CS 161: Computer Security

Lecture 3

September 9, 2014

Where we are

- How did NSA break SSL?
- Basic number theory
- RSA
- Digital certificates
- Shamir secret sharing
- Rabin signatures
- Secure hashing
- Elliptic curve cryptography
- Pseudo-random number generation
- SSL protocol

Review: Homomorphism

- Homomorphism is a mathematical property
  - Preserves operation under a function
  - Example: let $f(x) = x \bmod n$
  - The f is homomorphic under addition & multiplication
  - $f(x+y) = f(x) + f(y) \pmod{n}$
  - $f(xy) = f(x)f(y) \pmod{n}$

Review: RSA is homomorphic

- RSA is homomorphic under multiplication
- $E(m) \leftarrow m^e \pmod{n}$
- Then $E(m)E(m')\bmod n = E(mm' \bmod n)$
- This is actually a huge problem for RSA
- Has potential to allow forged messages or signatures
- To solve this, we usually add padding

Review: Shamir secret sharing

$f(x) = \ a_{q-1}x^{q-1} + \cdots + a_1 x + a_0 \pmod{m}$

Shares: $f(1), f(2), \ldots, f(n)$

q points →we can solve for $a_{q-1}, \ldots, a_1, a_0$

$f(0) = \ a_0$ =secret

Review: Shamir is (sort-of) homomorphic

- We can add together secret shares

$$f(x) = a_{q-1}x^{q-1} + \cdots + a_1 x + a_0 \pmod{m}$$
$$g(x) = b_{q-1}x^{q-1} + \cdots + b_1 x + b_0 \pmod{m}$$
$$h(x) = c_{q-1}x^{q-1} + \cdots + c_1 x + c_0 \pmod{m}$$

We can define

$$SUM(x) = (a_{q-1}+b_{q-1} + c_{q-1})x^{q-1} +$$
$$\ldots + (a_1+b_1 + c_1)x + (a_0 + b_0 + c_0) \pmod{m}$$

$SUM(0) = a_0 + b_0 + c_0 \pmod{m}$ (sum of secrets)

Review: Homomorphic (secret) addition

- Want to add secret values $a_0 + b_0 + c_0$

- Make three sets of secret shares

$$f(\quad), g(\quad), h(\quad)$$

- Give agent i: $f(i), g(i), h(i)$
- Agent i computes:

$$SUM(i) = f(i) + g(i) + h(i)$$

- Recover $SUM(0)$

Review: Chinese remainder theorem (CRT)

- Radically different way of representing integers modulo n

- If $n = n_1 n_2 \ldots n_k$ and all n_i are relatively prime
- We can represent $x \bmod n$ two different ways

$$x \bmod n$$
$$\langle x \bmod n_1, x \bmod n_2, \ldots, x \bmod n_k \rangle$$

Review: CRT is homomorphic (addition)

$$(x+y) \bmod n =$$
$$\langle x \bmod n_1, x \bmod n_2, \ldots, x \bmod n_k \rangle$$
$$+$$
$$\langle y \bmod n_1, y \bmod n_2, \ldots, y \bmod n_k \rangle$$
$$=$$
$$\langle (x+y) \bmod n_1, (x+y) \bmod n_2, \ldots, (x+y) \bmod n_k \rangle$$

Review: CRT is homomorphic (multiplication)

$$(xy) \bmod n =$$
$$\langle x \bmod n_1, x \bmod n_2, \ldots, x \bmod n_k \rangle$$
$$*$$
$$\langle y \bmod n_1, y \bmod n_2, \ldots, y \bmod n_k \rangle$$
$$=$$
$$\langle (xy) \bmod n_1, (xy) \bmod n_2, \ldots, (xy) \bmod n_k \rangle$$

Review: Squares modulo pq

- Let p, q be odd primes
- Some integers mod pq are squares (*quadratic residues*) and some are not

$1^2 = 1 \pmod{15}$; $2^2 = 4 \pmod{15}$; $4^2 = 1 \pmod{15}$;
$7^2 = 4 \pmod{15}$; $8^2 = 4 \pmod{15}$; $11^2 = 1 \pmod{15}$;
$13^2 = 4 \pmod{15}$; $14^2 = 1 \pmod{15}$

