Using Encryption to Enforce an Information Flow Policy – Research Directions

Jason Crampton

Information Security Group
Royal Holloway, University of London

The problem

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 $y \nleqslant x$ it is not possible to derive secret key $k(y)$

Outline of talk

- Review of yesterday's talk
- A hybrid scheme
- Embedding a poset into a lattice of divisors
- Policies and schemes based on directed graphs
- Future work

The Akl-Taylor scheme – Key generation

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

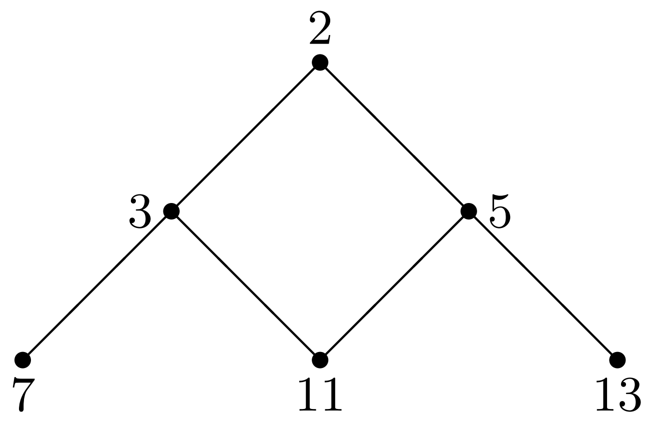
(2) Choose $\kappa \in [2, n-1]$ such that $(\kappa, n) = 1$

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

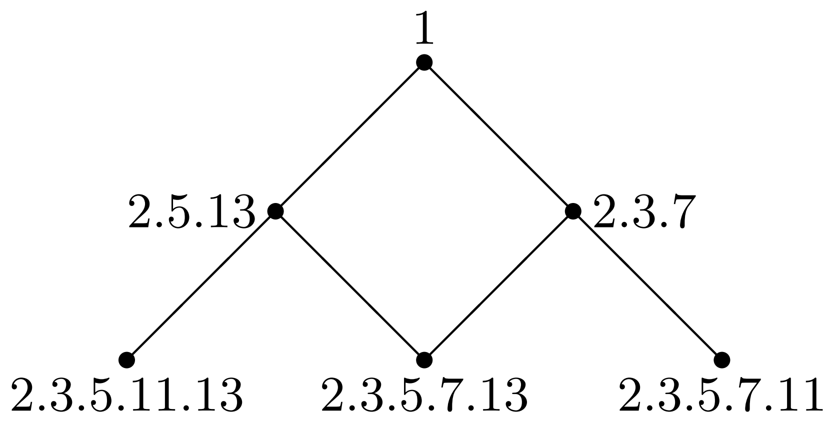
(4) For each $x \in X$, define and publish $\mathbf{e}(x) = \prod_{y \not\leqslant x} e(y)$

(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 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 $(\kappa, n) = 1$

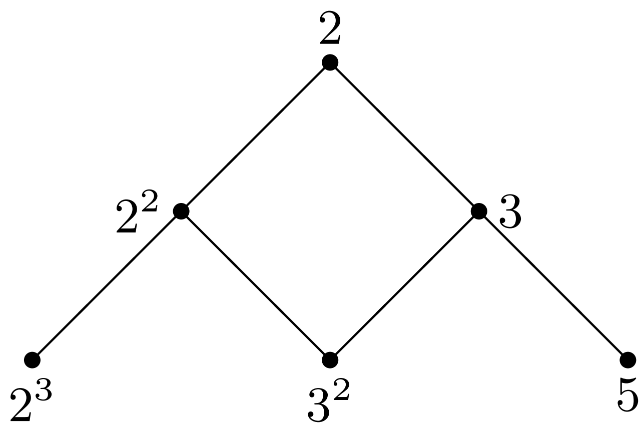
(3) Assign a prime e_i to the ith chain and, starting with the maximal element of each chain, define $e(x) = e_i^j$, where x is the jth element of the ith chain

