

Cryptography CS 555

Lecture 14-b



Primality Testing and Attacks on RSA

Review of Last Lecture

- Goal: testing whether a large number is a prime number or not without factorizing the number
- Cyclic group
 - has a generator, whose order is the size of the group
- The cyclic group Z_p^*
 - generators also known as primitive elements modulo p (how many are these?)

Lecture Outline

- Quadratic residues
- Primality testing
- Factorization
- Attacks on RSA



Testing Primitive Elements Modulo p

- The number of primitive elements modulo p is $\phi(p-1)$.

Theorem: Let p be a prime, $a \in \mathbb{Z}_p^*$ is a primitive element modulo p iff. $a^{(p-1)/q} \not\equiv 1 \pmod{p}$ for all primes q such that $q|(p-1)$.

Proof. The only if direction is straightforward.

For the if direction. If a is not primitive, it has order $d < (p-1)$. Then d is a divisor of $(p-1)$. Let q be a prime factor of $(p-1)/d$, i.e., $(p-1)/d = cq$. Then $(p-1)/q = cd$. Then $a^{(p-1)/q} = 1 \pmod{p}$.

Quadratic Residues Modulo A Prime

Definition

- a is a **quadratic residue** modulo p if $\exists b \in \mathbb{Z}_p^*$ such that $b^2 \equiv a \pmod{p}$,
- otherwise when $a \neq 0$, a is a **quadratic nonresidue**
- Q_p is the set of all quadratic residues
- \overline{Q}_p is the set of all quadratic nonresidues
- If p is prime there are $(p-1)/2$ quadratic residues in \mathbb{Z}_p^* ,
 $|Q_p| = (p-1)/2$
 - let g be generator of \mathbb{Z}_p^* , then $a=g^j$ is a quadratic residue iff. j is even.

How Many Square Roots Does an Element in \mathbb{Q}_p have

- A element a in \mathbb{Q}_p has exactly two square roots
 - a has at least two square roots
 - if $b^2 \equiv a \pmod{p}$, then $(p-b)^2 \equiv a \pmod{p}$
 - a has at most two square roots in \mathbb{Z}_p^*
 - if $b^2 \equiv a \pmod{p}$ and $c^2 \equiv a \pmod{p}$, then $b^2 - c^2 \equiv 0 \pmod{p}$
 - then $p \mid (b+c)(b-c)$, either $b=c$, or $b+c=p$

Legendre Symbol

- Let p be an odd prime and a an integer. The Legendre symbol is defined

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{if } p \mid a \\ 1, & \text{if } a \in Q_p \\ -1, & \text{if } a \in \overline{Q}_p \end{cases}$$

Euler's Criterion

Theorem: If $a^{(p-1)/2} \equiv 1 \pmod{p}$, then a is a quadratic residue (if $\equiv -1$ then a is a quadratic nonresidue)

I.e., the Legendre symbol $\left(\frac{a}{p}\right) = a^{(p-1)/2} \pmod{p}$

Proof. If $a = y^2$, then $a^{(p-1)/2} = y^{(p-1)} = 1 \pmod{p}$

If $a^{(p-1)/2} = 1$, let $a = g^j$, where g is a generator of the group Z_p^* . Then $g^{j(p-1)/2} = 1 \pmod{p}$. Since g is a generator, $(p-1) \mid j(p-1)/2$, thus j must be even. Therefore, $a = g^j$ is QR.

Jacobi Symbol

- let $n \geq 3$ be odd with prime factorization

$$n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$$

- the Jacobi symbol is defined to be

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k}$$

- the Jacobi symbol can be computed without factoring n (see the textbook for details)

Euler Pseudo-prime

- For any prime p , the Legendre symbol $\left(\frac{a}{p}\right) = a^{(p-1)/2} \pmod{p}$
- For a composite n , if the Jacobi symbol $\left(\frac{a}{n}\right) = a^{(n-1)/2} \pmod{n}$ then n is called an Euler pseudo-prime to the base a ,
 - i.e., a is a “pseudo” evidence that n is prime
- For any composite n , the number of “pseudo” evidences that n is prime for at most half of the integers in Z_n^*

