Secret sharing scheme and privacy homomorphism

Secret Sharing : Motivation

- Suppose you and your friend accidentally discovered a map that you believe would lead you to an island full of treasure.
- You and your friend are very excited and would like to go home and get ready for the exciting journey to the great fortune.
- But the problem is,
  - Who is going to keep the map?

Secret Sharing : Motivation…

- As they don't trust each other
- Need a scheme that could make sure that the map is shared in a way so that no one would be left out in this trip.
- What would you suggest?

Secret Sharing : Motivation…

- To split the map into two pieces and make sure that both pieces are needed in order to find the island.
- You can happily go home and be assured that your friend has to go with you in order to find the island.
- This illustrates the basic concept of *secret sharing*.

Generalization

- Desired Properties:
  - All n parties can get together and recover secret s.
  - Less than n parties cannot recover s.
- To achieve such a sharing,
  - Split the secret into n pieces $s_1, s_2, \ldots, s_n$ and give one piece to each party.
- Each piece here is called a *share.*
- Called *secret splitting* in some literature.

5

Generalization…

- Every piece of information is stored as a bit string or a number on a computer.
  - need to share a bit string or a number
- For example, assume that your salary is stored as a number 12345678.
  - Now you want to split your salary into two shares for two parties.
  - A naïve approach
    - To split the salary into two parts…
  - Is the scheme satisfies the two properties ????

6

Attacks

- However, there is a problem !!!
  - Suppose I am the first party who gets the most significant 4 digits of your salary.
  - It is true that I don't know exactly how much your salary is, but I have a pretty good idea about the range of your salary (>= 12340000), because I have the 4 most significant digits.
- This is called Partial Information Disclosure
- What about Brute Force Attack???
  - Consider the launch of a nuclear missile where the password is shared between two generals

7

Attacks…

A naive way of splitting a secret could cause partial information disclosure, which might be undesirable in certain cases and fatal in others.

8

Partial Information Disclosure: solution

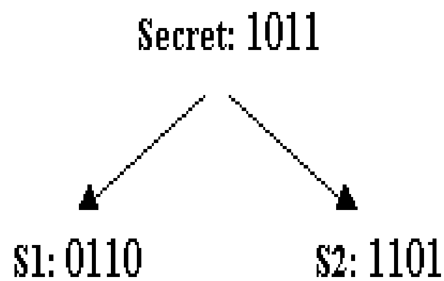
- We would like to solve the partial information disclosure problem:
  - Strengthen property 2
  - Seems counter-intuitive !!!
- But we have a solution to this…..

Partial Information Disclosure: Solution…

- Suppose two parties are going to share a secret bit string 1011. The two shares are generated as follows:
  - To generate the first bit of the two shares,
    - flip a coin
    - If the result is head, then set the first bit of the first share to 0;
    - Else set the first bit of the first share to 1
  - To generate the first bit of the second share.
  - If the result of the previous coin flipping was a head, then copy the first bit of the secret.
  - Else flip the first bit of the secret and use that.
  - Repeat this random process for each bit of the secret.

Partial Information Disclosure:Solution…

- Suppose for our example where the secret bit string is 1011,
- We flip the coin 4 times and get the sequence head, tail, tail, and head.
- Now think of the two properties !!!



Modifying Disclosure Conditions

- Now we have this nice secret splitting scheme.
- But such a secret splitting scheme may not suffice in certain cases !!!!
- Recall again the control system of a nuclear missile launch
  - There are three generals who are in charge of a missile launch.
  - A simple solution would be to give the secret code to these three generals,
  - But then it is possible for a compromised general to start a war and destroy the planet.
  - We need some sort of secret sharing here.
    - Generate 3 shares from the secret code and give one share to each general.

Modifying Disclosure Conditions…

- Now think about the attacks !!!
  - Partial Information Disclosure??
  - Brute Force Attack???
  - Attack on Availability ????
- What can happen if one general is a spy from a hostile country?
  - We're not worried about him launching the missile by himself.
  - But he can disable the missile launch capability by throwing away his share !!!!

