A Practical Application of POSET: Hierarchical Access Control

Dr. Ashok Kumar Das

IEEE Senior Member
Associate Professor
Center for Security, Theory and Algorithmic Research
International Institute of Information Technology, Hyderabad

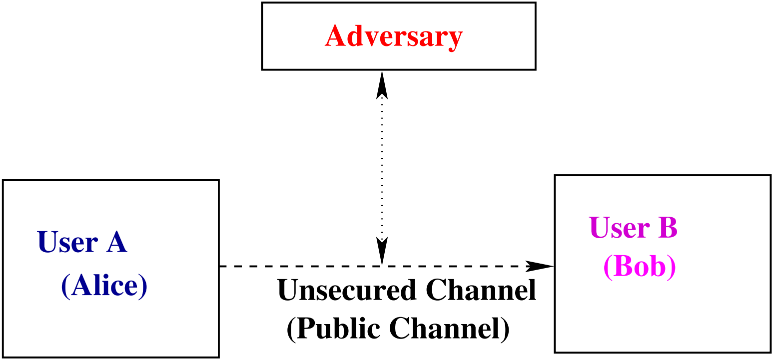
E-mail: *ashok.das@iiit.ac.in*
URL: `http://www.iiit.ac.in/people/faculty/ashokkdas`
`https://sites.google.com/site/iitkgpakdas/`

Overview of Cryptography

What is Cryptography?

- Cryptography is the study of **mathematical techniques** related to aspects of information security such as confidentiality, data integrity, entity authentication, message authentication (data origin authentication) and non-repudiation.
- Cryptography is not the only means of providing information security, but rather one set of techniques.
- Now-a-days, cryptography has moved from an art to a science. Thus, cryptography is the science of keeping secrets secret.

Consider the following simple two-party communication model:



Introduction to Cryptography

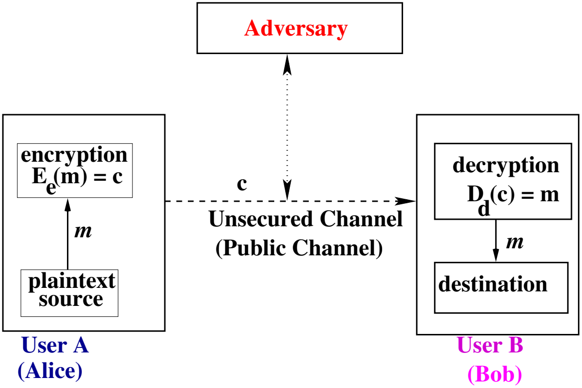
- An "**adversary**" is an entity in a two-party communication which is neither the sender nor the receiver, and which tries to defeat the information security service being provided between the sender and the receiver.
- A "**channel**" is a means of conveying information from one entity to another entity.
- An "**unsecured channel**" is one from which parties other than the sender and the receiver can reorder, delete, insert, or read the data being transmitted.
- A "**secured channel**" is one from which an adversary does not have the ability to reorder, delete, insert, or read the data being transmitted.

Types of adversary

- A **"passive adversary"** is an adversary who is only capable of reading information from an unsecured channel.
- An **"active adversary"** is an adversary who is capable to transmit, alter, or delete information on an unsecured channel.

Introduction to Cryptography

Consider the following simple two-party communication model with encryption:



$E_e(\cdot)/D_d(\cdot)$: encryption/decryption transformation using the encryption key e and decryption key d; $D_d = E_e^{-1}$; m: plaintext message and c: ciphertext message

Cryptology = Cryptography + Cryptanalysis

Definition

An encryption scheme (cipher or cryptosystem) is said to be ***breakable*** if a third party, without prior knowledge of the key pair (e, d) where e is the encryption key and d is the corresponding decryption key, can systematically recover plaintext from corresponding ciphertext within some appropriate time frame.

Goal: We want this problem for an adversary (attacker) to be NP-hard (computationally infeasible).

