Using Encryption to Enforce an Information Flow Policy – An Introduction

Jason Crampton

Information Security Group
Royal Holloway, University of London

Introduction – Information flow policies

Access control policy based on relative security clearance of users and data

- $\langle X, \leqslant \rangle$ is a partially ordered set of security labels
- U is a set of users
- O is a set of data objects and
- $\lambda : U \cup O \to X$ is a function that assigns a security label to each user and data object

Introduction – Information flow policies

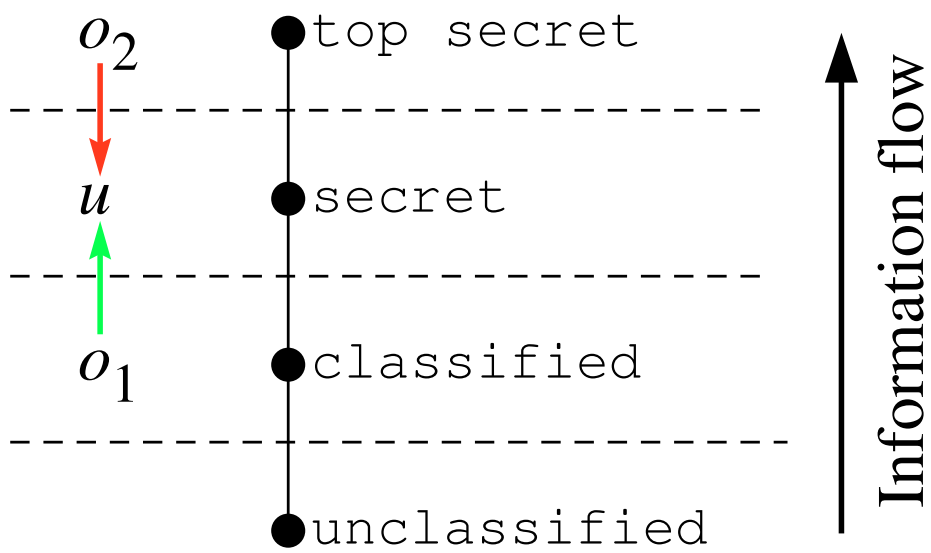
Information flow between user and object must respect ordering of respective labels

- A user u is permitted to read object o iff $\lambda(u) \geqslant \lambda(o)$
- Best known in context of Bell-LaPadula model (simple security property) for military security classifications

Introduction – Information flow policies

u can read o_1

u cannot read o_2



Introduction – Potential applications

Password protected file system

- Each directory in the directory tree is associated with a key
- Each file in a directory is password protected
- The password is a hash of the parent directory's key

Introduction – Potential applications

Private and shared mail boxes

- Mail is encrypted, each user has a private mail box and has access to a number of shared mail boxes
- b and b' are two mail boxes associated with sets of users V and V'
  - Define $b \leqslant b'$ iff $V \supseteq V'$
  - Maximal elements are those boxes associated with a single user

Introduction – Potential applications

Broadcast messages

- Users are arranged in a hierarchy
- Any user can send encrypted messages that can be read by other users who are at least as senior

Controlling access to broadcast XML documents

- More later ...

Introduction – A simple cryptographic solution

Associate each label x with a unique key $k(x)$

- Distribute keys so that user u has keys $\{k(x) : x \in X, x \leqslant \lambda(u)\}$
- Encrypt data (messages, database tables, etc.) with appropriate key
- To decrypt object o, u must have the key $k(\lambda(o))$, which implies that $\lambda(o) \leqslant \lambda(u)$

Users must have (knowledge of) many keys

Introduction – The problem defined

Given a poset X, find a method of assigning keys to elements of X with the following properties:

- For each $x \in X$, there is a single key $k(x)$
- For each key $k(x)$, it is possible to derive $k(y)$ for all $y \leqslant x$

We must consider the following issues:

- Key generation
- Key derivation
- Security – resistance to collaborative attacks by keyholders
- Computational and key storage overheads

Introduction – Generic solution

Associate certain public information with each element $x \in X$

Compute secret key $k(x)$ for each element $x \in X$ using one-way function

Publish information for each element of X such that

- Given $k(x)$ and $y \leqslant x$ it is possible to use public information to derive secret key $k(y)$
- Given $k(x)$ and $y \not\leqslant x$ it is not possible to derive secret key $k(y)$

