CS443 - More Crypto

Note Title 1/18/2013

Announcements

- Lab 1 due tomorrow
- No class Thursday
  Instead, read "No Tech Hacking"
  & essay due in class Tuesday.
- HW2 - over cryptography.
  due in 2 weeks

Last time: DES & AES

Symmetric key cryptography:

- based on shared knowledge of a private key
- very secure: 128-bit key in AES would take roughly 10 billion billion (est. 1.02×10^{18}) years to crack on a 2012 supercomputer

Assymetric key Cryptography

- based on 1-way communication
- huge development, first published in "New directions in cryptography" by Diffie & Hellman, 1976

Daily conspiracy:
Actually, secretly developed by government researchers in the U.K. in 1973.

General overview

(1) Alice looks up Bob's public key E_B in a directory.
Uses an algorithm to encode message m: written $E_B(m)$

(2) Bob then uses private key D_B to decode: $D_B(E_B(m)) = m$

Idea: Alice can give away m & $E_B(m)$, but no one else can decode without D_B.

Number Theory

2 & 8 are not relatively prime

3 and 11 are

The Euler phi function $\phi(n)$ is defined so that

$\phi(n) = \#$ of integers $\leq n$ that are $\underbrace{\text{relatively prime}}_{\text{no common divisors}}$ to n.

Ex: $\phi(7) = 6$ $\quad (\phi(7) = 7)$

$\phi(6) = \cancel{3}\, 2$

1 and 5

Lemma: If p is prime, $\phi(p) = p-1$

Lemma: If p & q are prime,

$$\phi(p \cdot q) = (p-1)(q-1)$$
$$= \phi(p)\,\phi(q)$$

Why? $p \cdot q$ divisors: $p, 2p, 3p, \ldots (q-1)\cdot p$

$q, 2q, \ldots (p-1)\cdot q$

Euler's thm: If n is a positive integer with $\underbrace{\gcd(a,n)=1}_{\substack{\uparrow \\ a \,\&\, n \text{ are relatively prime}}}$,

then $\underbrace{a^{\phi(n)} = 1}$ mod n

Cor: If a is relatively prime to p & q (both primes), then

$$\underbrace{a^{(p-1)(q-1)}}_{\substack{\| \\ a^{\phi(pq)}}} = 1 \pmod{pq}$$

Remember $\mathbb{Z}_n$?

Key: If a is relatively prime to n, then $\exists b$ with $ab = 1 \mod n$

Ex: $n = 8$

$a = 2$: $2x = 1 \mod 8 \rightarrow$ NO inverse

$a = 3$: $3x = 1 \mod 8 \Rightarrow x = 3$

$a = 4$: No inverse

$a = 5$: $5 \cdot x = 1 \mod \Rightarrow x = 5$

$a = 6$: No inverse

$a = 7$: $7 \cdot x = 1 \mod 8 \Rightarrow x = 7$

RSA: [Rivest - Shamir - Adelmann 1978]

① Bob generates 2 primes p & q and computes $n = p \cdot q$
$\phi(n) = (p-1)(q-1)$

② Bob picks e relatively prime to $\phi(n)$, & then finds d s.t. $ed \equiv 1 \bmod \phi(n)$
(via Euclidean algorithm)

→ d is private key

→ (e, n) are public key

Side note: Euclidean Algorithm

Input: a, b
While $b \neq 0$
    $r \leftarrow a \bmod b$
    $a \leftarrow b$
    $b \leftarrow r$
Return a

What does it do?

calculates gcd of a & b

So: given e & $\phi(n)$ relatively prime,

then $\gcd(e, \phi(n)) = 1$

By tracking variables in this algorithm, get value of d.

Now: Alice has a message m.
To send it to Bob:

1. Compute $c = m^e \bmod n$ — (e, n) is public
2. Send to Bob

Bob decodes:

$$c^d = (m^e)^d \bmod n$$

$$= m^{ed} \bmod n \stackrel{?}{=} m$$

Why it works:

- know $ed \equiv 1 \mod \phi(n)$

  so $ed = \boxed{1 + k\phi(n)}$ for some k

  $= 1 + k(p-1)(q-1)$

- Then $m^{ed} \equiv m^{1+k\phi(n)} \mod n$

  $\equiv (m^1)(m^{k\phi(n)}) \mod n$

  $\equiv m \cdot \left(m^{\phi(n)}\right)^k \mod n$

  $\equiv m \, (1)^k \mod n \equiv m$

Public: (e, n)

How hard to get d?

- Easy if you know $\phi(n) = (p-1)(q-1)$

need $ed \equiv 1 \bmod \phi(n)$

Need to factor n to find p & q

Notes

- Actually, in general no one will know p or q - use central certificate authority.

Public: (e, n)
Private: d

How do we get d again?

↳ Simple with p & q.
"Hard" without.

This is why the effectiveness of RSA is based on factoring!

If we knew $\phi(n) = (p-1)(q-1)$, could break the system.

How hard is factoring?

No 512-bit number has (yet) been factored.

Diffie-Hellman key exchange [1976]

Most common use of public key cryptography is to exchange private keys!

Why?

Symmetric encryption is more secure.

Diffie-Hellman basics:

Consider a prime number q (or $q = p^k$, with p prime).

