# CS 556 – Computer Security
# Spring 2018

Dr. Indrajit Ray
Email: indrajit@cs.colostate.edu

Department of Computer Science
Colorado State University
Fort Collins, CO 80523, USA

# RSA CRYPTOSYSTEM

# RSA Public Key Cryptosytem



- Best known and widely regarded as practical public-key scheme
- Proposed by Rivest, Shamir, Adleman in 1978
  R L Rivest, A Shamir, L Adleman, "On Digital Signatures and Public Key Cryptosystems", Communications of the ACM, vol 21(2), Feb 1978
- It is a public-key scheme which may be used for encrypting messages, exchanging keys, and creating digital signatures

# RSA Cryptosystem



- RSA is a public key encryption algorithm based on exponentiation using modular arithmetic
- Its security relies on the difficulty of calculating factors of large numbers
- The algorithm is patented in North America (although algorithms cannot be patented elsewhere in the world)
  - This is a source of legal difficulties in using the scheme

# The RSA Algorithm

RSA CRYPTOSYSTEM

- Select two large primes, p and q ($\sim$ 100 digits or larger) at random; keep these private
- Multiply p and q to calculate the system modulus n
- To obtain your public key <e,n>, select e $<$ n, relatively prime to $\phi$(n)
- To obtain your private key <d,n>, find d that is the multiplicative inverse of e $\mod$ $\phi$(n)
- To encrypt m $\ll$ n, compute c $=$ m$^e$ $\mod$ n
- To decrypt c, compute m $=$ c$^d$ $\mod$ n

# Security of RSA



- To break RSA we have to compute d such that e.d $=$ 1 mod (p-1).(q-1)
- How hard is it to compute d given (e,n)?
  - Do not know
  - But it is no harder than factorizing n into p and q
  - So security of RSA is <u>no better</u> than the complexity of the factoring problem

# Security of RSA



- Until 1989 factorization attacks were based on “high school mathematics”. Since then sophisticated attacks have extended factorization to large numbers. At present it appears that 130 digit numbers can be factored in several months using lots of idle workstations

# Security of RSA



- Best known factorization algorithm (Brent-Pollard) takes

$$O\left(\frac{e^{\sqrt{2.ln(p).ln(ln(p))}}}{ln(p)}\right)$$

operations on number n whose largest prime factor is p

# Security of RSA



| Decimal digits in n | # of operations to factor n |
|---|---|
| 20 | 7200 |
| 40 | $3.11.e^{+06}$ |
| 60 | $4.63.e^{+08}$ |
| 80 | $3.72.e^{+10}$ |
| 100 | $1.97.e^{+12}$ |
| 120 | $7.69.e^{+13}$ |
| 140 | $2.35.e^{+15}$ |
| 160 | $5.92.e^{+16}$ |
| 180 | $1.26.e^{+18}$ |

# Security of RSA



- Cryptographers can stay ahead of (factorization) cryptanalysts by increasing the key size
- Many implementations choose n to be 512 bits and the key to be 1024 bits

# Generating Big Prime Numbers



- The probability of a randomly chosen number being prime is 1/(ln n)
  - For a 100 digit number, the chance is about 1/230
  - Guess and check, should take 230 tries on the average
- Does it REALLY have to be prime?
  - Good algorithms exist for finding “probably prime” numbers

# Digital Signatures in RSA



- RSA has an important property, not shared by other public key cryptosystems: Encryption and Decryption are commutative, that is
  - Encryption followed by decryption yields the original message
  - Decryption followed by encryption yields the original message
- Any cryptosystem that preserves message length will have this commutative property

# RSA versus DES



- Fastest implementation of RSA can encrypt in the range of kilobits/second
- Fastest implementation of DES can encrypt in the range of megabits/second
  - This 1000-fold difference in speed is likely to remain independent of technology advances
- In software DES is about 100 times faster than RSA
- It is often proposed that RSA be used for secure exchange of DES keys