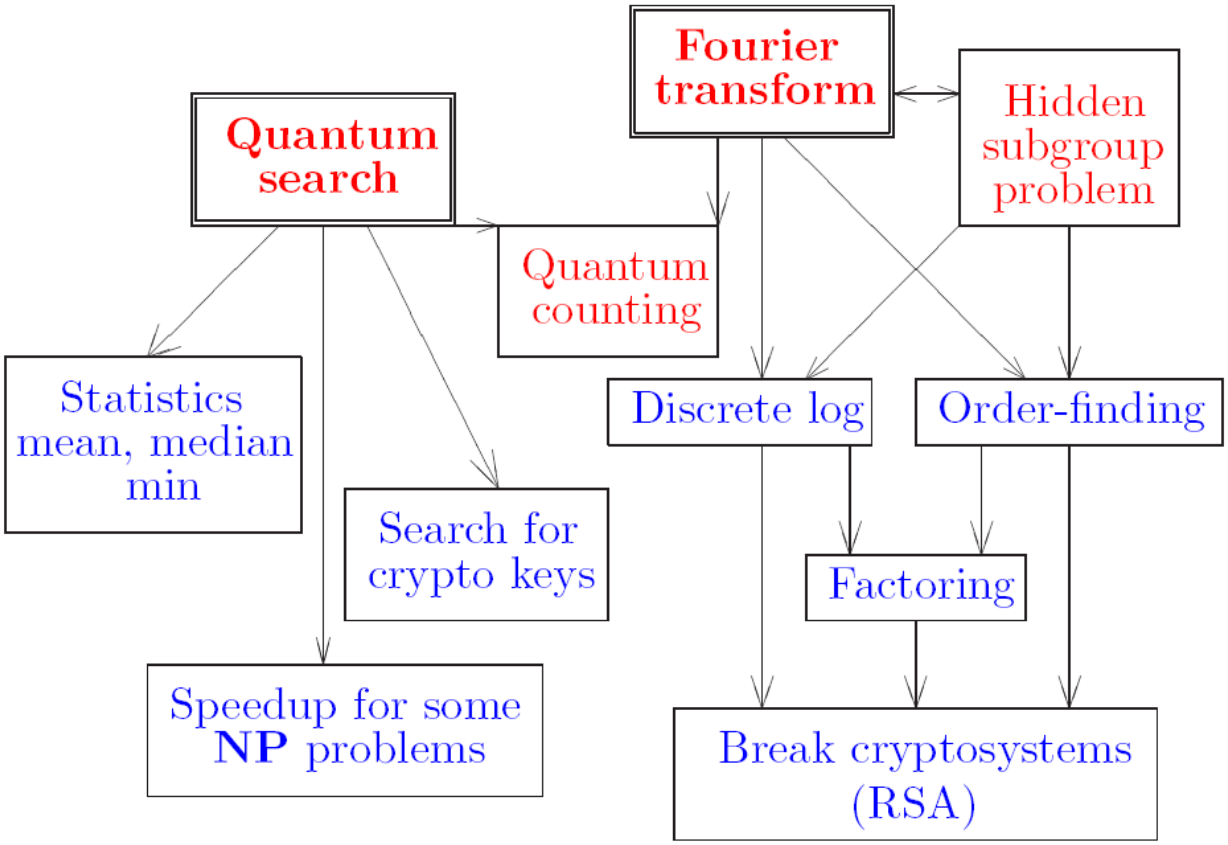
Lecture 13 Algorithms and cryptography



Note: The complexity class NP
Problems in NP: yes answer can be easily verified with the aid of an appropriate witness.

Classical factoring algorithm

Modular arithmetic

Arithmetic of remainders (p.626 of the textbook)

$18 = 2 \cdot 7 + 4$ $\qquad$ $18 \pmod 7$

$=$

For any positive integers x & n

$$\boxed{x = kn + r} \leftarrow \text{unique representation}$$

k is a non-negative integer and $0 \le r \le n-1$.

