CS 161: Computer Security

Lecture 2

September 3, 2015

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: RSA

- From last lecture
  - $n = pq$ (p, q large primes)
  - Suppose $ed \equiv 1 \pmod{(p-1)(q-1)}$
  - Then $m^{ed} \equiv m \pmod{n}$
  - Encryption key: e; Decryption key: d
  - Encryption: $E(m) \leftarrow m^e \pmod{n}$
  - Decryption: $D(c) \leftarrow c^d \pmod{n}$
- So $D\big(E(m)\big) \equiv m^{ed} \equiv m \pmod{n}$
- Beauty: we can disclose only one of e, d
- Asymmetric (public-key) cryptography

Review: RSA and factoring

- If we can factor large numbers, we can break RSA
- Why?
- Because we can solve $n = pq$ and then
$$ed \equiv 1 \pmod{(p-1)(q-1)}$$

- If we can break RSA, can we factor?
- Unknown!!!!

Review: RSA and factoring

Factoring algorithm → RSA cryptanalysis algorithm

but

RSA cryptanalysis algorithm → ???

Review: Digital signatures

- Remember $m^{ed} \equiv m \pmod{n}$
- Before we published e, n and kept d secret

- Suppose we publish d, n and keep e secret

- To sign m we send $\langle m, E(m) \rangle$
- Verifier receives $\langle m, c \rangle$
- To verify signature, check $m \stackrel{?}{=} D(c)$

Digital certificates

$$\begin{bmatrix} \text{Name:} & \text{Alice} \\ \text{Verification key:} & \langle \text{d,n} \rangle \\ \text{Expiration date:} & \text{Dec 31, 2020} \end{bmatrix}_{CA}$$

- Certificate is signed by Certificate Authority
  - Symantec (Verisign/Thawte/Geotrust): 38%
  - Comodo SSL: 29%
  - GoDaddy: 13%
  - GlobalSign: 10%
  - everyone else (combined): 10%

RSA exponents are always odd

- Recall: p, q large primes (and thus odd numbers)
- Recall: $ed \equiv 1 \pmod{(p-1)(q-1)}$
- So $(p-1)(q-1)$ will be even
- Thus ed is odd, and that means both e & d are odd

- Shortly, we find out what happens when we use an even exponent … (Rabin signatures)

Today's lecture

- Homomorphism
- Shamir secret sharing
- Secure computation
- Chinese remainder theorem
- Rabin signatures

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
  - $f(x+y) = f(x) + f(y) \pmod{n}$
  - $f(xy) = f(x)f(y) \pmod{n}$

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

Secret sharing

- Suppose we want to share a secret
  - Share among n users
  - Allow a quorum q of users to recover a secret
- Example
  - Corporate bank account
  - Requires three out of six corporate officers to access
- Shamir secret sharing allows to realize this
- But leaks no further information

Shamir secret sharing

- Key idea:
  - Make a random curve of degree $q-1$: $f(x)$
  - Distribute n points on curve: $f(1), f(2)$ ... , $f(n)$
  - q points determine the curve
  - $q-1$ points do <u>not</u> determine the curve
  - Secret is $f(0)$
  - If we do it mod m, then $q-1$ points give no info
  - $f(0)$ can be any integer mod m

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

Finding the secret

- This reduces to solving linear equations
- High School algebra techniques (but modulo n)
- Example ($q = 3$):

$$f(1) = a_2 + a_1 + a_0 \pmod{m}$$
$$f(2) = 4a_2 + 2a_1 + a_0 \pmod{m}$$
$$f(3) = 9a_2 + 3a_1 + a_0 \pmod{m}$$
$$f(4) = 16a_2 + 4a_1 + a_0 \pmod{m}$$
$$f(5) = 25a_2 + 5a_1 + a_0 \pmod{m}$$

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)

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 $SUM(0)$

Secure multi-party computation

- Using a variety of techniques, we can extend to all functions (not just addition)
- This is a super-hot area of security today
- Example: Prof. Raluca Popa (joining Berkeley next year) is working on making database search secure

Database secure computation

- If we encrypt entries on database, how do we search and change them
- If we decrypt and re-encrypt database, super-expensive
- Need some very advanced techniques to perform computation while encrypted

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

$$x \bmod n$$

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

CRT is homomorphic (addition)!

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

CRT is homomorphic (multiplication)!

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

Reading on CRT

- See reading (on Piazza for) algorithm to convert from CRT form to modular form

- http://www.cut-the-knot.org/blue/chinese.shtml

CRT with two primes

- We are especially interested in $n = pq$
  - Where p & q are large primes

CRT $15 = 3 * 5$

$$
\begin{aligned}
0 \bmod 15 &= \langle 0 \bmod 3, 0 \bmod 5\rangle\\
1 \bmod 15 &= \langle 1 \bmod 3, 1 \bmod 5\rangle\\
2 \bmod 15 &= \langle 2 \bmod 3, 2 \bmod 5\rangle\\
3 \bmod 15 &= \langle 0 \bmod 3, 3 \bmod 5\rangle\\
4 \bmod 15 &= \langle 1 \bmod 3, 4 \bmod 5\rangle\\
5 \bmod 15 &= \langle 2 \bmod 3, 0 \bmod 5\rangle\\
6 \bmod 15 &= \langle 0 \bmod 3, 1 \bmod 5\rangle\\
7 \bmod 15 &= \langle 1 \bmod 3, 2 \bmod 5\rangle\\
8 \bmod 15 &= \langle 2 \bmod 3, 3 \bmod 5\rangle
\end{aligned}
$$

Squares

- Let's think about squares: x^2
- Some integers are squares $\{1, 4, 9, 16, 25, ...\}$
- Some integers are not squares $\{2, 3, 5, 6, , ...\}$
- Non-zero integer squares: two square roots
  - $\sqrt{4} = \{2, -2\}$
  - $\sqrt{9} = \{3, -3\}$

Squares modulo prime p

- Let p be an odd prime
- Some integers mod p are squares (*quadratic residues*) and some are not
  - $1^2 = 1 \pmod 5$
  - $2^2 = 4 \pmod 5$
  - $3^2 = 4 \pmod 5$
  - $4^2 = 1 \pmod 5$

  - $\sqrt{1} = \{1, -1\} = \{1, 4\} \pmod 5$
  - $\sqrt{4} = \{2, -2\} = \{2, 3\} \pmod 5$

Squares modulo pq

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

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

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

What is going on here?

- Need to see CRT view to understand

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

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

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

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

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

Next lecture

- How to test if a number is a square mod pq
- How to take square roots mod pq

- and then – crypto-hashing