Randomized Algorithms

- A yes-biased Monte Carlo algorithm is a randomized algorithm for a decision problem in which a “yes” answer is (always correct), but a “no” answer may be incorrect
 - error probability for an instance is the probability that instance is answered incorrectly
 - error probability for the algorithm is the max among all instance error probabilities
- A no-biased Monte Carlo algorithm is defined similarly
- A Las Vegas algorithm may not give an answer, but any answer it gives is correct

The Solovay-Strassen Algorithm

Solovay-Strassen(n)

choose a random integer a s.t. $1 \leq a \leq n-1$

$x \leftarrow \left(\frac{a}{n}\right)$

if $x=0$ then return (“ n is composite”) // $\gcd(x,n) \neq 1$

$y \leftarrow a^{(n-1)/2} \bmod n$

if $(x=y)$ then return (“ n is prime”)

// either n is a prime, or a pseudo-prime

else return (“ n is composite”)

// violates Euler’s criterion

If n is composite, it passes the test with at most $\frac{1}{2}$ prob.

Use multiple tests before accepting n as prime.

Rabin-Miller Test

- Another efficient probabilistic algorithm for determining if a given number n is prime.
 - Write $n-1$ as $2^k m$, with m odd.
 - Choose a random integer a , $1 \leq a \leq n-1$.
 - $b \leftarrow a^m \bmod n$
 - if $b=1$ then return “ n is prime”
 - compute $b, b^2, b^4, \dots, b^{2^{(k-1)}}$, if we find -1 , return “ n is prime”
 - return “ n is composite”
- A composite number pass the test with $\frac{1}{4}$ prob.
- When t tests are used with independent a , a composite passes with $(\frac{1}{4})^t$ prob.
- The test is fast, used very often in practice.

Why Rabin-Miller Test Work

Claim: If the algorithm returns “n is composite”, then n is not a prime.

Proof: if we choose a and returns composite on n, then

- $a^m \neq 1, a^m \neq -1, a^{2^m} \neq -1, a^{4^m} \neq -1, \dots, a^{2^{k-1}m} \neq -1 \pmod{n}$
- suppose, for the sake of contradiction, that n is prime,
- then $a^{n-1} = a^{2^k m} = 1 \pmod{n}$
- then there are two square roots modulo n, 1 and -1
- then $a^{2^{k-1}m} = a^{2^{k-2}m} = a^{2^m} = a^m = 1$ (contradiction!)
- so if n is prime, the algorithm will not return “composite”

Quadratic Residues Modulo a Composite

Definition: a is a **quadratic residue** modulo n ($a \in Q_n$) if $\exists b \in \mathbb{Z}_n^*$ such that $b^2 \equiv a \pmod{n}$, otherwise when $a \neq 0$, a is a **quadratic nonresidue**

Fact: $a \in Q_n^*$, where $n=pq$, iff. $a \in Q_p$ and $a \in Q_q$

- If $b^2 \equiv a \pmod{n}$, then $b^2 \equiv a \pmod{p}$ and $b^2 \equiv a \pmod{q}$
- If $b^2 \equiv a \pmod{p}$ and $c^2 \equiv a \pmod{q}$, then the solutions to
 - $x \equiv b \pmod{p}$ and $x \equiv c \pmod{q}$
 - $x \equiv b \pmod{p}$ and $x \equiv -c \pmod{q}$
 - $x \equiv -b \pmod{p}$ and $x \equiv c \pmod{q}$
 - $x \equiv -b \pmod{p}$ and $x \equiv -c \pmod{q}$satisfies $x^2 \equiv a \pmod{p}$

Quadratic Residues Modulo a Composite

- $|\underline{Q}_n| = |\underline{Q}_p| \cdot |\underline{Q}_q| = (p-1)(q-1)/4$
- $Q_n = 3(p-1)(q-1)/4$
- Jacobi symbol does not tell whether a number a is a QR

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right) \left(\frac{a}{q}\right)$$