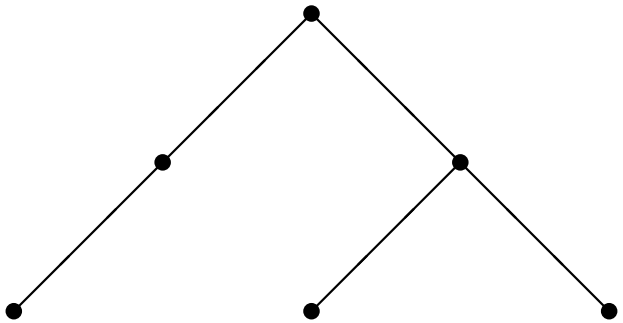
Outline of talk

- Simple schemes for trees
- The Akl-Taylor scheme (1983)
- The MacKinnon-Taylor-Meijer-Akl scheme (1985)
- The Harn-Lin scheme (1990)
- Application to broadcast XML documents
- Areas for future research

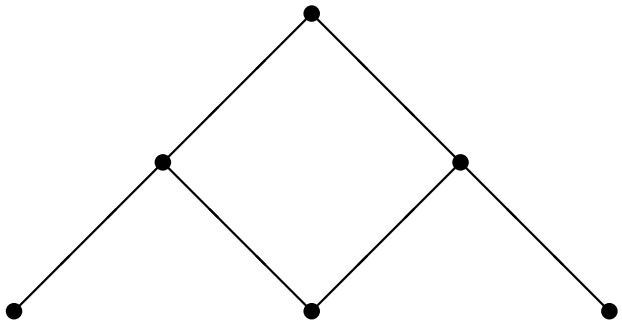
Trees

We assume that (the Hasse diagram of) X is a tree.

- X has a unique maximal element $\widehat{x}$ (the root of the tree)
- For all $x \neq \widehat{x}$, there exists a unique $y \in X$ such that $x \lessdot y$



A tree



Not a tree

The Gudes scheme (1980) – Encryption

Assign a secret key $k(x)$ to each $x \in X$

Publish $e(x) = E_{k(y)}(k(x))$, where y is the parent of x and E denotes some (symmetric) encryption method

Encrypt data with security label x using $k(x)$

The Gudes scheme – Decryption

Let $x, x' \in X$ with $x' \leqslant x$

- How does user u with security label x use $k(x)$ (and public information) to decrypt data with security label x'?

There exists a unique chain $x' = x_0 \lessdot x_1 \lessdot \ldots x_{m-1} \lessdot x_m = x$ since X is a tree

- u decrypts $e(x_{m-1})$ with $k(x_m) = k(x)$ to obtain $k(x_{m-1})$
- u decrypts $e(x_{m-2})$ with $k(x_{m-1})$ to obtain $k(x_{m-2})$
- $\ldots$
- u decrypts $e(x_0)$ with $k(x_1)$ to obtain $k(x_0) = k(x')$
- u decrypts the data encrypted with $k(x')$

The RSA cryptosystem

Let $n = pq$ where p and q are large primes

A user u has a public key e and a private key d such that $(e, \phi(n)) = 1$ and $e \cdot d = 1 \mod \phi(n)$

A message M to be read only by u is encrypted by computing $C = M^e \mod n$

u can decrypt C by computing

$$C^d = (M^e)^d = M^{de} = M^{k\phi(n)+1} = M$$

The RSA problem and assumption

Given a ciphertext C and the public key e an adversary a can recover M by computing $C^{1/e} \mod n$

- In other words, a has to compute integral roots of $C \mod n$
- This is the *RSA problem*

The *RSA assumption* is that the RSA problem is computationally hard when n is sufficiently large and M is a randomly chosen integer between 0 and $n-1$

- It is known to be as difficult as factoring n when $e = 2$ (Rabin 1979)

The Akl-Taylor scheme

Its security depends on the RSA assumption

Hinges on the definition of a public parameter $\mathbf{e}(x)$

- Used to derive a secret key $k(x)$
- Has the property that $\mathbf{e}(x) \mid \mathbf{e}(y)$ iff $x \geqslant y$

