Practical Constructions for the Efficient Cryptographic Enforcement of Interval-Based Access Control Policies

Jason Crampton

8 March 2011
Problem Summary

Given $V = \{[i, j] : 1 \leq i \leq j \leq m\}$, 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

Why is this interesting?

How do we tackle it?
Cryptographic Access Control

Temporal Access Control
  Binary Decomposition
  Multiplicative Decomposition
  Related Work

Extensions to Higher Dimensions

Concluding Remarks
Graph-Based Authorization Policies

Let $G = (V, E)$ be a directed acyclic graph, where $V$ is some set of security-relevant attributes.

- Each user and object is associated with some vertex in $V$.
- Let $\lambda(u) \in V$ and $\lambda(o) \in V$ denote the vertices associated with $u$ and $o$, respectively.
- $u$ is authorized to read $o$ if there is a path from $\lambda(u)$ to $\lambda(o)$.

The information flow policies of Denning and of Bell-LaPadula are examples of graph-based authorization policies.
The natural cryptographic enforcement model is to define a key $\kappa(v)$ for each vertex $v \in V$

- We encrypt resources associated with vertex $v$ with key $\kappa(v)$
- $u$ must be able to obtain the key $\kappa(y)$ whenever there is a directed path from $\lambda(u)$ to $y$

The “trivial” enforcement mechanism simply gives each user every key that they require.

However, there are good reasons to require that a user is given a single key.
An Iterative Enforcement Mechanism

**Requirement**  
For every edge \((x, y)\) it should be possible to derive \(\kappa(y)\) from \(\kappa(x)\)

**Solution**  
Publish information associated with each edge that allows the recovery of \(\kappa(y)\) only if \(\kappa(x)\) is known

**Example**  
For each edge \((x, y)\), publish \(\kappa(y) \oplus h(\kappa(x) \parallel y)\)

For convenience, we will refer to “edge-based encryption”
Security Considerations

- It should be computationally hard for \( u \) to derive \( \kappa(y) \) if there is no path from \( \lambda(u) \) to \( y \)
- More generally, it should be computationally hard for a group of users \( U_{Collude} \subseteq U \) to pool key information to obtain \( \kappa(y) \) unless there exists \( u \in U_{Collude} \) 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
- Stronger security guarantees (such as key indistinguishability) can be obtained if required
Space-Time Trade-Offs

Edge-based encryption requires iterative key derivation

- In the worst case a user will need to derive $d$ keys, where $d$ is the diameter of $G$

$$|E| = 12; \quad d = 3$$

Alternatively, publish information for the policy defined by $G^* = (V, E^*)$

- Key derivation always requires a single step
- The amount of public information increases

$$|E^*| = 25; \quad d = 1$$
More Complex Trade-Offs

Given an authorization graph \( G_{auth} = (V, E_{auth}) \), we say \( E_{enf} \) is (policy-)enforcing if and only if \( E^*_{auth} = E^*_{enf} \)

- In other words, \( G_{auth} \) and \( G_{enf} \) contain exactly the same paths

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

\[
\begin{array}{|c|c|}
\hline
|E_{enf}| & d(G_{enf}) \\
\hline
\frac{1}{2}n(n-1) & 1 \\
\Theta(n \log n) & 2 \\
\Theta(n \log \log n) & 3 \\
\Theta(n \log^* n) & 4 \\
n - 1 & n - 1 \\
\hline
\end{array}
\]
More Complex Trade-Offs: An Illustration

Consider a total order of 16 elements, for which we will construct a two-hop scheme.
More Complex Trade-Offs: 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.
More Complex Trade-Offs: 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
More Complex Trade-Offs: 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
More Complex Trade-Offs: 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
More Complex Trade-Offs: 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

Temporal Access Control

Extensions to Higher Dimensions

Concluding Remarks
Introduction

Protected data is released periodically

- Each period is regarded as a time point
- An interval is a consecutive sequence of time points
- Each user is authorized for some interval
- The authorization graph resembles a triangular mesh

One possible application is subscription-based services
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 $V = \{[i, j] : 1 \leq i \leq j \leq m\}$, 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
Cryptographic Access Control

Temporal Access Control
  Binary Decomposition
  Multiplicative Decomposition
  Related Work

Extensions to Higher Dimensions

Concluding Remarks
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

- We require $\frac{1}{6} m(m - 1)(m + 4)$ edges
  - $e_m - e_{m-1} = (t_m - 1)$, where $t_m = \frac{1}{2} m(m + 1)$
  - whence $e_m = \sum_{i=1}^{m} (t_m - 1)\ldots$
Two Results

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

Proposition

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

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

Proposition

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

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 $T_7$
The Construction

- Let $\ell = \lfloor m/2 \rfloor$ and $r = \lceil m/2 \rceil$
- Then $T_m$ comprises:
  - a copy of $T_\ell$, containing the minimal elements $[1, 1], \ldots, [\ell, \ell]$
  - a copy of $T_r$, containing the minimal elements $[\ell + 1, \ell + 1], \ldots, [m, m]$
  - a copy of rectangle $R_{\ell, r}$, containing the remaining nodes in $T_m$

1. Include an edge from every node in $R_{\ell, r}$ to one node in $T_\ell$ and one node in $T_r$
   - In particular, for node $[x, y]$ such that $x \leq \ell < y$, we add edges from $[x, y]$ to $[x, \ell]$ and from $[x, y]$ to $[\ell + 1, y]$