- when it is -1, then either $a \in \underline{Q}_p \wedge a \notin \underline{Q}_q$ or $a \notin \underline{Q}_p \wedge a \in \underline{Q}_q$
- when it is 1, then either $a \in \underline{Q}_p \wedge a \in \underline{Q}_q$ or $a \notin \underline{Q}_p \wedge a \notin \underline{Q}_q$
- it is widely believed that determining QR modulo n is equivalent to factoring n , no proof is known
 - without factoring, one can guess correctly with prob. $1/2$

Summary of Number Theory Results Covered

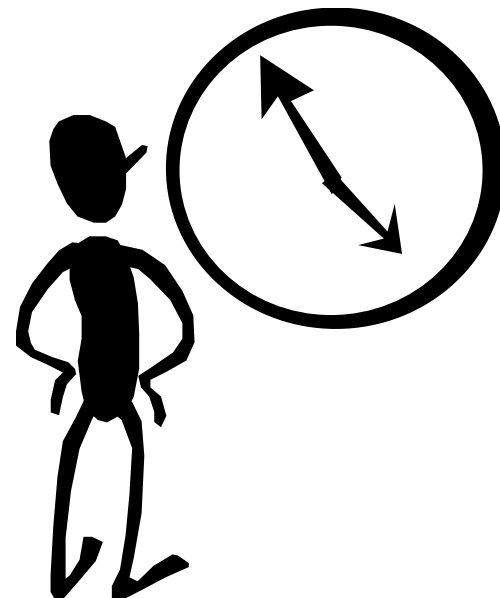
- Z_p^* is a cyclic group
 - has generators
- QR and QNR in Z_p^* can be easily determined by computing the Legendre symbol
- Jacobi symbol (generalizes Legendre symbol to composites)
 - can be computed without factoring n
 - Jacobi symbol does not determine QR in Z_n^*
 - QR in Z_n^* is hard
- Primality Testing
 - Solovay-Strassen
 - Rabin-Miller

Attacks on RSA

- Goals:
 - recover secret key d
 - Brute force key search
 - infeasible
 - Timing attacks
 - Mathematical attacks
 - decrypt one message
 - learn information from the cipher texts

Timing Attacks

- *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems (1996), Paul C. Kocher*
- By measuring the time required to perform decryption (exponentiation with the private key as exponent), an attacker can figure out the private key
- Possible countermeasures:
 - use constant exponentiation time
 - add random delays
 - blind values used in calculations



Timing Attacks (cont.)

- Is it possible in practice? YES.

OpenSSL Security Advisory [17 March 2003]

Timing-based attacks on RSA keys

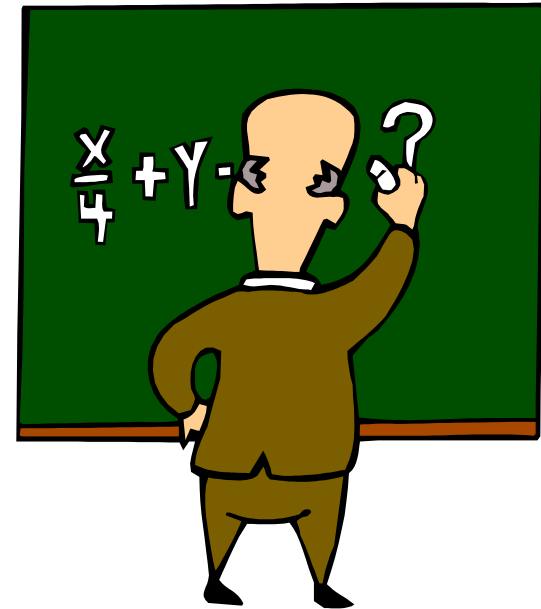
=====

OpenSSL v0.9.7a and 0.9.6i vulnerability

Researchers have discovered a timing attack on RSA keys, to which OpenSSL is generally vulnerable, unless RSA blinding has been turned on.