The Akl-Taylor scheme – Key generation

(1) Choose large primes p and q and publish $n = pq$

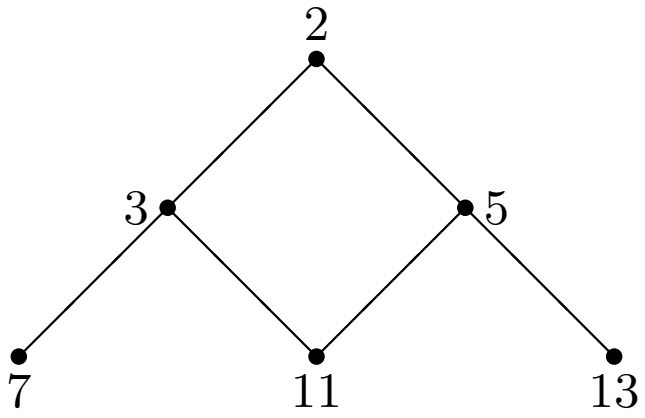
(2) Choose $\kappa \in [2, n-1]$ such that $(\kappa, n) = 1$

(3) For each $x \in X$, choose a distinct prime $e(x)$

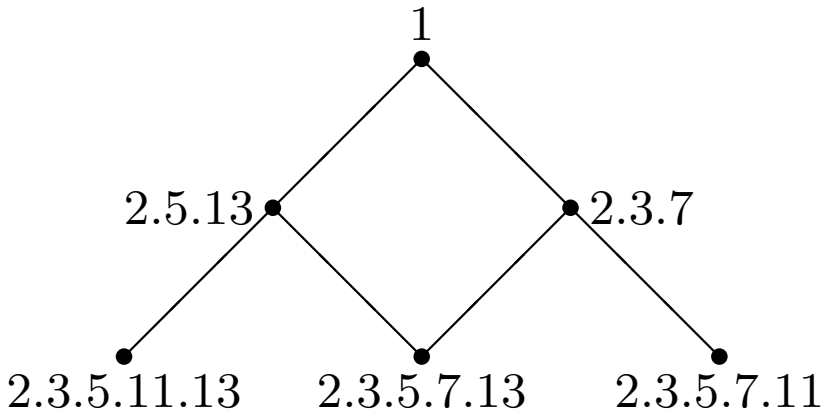
(4) For each $x \in X$, define and publish $\mathbf{e}(x) = \prod_{y \not\leqslant x} e(y)$

(5) For each $x \in X$, compute secret key $k(x) = \kappa^{\mathbf{e}(x)} \mod n$

The Akl-Taylor scheme – A simple example



$e(x)$



$\mathbf{e}(x)$

The Akl-Taylor scheme – Key derivation

Let $y \leqslant x$ and suppose the holder of $k(x)$ wishes to compute $k(y)$

Then he computes

$$\begin{aligned}(k(x))^{\mathbf{e}(y)/\mathbf{e}(x)} \mod n &= \left(\kappa^{\mathbf{e}(x)}\right)^{\mathbf{e}(y)/\mathbf{e}(x)} \mod n \\ &= \kappa^{\mathbf{e}(y)} \mod n \\ &= k(y)\end{aligned}$$

Note that this computation is feasible (given the RSA assumption) only if $\mathbf{e}(x) \mid \mathbf{e}(y)$ and that by construction $y \leqslant x$ iff $\mathbf{e}(x) \mid \mathbf{e}(y)$

Hence the holder of $k(x)$ can always compute $k(y)$ if $y \leqslant x$

The Akl-Taylor scheme – Security considerations

Lemma 1 *A key $k(x)$ can be feasibly computed from a set of keys $\{k(y) : y \in Y, Y \subseteq X\}$ iff*

$$\gcd\{\mathbf{e}(y) : y \in Y\} \mid \mathbf{e}(x)$$

Proposition 2 *Let $Y \subseteq X$ such that for all $y \in Y$, $y \not\geqslant x$. Then $\gcd\{\mathbf{e}(y) : y \in Y\} \nmid \mathbf{e}(x)$.*

Corollary 3 *Let $V \subseteq U$ be a set of users and $x \in X$ such that for all $u \in V$, $\lambda(u) \not\geqslant x$. Then it is not feasible for the members of V to obtain $k(x)$ by pooling key information.*