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

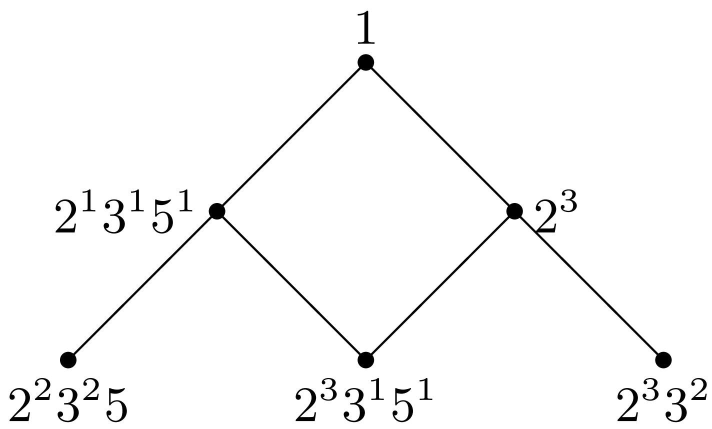
(5) For each $x \in X$, compute secret key $k(x) = \kappa^{\mathbf{e}(x)} \mod n$

Key derivation is similar to Akl-Taylor scheme

The MTMA scheme – A simple example



$e(x)$



$\mathbf{e}(x)$

The 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 $(\kappa, n) = 1$

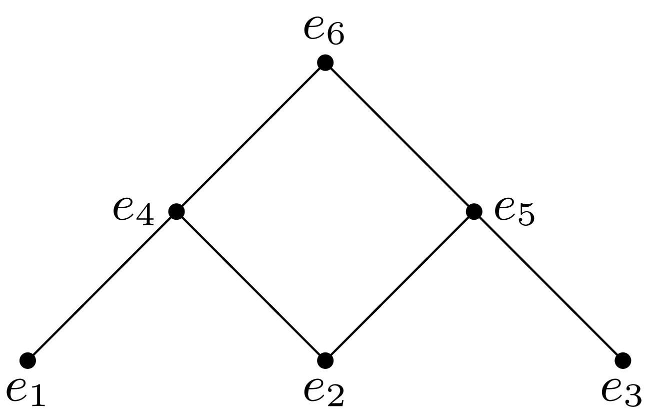
(3) For each $x \in X$, choose a prime $e(x)$ and compute $d(x)$, where $e(x) \cdot d(x) = 1 \mod \phi(n)$

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

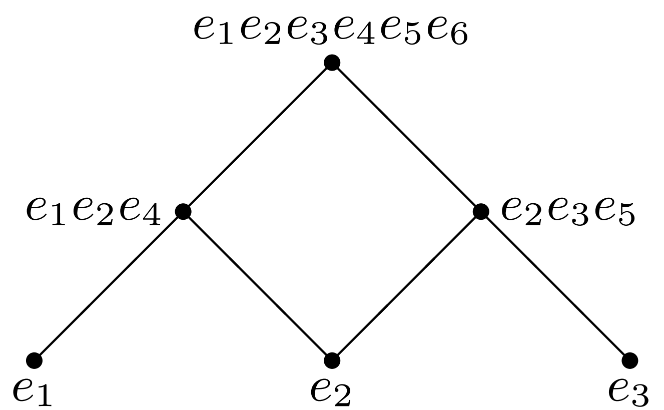
$$\mathbf{e}(x) = \prod_{y \leqslant x} e(y) \quad \text{and} \quad \mathbf{d}(x) = \prod_{y \leqslant x} d(y) \mod \phi(n)$$

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

The Harn-Lin scheme – 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 $y \leqslant x$

A hybrid scheme (Crampton)

Combine elements of the MTMA and the Harn-Lin schemes

- Reduce the number of primes required in the Harn-Lin scheme
- Reduce the difficulty of updates in the MTMA scheme