Definition (Brute-force attack)

An encryption scheme can be broken by trying all possible keys to see which one the communicating parties are using (assuming that the class of encryption functions is public knowledge).
This is called an exhaustive search of the key space.

What is meant by “Security lies in the keys” (using brute-force attack)

Key size (bits)	**Number of alternative keys**	**Time required at 10^6 decryptions per microsecond**
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.9×10^{30} years

Symmetric-Key Encryption

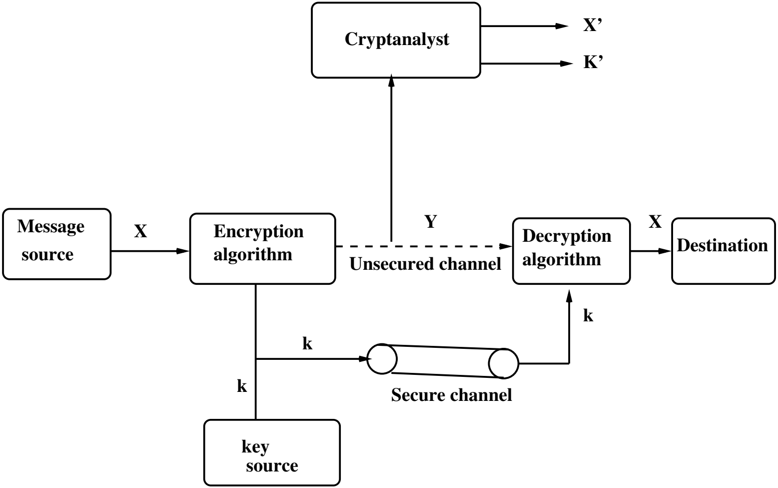


Figure: Model of conventional encryption

Public-Key Cryptography

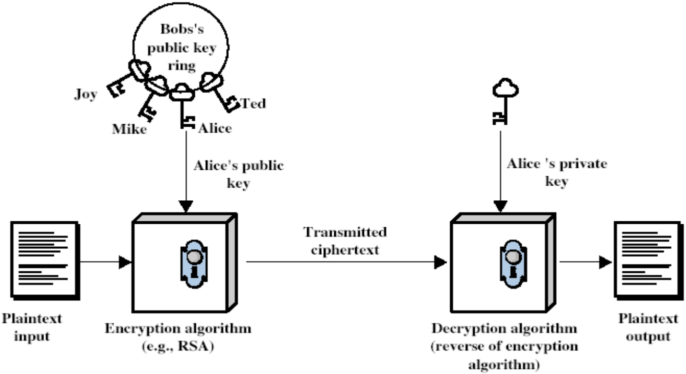


Figure: Model of public key encryption

Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC)

- ECC makes use of the elliptic curves (not ellipses) in which the variables and coefficients are all restricted to elements of a finite field.
- Two family of elliptic curves are used in ECC:
  - prime curves defined over Z_p, that is, $GF(p)$, p being a prime.
  - binary curves constructed over $GF(2^n)$.

Elliptic Curve Cryptography (ECC)

Elliptic curves over modulo a prime $GF(p)$

Definition

Let $p > 3$ be a prime. The elliptic curve $y^2 = x^3 + ax + b$ over Z_p is the set $E_p(a, b)$ of solutions $(x, y) \in E_p(a, b)$ to the congruence

$$y^2 \quad = \quad x^3 + ax + b \pmod{p},$$

where $a, b \in Z_p$ are constants such that $4a^3 + 27b^2 \neq 0 \pmod{p}$, together with a special point $\mathcal{O}$ called the point at infinity (or zero point).

Elliptic curves over modulo a prime $GF(p)$

Properties of Elliptic Curves

- An elliptic curve $E_p(a, b)$ over Z_p (p prime, $p > 3$) will have roughly p points on it.
- More precisely, a well-known theorem due to Hasse asserts that the number of points on $E_p(a, b)$, which is denoted by $\#E$, satisfies the following inequality:

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}.$$

- In addition, $E_p(a, b)$ forms an abelian or commutative group under addition modulo p operation.