The MacKinnon-Taylor-Meijer-Akl scheme

We assume that there exists a partition of X into w disjoint chains

(1) Choose large primes p and q and publish $n = pq$

(2) Choose $\kappa \in [2, n-1]$ such that $(\kappa, n) = 1$

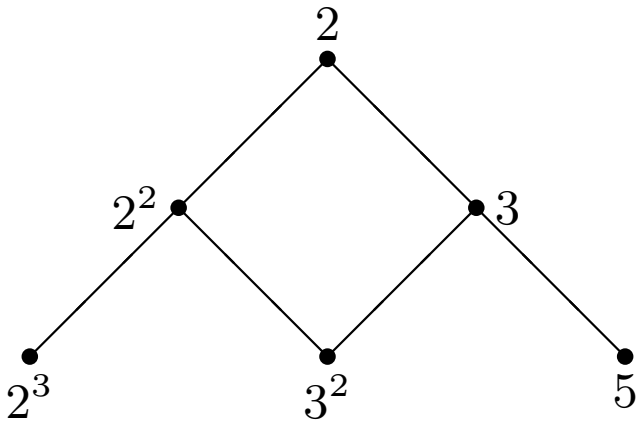
(3) Assign a prime e_i to the ith chain and, starting with the maximal element of each chain, define $e(x) = e_i^j$, where x is the jth element of the ith chain

(4) For each $x \in X$, define $\mathbf{e}(x) = \text{lcm}\{e(y) : y \not\leqslant x\}$

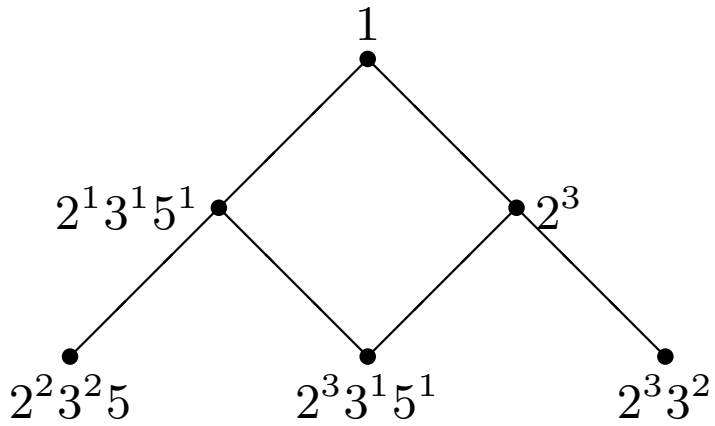
(5) For each $x \in X$, compute secret key $k(x) = \kappa^{\mathbf{e}(x)} \mod n$

Key derivation is similar to Akl-Taylor scheme

The MTMA scheme – A simple example

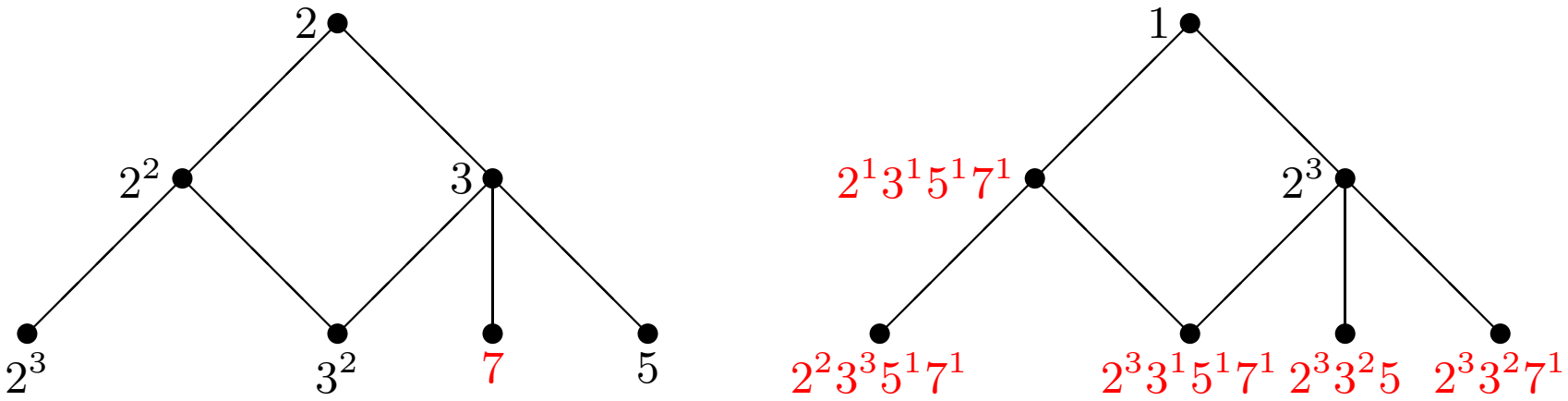


$e(x)$



$\mathbf{e}(x)$

The MTMA scheme – Updating public parameters



Any element not in the order filter generated by the new element must have its public parameter updated