2. Recursively apply step 1 to $T_\ell$ and $T_r$, terminating when $\ell, r \leq 1$
Cryptographic Access Control

Temporal Access Control

Binary Decomposition

Multiplicative Decomposition

Related Work

Extensions to Higher Dimensions

Concluding Remarks
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 \( D_a \)

- Divide \( T_m \) into \( a^2 \) blocks so that each block contains a single node from each \( D_a \)
- Each interval in \( D_a \) is the union of no more than \( b \) intervals in copies of \( T_a \)
- Construct \( a^2 \) copies of a 1-hop scheme for \( T_b \)
- Construct a 1-hop scheme for each copy of \( T_a \)
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 $T_{36}$
Theorem

Let \( m = \prod_{i=1}^{d} a_i \), where \( a_i \) is an integer and \( 2 \leq a_i \leq a_{i+1} \) 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

- The difference between successive terms in the summation is approximately equal to zero when $a_{i+1} \approx a_i^2$ (minimize $d$)
- The $i$th term in the summation is minimized when $a_i = 2$ (minimize $|E|$)
- For $m = 36$ we have

| 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 $d \geq 2$. 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 $\log \log m$. 
Hybrid Decomposition

Note that we cannot use multiplicative decomposition directly if $m$ is a prime

- Apply binary decomposition first
- $T_{41}$, for example, can be split into $T_{20}$ and $T_{21}$
- Now construct 2-hop schemes for both $T_{20}$ and $T_{21}$ (since $20 = 4 \times 5$ and $21 = 3 \times 7$)
- Hence we derive a 3-hop scheme for $T_{41}$
Cryptographic Access Control

Temporal Access Control
  Binary Decomposition
  Multiplicative Decomposition
  Related Work

Extensions to Higher Dimensions

Concluding Remarks
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 | $O\left(m^2 \log m\right)$                | $O\left(\log^* m\right)$ |
|                      | $O\left(m^2\right)$                        |                      |
| De Santis et al., 2008 | $O\left(m^2\right)$                        | $O\left(\log m \log^* m\right)$ |
|                      | $O\left(m^2 \log m\right)$                | $O\left(\log^* m\right)$ |
|                      | $O\left(m^2 \log m \log \log m\right)$   |                      |
| Crampton, 2009       | $m(m - 1)$                                 | $\lceil \log m \rceil$ |
|                      | $\frac{1}{6} m(m - 1)(\sqrt{m} + 4)$       | $2$                  |
| Crampton, 2010       | $m^2 \left(1 + \frac{1}{6} \lceil \log \log m \rceil\right)$ | $\lceil \log \log m \rceil$ |
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.
Cryptographic Access Control

Temporal Access Control

Extensions to Higher Dimensions

Concluding Remarks
"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

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

Corollary

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

\[
|E| < 2 |T_{m,km}| \left(1 + \frac{1}{3k}\right) \leq \frac{8}{3} |T_{m,km}|.
\]
Interval-Based Access Control Policies

Define $T_n^k = T_n \times \cdots \times T_n$ $k$ 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**

$|E| \in \Theta \left( \left( \frac{3}{2} \right)^k |T_n^k| \right)$. 
Sketch Proof: $k = 1$

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

- $x$ and $y$ can be regarded as the “corners” of the interval $[x, y]$
- The corners can be labelled with a binary digit, where 0 indicates it is less than equal to $m$ and 1 indicates it is greater than $m$
- If the corner labels are the same, then the interval $[x, y]$ is completely contained in a subinterval of length $m$
- 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 (in this case $m^2$)
Sketch Proof: $k = 2$

Consider the proof for $k = 2$, for a square of side $2m$

- The bottom left-hand and top right-hand corners of a rectangle can be associated with a pair $(b_1, b_2) \in \{0, 1\}^2$
- A rectangle straddles $2^d$ squares of side $m$, where $0 \leq d \leq 2$ is the Hamming distance between these corners; for $d > 0$ this gives the number of edges required from that rectangle in the recursive step
- The number of choices for the co-ordinates of the corners is also determined by the Hamming distance

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

- Finally, the number of corner pairs with Hamming distance $d$ is given by $2^{2-d} \binom{2}{d}$
Sketch Proof: The General Case

- Any “hyperinterval” \( \mathcal{I} \) in \( T_{2m}^k \) 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 \( 0 \leq d \leq k \) 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 \( \alpha^{k-d} \beta^d \), where \( \alpha = \frac{1}{2} m(m + 1) \) and \( \beta = m^2 \);
  - choices for hypercubes containing the hypercorners, which equals \( 2^{k-d} \binom{k}{d} \)
- We deduce the following recurrence relation

\[
e(n, k) = \begin{cases} 
2^k e\left(\frac{n}{2}, k\right) + \sum_{i=1}^{k} \binom{k}{i} (3^i - 1) \left(\frac{n}{2}\right)^{i+k} & n > 1 \\
0 & n = 1
\end{cases}
\]
Cryptographic Access Control

Temporal Access Control

Extensions to Higher Dimensions

Concluding Remarks
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.
Practical constructions for the efficient cryptographic enforcement of interval-based access control policies.
To appear in ACM Transactions on Information and System Security.

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