CSCI361
Computer Security

Secret sharing and its applications

Outline

- Motivation
- Secret sharing: model
  - Threshold schemes.
  - General schemes.
- Verifiable secret scheme
- Application

Motivation

- Principle of **reduced trust**:
  - to keep a secret safe and also make the system more robust, it is best if less power is given to a single entity
    - A secret key used to encrypt a file system should not be entrusted to one person
      - What if he looses the secret?
      - He leak the secret

- Distributing trust gives a solution to both of the above problems.
  - Key recovery system
  - Dishonest user

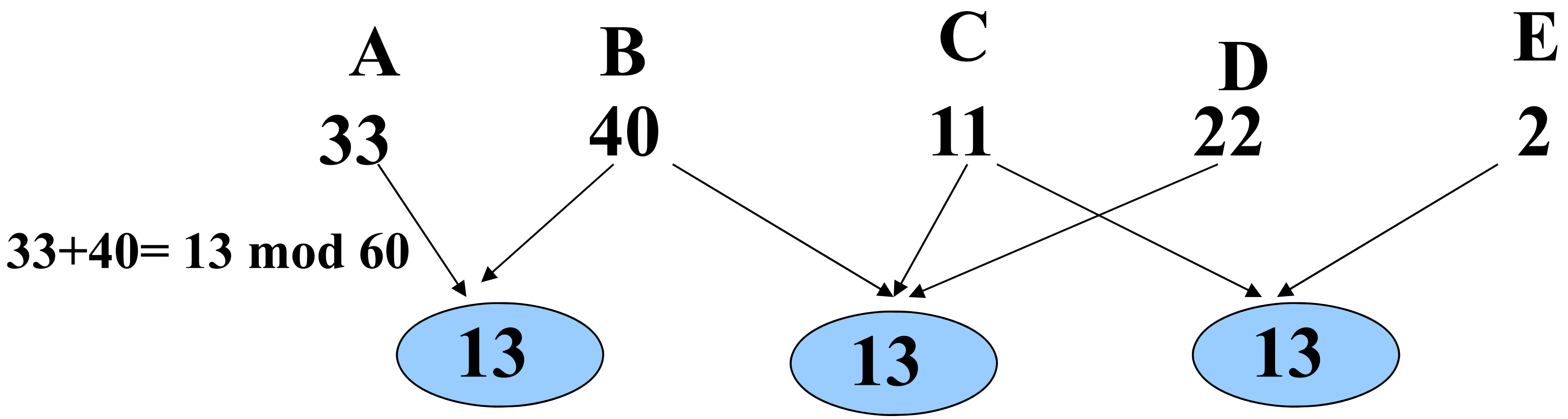
Key Escrow /Key Backup

To provide key backup:

- Divide the secret key into pieces
- Distribute the pieces to different servers such that certain subgroups of servers can recover the key

$A \wedge B$, $B \wedge C \wedge D$, $C \wedge E$ can recover the secret

- Consider RSA system.
- $N= 7x11=77$, $\varphi(N)=6\times10=60$
- $d= 13$, $e = d^{-1} = 37 \bmod 60$



Key escrow can be (mis)used :

- In 1991 the U.S. government attempted to introduce a new standard which would enable the government to read all private communications
  - Private key is broken into two halves:
  - The government keeps one half
  - Another authority the other half
  - A court order allows an agency to access both halves
- This standard was not successful.

A numerical example

- Consider a six digit combination lock.
  - The combination can be shared among 4 people.
  - Any three can calculate the combination.
  - No two people can calculate the combination.

Person	c_1	c_2	c_3	c_4	c_5	c_6
One	1	1	1	0	0	0
Two	0	0	1	1	1	0
Three	1	0	0	0	1	1
Four	0	1	0	1	0	1

Each c_i appears twice. As long so no more than one person is missing, somebody present knows c_i.

This is a *threshold secret sharing scheme*.

Shamir's Secret Sharing (1979)

- A threshold scheme using polynomial interpolation.

- An honest dealer *D* distributes a secret s among n users, such that at least t users must collaborate to find the secret
  - less than t players cannot have **any** information about the secret

The scheme

- We want to share a secret *s* among users $U_1, U_2..U_n$, such that any *t* users can reconstruct the secret.

