
Practical and Efficient Enforcement

- My approach attacks the problem directly and makes use of specific characteristics of the application
- My constructions yield explicit formulae (rather than asymptotic behaviour) for the number of edges and the number of hops required
- My schemes can be implemented directly using existing iterative key encrypting schemes

"Geo-Spatial" Access Control Policies

- Data objects are associated with a point in a two-dimensional grid

- Users are authorized for rectangles covering a set of points in the grid
- The set of rectangles ordered by subset inclusion forms a partially ordered set
- The set of nodes in the authorization graph is $T_m \times T_n$
- We will write $T_{m,n}$ to denote $T_m \times T_n$



The Main Results

Theorem

There exists an enforcing set of edges E such that the diameter of the graph $(T_{n,n}, E)$ is bounded by $\lceil \log n \rceil$ and

$$|E| = \frac{1}{3} n^2 (n-1)(2n+5) < \frac{8}{3} |T_{n,n}| .$$

Theorem

There exists an enforcing sets of edges E such that the diameter of $(T_{m,km}, E)$ is $\log m + \log k = \log km$ *and*

$$|E| = \frac{1}{6} km^2 \big(3(k-1)m(m+1) + 2(m-1)(2m+5)\big).$$

Corollary

For $k \geqslant 1$, there exists an enforcing set of edges E such that the diameter of $(T_{m,km}, E)$ is $\log km$ *and*

$$|E| < 2 |T_{m,km}| \left(1 + \frac{1}{3k}\right) \leqslant \frac{8}{3} |T_{m,km}| .$$

Interval-Based Access Control Policies

Define $T_n^k = \underbrace{T_n \times \cdots \times T_n}_{k \text{ times}}$

Theorem

There exists a set of enforcing edges E for T_n^k such that the diameter of (T_n^k, E) is $\log n$ and

$$|E| = \frac{n^k}{2^k} \sum_{i=1}^{k} \binom{k}{i} \frac{(3^i - 1)(n^i - 1)}{2^i - 1}.$$

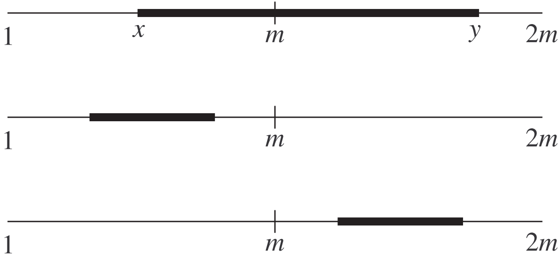
Corollary

$|E|$ is $\Theta\left(\left(\frac{3}{2}\right)^k \left|T_n^k\right|\right)$.

Sketch Proof: $k = 1$

Consider $[x, y]$, $1 \leqslant x \leqslant y \leqslant 2m$

- x and y can be regarded as the "corners" of the interval $[x, y]$
- Each corner can be labelled with a bit, where 0 indicates it is less than or equal to m and 1 indicates it is greater than m
- If x and y's labels are the same, then the interval $[x, y]$ is completely contained in a subinterval of length m



Sketch Proof: $k = 2$

- We only need to add (two) edges in the recursive step if the corner labels are different

- Hence, the recurrence relation for the number of edges has the form

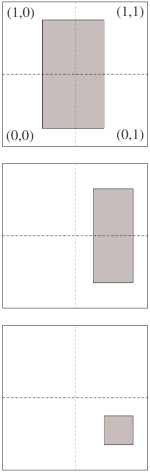
$$e(2m) = 2a + 2e(m)$$

where a is the number of intervals whose corner labels are different

- If the corner labels are different we have m choices for each of x and y, whence $a = m^2$