Elliptic Curve Cryptography (ECC)

References

- N. Koblitz. Elliptic Curve Cryptosystems. Mathematics of Computation, Vol. 48, pp. 203-209, 1987.
- V. Miller. Uses of elliptic curves in cryptography. Advances in Cryptology - CRYPTO'85, Lecture Notes in Computer Science (LNCS), Springer, Vol. 218, pp. 417-426, 1986.
- Douglas R. Stinson. Cryptography: Theory and Practice, Chapman & Hall/CRC, 2^{nd} Edition, 2005.

Elliptic Curve Cryptography (ECC)

Elliptic curves over modulo a prime $GF(p)$

Finding an inverse

- The inverse of a point $P = (x_P, y_P) \in E_p(a, b)$ is $-P = (x_P, -y_P)$, where $-y$ is the additive inverse of y.
- For example, if $p = 13$, the inverse of $(4, 2)$ is $(4, -2) \pmod{13}$ $= (4, 11)$.

Elliptic Curve Cryptography (ECC)

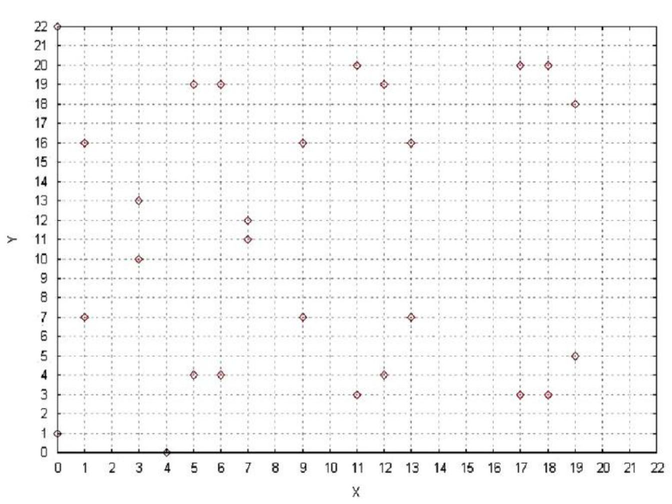
Finding all points on an elliptic curve

Algorithm: EllipticCurvePoints (p, a, b)

$x \leftarrow 0$
while $x < p$ **do**
  $w \leftarrow (x^3 + ax + b) \pmod{p}$
  if w is a perfect square in Z_p) **then**
    Output $(x, \sqrt{w}), (x, -\sqrt{w})$
  end if
  $x \leftarrow x + 1$
end while

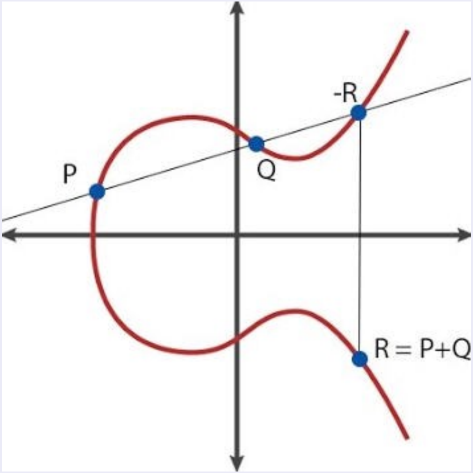
Elliptic Curve Cryptography (ECC)

Example of elliptic curve in case of $y^2 = x^3 + x + 1 \pmod{23}$.