Modifying Disclosure Conditions…

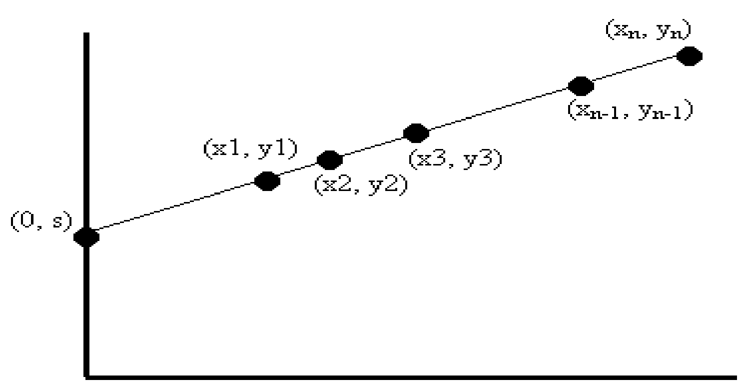
- The problem is really the availability of the secret code.
- An essential issue in this example because,
  - the capability to launch a missile depends on the availability of the secret code.
- Assuming that it is unlikely that more than 1 general could be compromised or unavailable
  - Now postulate the policy of your secret sharing scheme ????

(k,n) Secret Sharing

- To generalize the properties, we get *(k,n) secret sharing*.
- Given a secret s, to be shared among n parties, that sharing should satisfy the following properties:
  - **Availability**: greater than or equal to k parties can recover s.
  - **Confidentiality**: less than k parties have no information about s.
- Can we consider secret splitting as a special case of secret sharing ????

15

- Let's start with the design of an (2,n) scheme.
- Let's say we want to share a secret s among n parties. We use some basic geometry



16

(k,n) Secret Sharing…

- Each point that is picked represents a share.
- We claim that these n shares constitute an (2,n) sharing of s.
- Now think about availability and confidentiality properties ????

(k,n) Secret Sharing …

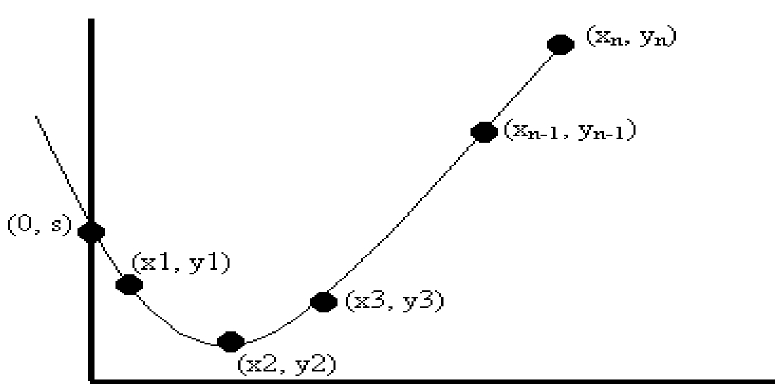
- To show availability, we need to prove that two parties can recover the secret.
- Two parties have two shares; that is two points.
- Given these two points, how can we recover the secret?
  - We know that two points determine a line, so we can figure out the line that goes through both points.
  - Once we know the line, we know the intersection of the line with the y axis.
  - Then, we get the secret.
  - So, it only takes us two points (shares) to make the secret available.

(k,n) Secret Sharing …

- What about confidentiality? We need to show that one share does not disclose any information about the secret.
- There are infinite possible lines that go through this point, and these lines intersect with the y-axis at different points, all of which yield different "secrets".
- In fact, given any possible secret, we can draw a line that goes through the secret and the given share.
- This means that with one point, no information about the secret is exposed.

19

- Using the same idea, can we design an (3,n) secret sharing scheme?
- Note that the key point in the (2,n) scheme is that a line is determined by two points, but not by 1.
- Now we need a curve that is determined by three points, but not 2.



20

(k,n) Secret Sharing …

- To generalize the scheme even further, we have a construction of an (k, n) secret sharing scheme. Now we use the curve that corresponds to a (k-1) degree polynomial
- We randomly select a curve corresponding to such a polynomial that goes through the secret on the y-axis.
- Then we select n points on the curve.
- Using the same arguments, we can show that this scheme satisfies both availability and confidentiality properties.

