
Public/Private Keys, RSA Public Key Algorithm

27 April 2010
Lecture 6

Slide Credits: Steve Zdancewic (UPenn)

Topics for Today

- Intro to Public/Private key encryption
- Modular Mathematics
 - Euler's Totient Function
 - Fermat's Little Theorem
 - Chinese Remainder Theorem
- RSA Algorithm
- Proving Primes
 - Rabin-Miller primality test
- Source: HAC 8.2

Public Key Cryptography

- Sender encrypts using a *public key*
- Receiver decrypts using a *private key*
- Only the private key must be kept secret
 - Public key can be distributed at will
- Also called *asymmetric* cryptography
- Can be used for digital signatures
- Examples: RSA, El Gamal, DSA, various algorithms based on elliptic curves

- Used in SSL, ssh, PGP, ...

Public Key Notation

- Encryption algorithm
 $E : \text{keyPub} \times \text{plain} \rightarrow \text{cipher}$
Notation: $K\{\text{msg}\} = E(K, \text{msg})$
- Decryption algorithm
 $D : \text{keyPriv} \times \text{cipher} \rightarrow \text{plain}$
Notation: $k\{\text{msg}\} = D(k, \text{msg})$
- D inverts E
 $D(k, E(K, \text{msg})) = \text{msg}$
- Use capital “K” for public keys
- Use lower case “k” for private keys
- Sometimes E is the same algorithm as D

Secure Channel: Private Key

Alice



K_A, K_B
 k_A

$K_B\{\text{Hello!}\}$



$K_A\{\text{Hi!}\}$



Bart



K_A, K_B
 k_B

Trade-offs for Public Key Crypto

- More computationally expensive than shared key crypto
 - Algorithms are harder to implement
 - Require more complex machinery
- More formal justification of difficulty
 - Hardness based on complexity-theoretic results
- A principal needs one private key and one public key
 - Number of keys for pair-wise communication is $O(n)$

RSA Algorithm

- Ron Rivest, Adi Shamir, Leonard Adleman
 - Proposed in 1979
 - They won the 2002 Turing award for this work
- Has withstood years of cryptanalysis
 - Not a guarantee of security!
 - But a strong vote of confidence.
- Hardware implementations: 1000 x slower than DES

RSA at a High Level

- Public and private key are derived from secret prime numbers
 - Keys are typically ≥ 1024 bits
- Plaintext message (a sequence of bits)
 - Treated as a (large!) binary number
- Encryption is modular exponentiation
- To break the encryption, conjectured that one must be able to factor large numbers
 - Not known to be in P (polynomial time algorithms)

So Far

- Intro to Public/Private key encryption
- Modular Mathematics
 - Euler's Totient Function
 - Fermat's Little Theorem
 - Chinese Remainder Theorem
- RSA Algorithm
- Proving Primes
 - Rabin-Miller primality test

Number Theory: Modular Arithmetic

- Examples:
 - $10 \bmod 12 = 10$
 - $13 \bmod 12 = 1$
 - $(10 + 13) \bmod 12 = 23 \bmod 12 = 11 \bmod 12$
 - $23 \equiv 11 \pmod{12}$
 - “23 is congruent to 11 (mod 12)”
- $a \equiv b \pmod{n}$ iff $a = b + kn$ (for some integer k)
- The *residue* of a number modulo n is a number in the range $0 \dots n-1$

Number Theory: Prime Numbers

- A *prime number* is an integer > 1 whose only factors are 1 and itself.
- Two integers are *relatively prime* if their only common factor is 1
 - gcd = greatest common divisor
 - $\text{gcd}(a,b) = 1$
 - $\text{gcd}(15,12) = 3$, so they're not relatively prime
 - $\text{gcd}(15,8) = 1$, so they are relatively prime
- Easy to compute GCD using Euclid's Algorithm

Finite Fields (Galois Fields)

- For a prime p , the set of integers mod p forms a *finite field*
- Addition $+$ Additive unit 0
- Multiplication $*$ Multiplicative unit 1
- Inverses: $n * n^{-1} = 1$ for $n \neq 0$
 - Suppose $p = 5$, then the finite field is $\{0,1,2,3,4\}$
 - $2^{-1} = 3$ because $2 * 3 \equiv 1 \pmod{5}$
 - $4^{-1} = 4$ because $4 * 4 \equiv 1 \pmod{5}$
- Usual laws of arithmetic hold for modular arithmetic:
 - Commutativity, associativity, distributivity of $*$ over $+$

