Encryption

Just because you're paranoid doesn't mean they're not out to get you.

Woody Allen

Security Risks of Internet Communication

Eavesdropping

- Intermediaries listen in on private conversations
- Solution: Encryption (public or private-key)

Manipulation

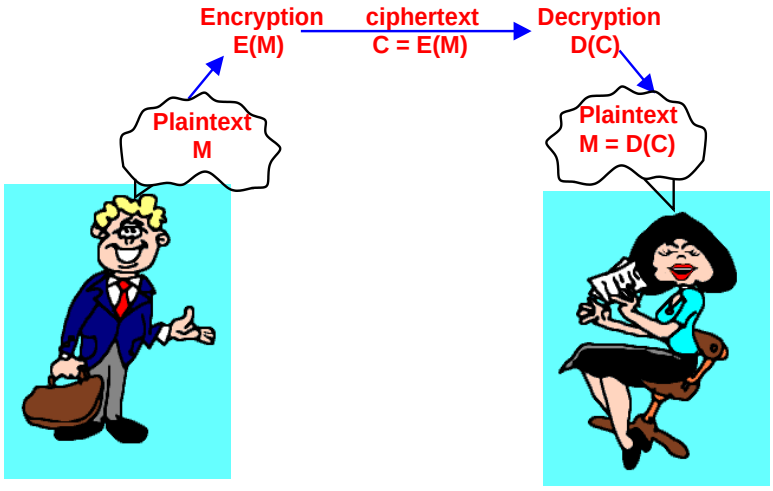
- Intermediaries change information in a private communication
- Solution: Methods for preserving data integrity (one-way hash functions and Message Authentication Codes (MACs))

Impersonation

- A sender or receiver communicates under false ID
- Solution: Authentication (digital signature, etc.)

Terminology

A sender (Bob) wants to send a message to a receiver (Alice) securely – wants to make sure no eavesdropper can read message.



plaintext – original message
encryption – process of disguising message to hide its contents
ciphertext – encrypted message
decryption – process of turning ciphertext back into plaintext
cryptography – science of keeping messages secure
cryptanalysis – science of breaking ciphertext

Background: Number-Theoretic Algorithms

Useful for public-key encryption schemes

Easy to find large primes

Difficult to factor products of large primes

Size of Inputs

- Few inputs of large integers
- Size of input = #bits

- An algorithm with integer inputs $a_1, a_2, \ldots, a_k$ is a **polynomial-time algorithm** if it runs in time polynomial in lg a_1, lg a_2, ..., lg a_k; i.e., polynomial in the lengths of the binary-encoded inputs

---

Cost of Operations

- Arithmetic on large integers takes time
- Cost is measured in terms of **bit operations**
- Multiplying two β-bit integers takes $\Theta(\beta^2)$ bit operations
- Dividing a β-bit integer by a shorter integer takes $\Theta(\beta^2)$ bit operations
- Faster methods do exist, but we will use the others in this lecture

---

Review of Number Theory

$$\text{Z} = \text{set of integers } \{\cdots, -2, -1, 0, 1, 2, \cdots\}$$
$$\text{N} = \text{set of natural numbers } \{0, 1, 2, \cdots\}$$

For two integers d and a, d | a (d **divides** a) if a = kd, k $\in$ Z. In this case, a is a **multiple** of d, and d is a **divisor** of a (if d >= 0). Every integer divides 0.

Examples: 2 | 8, 3 | 9, 2 | 10

Every integer a has the **trivial divisors** 1 and a.

Nontrivial divisors are called **factors**.

Examples: 2 is a factor of 8 and 10, 3 is a factor of 9.

An integer a > 1 with only trivial divisors is a **prime** number; otherwise, a is a **composite**. The integers $\{\cdots, -2, -1, 0, 1\}$ are neither prime nor composite. There are infinitely many prime numbers.

Division Theorem

For any integer a and positive integer n, there are unique integers q and r such that $0 \leq r < n$ and $a = qn + r$

q ($= \lfloor a/n \rfloor$) is the _____ of the division

r ($= a \bmod n$) is the _____

$a = \lfloor a/n \rfloor n + (a \bmod n)$ or
$a \bmod n = a - \lfloor a/n \rfloor n$

If $(a \bmod n) = (b \bmod n)$, then $a \equiv b \pmod n$

Example

$22 \bmod 5 =$ _
$-13 \bmod 5 =$ _

Equivalence

If $(a \bmod n) = (b \bmod n)$, then a is **equivalent** to b, modulo n, denoted $a \equiv b \pmod n$.