The Harn-Lin scheme

The Akl-Taylor and MTMA schemes are "top-down" schemes

Expensive to update keys in tree-like posets when new minimal elements are added

The Harn-Lin scheme aims to address this issue by assigning key material in a "bottom-up" fashion

A public parameter is used to derive a secret value that is used as the exponent in the one-way function (as in Akl-Taylor)

- Based on ideas used in Akl-Taylor and RSA

The Harn-Lin scheme – Key generation

(1) Choose large primes p and q and publish $n = pq$

(2) Choose $\kappa \in [2, n-1]$ such that $(\kappa, n) = 1$

(3) For each $x \in X$, choose a prime $e(x)$ and compute $d(x)$, where $e(x) \cdot d(x) = 1 \mod \phi(n)$

(4) For each $x \in X$, define

$$\mathbf{e}(x) = \prod_{y \leqslant x} e(y) \quad \text{and} \quad \mathbf{d}(x) = \prod_{y \leqslant x} d(y) \mod \phi(n)$$

(5) For each $x \in X$, compute secret key $k(x) = \kappa^{\mathbf{d}(x)} \mod n$

The Harn-Lin scheme – Key derivation

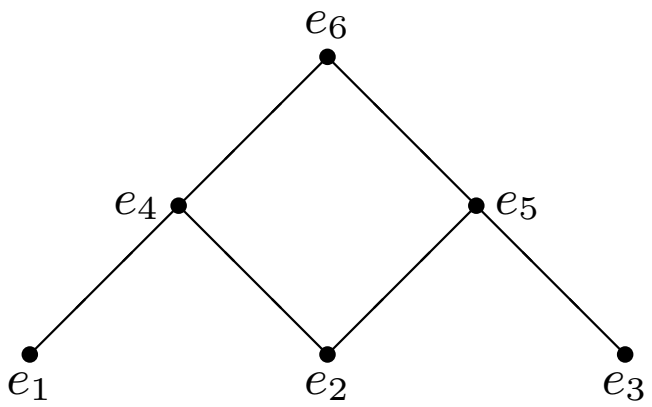
Let $y \leqslant x$ and suppose the holder of $k(x)$ wishes to compute $k(y)$

Then compute

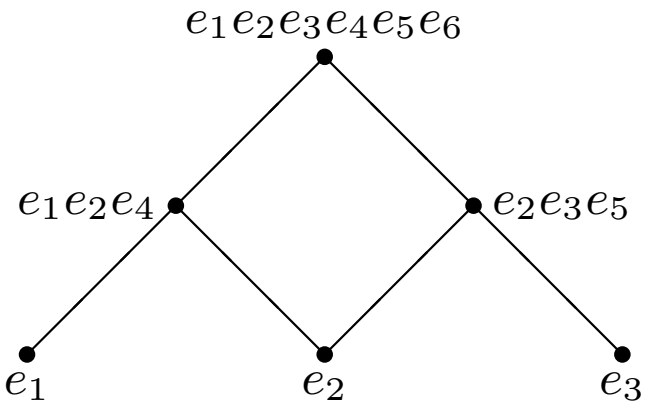
$$\begin{aligned}(k(x))^{\mathbf{e}(x)/\mathbf{e}(y)} \mod n &= \left(\kappa^{\mathbf{d}(x)}\right)^{\mathbf{e}(x)/\mathbf{e}(y)} \mod n \\ &= \kappa^{\mathbf{d}(x)\mathbf{e}(x)\mathbf{d}(y)} \mod n \\ &= \kappa^{\mathbf{d}(y)} \mod n \\ &= k(y)\end{aligned}$$

Note that this computation is only feasible if $\mathbf{e}(y) \mid \mathbf{e}(x)$ and that $y \leqslant x$ iff $\mathbf{e}(y) \mid \mathbf{e}(x)$, by definition of $\mathbf{e}$

The Harn-Lin scheme – A simple example



$e(x)$



$\mathbf{e}(x)$

Each $\mathbf{e}(x)$ includes a factor that is not included in $\mathbf{e}(y)$ for any $y \leqslant x$

The Harn-Lin scheme – Security considerations

Consider the simplest case, where $X = \{x_1, x_2\}$ with $x_1 < x_2$

- Let $e(x_i) = e_i$ and $d(x_i) = d_i$
- If the holder of key $k(x_1)$ wishes to derive $k(x_2)$, then he has to compute $\kappa^{\mathbf{d}_2} = \kappa^{d_1 d_2}$ given $\kappa^{\mathbf{d}_1} = \kappa^{d_1}$
- In other words, he has to compute d_2 from e_2 and n, since $\kappa^{d_1 d_2} = (\kappa^{d_1})^{d_2}$