Key generation

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

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

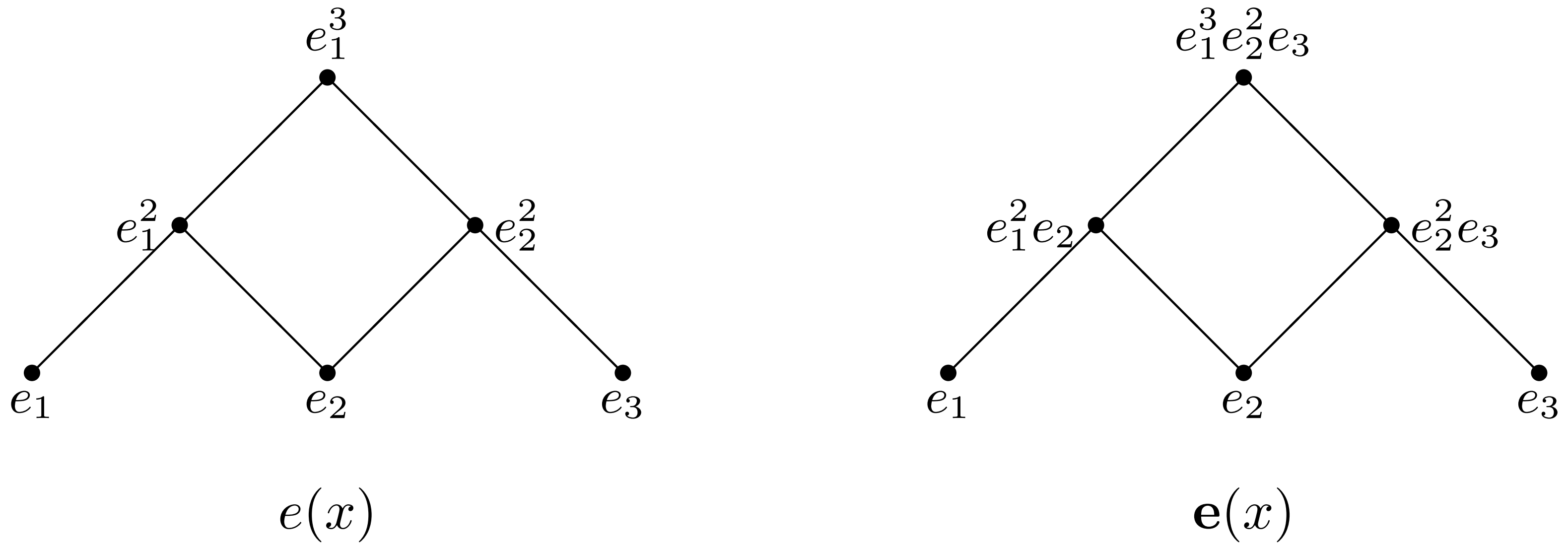
(3) Choose primes $e_1, \ldots, e_w$ and compute d_i, where $e_i \cdot d_i = 1 \mod \phi(n)$

(4) Assign e_i to the the ith chain and, starting with the *minimal* element of each chain, define $e(x) = e_i^j$, where x is the jth element in the ith chain

(5) For each $x \in X$, define $\mathbf{e}(x) = \mathrm{lcm}\{e(y) : y \leqslant x\}$ and $\mathbf{d}(x) = \mathrm{lcm}\{d(y) : y \leqslant x\} \mod \phi(n)$

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

A simple example



If the holders of keys κ^{d_1} and κ^{d_2} wish to compute $\kappa^{d_1^2 d_2}$ (say) then they must solve the equation $e_1 d_1 = 1 \mod \phi(n)$

Security considerations

Claim: Security of hybrid scheme is equivalent to that of Harn-Lin scheme

Question: Is the Harn-Lin scheme secure against *all* collaborative attacks?

Minimizing the number of primes

The Akl-Taylor and Harn-Lin schemes require n primes (where $n = |X|$)

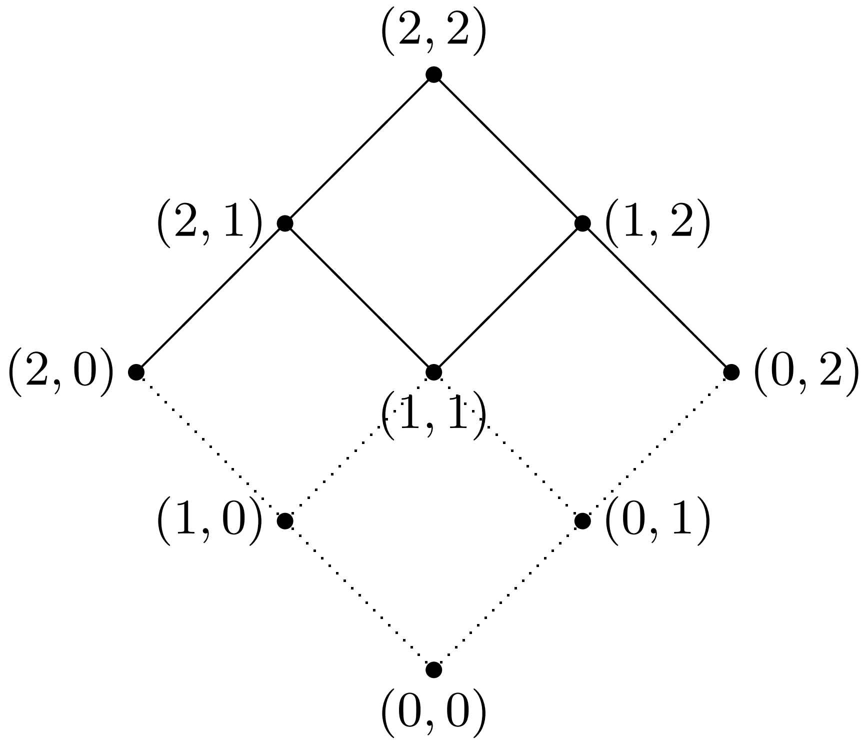
The MTMA and hybrid schemes require w primes (where w is the width of X)

Can we do better?