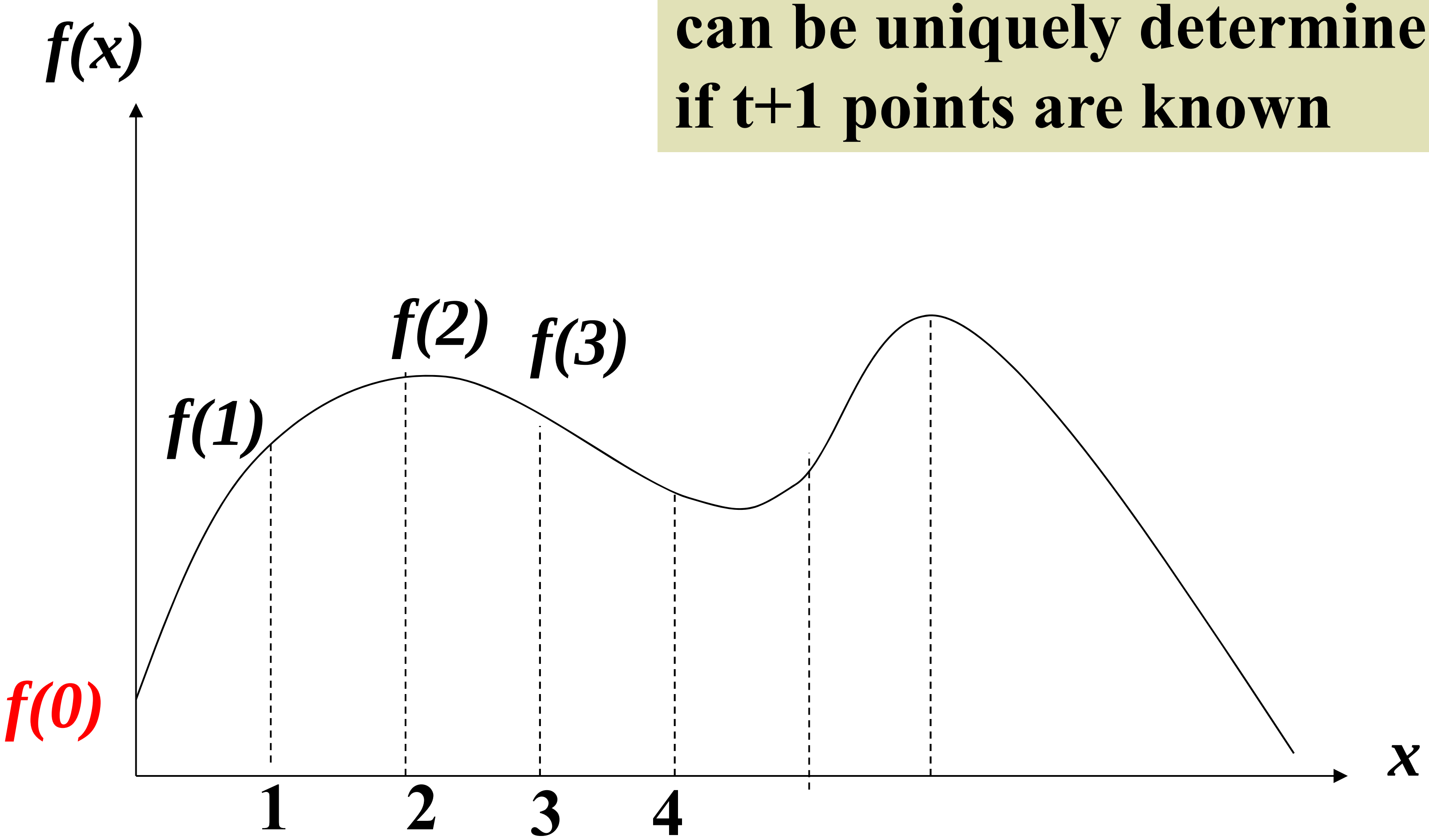
- Dealer *D* constructs a random polynomial *f*(*x*) of degree *t*-1 such that $a_0 = s$.