We saw last time that $\mathbb{Z} \bmod q$ is a finite field:

- nice + & × operation
- has <u>multiplicative</u> <u>inverse</u>:

<u>Ex</u>: $2 \cdot x = 1 \mod 5$

$$\Rightarrow x = 3$$

The protocol: p prime, $s < p$ (both public)

- Alice chooses $a < p$
  Bob chooses $b < p$

- Alice computes $\alpha = s^a \bmod p$
  Bob computes $\beta = s^b \bmod p$

- They exchange α and β

- Alice computes $\beta^a \bmod p$
  Bob computes $\alpha^b \bmod p$

$$\beta^a = (s^b)^a \bmod p = s^{ab} \bmod p$$
$$\alpha^b = (s^a)^b \bmod p = s^{ab} \bmod p$$

secret key

Ex: Let $s = 2$, $p = 29$

Alice likes $a = 14$

Bob likes $b = 12$

$$\alpha = 2^{14} \bmod 29 = 28$$

$$\beta = 2^{12} \bmod 29 = 7$$

$$\alpha^{12} \bmod 29 = 28^{12} \bmod 29 = 1$$

$$\beta^{14} \bmod 29 = 7^{14} \bmod 29 = 1$$

Common key $k = s^{ab} \bmod p$

Recap: Public info: • p and s

• $\alpha = s^a \bmod p$

• $\beta = s^b \bmod p$

Private: a (to Alice)

b (to Bob)

k

$$\alpha \cdot \beta = (s^a)(s^b) \bmod p = \underbrace{s^{a+b}}_{\neq k} \bmod p$$

Why is it hard to break?

At its base, the key is logarithms.
(Remember those?)

$\log_{10} 1000 = 3$

$\log_2 1024 = 10$

We want discrete logs:
given α, find a.

$\log_s(s^a) \bmod p$

The Discrete Log Problem

This is another one we "think" is hard.

Similar to factoring.

Note: There are ways to attack this! Not NP-Hard - just no known fast algorithms.

Stonger generalizations work in groups other than $\mathbb{Z}_p$.

(But elliptic curves are a bit beyond us now...)

Bigger Picture : NSA Suite B

The NSA has published a set of recommended algorithms (for both unclassified information as well as info up to SECRET).

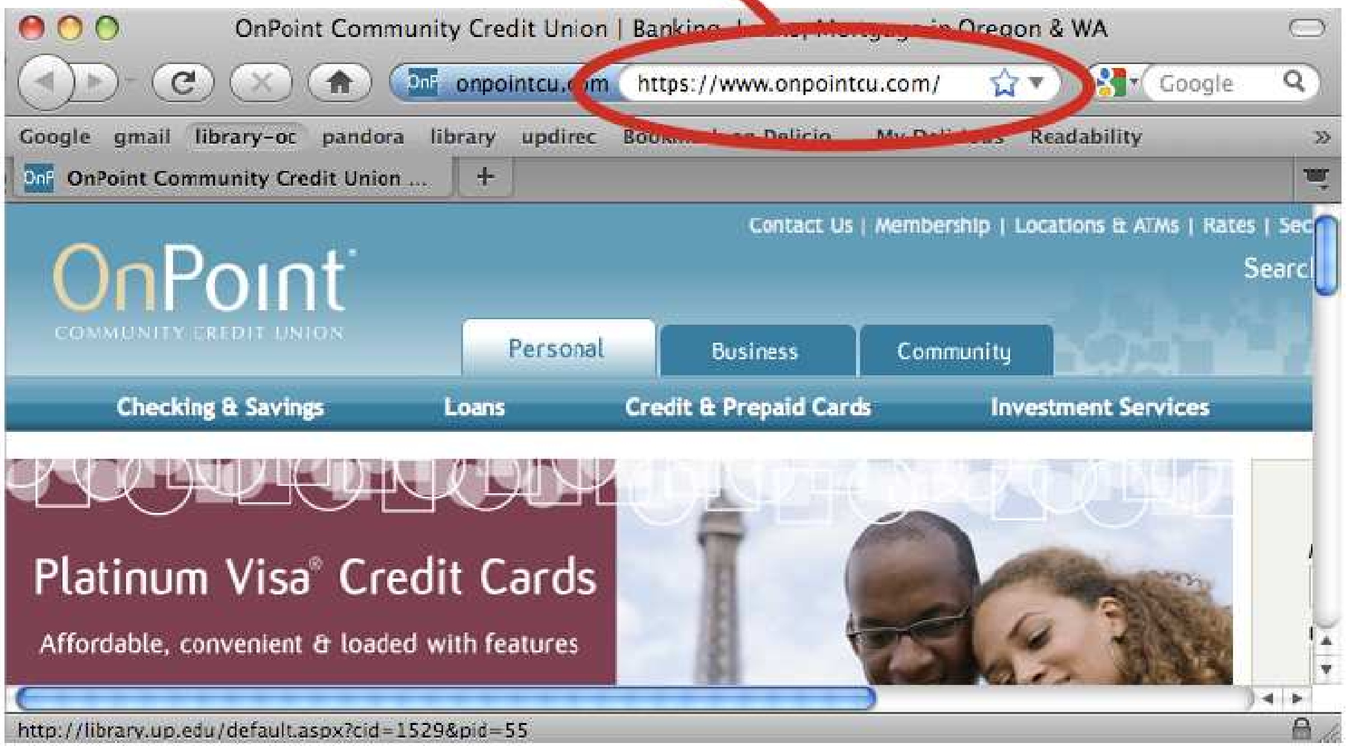
- Encryption : AES
- Signatures : Elliptic curve Diffie-Hellman
- Hashing : SHA (Secure hashing algorithm)

So- why study RSA??

Whenever you see "https", that's TLS at work.

Still in use!

(RSA is the basis of the Transport Layer Security in browsers.)



Why RSA: Cost

(Also used in smart cards, operating systems, etc.)

But why, if ECDH is better??

Most companies can't afford it, since the patents are still largely held by one company.

certicom