Minimizing the number of primes

Let m be the maximal out-degree or in-degree of a node in the Hasse diagram of X

Claim: X can be embedded in a fragment of the poset $S(a_1, \ldots, a_m)$ for suitable values of a_i



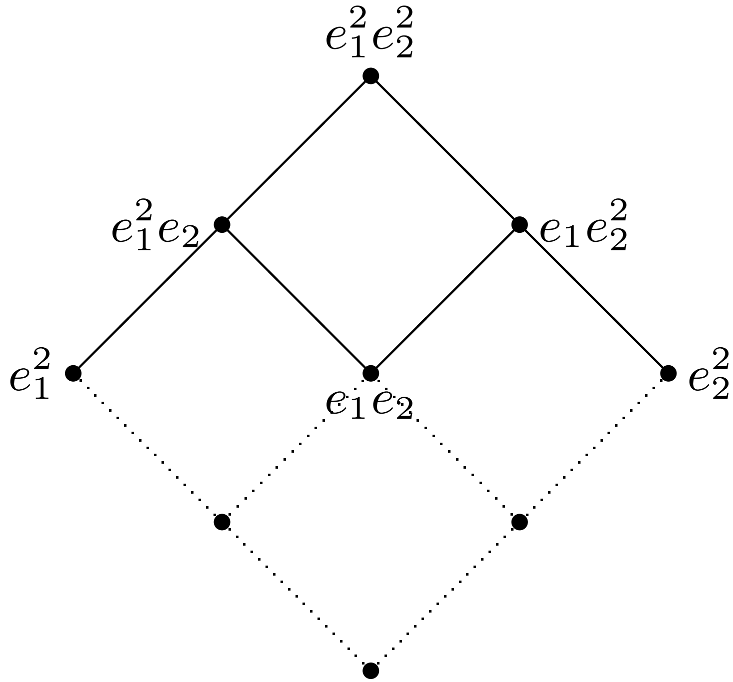
$S(2,2)$

Minimizing the number of primes

Note that $S(a_1, \ldots, a_m)$ is order isomorphic to the lattice of divisors of $\prod_{i=1}^{m} e_i^{a_i}$ for suitable choices of primes e_i

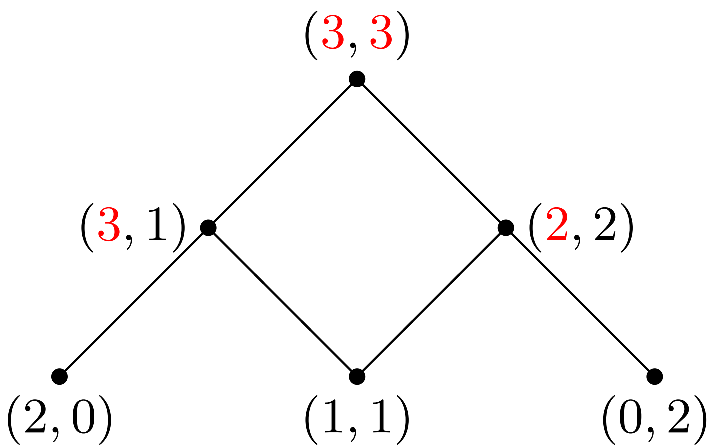
$$(b_1, \ldots, b_m) \mapsto e_1^{b_1} \ldots e_m^{b_m}$$