  $f(x) = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1}$
- This polynomial is constructed over numbers modulo a prime p, p is public.
- For user U_j, Dealer does the following
  – *Choose* x_j
  – *Calculate* $f(x_j)$
  – *Such that all* x_i $i=1,\ldots n$, *are distinct*
  – *User* U_j gets $(x_j, f(x_j))$
- (U_j, x_j) is public
  – Without losing generality, we can assume $x_j = j$

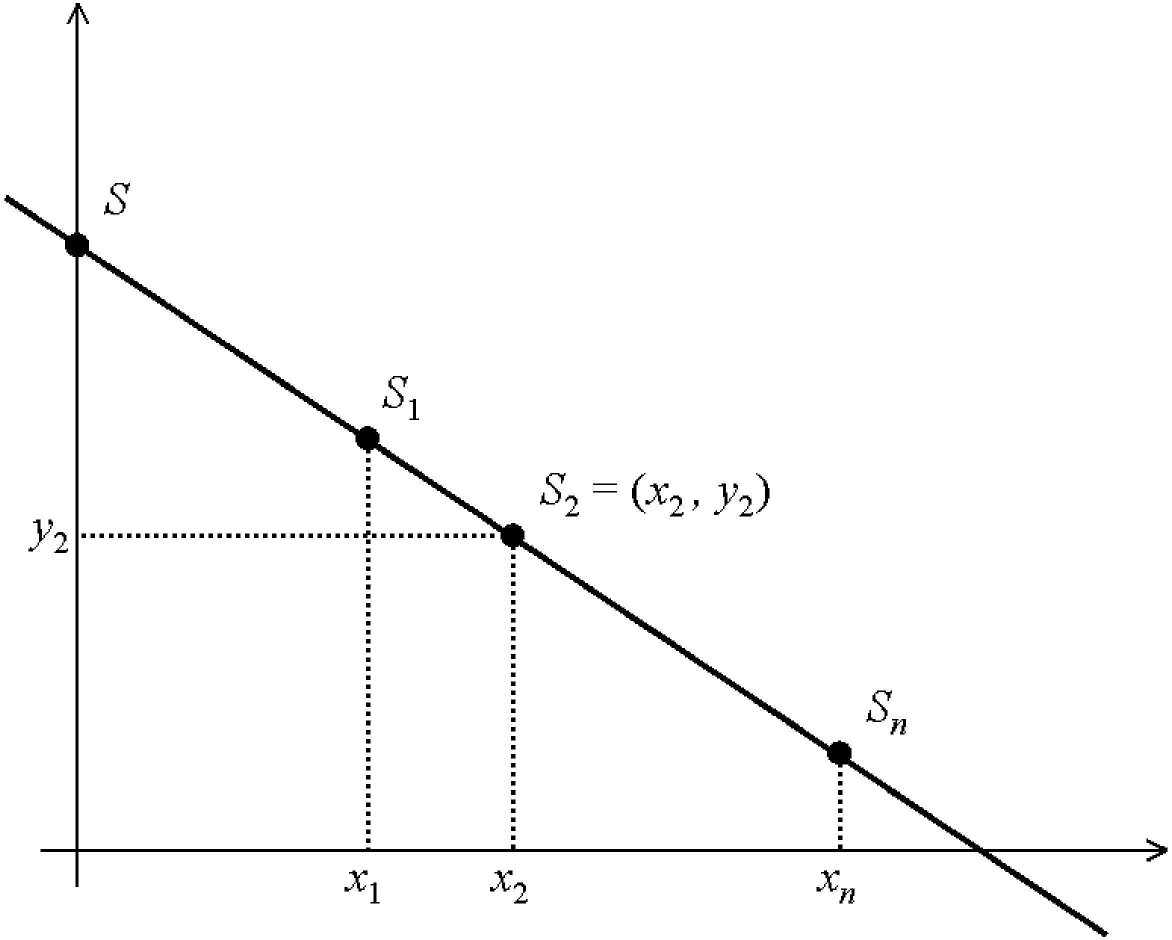
The Reconstruction Protocol

- Find the unique polynomial $f(x)$ such that $f(x)= f(j)$ and for $j=1,2,..t$
- Reconstruct the secret to be $f(0)$.