The Harn-Lin scheme – Security considerations

Any attempt to compute $k(x)$ given $k(y)$, with $y \leqslant x$, will require the solution of one or more equations of the form $e \cdot z = 1 \mod \phi(n)$ (given n and e)

- In this case, the Harn-Lin scheme is as secure as the RSA cryptosystem

The security of the general case, in which users collaborate, is not explicitly solved

Controlling access to broadcast XML documents

Recent work to appear in ACM Workshop on Secure Web Services (October 2004)

Takes advantage of tree-like structure of XML documents

Expresses access control policy in terms of subtrees of XML document

An example XML document

```
<acm-catalog issue-date="29/09/04" issue-number="1">
    <journal>...</journal>
        ⋮
    <journal>...</journal>
    <proceedings>...</proceedings>
        ⋮
    <proceedings>...</proceedings>
</acm-catalog>
```

The journal element

```
<journal>
    <name>...</name>
    <date>...</date>
    <volume>...</volume>
    <number>...</number>
    <table-of-contents>
        <item>...</item>···<item>...</item>
    </table-of-contents>
    <paper>...</paper>
        ⋮
    <paper>...</paper>
</journal>
```

The paper element

```
<paper>
    <title>...</title>
    <pages>...</pages>
    <author>...</author>
    <abstract>...</abstract>
    <body>...</body>
    <references>...</references>
    <bibtex-entry>...</bibtex-entry>
</paper>
```

A schematic view of the ACM catalog

A ACM catalog

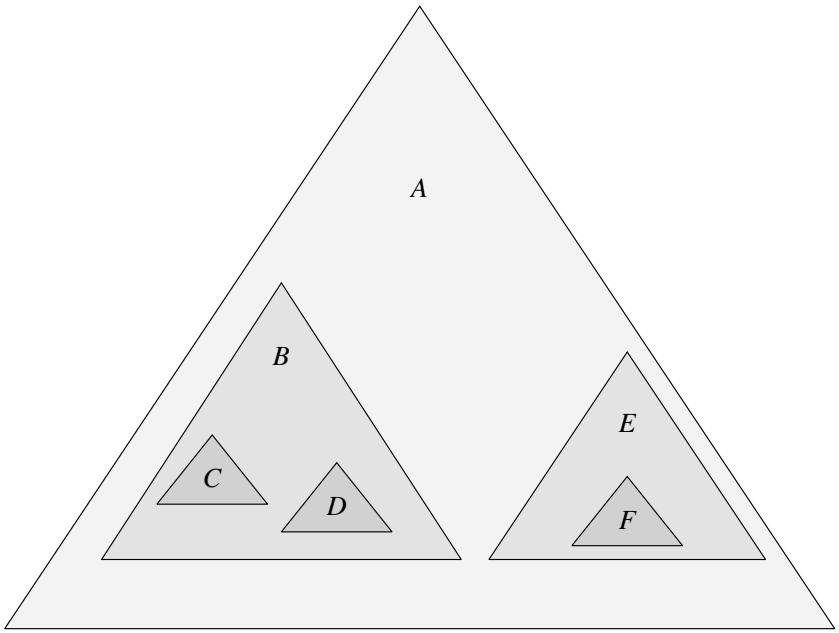
B Journals

C Journal papers

D Tables of contents

E Conference proceedings

F Conference papers



A simple access control policy

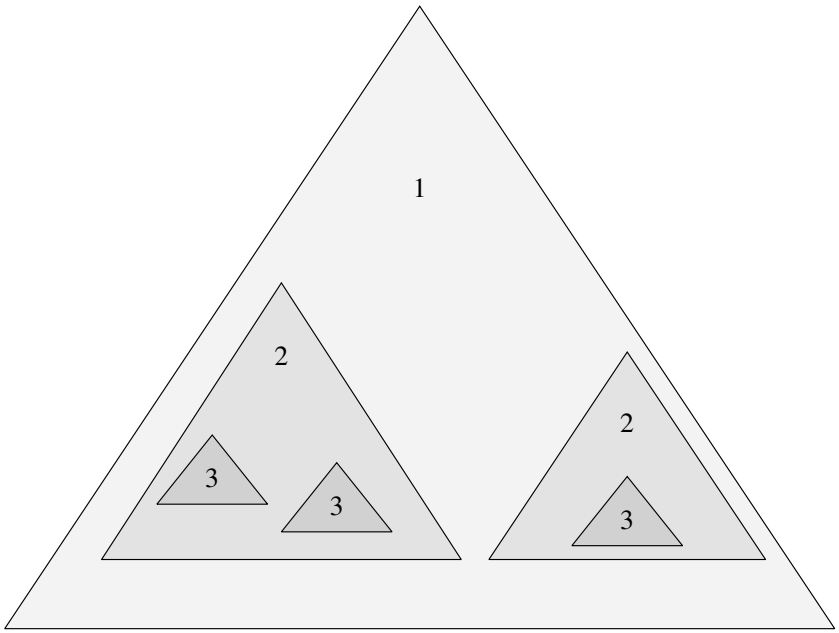
Subscriber	Permitted to access	Forbidden to access
`full`	A, B, C, D, E, F	
`restricted`	A, B, D, E	C, F
`journal`	A, B, C, D	E, F
`proceedings`	A, E, F	B, C, D