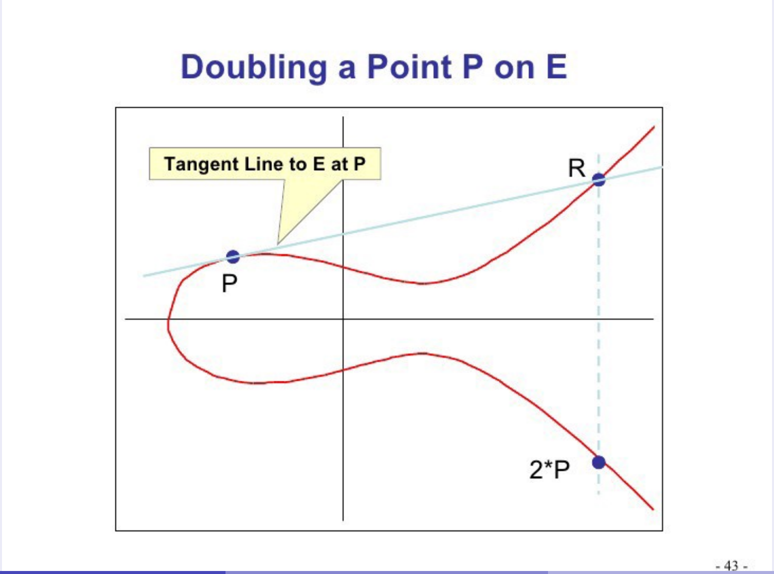


Elliptic Curve Cryptography (ECC)

Point addition on elliptic curve over finite field $GF(p)$



Doubling on elliptic curve over finite field $GF(p)$



Elliptic Curve Cryptography (ECC)

Point addition on elliptic curve over finite field $GF(p)$

Let G be the base point on $E_p(a, b)$ whose order be n, that is, $nG = G + G + \ldots + G(n\, times) = \mathcal{O}$.
If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two points on elliptic curve $y^2 = x^3 + ax + b \pmod p$, $R = (x_R, y_R) = P + Q$ is computed as follows:

$$x_R = (\lambda^2 - x_P - x_Q) \pmod p,$$

$$y_R = (\lambda(x_P - x_R) - y_P) \pmod p,$$

$$\text{where } \lambda = \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} \pmod p, \text{if } P \neq -Q & \textbf{[Point Addition]} \\ \frac{3{x_P}^2 + a}{2y_P} \pmod p, \text{if } P = Q. & \textbf{[Point Doubling]} \end{cases}$$

Elliptic Curve Cryptography (ECC)

Scalar multiplication on elliptic curve over finite field $GF(p)$

If $P = (x_P, y_P)$ be a point on elliptic curve $y^2 = x^3 + ax + b \pmod{p}$, then then $5P$ is computed as $5P = P + P + P + P + P$.
Think about optimization method?

Reference: N Tiwari, S Padhye. Provable Secure Multi-Proxy Signature Scheme without Bilinear Maps. International Journal of Network Security, Vol. 17, No. 1, pp. 288-293, 2015.

Elliptic Curve Cryptography (ECC)

Problem: Consider two points $P = (11, 3)$ and $Q = (9, 7)$ in the elliptic curve $E_{23}(1, 1)$. Compute $P + Q$ and $2P$.

In order to compute $R = P + Q = (x_R, y_R)$, we first compute λ as

$$\begin{aligned} \lambda &= \frac{7-3}{9-11} \pmod{23} \\ &= -2 \pmod{23} \\ &= 21. \end{aligned} \tag{1}$$

Thus, x_R and y_R are derived as

$$x_R = (21^2 - 11 - 9) \pmod{23} = 7,$$
$$y_R = (21(11-7) - 3) \pmod{23} = 12.$$

As a result, $P + Q = (7, 12)$.

Elliptic Curve Cryptography (ECC)

Problem: Consider two points $P = (11, 3)$ and $Q = (9, 7)$ in the elliptic curve $E_{23}(1, 1)$. Compute $P + Q$ and $2P$.

