CS 161 Computer Security
Spring 2008 Dawn Song — Notes 7

In a secret sharing scheme there is a trusted authority TA and n users $U_1, \ldots, U_n$. The TA has a secret value K called the secret or key. The TA uses a share generation algorithm to split K into n shares $s_1, \ldots, s_n$. Each share s_i is then transmitted to user U_i by a secure channel. The secret sharing protocol guarantees that two properties hold:

- A reconstruction algorithm can be used to efficiently reconstruct the secret K from any t of the n shares.
- Any $t-1$ of the n shares reveal *no* information about the secret K.

Such a scheme is called an (n,t) threshold scheme.

For example, if the secret K is an integer between 0 and $M-1$, then an (n,n) threshold scheme can be obtained by selecting $s_1, \ldots, s_{n-1}$ uniformly at randomly between 0 and $M-1$, and setting $s_n = K - \sum_{i=i}^{n-1} s_i \bmod M$. Now,

- $K = \sum_{i=1}^{n} s_i \bmod M$.
- Given all shares except s_j, K can take on any value modulo M.

To understand how to implement a general (n,t) threshold scheme we need to understand some properties of polynomials modulo a prime p. While working modulo a prime p, we can add, subtract and multiply numbers, as well as divide numbers as long as we are not dividing by 0. So we can consider polynomials whose coefficients are elements modulo p. For example $f(x) = x^2 + 2x + 4 \bmod 5$. It turns out that such polynomials have many of the same properties as polynomials with real coefficients:

Polynomials

- A polynomial of degree n over a field F has at most n roots (this can be proved by induction).
- A polynomial P of degree n is uniquely determined by any $n+1$ distinct pairs (x_i, y_i) such that $P(x_i) = y_i$ (this follows immediately from the previous property).

Suppose that we are given the value of a polynomial $P(x)$ of degree n at $n+1$ points: $P(a_i) = b_i$ for $i = 1 \ to \ n+1$. How do we reconstruct the unique polynomial $P(x)$ of degree n satisfying these $n+1$ constraints?

Consider the following polynomials of degree n:

For $i = 1, 2, \ldots, n+1$, define

$$\Delta_i(x) = \left(\prod_{j \neq i}(a_i - a_j)\right)^{-1} \prod_{j \neq i}(x - a_j).$$

Notice that $\Delta_i(a_i) = 1$ and for $1 \leq j \leq n+1$, $j \neq i$ $\Delta(a_j) = 0$. It follows that the desired polynomial $P(x) = \sum_{i=1}^{n+1} b_i \Delta_i(x)$.

The process we have just gone through—explicitly constructing a polynomial that passes through a number of given points—is called *Lagrange interpolation*.

If $n = 3$, and $a_i = i$, for instance, then

$$\Delta_1(x) = ((1-2)(1-3))^{-1}(x-2)(x-3) = 2^{-1}(x-1)(x-2)$$
$$\Delta_2(x) = ((2-1)(2-3))^{-1}(x-1)(x-3) = (-1)^{-1}(x-1)(x-3)$$
$$\Delta_3(x) = ((3-1)(3-2))^{-1}(x-1)(x-2) = 2^{-1}(x-1)(x-2).$$

Secret Sharing

Suppose the U.S. government finally decides that a nuclear strike can be initiated only if at least $t > 1$ major officials agree to it (what a "major official" is doesn't really matter to us). We want to devise a scheme such that (1) any group of t of these officials can pool their information to figure out the launch code and initiate the strike but (2) no group of $t-1$ or fewer can conspire to find the code. How can we accomplish this?

Suppose that there are n officials and that launch code is some natural number K. Let p be a prime number larger than n and s —we will work with numbers modulo p from now on.

Now pick a random polynomial f of degree $t-1$ such that $f(0) = K$. The share $s_i = f(i)$ for $i = 1$ *to* n.

- Any t officials, having the values of the polynomial at t points, can use Lagrange interpolation to reconstruct the polynomial f, and once they know f, they can compute $f(0) = K$ to learn the secret.
- Any group of $t-1$ officials has no information about K. All they know is that there is a polynomial of degree $t-1$ passing through their $t-1$ points such that $f(0) = K$. However, for each possible value $f(0) = b$, there is a unique polynomial that is consistent with the information of the $t-1$ officials, and satisfies the constraint that $f(0) = b$.