RSA Key Generation

- Choose large, distinct primes p and q .
 - Should be roughly equal length (in bits)
- Let $n = p \cdot q$
- Choose a random encryption exponent e
 - With requirement: e and $(p-1) \cdot (q-1)$ are relatively prime.
- Derive the decryption exponent d
 - $d = e^{-1} \text{ mod } ((p-1) \cdot (q-1))$
 - d is e 's inverse mod $((p-1) \cdot (q-1))$
- Public key: $K = (e, n)$ pair of e and n
- Private key: $k = (d, n)$
- Discard primes p and q (they're not needed anymore)

RSA Encryption and Decryption

- Message: m
- Assume $m < n$
 - If not, break up message into smaller chunks
 - Good choice: largest power of 2 smaller than n
- Encryption: $E((e,n), m) = m^e \bmod n$
- Decryption: $D((d,n), c) = c^d \bmod n$

Example RSA

- Choose $p = 47$, $q = 71$
- $n = p * q = 3337$
- $(p-1)*(q-1) = 3220$
- Choose e relatively prime with 3220: $e = 79$
 - Public key is $(79, 3337)$
- Find $d = 79^{-1} \bmod 3220 = 1019$
 - Private key is $(1019, 3337)$
- To encrypt $m = 688232687966683$
 - Break into chunks < 3337
 - 688 232 687 966 683
- Encrypt: $E((79, 3337), 688) = 688^{79} \bmod 3337 = 1570$
- Decrypt: $D((1019, 3337), 1570) = 1570^{1019} \bmod 3337 = 688$

Euler's *totient* function: $\phi(n)$

- $\phi(n)$ is the number of positive integers less than n that are relatively prime to n
 - $\phi(12) = 4$
 - Relative primes of 12 (less than 12): $\{1, 5, 7, 11\}$
 - For p a prime, $\phi(p) = p-1$. Why?
 - For p, q two distinct primes, $\phi(p \cdot q) = (p-1)(q-1)$
 - There's $p \cdot q - 1$ numbers less than $p \cdot q$
 - Factors of $p \cdot q =$
 - $\{1 \cdot p, 2 \cdot p, \dots, q \cdot p\}$ for a total of q of them
 - $\{1 \cdot q, 2 \cdot q, \dots, p \cdot q\}$ for another p of them
 - No other numbers
 - $\phi(p \cdot q) = (p \cdot q) - (p + q - 1) = pq - p - q + 1 = (p-1)(q-1)$
- Handwritten annotations:*
- Red arrow from "All #s $\leq p \cdot q$ " points to $(p \cdot q)$ in the formula.
 - Red arrow from "p many multiples of q" points to $(p + q - 1)$ in the formula.
 - Red arrow from "q many multiples of p" points to $(p + q - 1)$ in the formula.
 - Red arrow from "don't double count $p \cdot q$ " points to $(p + q - 1)$ in the formula.

Fermat's Little Theorem

- Generalized by Euler.
- Theorem: If p is a prime then $a^p \equiv a \pmod{p}$.
- Corollary: If $\gcd(a,n) = 1$ then $a^{\phi(n)} \equiv 1 \pmod{n}$.
- Corollary 2: If $r \equiv s \pmod{p-1}$ then $a^r \equiv a^s \pmod{p}$
 - In other words, when working modulo a prime p , exponents can be reduced modulo $p-1$
- Easy to compute $a^{-1} \pmod{n}$
 - $a^{-1} \pmod{n} = a^{\phi(n)-1} \pmod{n}$
 - Why? $a * a^{\phi(n)-1} \pmod{n}$
 - $= a^{\phi(n)-1+1} \pmod{n}$
 - $= a^{\phi(n)} \pmod{n}$
 - $= 1$

Example of Fermat's Little Theorem

- What is the inverse of 5, modulo 7?
- 7 is prime, so $\phi(7) = 6$
- $5^{-1} \text{ mod } 7 = 5^{6-1} \text{ mod } 7$
 $= 5^5 \text{ mod } 7$
 $= (5^2 * 5^2 * 5) \text{ mod } 7$
 $= ((5^2 \text{ mod } 7) * (5^2 \text{ mod } 7) * (5 \text{ mod } 7)) \text{ mod } 7$
 $= ((4 \text{ mod } 7) * (4 \text{ mod } 7) * (5 \text{ mod } 7)) \text{ mod } 7$
 $= ((16 \text{ mod } 7) * (5 \text{ mod } 7)) \text{ mod } 7$
 $= ((2 \text{ mod } 7) * (5 \text{ mod } 7)) \text{ mod } 7$
 $= (10 \text{ mod } 7) \text{ mod } 7$
 $= 3 \text{ mod } 7$
 $= 3$