In order to compute $R = 2P = (x_R, y_R)$, we must first derive λ as follows:

$$\lambda = \frac{3(11^2) + 1}{2 \times 3} \pmod{23} = 7.$$

Hence, $R = P + P = (x_R, y_R)$ is computed as

$$x_R = (7^2 - 11 - 11)(\bmod\ 23) = 4,$$
$$y_R = (7(11 - 4) - 3)(\bmod\ 23) = 0,$$

and, thus $2P = (4, 0)$.

Elliptic Curve Computational Problems

Elliptic Curve Discrete Logarithm Problem (ECDLP)

- Let $E_p(a, b)$ be an elliptic curve modulo a prime p.
- Given two points $P \in E_p(a, b)$ and $Q = kP \in E_p(a, b)$, for some positive integer k, where $Q = kP$ represent the point P on elliptic curve $E_p(a, b)$ be added to itself k times.
- Then the elliptic curve discrete logarithm problem (ECDLP) is to determine k given P and Q.
- It is computationally easy to calculate Q given k and P, but it is computationally infeasible to determine k given Q and P, when the prime p is large.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

In other words, ECDLP can be also formally defined as follows. For any PPT algorithm, say A (in the security parameter l), $Pr[A(P, Q) = k] < \epsilon(l)$, where $\epsilon(l)$ is a negligible function depending on l.

References:

- Vanga Odelu, **Ashok Kumar Das**, and Adrijit Goswami. "A secure effective key management scheme for dynamic access control in a large leaf class hierarchy," in ***Information Sciences (Elsevier)***, Vol. 269, No. C, pp. 270-285, 2014. (2020 SCI Impact Factor: 6.795) [This article has been downloaded or viewed 484 times since publication during the period October 2013 to September 2014]
- **Ashok Kumar Das**, Nayan Ranjan Paul, and Laxminath Tripathy. "Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem," in ***Information Sciences (Elsevier)***, Vol. 209, No. C, pp. 80 - 92, 2012. (2020 SCI Impact Factor: 6.795)

Elliptic Curve Cryptography (ECC)

Definition (Elliptic curve computational Diffie-Hellman problem (ECCDHP))

Let $P \in E_p(a,b)$ be a point in $E_p(a,b)$. The ECCDHP states that given the points $k_1.P \in E_p(a,b)$ and $k_2.P \in E_p(a,b)$ where $k_1, k_2 \in Z_p^*$, it is computationally infeasible to compute $k_1 k_2.P$, where $Z_p^* = \{1, 2, \cdots, p-1\}$.

Elliptic Curve Cryptography (ECC)

Definition (Elliptic curve decisional Diffie-Hellman problem (ECDDHP))

Let $P \in E_p(a,b)$ be a point in $E_p(a,b)$. The ECDDHP states that given a quadruple $(P,\ k_1.P,\ k_2.P,\ k_3.P)$, decide whether $k_3 = k_1 k_2$ or a uniform value, where $k_1, k_2, k_3 \in Z_p^*$.

The ECDLP, ECCDHP and ECDDHP are computationally infeasible when p is large. To make ECDLP, ECCDHP and ECDDHP intractable, p should be chosen at least 160-bit prime.

Hierarchical Access Control

Hierarchical Access Control

Overview of Hierarchical Access Control

- Hierarchical access control is a fundamental problem in computer and network systems.
- In a hierarchical access control, a user of higher security level class has the ability to access information items (such as message, data, files, etc.) of other users of lower security classes.
- A user hierarchy consists of a number n of disjoint security classes, say, SC_1, SC_2, ..., SC_n. Let this set be $SC = \{SC_1, SC_2, \ldots, SC_n\}$.
- A binary partially ordered relation $\geq$ is defined in SC as $SC_i \geq SC_j$, which means that the security class SC_i has a security clearance higher than or equal to the security class SC_j.