An **equivalence class modulo n** containing an integer a is $[a]_n = \{a + kn \mid k \in Z\}$.

Example: if a=8, n=3, then q = _, r = ___, and some b $\equiv$ a are _____.

The equivalence classes modulo n are

$[0]_3$ = _____ = $[3]_3$ = $[6]_3$

$[1]_3$ = _____

$[2]_3$ = _____

Common Divisors

If d | a and d | b, then d is a **common divisor** of a and b. The **greatest common divisor** gcd(a,b) is the largest such common divisor d.

$$gcd(a,b) = \begin{cases} 0 & \text{if } a = b = 0 \\ |\ a\ | & \text{if } |\ a\ | > 0, b = 0 \\ |\ b\ | & \text{if } |\ b\ | > 0, a = 0 \\ 1 \leq x \leq min(|\ a\ |, |\ b\ |) & \text{otherwise} \end{cases}$$

For example, some common divisors of 12 and 18 are 1, 2, 3, and 6. The greatest common divisor of 12 and 18 is 6.

Euclid's Theorem

If a and b are any integers, not both zero, then gcd(a,b) is the smallest positive element of the set {ax + by: x,y $\in$ Z} of linear combinations of a and b.

Example: gcd(9,15) = 3

9x + 15y = 3

x = 2, y = -1

Relative Primes

Two integers a and b are **relatively prime** if gcd(a,b) = 1.

Integers $n_1, n_2, \ldots, n_k$ are **pairwise relatively prime** if gcd(n_i, n_j)= 1 for all $i \neq j$.

Example: 8, 9, and 25 are pairwise relatively prime.

Unique Factorization

Theorem 33.7 For all primes p and all integers a and b, if p | ab, then p | a or p | b.

Theorem 33.8 A composite integer a can be written in exactly one way as a product of the form $a = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$, where the p_i are prime, $p_1 < p_2 < \cdots < p_r$, and the e_i are positive integers.

Examples: $675 = 3^3 * 5^2$

$1350 = 2 * 3^3 * 5^2$

$255 = 3 * 5 * 17$

Finding the gcd

Given prime factorizations of positive integers a and b,

$$a = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}, \qquad b = p_1^{f_1} p_2^{f_2} \cdots p_r^{f_r},$$

where some e_i, f_i may be 0.

Then gcd(a,b) = $p_1^{min(e_1,f_1)} p_2^{min(e_2,f_2)} \cdots p_r^{min(e_r,f_r)}$.

Example: gcd(255, 675) = $3^1 * 5^1 * 17^0 = 3 * 5 = 15$

However, factoring is not a polynomial time algorithm.

Euclid's Algorithm

For any non-negative integer a and any positive integer b, gcd(a, b) = gcd(b, a mod b)

```
Euclid(a,b)              ; second argument is strictly decreasing
1     if b = 0
2     then return a
3     else return Euclid(b, a mod b)
```

Example

```
Let a = 2322, b = 654.
2322 = 654*3 + 360          gcd(2322,654) = gcd(654,360)
654 = 360*1 + 294           gcd(654,360) = gcd(360,294)
360 = 294*1 + 66            gcd(360,294) = gcd(294,66)
294 = 66*4 + 30             gcd(294,66) = gcd(66,30)
66 = 30*2 + 6               gcd(66,30) = gcd(30,6)
30 = 6*5                    gcd(30,6) = 6
                            gcd(6,0) = 6 (Elementary property of
Therefore, gcd(2322,654) = 6.
```

Analysis

For any integer $k \geq 1$, if $a > b \geq 0$ and $b < F_{k+1}$, then Euclid(a,b) makes fewer than k recursive calls.

Remember that $F_k = F_{k-1} + F_{k-2}$, $F_k \approx \frac{\phi^k}{\sqrt{5}}$, $\phi = \frac{1+\sqrt{5}}{2}$

$$b < F_{k+1} = \frac{\phi^{k+1}}{\sqrt{5}},\ b < \frac{\phi\phi^k}{\sqrt{5}}$$

$$\phi^k > \frac{\sqrt{5}}{\phi}b,\ k > log_\phi \frac{\sqrt{5}}{\phi}b$$

k = O(lg b) recursive calls

$O(\beta)$ arithmetic operations
$O(\beta^3)$ bit operations

Extended Euclid

Since gcd(a,b) = ax + by, x, y $\in$ Z, finding x and y will be useful for computing modular multiplicative inverses

Extended-Euclid(a,b)

if b = 0
then return(a, 1, 0)
(d', x', y') = Extended-Euclid(b, a mod b)
(d, x, y) = (d', y', x' - $\lfloor \frac{a}{b} \rfloor$y')
return(d, x, y)

Running time same as Euclid algorithm.