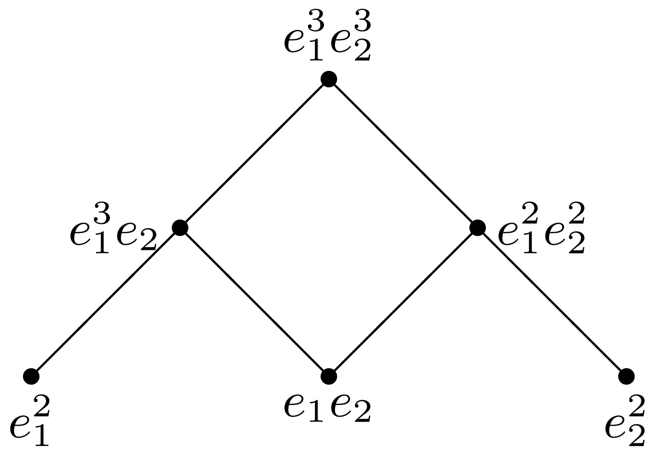
However, keyholders can collaborate to derive keys



Minimizing the number of primes

Ensure that the value of at least one co-ordinate in the parent node exceeds the corresponding value in each of the child nodes

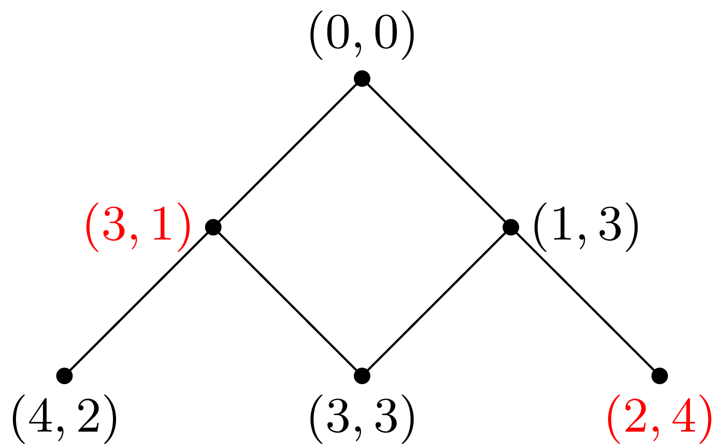




The MTMA scheme revisited

A similar method can be used for the assignment of public parameters for top-down schemes

Note that each co-ordinate in the ith level must be at least one greater than each of the corresponding co-ordinates in the $(i-1)$th level



Embedding posets in $S(a_1, \ldots, a_m)$

Is there a systematic way of assigning public values to elements of an arbitrary poset X?

Construct a mapping $\phi : X \rightarrow S(a_1, \ldots, a_m)$ such that

- ϕ is injective
- ϕ is order-preserving
- ϕ^{-1} is order-preserving

Minimizing the size of public values (and keys)

Scheme	**Largest public value**
Akl-Taylor	2.3.5.11.13
MTMA	$2^2 3^2 5$
Harn-Lin	$e_1 e_2 e_3 e_4 e_5 e_6$
Hybrid Harn-Lin-MTMA	$e_1^3 e_2^2 e_3$
Modified Harn-Lin	$e_1^3 e_2^3$
Modified MTMA	$e_1^4 e_2^2$

Minimizing the size of public values (and keys)

- It would seem that at least one public value must contain at least $n-1$ factors, where $n = |X|$
- This is intuitively reasonable ...
- ... but can it be proved?

Information flow policies for directed graphs

A poset can be thought of as the (acyclic) directed graph of the transitive closure of its Hasse diagram

Some information flow policies may

- not wish to have transitivity
- want cyclic information flow

May be important in formulating complex access control policies in non-military applications

The work of de Santis *et al*

Paper to appear in *Information Processing Letters*

Extension of Akl-Taylor to directed graphs

- Graph is transformed into a poset of height 2 and width equal to the number of nodes in the graph
- Akl-Taylor is applied to poset

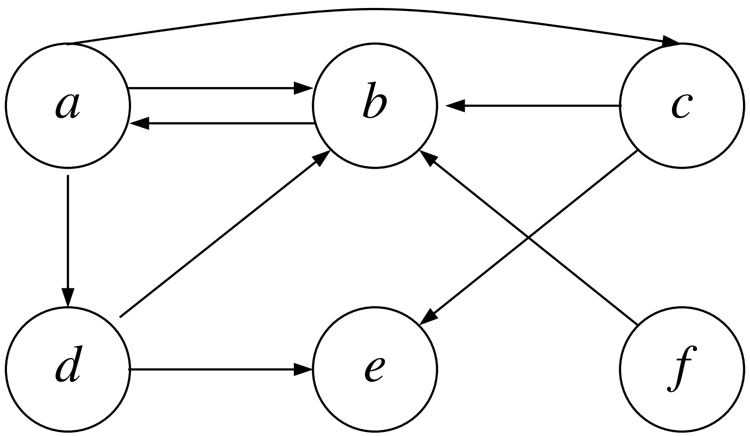
Each node x is associated with a key $k(x)$ and a secret value $s(x)$