Encryption levels

Subtrees have different protection requirements

Want to distinguish between access for `full` subscribers and other types of subscribers

Integers denote depth of encryption required to protect each region

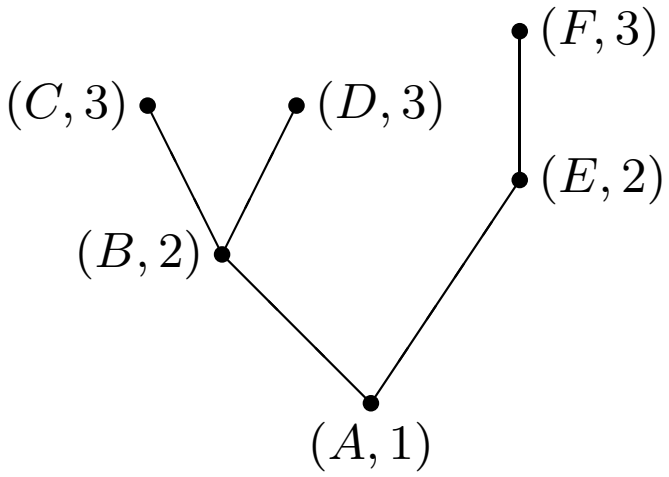


Policy hierarchies

Policy statement is a pair (o, d), where o is the subtree and d is the depth of encryption

Partial order on set of policy statements forms a policy hierarchy

- $(o, d) \leqslant (o', d')$ iff $d \leqslant d'$ and o is contained in o'