Example

```
(d, x, y) = Extended-Euclid(6, 3)
   (d', x', y') = Extended-Euclid(3, 0)
                = (3, 1, 0)
   d = 3, x = 0, y = 1 - 6/3 * 0 = 1
   = (3, 0, 1)

d = 3, x = 0, y = 1

6x + 3y = d
6*0 + 3*1 = 3
```

Correctness of Extended-Euclid

$$\begin{aligned} &\text{d' = gcd(b, a mod b)} \\ &\text{= bx' + (a mod b)y'} \quad \text{; Euclid's Theorem} \\ &\text{= d = gcd(a, b)} \quad \text{; Euclid's Algorithm} \end{aligned}$$

$$\begin{aligned} &\text{d = bx' + (a - } \lfloor \tfrac{a}{b} \rfloor \text{b) y'} \quad \text{; Division Theorem} \\ &\text{= ay' + b(x' - } \lfloor \tfrac{a}{b} \rfloor \text{y')} \quad \text{; Rearrange terms} \end{aligned}$$

$$\begin{aligned} \text{d = d' = ax + by} \\ \text{x = y'} \\ \text{y = x' - } \lfloor \tfrac{a}{b} \rfloor \text{y'} \end{aligned}$$

Modular Arithmetic

$$\begin{aligned} \text{(a mod n) + (b mod n) = (a + b) mod n} \\ \text{(a mod n) * (b mod n) = ab mod n} \\ \text{a}^{-1} \text{ mod n = b} \leftrightarrow \text{ab mod n = 1 (inverse)} \end{aligned}$$

Uses identities $a^{2c} mod n = (a^c)^2 mod n$ and $a^{2c+1} mod n = a * (a^c)^2 mod n$.

Solving Modular Linear Equations

$$\text{ax} \equiv \text{b (mod n)}$$

Given a, b, n > 0; find x.

Let d = gcd(a, n)

Solvable iff d | b

Theorem 33.23 If d | b and d = ax' + ny' (as computed by Extended-Euclid) then one solution is x_0 = x'(b/d) mod n.

Theorem 33.24 Given one solution x_0, there are exactly d distinct solutions, modulo n, given by $x_i = x_0 + i(n/d)$ for i = 0, 1, 2, . . ., d-1.

Pseudocode

```
ModularLinearEquationSolver(a, b, n)      ; O(lgn + gcd(a,n))
    (d, x', y') = Extended-Euclid(a, n)   ; arithmetic operations
    if (d | b)
    then x_0 = x'(b/d) mod n
        for i = 0 to d-1
            print (x_0 + i(n/d)) mod n
    else print "no solutions"
```

Note: Solving ax $\equiv$ 1 (mod n) gives a^{-1} mod n (only one solution).

Chinese Remainder Theorem

Find integers x that leave remainders 2, 3, 2 when divided by 3, 5, 7, respectively. [Sun-Tsu, 100 A.D.]

Theorem 33.27 Let $n = n_1 n_2 \cdots n_k$, where n_i are pairwise relatively prime and consider the correspondence

$$a \leftrightarrow (a_1, a_2, \ldots, a_k),$$

where $a \in Z_n$, $a_i \in Z_{n_i}$, and a_i = a mod n_i for i = 1, $\cdots$,k.

Chinese Remainder Theorem (33.27 cont.)

If $a \leftrightarrow (a_1, a_2, \cdots, a_k)$ and $b \leftrightarrow (b_1, b_2, \cdots, b_k)$

Then

(a+b) mod n $\leftrightarrow$ $((a_1 + b_1) \bmod n_1, \ldots, (a_k + b_k) \bmod n_k)$

(a-b) mod n $\leftrightarrow$ $((a_1 - b_1) \bmod n_1, \ldots, (a_k - b_k) \bmod n_k)$

ab mod n $\leftrightarrow$ $(a_1 b_1 \bmod n_1, \ldots, a_k b_k \bmod n_k)$.

