CS70, Fall 2003 **Discussion #7** **Amir Kamil**
UC Berkeley **10/17/03**

Topics: Fingerprinting, Secret Sharing

1 Fingerprinting

Consider the fingerprinting function $F_p(x) = x \mod p$, where p is some large prime. Suppose Alice and Bob share a large prime p, and Alice wants to send an important but public message m to Bob. In order to make sure that he receives the correct, she fingerprints the message as follows. She breaks it into $n = \sqrt[3]{p}$ bit pieces, then sends each piece along with its fingerprint. When Bob receives the pieces, he computes the fingerprint and checks against the received fingerprint. If a mismatch occurs, he asks Alice to resend the piece and its fingerprint.

Now suppose an adversary intercepts the communication, but doesn't know p. He wants to be able to change both the pieces of the message and their fingerprints so that Bob receives an incorrect message but believes he received the right one. Can the adversary do this with success guaranteed? For example, adding 1 to both a piece and its fingerprint works most of the time, but not all the time. Is there a scheme that always works?

Since the adversary is listening in on the transmission, he learns pairs $a_i, F_p(a_i)$, where the a_i are the n bit pieces. By the definition of the fingerprinting function, $a_i \equiv F_p(a_i) \pmod{p}$, so p divides $x_i = a_i - F_p(a_i)$. However, x_i may be a large multiple of p, and so may be hard to factor. But note that $\gcd(x_i, x_j)$ is also a multiple of p, and is very likely to be a smaller multiple of p than either x_i or x_j (it is quite unlikely that one is a multiple of the other). And then $\gcd(x_i, x_j, x_k)$ is an even smaller multiple of p. So by listening to y pieces of the message, the adversary can compute $\gcd(x_1, x_2, \cdots, x_y)$, which we expect to either be p or a small, easily factorable multiple of p.

Thus the adversary can learn p, allowing him to change the message as much as he wants. As a result, this fingerprinting scheme is insecure. This is why the RSA fingerprinting scheme is a better alternative.

2 Secret Sharing

Suppose I want to encode a secret m using the secret sharing protocol discussed in class: I'll pick a prime $p > m$ and a degree k polynomial $f(x)$ such that $f(0) \equiv m \pmod{p}$. Suppose you know k points on this polynomial (i.e. you know the value of $f(x) \mod p$ for k different x). Do you know any information about my secret?

Let's do a concrete example. Suppose I pick $p = 11$, and a degree 2 polynomial. I tell you that $f(6) \equiv 7 \pmod{11}$ and $f(7) \equiv 5 \pmod{11}$. Does this tell you anything about my secret?

In order to answer this question, first let's decide what information you knew before you learned the value of $f(6)$ and $f(7)$, and what you know now. The only thing you knew before was that my secret is a value between 0 and 10. What you know now depends on how many degree 2 polynomial satisfy the given values for $f(6)$ and $f(7)$. In fact, the following degree 2 polynomials all work:

$$f(x) = 4x^2 + x + 0$$

$$f(x) = 9x^2 + 2x + 1$$

$$f(x) = 3x^2 + 3x + 2$$

$$f(x) = 8x^2 + 4x + 3$$

$$f(x) = 2x^2 + 5x + 4$$

$$f(x) = 7x^2 + 6x + 5$$

$$f(x) = 1x^2 + 7x + 6$$

$$f(x) = 6x^2 + 8x + 7$$

$$f(x) = 0x^2 + 9x + 8$$

$$f(x) = 5x^2 + 10x + 9$$

$$f(x) = 10x^2 + 0x + 10.$$

These polynomials cover all possible values of the secret! And each value has exactly one corresponding polynomial. Thus you still have no idea what value my secret is, besides that it is between 0 and 10.

Now what if I also told you that $f(8) \equiv 7 \pmod{11}$? Now there is only a single degree 2 polynomial that satisfies all three points, namely

$$f(x) = 2x^2 + 5x + 4.$$

Thus my secret is the value 4.

You may be wondering how I came up with the above polynomials. Well, for the first set, I used an inefficient program to compute them. For the actual correct polynomial, I used the polynomial interpolation algorithm you proved in your homework. Recall that the polynomial $f_k(x) = b_k F_k(x)$, where $F_k(x) = (x-0)(x-1)\cdots(x-k+1)(x-k-1)\cdots(x-10)$ and $b_k = F_k(k)^{-1} \pmod{11}$, has the value 0 for $f(x \neq k)$ mod 11 and 1 for $f(k) \mod 11$.

But this isn't quite what we want. If we interpolated using these polynomials, we'd end up with a polynomial such that $f(x) \equiv 0 \pmod{11}$ for $x \notin \{6,7,8\}$. We need polynomials $f_k(x)$ for $k \in \{6,7,8\}$ such that $f_k(k) \equiv 1 \pmod{11}$ and $f_k(j) \equiv 0 \pmod{11}$ for $j \in \{6,7,8\} - \{k\}$, but $f_k(j)$ is unforced for $j \notin \{6,7,8\}$. So the polynomials we are looking for are

$$f_6(x) = b_6 F_6(x), \quad F_6(x) = (x-7)(x-8), \quad b_6 = F_6(6)^{-1} \pmod{11}$$

$$f_7(x) = b_7 F_7(x), \quad F_7(x) = (x-6)(x-8), \quad b_7 = F_7(7)^{-1} \pmod{11}$$

$$f_8(x) = b_8 F_8(x), \quad F_8(x) = (x-6)(x-7), \quad b_8 = F_8(8)^{-1} \pmod{11}.$$

Then our desired polynomial is $f(x) = f(6)f_6(x) + f(7)f_7(x) + f(8)f_8(x)$. This polynomial has the correct values at $x = 6$, $x = 7$, and $x = 8$.

Using the above method, we recover the polynomial $f(x) = 2x^2 + 5x + 4$. Thus the secret is 4.

Now suppose I did not tell you the degree of the polynomial that encodes my secret. How many points would be required for you to figure out the secret?

In this case, it is impossible for you to figure out the secret. Suppose I pick a degree 100 polynomial, and carefully choose 10 points to give you such that there is a degree 3 polynomial that satisfies all 10 points. You can try all possible degrees from 1 to 9, and even if that degree 3 polynomial is the only polynomial in this range that satisfies the 10 points, you can't be sure that the polynomial is correct. In fact, in this case it is not, so you would recover the wrong secret.