

Cryptography CS 555

Lecture 14-c



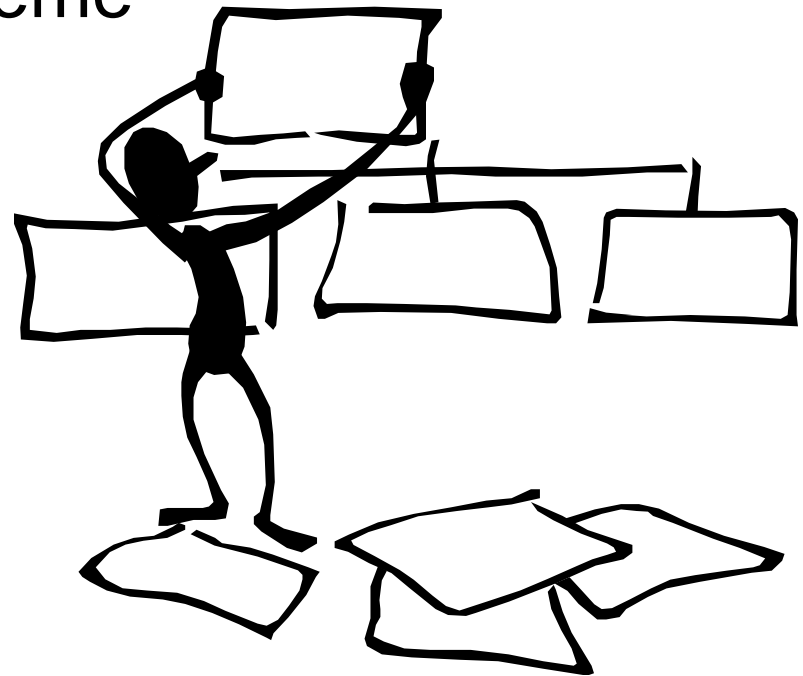
Attacks on RSA and Rabin Encryption

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

Lecture Outline

- Attacks on RSA
- Rabin encryption scheme



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

Factoring Large Numbers

- 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

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 $x^2 \equiv 1 \pmod{15}$
 - $\gcd(4+1, 15) = 5$
 - $\gcd(4-1, 15) = 3$
 - $\gcd(11+1, 15) = 3$
 - $\gcd(11-1, 15) = 5$

Factoring when knowing e and d

- Knowing ed such that $ed \equiv 1 \pmod{\Phi(n)}$
 - write $ed - 1 = 2^s r$ (r odd)
 - choose w at random such that $1 < w < n-1$
 - if w not relative prime to n then return $\gcd(w, n)$
 - (if $\gcd(w, n) = 1$, what value is $(w^{2^s r} \pmod n)$?)
 - compute $w^r, w^{2r}, w^{4r}, \dots$, by successive squaring until find $w^{2^t r} \equiv 1 \pmod n$
 - Fails when $w^r \equiv 1 \pmod n$ or $w^{2^t r} \equiv -1 \pmod n$
 - Failure probability is less than $\frac{1}{2}$ (Proof is complicated)

Example: Factoring n given d

- Input: $n=2773$, $e=17$, $d=157$
- $ed-1=2668=2^2 \cdot 667$ (r=667)
- Pick random w, compute $w^r \bmod n$
 - $w=7$, $7^{667}=1$ no good
 - $w=8$, $8^{667}=471$, and $471^2=1$, so 471 is a nontrivial square root of 1 mod 2773
 - compute $\gcd(471+1, 2773)=59$
 - $\gcd(471-1, 2773)=47$.
 - $2773=59 \cdot 47$

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?)

Decryption attacks on RSA

- The RSA Problem: Given a positive integer n that is a product of two distinct large primes p and q , a positive integer e such that $\gcd(e, (p-1)(q-1))=1$, and an integer c , find an integer m such that $m^e \equiv c \pmod{n}$
 - widely believed that the RSA problem is computationally equivalent to integer factorization; however, no proof is known
- The security of RSA encryption's scheme depends on the hardness of the RSA problem.

Other Decryption Attacks on RSA

Small encryption exponent e

- When $e=3$, Alice sends the encryption of message m to three people (public keys (e, n_1) , (e, n_2) , (e, n_3))
 - $C_1 = M^3 \bmod n_1$, $C_2 = M^3 \bmod n_2$, $C_3 = M^3 \bmod n_3$,
- An attacker can compute a solution to the following system

$$x \equiv c_1 \pmod{n_1}$$

$$x \equiv c_2 \pmod{n_2}$$

$$x \equiv c_3 \pmod{n_3}$$

- The solution x modulo $n_1 n_2 n_3$ must be M^3
 - (No modulus!), one can compute integer cubit root
- Countermeasure: padding required

Other Attacks on RSA

Forward Search Attack

- If the message space is small, the attacker can create a dictionary of encrypted messages (public key known, encrypt all possible messages and store them)
- When the attacker 'sees' a message on the network, compares the encrypted messages, so he finds out what particular message was encrypted

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

Other Attacks on RSA

Small decryption exponent d

- Choosing a small exponent helps efficiency **BUT**
- If size of d is 1/4 size of n (in bits) and $\gcd(p-1, q-1)$ is small, there is a way to compute d only from e and n .
- Countermeasure: d should be about the same size as n .

The Rabin Encryption Scheme

- Motivation: The security of RSA encryption depends on the difficulty of computing the e 'th root modulo n , i.e., given C , it is difficult to find M s.t. $M^e = C \pmod n$.
- It is not known that this encryption is as difficult as factoring.
- The Rabin encryption scheme is provably “secure” if factoring is hard
 - here “secure” means to recover the plaintext from a ciphertext
- Idea: rather than using an odd prime as e , uses 2
 - $f(x) = x^2 \pmod n$
 - this is not a special case of RSA as this function is not 1-to-1.

The Rabin Encryption Scheme

- Public key: n
- Privacy key: p, q s.t. $n=pq$
- Encryption: compute $c=m^2 \bmod n$
- Decryption: compute the square roots of c .
 - how many are there?
- **Fact:**
 - when $p \equiv q \equiv 3 \pmod{4}$, deterministic algorithms exist to compute the square roots
 - otherwise, efficient randomized algorithms exist to compute the square roots

Computing Square Roots is as hard as Factoring

- Given an algorithm A that can compute one square root of a number a modulo n ,
- One can use A to factor n as follows
 - randomly pick x , compute $z = x^2 \bmod n$
 - ask A to compute the square root of z , A returns y
 - if $y=x$ or $y=n-x$, then try again, otherwise, compute $\gcd(x,y)$ gives us a prime factor of n
 - as A has no way to tell which x we've picked, with prob. $\frac{1}{2}$, A returns a square root that allows us to factor n

Pragmatic Considerations for the Rabin Encryption Scheme

- Normally, one picks $p \equiv q \equiv 3 \pmod{4}$
- Redundency is used to ensure that only one square root is a legitimate message
- Encryption very fast, only one exponentiation
- Decryption comparable to RSA decryption

Summary

- Efficient probabilistic algorithms for primality testing exist
- The following are equivalent
 - factoring n
 - computing $\phi(n)$
 - find d for the corresponding e
- The Rabin cryptosystem is “provably secure”



Next ...

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