From $a \longrightarrow (a_1, \cdots, a_k)$

(a mod n_i, $\cdots$, a mod n_k)

From $a_1, \cdots, a_k \longrightarrow a$

$m_i = n/n_i$ for i = 1, ..., k

$c_i = m_i(m_i^{-1} \bmod n_i)$

$a \equiv (a_1 c_1 + \cdots + a_k c_k) \pmod{\text{n}}$

Example

Given $a \equiv 2 \pmod 5$

$a \equiv 3 \pmod{11}$

Find $a \equiv x \pmod{55}$

$$a_1 = 2,\ a_2 = 3$$

$$m_1 = 11,\ m_2 = 5$$

$$n_1 = 5,\ n_2 = 11$$

$$m_1^{-1} \bmod n_1 = 11^{-1} \bmod 5 = 1$$

$$m_2^{-1} \bmod n_2 = 5^{-1} \bmod 11 = 9$$

$$c_1 = m_1(m_1^{-1} \bmod n_1) = 11(1) = 11$$

$$c_2 \;=\; m_2(m_2^{-1} \bmod n_2) = 5(9) = 45$$

$$\begin{aligned} \text{a} &\equiv 2^*11 + 3^*45 \pmod{55} \\ &\equiv 22 + 135 \pmod{55} \\ &\equiv 157 \pmod{55} \\ &\equiv 47 \pmod{55} \end{aligned}$$

Thus, we can work in modulo n or modulo n_i.

Corollary 33.29

If $n_1,\ n_2,\ \cdots\ n_k$ are pairwise relatively prime and $n \;=\; n_1 n_2 \cdots\ n_k$, then for all integers x and a $x \;\equiv\; a \bmod n_i$ iff $x \;\equiv\; a \bmod n$.

Euler's phi function

Euler's phi function ϕ(n) is the size of $Z_n^* = \{[a]_n \;\in\; Z_n$: gcd(a,n) = 1$\}$, the multiplicative group mod n. ϕ(p) = p-1 if p is prime.

Euler's Theorem

For any integer n > 1, $a^{\phi(n)} \;\equiv\; 1 \pmod{\text{n}}$ for all $a \in Z_n^*$.

Fermat's Theorem

If p is prime, then $a^{p-1} \;\equiv\; 1 \pmod{\text{p}}$ for all $a \in Z_p^*$.

Repeated Squaring

Compute: a^b mod n, where a and b are nonnegative integers and n is a positive integer.

Let $b = < b_k, b_{k-1}, \ldots, b_1, b_0 >$.

Compute a^c mod n by doubling c for each i and incrementing c when b_i= 1.

Pseudocode

Modular-Exponentiation(a, b, n)

c = 0
d = 1
let $b = < b_k b_{k-1}, \ldots, b_1, b_0 >$ be binary encoding of b
for i = k downto 0
c = 2c
d = (d*d) mod n
if $b_i = 1$
then c = c + 1
d = (d*a) mod n return d

`Modular-Exponentiation(a=5, b=501, n=6)  b = 111110101`

```
i = 8   | 7   | 6   | 5     | 4     |  3   |   2     |   1   |
c = 0 1 | 2 3 | 6 7 | 14 15 | 30 31 | 62 - | 124 125 | 250 - |
d = 1 5 | 1 5 | 1 5 |  1  5 |  1  5 |  1 - |   1   5 |   1 - |
```

Analysis

If a, b, n are β-bit numbers, there are $O(\beta)$ arithmetic operations and $O(\beta^3)$ bit operations.

Encryption - Symmetric Cryptography

Private Key

Alice and Bob share a key K the adversary does not know

Alice and Bob agree on cryptosystem and key

Bob encrypts plaintext using key, sends ciphertext to Alice

Alice decrypts ciphertext with same key and reads the message

Advantage: fast

Disadvantage: keys must be distributed secretly

Disadvantage: if key is compromised, all is lost

Disadvantage: number of needed keys is n^2

Encryption - Public-Key Cryptography

Encryption key is public

Decryption key is private (secret)

Private key cannot be calculated from public key in a reasonable amount of time

---

RSA Public-Key Cryptosystem

- Rivest, Shamir, and Adleman, 1977
- Most commonly used encryption and authentication algorithm today
- Used in Netscape, Microsoft browsers, Internet and computing standards
- Send encrypted messages
- Append unforgeable digital signature
- Based on ease of finding large primes and difficulty of factoring their products