  - $\sqrt{1} = \{1, -1, 4, -4\} = \{1, 4, 11, 14\} \pmod{15}$
  - $\sqrt{4} = \{2, -2, 7, -7\} = \{2, 7, 8, 13\} \pmod{15}$

Review: Square-rooting → factoring

$$\sqrt{x^2} \pmod{pq}$$

$$\langle x \bmod p, x \bmod q \rangle$$
$$\langle x \bmod p, -x \bmod q \rangle$$
$$\langle -x \bmod p, x \bmod q \rangle$$
$$\langle -x \bmod p, -x \bmod q \rangle$$

If we have two random square roots x_1 & x_2

then sometimes $\gcd(x_1 + x_2, pq) = p$ or q

Review: Square-rooting → factoring

If we have two random square roots x_1 & x_2
then sometimes $\gcd(x_1 + x_2, pq) = p$ or q

$$\langle x \bmod p, x \bmod q\rangle + \langle x \bmod p, -x \bmod q\rangle =$$
$$\langle (x+x) \bmod p, (x-x) \bmod q\rangle =$$
$$\langle 2x \bmod p, 0 \bmod q\rangle$$

which is a multiple of q

This lecture

- Rabin signatures
- Cryptographic hashing

Rabin signatures

- To compute a Rabin signature
    - Adjust message so that it is a square
- Compute square root modulo pq
- Anyone can verify signature (just square)
- But if we can take square roots, we can factor

Rabin signatures

- Pick a random r, compute $r^2 \bmod n$.
- We will have four square roots

$$\begin{aligned} r &= \langle r \bmod p, r \bmod q \rangle \\ s &= \langle r \bmod p, -r \bmod q \rangle \\ -s &= \langle -r \bmod p, r \bmod q \rangle \\ -r &= \langle -r \bmod p, -r \bmod q \rangle \end{aligned}$$

- If we have a square-root taking machine, with 50% probability we will get s or $-s$.
- So, with 50% probability

$$\gcd\left(r + \sqrt{r^2}, n\right) = p \text{ or } q$$

Fermat's Little Theorem

- Fermat-Euler theorem
- If m, n relatively prime, then $m^{\varphi(n)} \equiv 1 \pmod{n}$
- If p prime, then $m^{(p-1)} \equiv 1 \pmod{p}$
- Suppose a is a square $\bmod p$, then it has a square root b - in other words $b^2 \equiv a \pmod{p}$

$$a^{(p-1)/2} \equiv (b^2)^{(p-1)/2} \equiv b^{(p-1)} \equiv 1 \pmod{p}$$

How to compute square roots

- Remember $a^{(p-1)/2} \equiv 1 \pmod p$
- Need to compute square root of $(a \bmod p)$.
- Assume $p \equiv 3 \bmod 4$ or $p = 4k + 3$
- (Look in posted notes if $p \equiv 1 \bmod 4$)
- Let $x \equiv a^{\frac{p+1}{4}} \equiv a^{k+1} \pmod p$
- Now a is a square so

$$x^2 \equiv a^{2k+2} \equiv a^{2k+1}a \equiv a^{(p-1)/2}a \equiv 1 \cdot a \equiv a \pmod p$$

- So $x \equiv a^{\frac{p+1}{4}} \pmod p$ is square root

Computing square roots mod pq

- To compute square root of $a \bmod pq$,
  - Compute square root $a \bmod p$
  - Compute square root $a \bmod q$
  - Combine using CRT

Hash functions

- You remember hash functions from 61B
- Properties
  - Variable input size
  - Fixed output size (e.g., 512 bits)
  - Efficient to compute
  - Psuedo-random (mixes up input well!)
- In this lecture $H()$ denotes a hash function

Collisions

- Collision occurs when
- $x \neq y$ but $H(x) = H(y)$
- Since input size > output size, collisions happen

Birthday paradox

- Ignore leapdays
- Probability that two people are born on same day is 1/365
- How many people until probability of at least one common birthday > 50%
- Surprising answer 23 (!)