Chinese Remainder Theorem

- (Or, enough of it for our purposes...)
- Suppose:
 - p and q are relatively prime
 - $a \equiv b \pmod{p}$
 - $a \equiv b \pmod{q}$
- Then: $a \equiv b \pmod{p \cdot q}$
- Proof:
 - p divides $(a-b)$ (because $a \pmod{p} = b \pmod{p}$)
 - q divides $(a-b)$
 - Since p, q are relatively prime, $p \cdot q$ divides $(a-b)$
 - But that is the same as: $a \equiv b \pmod{p \cdot q}$

Proof that D inverts E

$$\begin{aligned} & c^d \bmod n \\ &= (m^e)^d \bmod n && \text{(definition of } c) \\ &= m^{ed} \bmod n && \text{(arithmetic)} \\ &= m^{k \cdot (p-1) \cdot (q-1) + 1} \bmod n && \text{(d inverts e)} \\ &= m^* m^{k \cdot (p-1) \cdot (q-1)} \bmod n && \text{(arithmetic)} \\ &= m \bmod n && \text{(C. R. theorem)} \\ &= m && \text{(} m < n \text{)} \end{aligned}$$

$$e \cdot d \equiv 1 \pmod{(p-1) \cdot (q-1)}$$


Finished Proof

- Note: $m^{p-1} \equiv 1 \pmod{p}$ (if p doesn't divide m)
 - Why? Fermat's little theorem.
- Same argument yields: $m^{q-1} \equiv 1 \pmod{q}$
- Implies:
 - $m^{p-1} \equiv m^{k * \phi(n)} \pmod{p}$
 - Since we are working mod p
 - and $\phi(n) = (p-1)(q-1) \equiv k * (p-1)(q-1) \pmod{p-1}$
 - $m^{k * \phi(n) + 1} \equiv m \pmod{p}$
- Similarly for q : $m^{k * \phi(n) + 1} \equiv m \pmod{q}$
- Chinese Remainder Theorem implies:
 $m^{k * \phi(n) + 1} \equiv m \pmod{n}$

So Far

- Intro to Public/Private key encryption
- Modular Mathematics
 - Euler's Totient Function
 - Fermat's Little Theorem
 - Chinese Remainder Theorem
- RSA Algorithm
- Proving Primes
 - Rabin-Miller primality test

How to Generate Prime Numbers

- Many strategies, but *Rabin-Miller* primality test is often used in practice.
 - $a^{p-1} \equiv 1 \pmod{p}$
- Efficiently checkable test that, with probability $\frac{3}{4}$, verifies that a number p is prime.
 - Iterate the Rabin-Miller primality test t times.
 - Probability that a composite number will slip through the test is $(\frac{1}{4})^t$
 - These are worst-case assumptions.
- In practice (takes several seconds to find a 512 bit prime):
 1. Generate a random n -bit number, p
 2. Set the high and low bits to 1 (to ensure it is the right number of bits and odd)
 3. Check that p isn't divisible by any "small" primes $3, 5, 7, \dots, < 2000$
 4. Perform the Rabin-Miller test at least 5 times.

Rabin-Miller Primality Test

- Is n prime?
- Write n as $n = (2^s) * r + 1$ (with odd r)
- Pick random number a , with $1 \leq a < n$ and $\gcd(a, n) = 1$ (relatively prime)
- If
 - $a^r \equiv 1 \pmod{n}$ and
 - for all j in $\{0 \dots s-1\}$, $a^{2^j r} \equiv -1 \pmod{n}$
- Then return composite
- Else return probably prime

Conclusion

- Modular Mathematics
 - Euler's Totient Function
 - Fermat's Little Theorem
 - Chinese Remainder Theorem
- RSA Algorithm
- Proving Primes
 - Rabin-Miller primality test

So Far

- Intro to Public/Private key encryption
- Modular Mathematics
 - Euler's Totient Function
 - Fermat's Little Theorem
 - Chinese Remainder Theorem
- RSA Algorithm
- Proving Primes
 - Rabin-Miller primality test