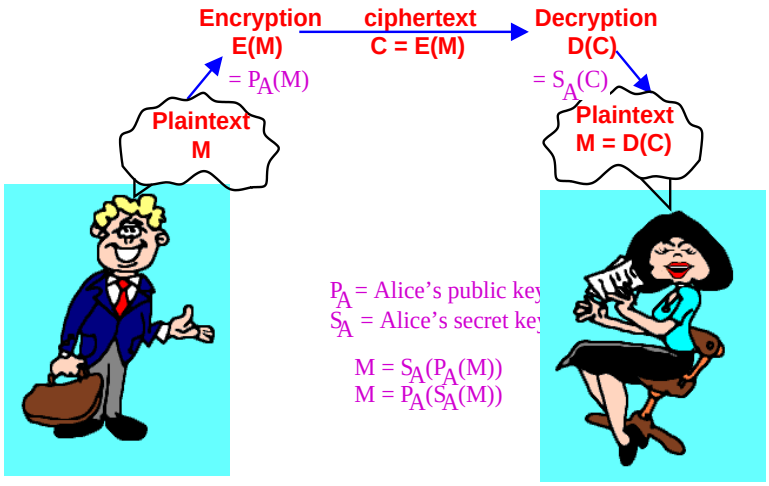
---

Public Key Cryptography

- Each participant has
  - public key – released to others
  - secret key – kept secret
  - Example Alice (P_A, S_A), Bob (P_B, S_B)
- Public and Secret functions are inverses
  - $\mathrm{M} = S_A(P_A(\mathrm{M}))$
  - $\mathrm{M} = P_A(S_A(\mathrm{M}))$
- Must be able to reveal P_A while S_A remains uncomputable (or at least very difficult to compute).
- Security depends on method of computing keys

- RSA – factoring large integers
- McEliecee – decoding linear code (NP-Complete)
- El Gamal – discrete logarithm problem
- Chor-Rivest – knapsack (NP-Complete)

Protocol for Sending Encrypted Message M



- Bob looks up Alice's public key P_A.
- Bob computes ciphertext message C = P_A(M) for his original message M.
- Bob sends C to Alice (eavesdroppers do not have S_A).
- Alice computes S_A(C) = S_A(P_A(M)) = M.

Protocol for Sending a Signed Message M'

- Alice computes digital signature $\sigma = S_A$(M').

- Alice sends (M', σ) to Bob.
- Bob checks that M' = $P_A(\sigma) = P_A(S_A(\text{M'})) = \text{M'}$.
- Message M' is not encrypted.

Protocol for Sending a Signed, Encrypted Message M

- Bob computes digital signature $\sigma = S_B(\text{M})$, and creates new message $M' = < M, \sigma >$.
- Bob computes C = P_A(M') and sends C to Alice.
- Alice computes $< M, \sigma > = S_A(\text{C})$ and then verifies signature using M = $P_B(\sigma)$.

RSA Cryptosystem

Public and secret keys are created as follows:

1. Select at random two large prime numbers p and q (say >100 decimal digits each).
2. Compute n = pq.
3. Select a small odd integer e that is relatively prime to $\phi(n)$ = (p-1)(q-1).
4. Compute d = $e^{-1} \text{mod } \phi(\text{n})$ (multiplicative inverse).
5. Publish pair P = (e, n) as RSA Public Key.

6. Keep pair S = (d, n) as RSA Secret Key.
7. P(M) = M^e (mod n)
8. S(C) = C^d (mod n)

Example of RSA Encryption

1. p = 41, q = 59
2. n = pq = 2419
3. ϕ(n) = (p-1)(q-1) = 40*58 = 2320
   Find e such that gcd(e, 2320) = 1 and e is small and odd
   e = 3 works
4. d = e^{-1} mod ϕ(n)
   = 3^{-1} mod 2320
   d = 1547
   d*e mod ϕ(n) = 1547*3 mod 2320 = 1
5. P = (e, n) = (3, 2419)
6. S = (d, n) = (1547, 2419)
   P(M) = M^3 (mod 2419)
   S(M) = M^{1547} (mod 2419)
   Note: Only 2419 different messages are possible.