Sketch Proof: $k = 2$

- The bottom left-hand and top right-hand corners of a rectangle can each be associated with a pair in $\{0,1\}^2$
- Moreover, if the two corners are represented by (b_1, b_2) and (t_1, t_2) then $b_1 \leqslant t_1$ and $b_2 \leqslant t_2$
- A rectangle straddles 2^d squares of side m, where $0 \leqslant d \leqslant 2$ is the Hamming distance between these corners
  - The Hamming distance is the number of places in which the two pairs differ
  - For $d > 0$, 2^d is the number of edges required from that rectangle in the recursive step

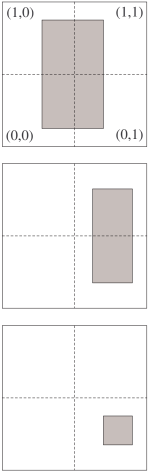


Sketch Proof: $k = 2$

- The number of choices for the co-ordinates of the corners is also determined by the Hamming distance

$$\left(\frac{1}{2}m(m+1)\right)^{(2-d)}\left(m^2\right)^d$$

  - If $b_i = t_i$ then there are $\frac{1}{2}m(m+1)$ choices for the endpoints of the ith interval
  - If $b_i < t_i$ then there are m^2 choices
- Finally, the number of corner pairs with Hamming distance d is given by $2^{2-d}\binom{2}{d}$
  - If $b_i = t_i$ then there are two choices for b_i
  - If $b_i < t_i$ then there is only once choice for b_i
  - There are $\binom{2}{d}$ ways in which we can choose corners with Hamming distance d



Sketch Proof: $k = 2$

- We deduce the recurrence relation

$$e(2m) = 4e(m) + \sum_{d=1}^{2} \alpha(d)\beta(d)\gamma(d)$$

  - $\alpha(d) = 2^d$ is the number of edges required to connect a rectangle with Hamming distance d to sub-rectangles contained with copies of a square of side m
  - $\beta(d) = \left(\frac{m+1}{2}\right)^{2-d} m^{d+2}$ is the number of rectangles with Hamming distance d
  - $\gamma(d) = 2^{2-d}\binom{2}{d}$ is the number of ways of fitting rectangles with Hamming distance d in a square of side $2m$
- That is

$$e(2m) = 4e(m) + m^2 \sum_{d=1}^{2} (2m)^d (m+1)^{2-d} \binom{2}{d}$$

Sketch Proof: The General Case

- Any "hyperinterval" $\mathcal{I}$ in T^k_{2m} can be represented as the union of at most 2^k hyperintervals in copies of the hypercube $[1, m]^k$
- $\mathcal{I}$ is associated with two k-tuples in $\{0, 1\}^k$, which identify the bottom left-hand and top right-hand "hypercorners" of $\mathcal{I}$
- The Hamming distance $0 \leqslant d \leqslant k$ determines the number of:
  - copies of $[1, m]^k$ that $\mathcal{I}$ straddles (and hence the out-degree of $\mathcal{I}$), which equals 2^d
  - choices for the co-ordinates of $\mathcal{I}$, which equals $(\frac{1}{2}m(m+1))^{k-d}(m^2)^d$
  - choices for hypercubes containing the hypercorners, which equals $2^{k-d}\binom{k}{d}$
- We deduce the following recurrence relation

$$e(2m, k) = 2^k e\left(m, k\right) + m^k \sum_{d=1}^{k} (2m)^d (m+1)^{k-d} \binom{k}{d}$$

Contributions

- First work in this area to develop techniques tailored for the problem
- First work to provide exact (and better) bounds for the number of edges
- First work to retain the simplicity of existing iterative schemes
  - Other constructions require auxiliary data structures
  - Other constructions require more complex key derivation algorithms
- First work to provide explicit constructions for higher dimensions that are natural extensions of those for lower dimensions

References

J. Crampton.
Trade-offs in cryptographic implementations of temporal access control.
In *Proceedings of NordSec 2009*.

J. Crampton.
Practical constructions for the efficient cryptographic enforcement of interval-based access control policies.
ACM Transactions on Information and System Security, 2011.

J. Crampton.
Time-storage trade-offs for cryptographically-enforced access control.
In *Proceedings of ESORICS 2011*.