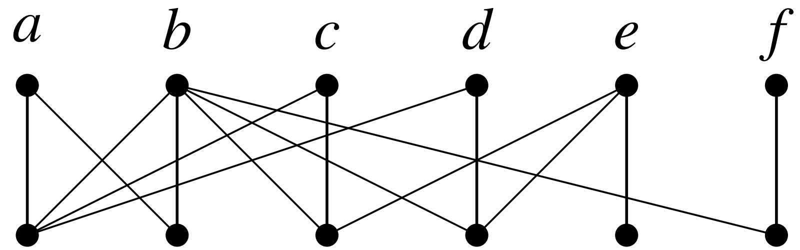
- $s(x)$ is used to derive $k(y)$ for any y such that (y, x) is an edge in the graph

The graph-poset transformation

Each node x in the graph (X, E) is associated with two elements in the poset – a lower element x_l and an upper element x_u

$x_l \lessdot y_u$ iff either $x = y$ or $(x, y) \in E$





Keys, secret values and public information

Apply Akl-Taylor scheme to poset

Define $k(x) = k(x_l) = \kappa^{\mathbf{e}(x_l)}$ and $s(x) = k(x_u) = \kappa^{\mathbf{e}(x_u)}$

Publish $e(x_l)$ and $e(x_u)$

Key derivation

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

Then he computes

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

Optimizing the scheme

de Santis *et al* note that their scheme requires $2n$ pairs of keys and secret values

They propose an optimization that requires only n pairs of keys and secret values

- Similar in style to MTMA optimization of Akl-Taylor

An alternative scheme (Crampton)

Does not require graph-poset transformation

Simpler to compute keys and secret values

Security comparable to that of Akl-Taylor and de Santis schemes

Key and secret value generation

- Choose large primes p and q and publish $n = pq$
- Choose $\kappa \in [2, n-1]$ such that $(\kappa, n) = 1$
- For each $x \in X$, choose a distinct prime $p(x)$ and define $P = \prod_{x \in X} p(x)$
- For each $x \in X$, publish $q(x) = P/p(x)$
- For each $x \in X$, define and publish $\mathbf{p}(x) = \prod_{\{y \in X : (x,y) \notin E\}} p(y)$
- For each $x \in X$, define secret value $s(x) = \kappa^{\mathbf{p}(x)} \mod n$
- For each $x \in X$, compute secret key $k(x) = \kappa^{q(x)} \mod n$

Key derivation

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

The keyholder computes

$$(s(x))^{q(y)/\mathbf{p}(x)} = \left(\kappa^{\mathbf{p}(x)}\right)^{q(y)/\mathbf{p}(x)} = \kappa^{q(y)} = k(y)$$

It can be shown that this scheme is secure against collaborative attacks

Proof is very similar to work by Akl-Taylor and de Santis *et al*

A comparison

		de Santis *et al*		Crampton	
Node x	$p(x)$	$e(x_u)$	$e(x_l)$	$\mathbf{p}(x)$	$q(x)$
a	2	5.7.11.13	3.5.7.11.13	5.7.11.13	3.5.7.11.13
b	3	11	5.7.11.13	11	2.5.7.11.13
c	5	3.7.11.13	2.3.7.11.13	3.7.11.13	2.3.7.11.13
d	7	3.5.11.13	2.3.5.11.13	3.5.11.13	2.3.5.11.13
e	11	2.3.13	2.3.13	2.3.13	2.3.5.7.13
f	13	2.3.5.7.11	2.3.5.7.11	2.3.5.7.11	2.3.5.7.11

Further research opportunities

Can we relax the restriction that no coalition of users should be able to derive keys to which they should not have access?

- Can we set some threshold value t such that no coalition of fewer than t users can derive keys they should not have access to?

Can we find other one-way functions to use as the basis for cryptographic schemes?

Can we find other applications in which these techniques are useful?

Partial orders and computer security

Role-based access control

- Central concept is role hierarchy (modelled as poset)
- Antichains are very important in RBAC
- Many interesting mathematical questions regarding lattice of antichains

Access control policies for hierarchical structures

- File systems
- XML documents

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

[3] 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.

[4] A. De Santis, A.L. Ferrara, and B. Masucci. Cryptographic key assignment schemes for any access control policy. *Information Processing Letters*. To appear.