Implementing RSA

- Bob generates two large primes, p and q
  Probabilistic primality testing $O((lgn)^3)$
- Bob computes n = pq and ϕ(n) = (p-1)(q-1)
- Bob chooses random e ($1 < e < \phi(n)$) such that gcd(e, ϕ(n)) = 1
  Euclidean algorithm
- Bob computes d = e^{-1} mod ϕ(n)
  Extended Euclidean Algorithm $O((lgn)^2)$
- Bob publishes n and e in a directory as his public key

RSA Computation

Using public key P = (e, n) to transform messages M:

$$P(M) \ = \ M^e (mod\ n)$$

Using secret key S = (d, n) to transform ciphertext C:

$$S(C) \ = \ C^d (mod\ n)$$

Use Modular-Exponentiation:
If $| e | = O(1)$, $| d | = | n | = \beta$
Then Public Key requires O(1) modular multiplications, $O(\beta^2)$ bit operations
Secret key requires $O(\beta)$ modular multiplications
$O(\beta^3)$ bit operations.

Correctness

P(S(M)) = $(M^d mod n)^e$ mod n = $M^{ed} mod\ n$
S(P(M)) = $(M^e mod n)^d$ mod n = $M^{ed} mod\ n$
Since d = e^{-1} mod [(p-1)(q-1)]
Then ed = 1 + k(p-1)(q-1)

If M $\equiv$ 0 (mod p), then $M^{ed} \equiv M(mod\ p)$
If M $\not\equiv$ 0 (mod p), then
$M^{ed} \equiv M(M^{p-1})^{k(q-1)}(mod\ p)$
$\equiv M(1)^{k(q-1)}(mod\ p)$ Fermat's Theorem
$\equiv M(mod\ p)$.
Similarly for q, thus
$M^{ed} \equiv M \quad (mod\ p)$
$M^{ed} \equiv M \quad (mod\ q)$
p and q are prime, n = pq.

Thus, by the Corollary to the Chinese Remainder Theorem, $M^{ed} \equiv M(mod\ n)$.

If the adversary can factor n into p and q, then the code is broken, but this is hard.

Primality Testing

Finding large primes.

Density Of Primes

The **prime distribution function** π(n) specifies number of primes $\leq$ n.

Theorem 33.37 $n \overset{lim}{\to} \infty \frac{\pi(n)}{n/lnn} = 1$

$\pi(n) \approx \frac{n}{lnn}$

For example, $\pi(10^9) \approx$ 48,254,942.

The probability that randomly-chosen n is prime $\approx \frac{1}{lnn}$. Thus, try $\frac{lnn}{2}$ odd numbers near n to find a prime with high probability.

For example, 100-digit number
$ln10^{100} \approx 230$. Try 115 odd numbers near 10^{100}.

About 1/230 100-digit numbers are prime.

Break input message M into numerical blocks smaller than n.

Trial Division

Try all odd numbers $3, \ldots, \sqrt{n}$ to test n for primality.

Running Time $\Theta(\sqrt{n})$, but $\beta = \lceil lg(n+1) \rceil \longrightarrow \sqrt{n} = \Theta(2^{\frac{\beta}{2}})$ (exponential).

This works well only for small n.

Pseudo

Z+ = nonzero elements of $Z_n = \{1, 2, \ldots, \text{n-1}\}$.

By Fermat's Theorem, if n is prime, then $a^{n-1} \equiv 1 (mod\ n)$ for every $a \in Z_n^+$.

If some a violates, then n is composite.

- Pseudo test tries formula for a=2. If satisfied, declare n prime.
- Does not always work, but the numbers errantly declared prime (base-a pseudoprimes) are rare.

- Carmichael Numbers are composites that satisfy formula for all $a \in Z_n^*$. Very rare.
- Miller-Rabin randomized primality test overcomes this deficiency in the pseudo test (tries random "a"s).

Integer Factorizations

- Trial division by all integers up to B to factor number up to B^2
- Pollard-Rho factors numbers up to B^4 (usually)
  - Works well in practice on numbers with **small** factors
  - Analysis: $\Theta(\sqrt{p})$ to find factor p
    To factor β-bit composite number n
    Try all prime factors $< \lfloor\sqrt{n}\rfloor$
    The run time is $n^{\frac{1}{4}} = 2^{\frac{\beta}{4}}$ arithmetic operations
    $n^{\frac{1}{4}}\beta^3 = 2^{\frac{\beta}{3}}$ bit operations

---

Hybrid Cryptosystems

In practice, public-key crypto used to secure and distribute session keys, which are then used with private-key crypto to secure message traffic.

Applications