Probability of a collision

- Suppose hash value range is n
- And k input points are hashed

- Probability of a collision is

$$P(n,k) = 1 - \frac{n!}{(n-k)!\,n^k} \approx 1 - e^{-k^2/2n}$$

Cryptographic hash functions

- Cryptographic hash functions add conditions
- Preimage resistance
    - Given h, intractable to find y such that $H(y) = h$
- Second preimage resistance
    - Given x, intractable to find $y \neq x$ such that $H(y) = H(x)$
- Collision resistance
    - Intractable to find $(x, y), y \neq x$ such that $H(y) = H(x)$

We have a hash function crisis

- Popular hash function MD5
  - Thoroughly broken
- Government standard function SHA-1, SHA-2
  - Theoretical weaknesses
- “New” cryptographic hash function SHA-3
  - Too new to fully evaluate
  - Maybe good enough

Review: issues w/ RSA signatures

- How does verifier check true value for d, n ?
  - Digital certificates Solved!
- What about large documents $(m > n)$?
  - Cryptographic hashes
- What if we want to **both** encrypt & sign?
  - Use two sets $\langle e, d, n \rangle$ and $\langle e', d', n' \rangle$ (later)

To compute RSA signature

- For large documents m
- Compute $H(m)$
- Sign $H(m)$
- Transmit $\langle m, Sign(H(m))\rangle$

- This is used in digital certificates (used in SSL)

MS and Google on SHA-1

- Microsoft turning off SHA-1 support (in SSL certificates) on 1/1/2017

- Google is issuing warnings on SHA1 SSL in Chrome

Expires in 2016

Expires after 2016

- Big pushback from CA industry

Modular division

- You've been lied to about fractions
- You learned that ½ = 0.5
- This is not quite true

- What does ½ really mean?
- A value that when multiplied by 2 gives 1.
- When we talk about real numbers ½ = 0.5
- But when talk in other contexts ½ has different values.

Modular division

- What is $½ \bmod 5$?
- Solution to $2x = 1 \bmod 5$
- $x = 3$
- $½ = 3 \bmod 5$
- What is $¼ \bmod 5$?
- Solution to $4x = 1 \bmod 5$
- $x = 4$
- $¼ = 4 \bmod 5$

Modular division mod p

- How to calculate $x^{-1} \bmod p$ (p prime)?
- Method 1:
  - Use Extended GCD to solve
  - $ax + bp = 1$
  - $ax \equiv 1 \pmod{p}$ so $a \equiv x^{-1} \pmod{p}$
- Method 2:
  - Use Fermat-Euler theorem to solve
  - $x^{\varphi(p)} \equiv x^{p-1} \equiv 1 \pmod{p}$
  - $x^{(p-2)}x \equiv x^{p-1} \equiv 1 \pmod{p}$
    so $x^{(p-2)} \equiv x^{-1} \pmod{p}$

Modular division mod n

- How to calculate $x^{-1} \bmod n$ (n composite)?
  - Note x, n must be relatively prime
  - Dividing by a modular factor like dividing by zero

- Method 1:
  - Use Extended GCD to solve
  - $ax + bn = 1$
  - $ax \equiv 1 \pmod{n}$ so $a \equiv x^{-1} \pmod{n}$

Modular division

- How to calculate $x^{-1} \bmod n$ (n composite)?
  - Note x, n must be relatively prime
  - Dividing by a modular factor like dividing by zero

- Method 2:
  - Use Fermat-Euler theorem to solve
  - $x^{\varphi(n)} \equiv 1 \pmod{n}$
  - $x^{\varphi(n)-1}x \equiv x^{\varphi(n)} \equiv 1 \pmod{n}$
    so $x^{\varphi(n)-1} \equiv x^{-1} \pmod{n}$

Next lecture

- Discrete logarithm problem
- Diffie-Hellman key exchange
- Man-in-the-middle attacks
- Elgamal signatures
- Elliptic curve cryptography
- Pseudo-random number generation