Modular arithmetic = ordinary arithmetic in which we pay attention to remainders only. Notation (mod n) is used to indicate that we are working in modular arithmetic.

Class exercise: Prove that 2=5=8=11 (mod 3)

x	$=$	$k \cdot n$	$+$	r
2	=	$0 \cdot 3$	+	2
5	=	$1 \cdot 3$	+	2
8	=	$2 \cdot 3$	+	2
11	=	$3 \cdot 3$	+	2

↑ mod 3 ↑ all the same

Class exercise: calculate $7^n \bmod 15$, $n = 1, 2, 3, 4$

$n=1$	$7^1 = 7 = 15 \cdot 0 + 7$	$\Rightarrow$	$7^1 \bmod 15 = 7$
$n=2$	$7^2 = 49 = 15 \cdot 3 + 4$	$\Rightarrow$	$7^2 \bmod 15 = 4$
$n=3$	$7^3 = 343 = 15 \cdot 22 + 13$	$\Rightarrow$	$7^3 \bmod 15 = 13$
$n=4$	$7^4 = 2401 = 15 \cdot 160 + 1$	$\Rightarrow$	$7^4 \bmod 15 = 1$

Classical factoring algorithm: How to factor 15?

(1) Pick a number less than 15 (for example 7).

(2) Calculate $7^n \bmod 15$

n	7^n	$15 \times m$	$7^n \bmod 15$
1	7	0	7
2	49	45	4
3	343	330	13
4	2401	2400	1
5	16807	16800	7

$\leftarrow R = 4$

pattern 7 4 13 1 is repeated

Question: is it a coincidence that pattern repeated itself after we got 1?
If you think that it is not, look for proof.

$7^5 = 7^4 \cdot 7 = (2400 + 1) \cdot 7 = (15 \cdot 160 + 1) \cdot 7$

$= 15 \cdot 160 \cdot 7 + 7 \Rightarrow 7^5 \bmod 15 = 7$ and so on...

The point of calculating $7^n \bmod 15$ was to find period R. This is the step that is hard for classical computers for large n.

3) Calculate greatest common divisor $\gcd\{7^{R/2} \pm 1, 15\}$

$7^{R/2} - 1 = 48; \quad \gcd\{48, 15\} = 3$

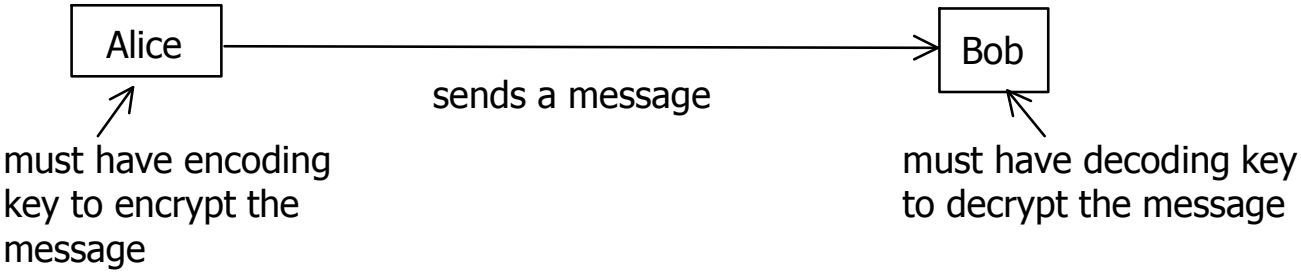
$7^{R/2} + 1 = 50; \quad \gcd\{50, 15\} = 5$

There is an efficient classical algorithm for finding gcd. See pages 627-629 of the textbook for description [Euclid's algorithm].