SHAMIR'S SECRET SHARING SCHEME

Mathematical Definition

- Goal is to divide some data *D* (e.g., the safe combination) into n pieces $D_1, D_2 \ldots D_n$ in such a way that:
  - Knowledge of any k or more D pieces makes D easily computable.
  - Knowledge of any k -1 or fewer pieces leaves D completely undetermined (in the sense that all its possible values are equally likely).
- This scheme is called (k,n) threshold scheme. If k=n then all participants are required together to reconstruct the secret.

Shamir's Secret Sharing

- To design (k,n) threshold scheme to share our secret S where k < n.
- Choose at random (k-1) coefficients $a_1, a_2, a_3 \ldots a_{k-1}$, and let S be the a_0

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots\ldots + a_{k-1} x^{k-1}$$

- Substituting a_0 by S

$$f(x) = S + a_1 x + a_2 x^2 + \ldots\ldots + a_{k-1} x^{k-1}$$

Shamir's Secret Sharing …

- Construct n points (i,f(i)) where i=1,2…..n
- Given any subset of k of these pairs, we can find the coefficients of the polynomial by interpolation, and then evaluate a_0=S , which is the secret.

- Let S=1234
- n=6 and k=3 and obtain random integers

  a_1=166 and a_2=94

$$f(x) = 1234 + 166x + 94x^2$$

- Secret share points

  (1,1494),(2,1942)(3,2598)(4,3402)(5,4414)(6,5614)

- We give each participant a different single point (both x and f(x)).

Reconstruction

- In order to reconstruct the secret any 3 points will be enough
- Let us consider

$$(x_0, y_0) = (2,1924), (x_1, y_1) = (4,3402), (x_2, y_2) = (5,4414)$$
$$l_0 = x - x_1 / x_0 - x_1 * x - x_2 / x_0 - x_2 = x - 4/2 - 4 * x - 5/2 - 5 = 1/6x^2 - 11/2x + 31/3$$
$$l_1 = x - x_0 / x_1 - x_0 * x - x_2 / x_1 - x_2 = x - 2/4 - 2 * x - 5/4 - 5 = -1/2x^2 - 31/2x - 5$$
$$l_2 = x - x_0 / x_2 - x_0 * x - x_1 / x_2 - x_1 = x - 2/5 - 2 * x - 4/5 - 4 = 1/3x^2 - 2x + 22/3$$

$$f(x) = \sum_{j=0}^{2} y_j l_j(x) = 1942(1/6x^2 - 11/2x + 31/3) + 3402(-1/2x^2 - 31/2x - 5) + 4414(1/3x^2 - 2x + 22/3)$$
$$f(x) = 1234 + 166x + 94x^2$$

Security discussion

- **Secrecy**
  - Secrecy: the adversary needs to corrupt at least k shareholders and collect their shares in order to learn the secret;
- **Availability**
  - For a given k, the secret Availability increases as n increases...
  - For a given n the secret's Secrecy increase as k increases.

Security discussion…

- Information theoretically secure
- Space Efficient: the size of each share does not exceed the size of the secret
- Keeping k fixed, shares can be easily added or removed, without affecting other shares
- It is easy to change the shares, keeping the same secret
- It is possible to provide more than one share per individual: hierarchy

Homomorphic property of secret sharing

- Similar to Encryption, secret sharing schemes have homomorphic properties
  - i.e. For operations on the secret, there are corresponding operations on shares that preserve the relation between the secret and shares
- Consider Shamir's scheme
  - Let s and t be two secrets with polynomials f and g respectively
  - Now consider the sum of the secret s+t
  - Since s+t = f(0) + g(0) = (f+g)(0)
  - What can you say about polynomial (f+g) ???
- Conversely, adding the shares $[s]_i$ and $[t]_i$ gives $[s]_i + [t]_i = f(i)+g(i) = (f+g)(i)$

Homomorphic property of secret sharing …

- Now think about multiplicative homomorphism using Shamir's Secret Sharing scheme ?????