A polynomial of degree t can be uniquely determined if t+1 points are known



$$t=2,\ \ f(x)=a+bx$$



Lagrange interpolation

- Suppose you have *n* pairs (x_i,y_i=$f(x_i)$) and want to find the polynomial *f*.
- The polynomial of degree n-1 through the data is given by Lagrange interpolation.

$$f(x) = \sum_{j=1}^{n} f_j(x) \qquad f_j(x) = y_j \prod_{k=1,k\neq j}^{n} \frac{(x - x_k)}{(x_j - x_k)}$$

Consider a (3,6)-SSS over Z_7.

1. Let $x_i=i$, $i=1\ldots6$.
2. The secret is 3.
3. $f(x)=3+3x+3x^2$.
4. Share table

Share	s_1	s_2	s_3	s_4	s_5	s_6
Value	2	..	4	..	..	3

5. Assume P_1, P_3 and P_6 cooperate, each giving an equation.

$$2=k+a_1+a_2$$
$$4=k+3a_1+2a_2$$
$$3=k+6a_1+a_2$$

Finding k with Lagrange interpolation

The data: Y_1=f(1)=2, y_3=f(3)=4, y_6=f(6)=3.

$$
\begin{aligned}
f(0) &= \frac{(-x_3)(-x_6)}{(x_1-x_3)(x_1-x_6)}y_1 \\
&\quad + \frac{(-x_1)(-x_6)}{(x_3-x_1)(x_3-x_6)}y_3 \\
&\quad + \frac{(-x_1)(-x_3)}{(x_6-x_1)(x_6-x_3)}y_6 \\
&= 2\times\frac{(-3)(-6)}{(1-3)(1-6)} + 4\times\frac{(-1)(-6)}{(3-1)(3-6)} + 3\times\frac{(-1)(-3)}{(6-1)(6-3)} \\
&= 3
\end{aligned}
$$

Properties of Shamir's SS

- **Perfect Security**
  - t users can find a unique secret ,
  - t-1 users cannot learn anything
- **Ideal**
  - Each share is exactly the same size as the secret.
-
- **Extendable**
  - More shares can be created
    - New users joining the system
- **Flexible**
  - can support different levels of trust
    - Given more share to more trusted people

Homomorphic property

- $f(1), f(2) .. f(n)$ are shares of polynomial $f(x)$
- $g(1), g(2) .. g(n)$ are shares of polynomial $g(x)$

- Then $f(1)+ g(1), f(2)+g(2) .... f(n) + g(n)$ are shares of $f(x)+ g(x)$
  - That is the secret f(0)+g(0)

➔ *to multiply a secret by a constant, each share holder has to multiply by the same constant*

Example

- Sharing s=5 among 7 people such that any three can find the secret
- $f(x)= 5+2x+3x^2 \ \ mod\ 11$

$f(1)=10, f(2)= 10, f(3)=5, f(4)=6, f(5)=2, f(6)=9, f(7)=1$

- Sharing s=7 among the same people
- $g(x)= 7+x+x^2 \ \ mod\ 11$

$g(1)=9, g(2)= 2, g(3)=8, g(4)=5, g(5)=4, g(6)=5, g(7)=8$

- Shares of s=1 for the same people
- $1\ (=5+7\ mod\ 11)$
- $u(x) = f(x)+g(x)= 1+3x+4x^2\ mod\ 11$

$u(1)=8, u(2)= 1$ ..

Verifiable secret sharing

- Dealer is not trusted
- Dealer needs to ‘prove’ that the shares are consistent shares
  - Every t-1 subset gives the same secret

- A verifiable secret sharing system allows users to check validity of their shares
- Two versions
  - Interactive proofs
    - Requires interaction between dealer and participants
      - costly
  - non Interactive proofs
    - dealer can send messages,
    - the shareholders cannot talk with each other or with the dealer (for share verification).
    - The can use public information to check validity of shares

Threshold signature

Threshold RSA

- Public key (e,N), secret key (d,N)
- Share secret key among users:
  - $d_1,d_2, \ldots d_n$ using an extension of Shamir's scheme

- For a message m that t users agree on, each user produces a partial signature
  $H(m)^{d1}, H(m)^{d2} \ldots H(m)^{dt}$

- Combiner combines these partial signatures (e.g. multiply them) to obtain
  $H(m)^d = H(m)^{d1} \times H(m)^{d2} \times \ldots H(m)^{dt}$
- The signed message is $(m, H(m)^d)$

- Verification is as usual
- Given (m,s), we check $H(m) = s^e \bmod N$