Math-Based Key Recovery Attacks

- Three possible approaches:
 1. Factor $n = pq$
 2. Determine $\Phi(n)$
 3. Find the private key d directly
- All the above are equivalent to factoring n
 - 1 implies 2
 - 2 implies 3
 - needs to show that 3 implies 1



$\Phi(n)$ implies factorization

- Knowing both n and $\Phi(n)$, one knows

$$n = pq$$

$$\Phi(n) = (p-1)(q-1) = pq - p - q + 1$$

$$= n - p - n/p + 1$$

$$p\Phi(n) = np - p^2 - n + p$$

$$p^2 - np + \Phi(n)p - p + n = 0$$

$$p^2 - (n - \Phi(n) + 1)p + n = 0$$

- There are two solutions of p in the above equation.
- Both p and q are solutions.

Factoring Large Numbers

- Three most effective algorithms are
 - quadratic sieve
 - elliptic curve factoring algorithm
 - number field sieve
- One idea many factoring algorithms use:
 - Suppose one find $x^2 \equiv y^2 \pmod{n}$ such that $x \not\equiv y \pmod{n}$ and $x \not\equiv -y \pmod{n}$. Then $n \mid (x-y)(x+y)$. Neither $(x-y)$ or $(x+y)$ is divisible by n ; thus, $\gcd(x-y, n)$ has a non-trivial factor of n

Time complexity of factoring

- quadratic sieve:
 - $O(e^{(1+o(1))\sqrt{\ln n \ln \ln n}})$ for n around 2^{1024} , $O(e^{68})$
- elliptic curve factoring algorithm
 - $O(e^{(1+o(1))\sqrt{2 \ln p \ln \ln p}})$, where p is the smallest prime factor
 - for $n=pq$ and p, q around 2^{512} , for n around 2^{1024} $O(e^{65})$
- number field sieve
 - $O(e^{(1.92+o(1)) (\ln n)^{1/3} (\ln \ln n)^{2/3}})$, for n around 2^{1024} $O(e^{60})$
- Multiple 512-bit moduli have been factored
- Extrapolating trends of factoring suggests that
 - 768-bit moduli will be factored by 2010
 - 1024-bit moduli will be factored by 2018

Factoring when knowing e and d

- **Fact:** if $n=pq$, then $x^2 \equiv 1 \pmod{n}$ has four solutions that are $<n$.
 - $x^2 \equiv 1 \pmod{n}$ if and only if
 - both $x^2 \equiv 1 \pmod{p}$ and $x^2 \equiv 1 \pmod{q}$
 - Two trivial solutions: 1 and $n-1$
 - 1 is solution to $x \equiv 1 \pmod{p}$ and $x \equiv 1 \pmod{q}$
 - $n-1$ is solution to $x \equiv -1 \pmod{p}$ and $x \equiv -1 \pmod{q}$
 - Two other solutions
 - solution to $x \equiv 1 \pmod{p}$ and $x \equiv -1 \pmod{q}$
 - solution to $x \equiv -1 \pmod{p}$ and $x \equiv 1 \pmod{q}$
 - E.g., $n=3 \times 5=15$, then $x^2 \equiv 1 \pmod{15}$ has the following solutions:
1, 4, 11, 14

Factoring when knowing e and d

- Knowing a nontrivial solution to $x^2 \equiv 1 \pmod{n}$
 - compute $\gcd(x+1, n)$ and $\gcd(x-1, n)$
- E.g., 4 and 11 are solution to $x^2 \equiv 1 \pmod{15}$
 - $\gcd(4+1, 15) = 5$
 - $\gcd(4-1, 15) = 3$
 - $\gcd(11+1, 15) = 3$
 - $\gcd(11-1, 15) = 5$

Summary of Key Recovery Math-based Attacks on RSA

- Three possible approaches:
 1. Factor $n = pq$
 2. Determine $\Phi(n)$
 3. Find the private key d directly
- All are equivalent
 - finding out d implies factoring n
 - if factoring is hard, so is finding out d
- Should never have different users share one common modulus
 - (why?)

Summary

- Efficient probabilistic algorithms for primality testing exist
- The following are equivalent
 - factoring n
 - computing $\phi(n)$
 - find d for the corresponding e



Next ...

- SRA Mental Poker Protocol
- Semantic Security of RSA Encryption
- OAEP
- Stinson, Chapter 5.8, 5.9