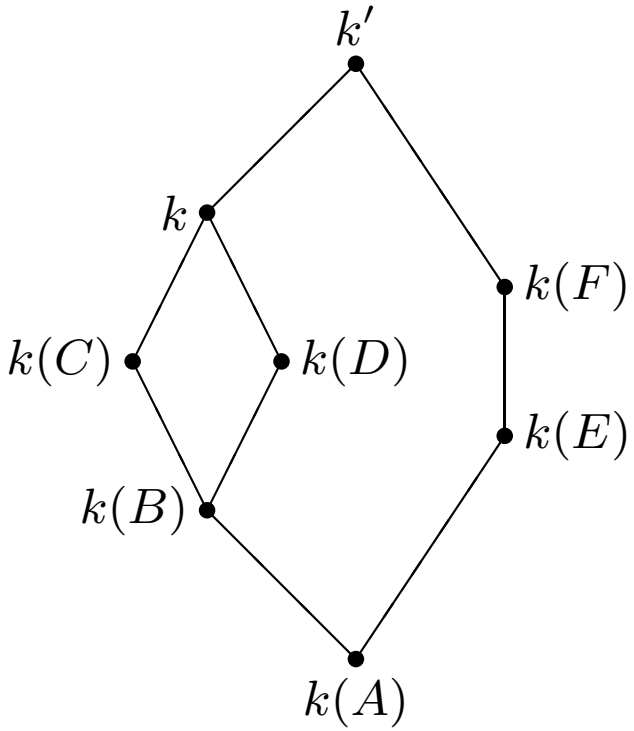


Key hierarchies

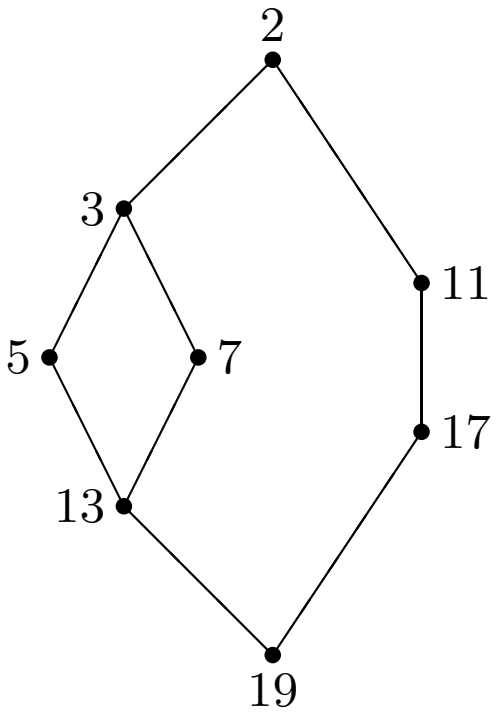
Associate keys with roles (subscriber classifications)

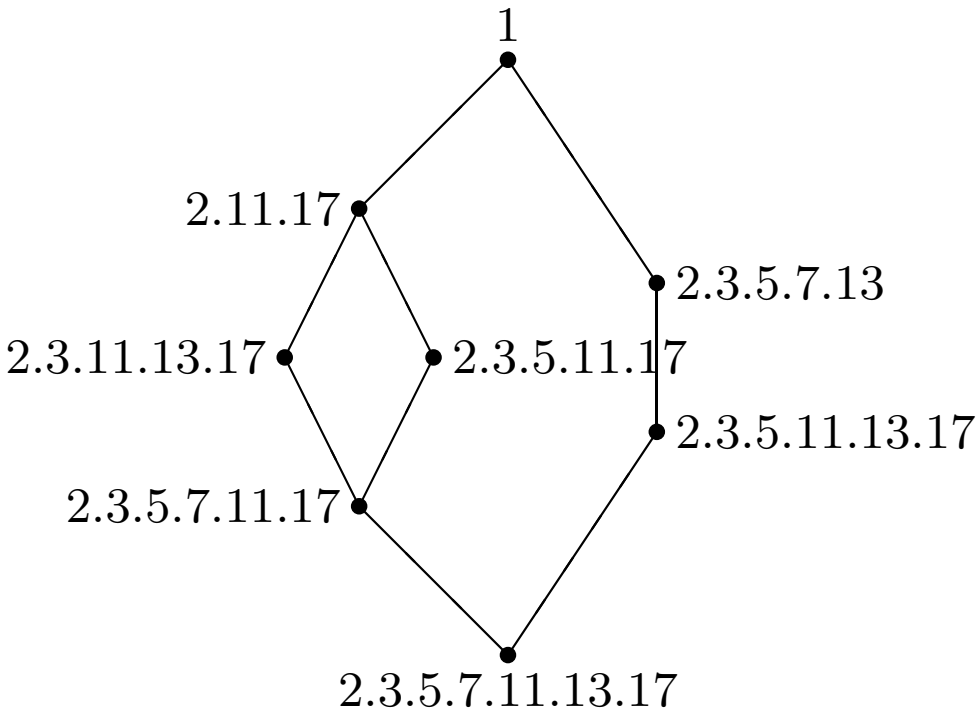
- k' for `full` subscribers
- k for `journal` subscribers
- ...

Apply Akl-Taylor to key hierarchy



Applying Akl-Taylor





Current and future research

Enforcing an information flow policy defined over an arbitrary directed graph

- Handling more complex access control policies

Improving efficiency

- Minimizing the size of keys
- Minimizing the number of primes

References

[1] S.G. Akl and P.D. Taylor. Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on Computer Systems*, 1(3):239–248, 1983.

[2] D.E. Bell and L. LaPadula. Secure computer systems: Unified exposition and Multics interpretation. Technical Report MTR-2997, Mitre Corporation, Bedford, Massachusetts, 1976.

[3] J. Crampton. Applying hierarchial and role-based access control to XML documents. In *Proceedings of 2004 ACM Workshop on Secure Web Services*, 2004.

[4] E. Gudes. The design of a cryptography based secure file system. *IEEE Transactions on Software Engineering*, 6(5):411–420, 1980.

[5] L. Harn and H.Y. Lin. A cryptographic key generation scheme for multilevel data security. *Computers and Security*, 9(6):539–546, 1990.

[6] S.J. MacKinnon, P.D. Taylor, H. Meijer, and S.G. Akl. An optimal algorithm for assigning cryptographic keys to control access in a hierarchy. *IEEE Transactions on Computers*, C-34(9):797–802, 1985.

[7] R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.