Hierarchical Access Control

Overview of Hierarchical Access Control

- In addition the relation $\geq$ satisfies the following properties:
  - **[Reflexive property]** $SC_i \geq SC_i, \forall SC_i \in SC$.
  - **[Anti-symmetric property]** If $SC_i, SC_j \in SC$ such that $SC_i \geq SC_j$ and $SC_j \geq SC_i$, then $SC_i = SC_j$.
  - **[Transitive property]** If $SC_i, SC_j, SC_k \in SC$ such that $SC_i \geq SC_j$ and $SC_j \geq SC_k$, then $SC_i \geq SC_k$.
- If $SC_i \geq SC_j$, we call SC_i as the predecessor of SC_j and SC_j as the successor of SC_i. If $SC_i \geq SC_k \geq SC_j$, then SC_k is an intermediate security class. In this case SC_k is the predecessor of SC_j and SC_i is the predecessor of SC_k.
- In a user hierarchy, the encrypted message by a successor security class is only decrypted by that successor class as well as its all predecessor security classes in that hierarchy.

Overview of Hierarchical Access Control

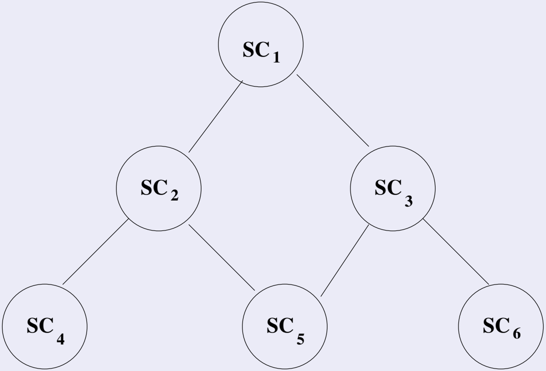


Figure: A small sample of poset in a user hierarchy.

Hierarchical Access Control

Applications of Hierarchical Access Control

- Military
- Government schools and colleges
- Private corporations
- Computer network systems
- Operating systems
- Database management systems

Chung et al.'s User Hierarchical Access Control Scheme

Reference

- Y. F. Chung, H. H. Lee, F. Lai and T. S. Chen, "Access control in user hierarchy based on elliptic curve cryptosystem", ***Information Sciences (Elsevier)***, vol. 178, no. 1, pp. 230-243, 2008 (2020 SCI Impact Factor: 6.795).

Chung et al.'s User Hierarchical Access Control Scheme

Relationship Building Phase

- CA (central authority) builds a hierarchical structure for controlling access according to the relationships among the nodes in the hierarchy.
- Let $U = \{SC_1, SC_2, \ldots, SC_n\}$ be a set of n security classes in the hierarchy. Assume that SC_i is a security class with higher clearance and SC_j a security class with lower clearance, that is, $SC_i \geq SC_j$.
- A legitimate relationship $(SC_i, SC_j) \in R_{i,j}$ between two security classes SC_i and SC_j exists in the hierarchy if SC_i can access SC_j.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Generation Phase

CA performs the following steps:

- **Step 1:** Randomly selects a large prime p.
- **Step 2:** Selects an elliptic curve $E_p(a, b)$ defined over Z_p such that the order of $E_p(a, b)$ lies in the interval $[p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$.
- **Step 3:** Selects a one-way function $h(\cdot)$ to transform a point into a number and a base point G_j from $E_p(a, b)$ for each security class SC_j $1 \leq j \leq n$.
- **Step 4:** For each security class SC_j $(1 \leq j \leq n)$, selects a secret key sk_j and a sub-secret key s_j.
- **Step 5:** For all $\{SC_i | (SC_i, SC_j)\} \in R_{i,j}$, computes the followings:
  $s_i G_j = (x_{j,i}, y_{j,i})$,
  $h(x_{j,i} || y_{j,i})$, where $||$ is a bit concatenation operator.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Generation Phase (Continued...)

- **Step 6:** Finally, computes the public polynomial $f_j(x)$ using the values of $h(x_{j,i}||y_{j,i})$ as

$$f_j(x) \;=\; \prod_{SC_i \geq SC_j} (x - h(x_{j,i}||y_{j,i})) + sk_j \pmod{p}$$

- **Step 7:** Sends sk_j and s_j to the security class SC_j via a secret channel.
- **Step 8:** Announces p, $h(\cdot)$, G_j, $f_j(x)$ as public.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Derivation Phase

In order to compute the secret keys sk_j of all successors, SC_j, the predecessor SC_i, for which the relationships $(SC_i, SC_j) \in R_{i,j}$ between SC_i and SC_j hold, proceeds as follows:

- Step 1: For $\{SC_i | (SC_i, SC_j)\} \in R_{i,j}$, computes the followings:
  $s_i G_j = (x_{j,i}, y_{j,i})$,
  $h(x_{j,i} || y_{j,i})$.
- Step 2: Computes the secret key sk_j using $h(x_{j,i} || y_{j,i})$ as follows:

$$\begin{aligned} f_j(x) &= \prod_{SC_i \geq SC_j} (x - h(x_{j,i} || y_{j,i})) + sk_j \pmod{p}, \\ f_j(h(x_{j,i} || y_{j,i})) &= sk_j \pmod{p}. \end{aligned}$$

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Derivation Phase (Continued...)

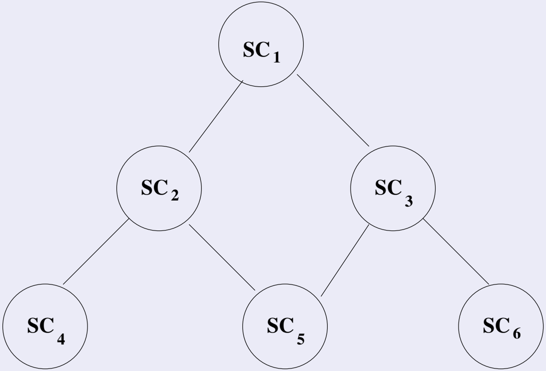


Figure: A small sample of poset in a user hierarchy.

Chung et al.'s User Hierarchical Access Control Scheme (Continued...)

Key Derivation Phase (Continued...)

$$
\begin{aligned}
f_j(x) &= \prod_{SC_i \geq SC_j} [x - h(x_{j,i}||y_{j,i})] + sk_j \pmod p, \\
SC_1 : f_1(x) &= [x - h(x_{1,0}||y_{1,0})] + sk_1 \pmod p, \text{ where } s_0 \text{ is given} \\
&\quad \text{by CA} \\
SC_2 : f_2(x) &= [x - h(x_{2,1}||y_{2,1})] + sk_2 \pmod p, \\
SC_3 : f_3(x) &= [x - h(x_{3,1}||y_{3,1})] + sk_3 \pmod p, \\
SC_4 : f_4(x) &= [x - h(x_{4,1}||y_{4,1})][x - h(x_{4,2}||y_{4,2})] + sk_4 \pmod p, \\
SC_5 : f_5(x) &= [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})][x - h(x_{5,3}||y_{5,3})] \\
&\quad + sk_5 \pmod p, \\
SC_6 : f_6(x) &= [x - h(x_{6,1}||y_{6,1})][x - h(x_{6,3}||y_{6,3})] + sk_6 \pmod p
\end{aligned}
$$

Chung et al.'s Scheme (Continued...)

Key Derivation Phase (Continued...)

To derive the secret key sk_5 of SC_5 by its predecessor class SC_2, SC_2 needs to do following:

- Computes $s_2 G_5 = (x_{5,2}, y_{5,2})$ and then $h(x_{5,2}||y_{5,2})$.
- Determines sk_5 using $h(x_{5,2}||y_{5,2})$ from the public polynomial $f_5(x) = [x - h(x_{5,1}||y_{5,1})][x - h(x_{5,2}||y_{5,2})][x - h(x_{5,3}||y_{5,3})] + sk_5 \pmod p$ as $sk_5 = f_5(h(x_{5,2}||y_{5,2})) \pmod p$.