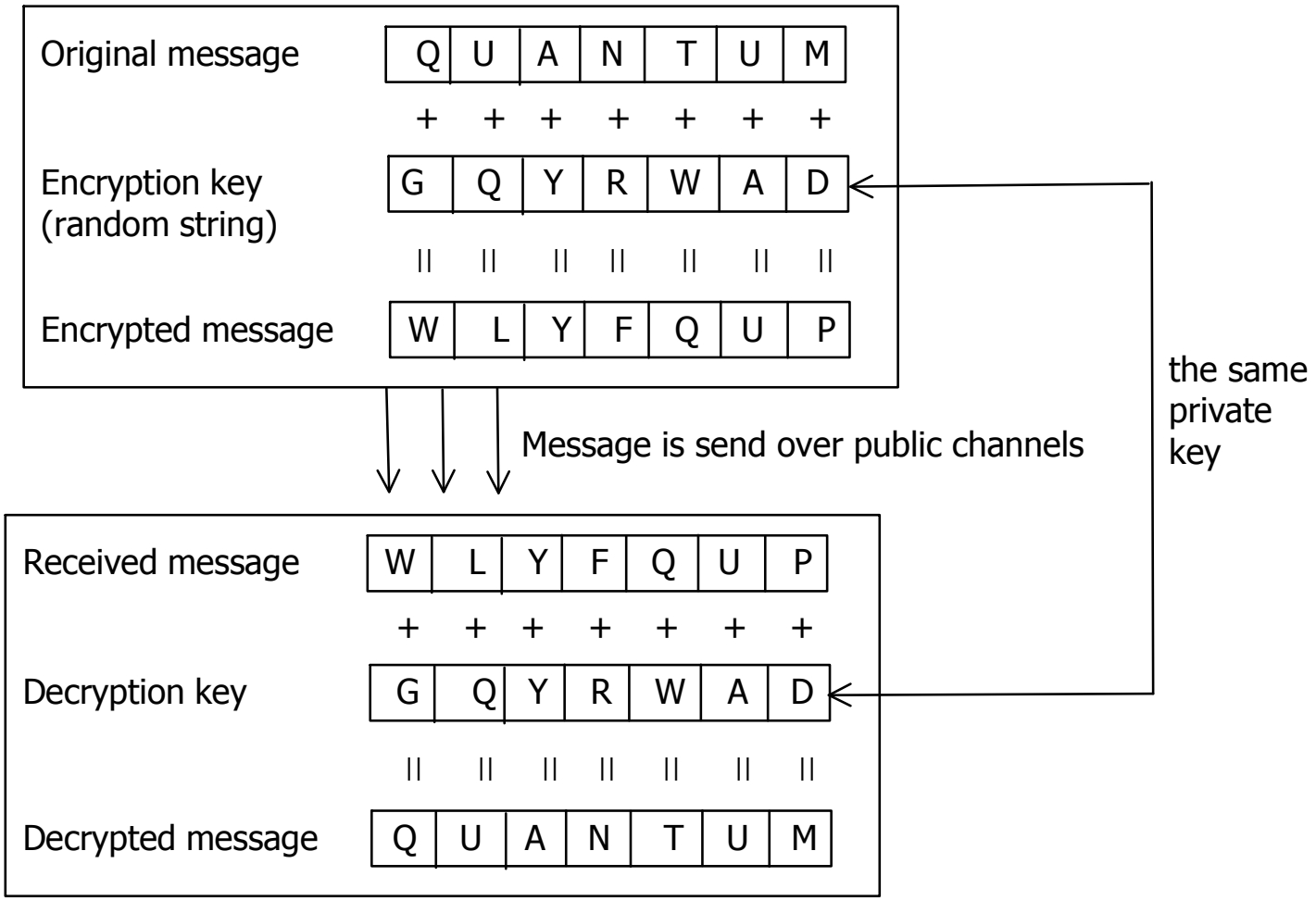
Cryptography (p.582-583 of the textbook)

Cryptography: art of enabling two parties to communicate in private

Private key cryptography



Example: Vernam cipher (or a one time pad)



Great feature: as long as the key string is a secret and in used ones, it is secure.

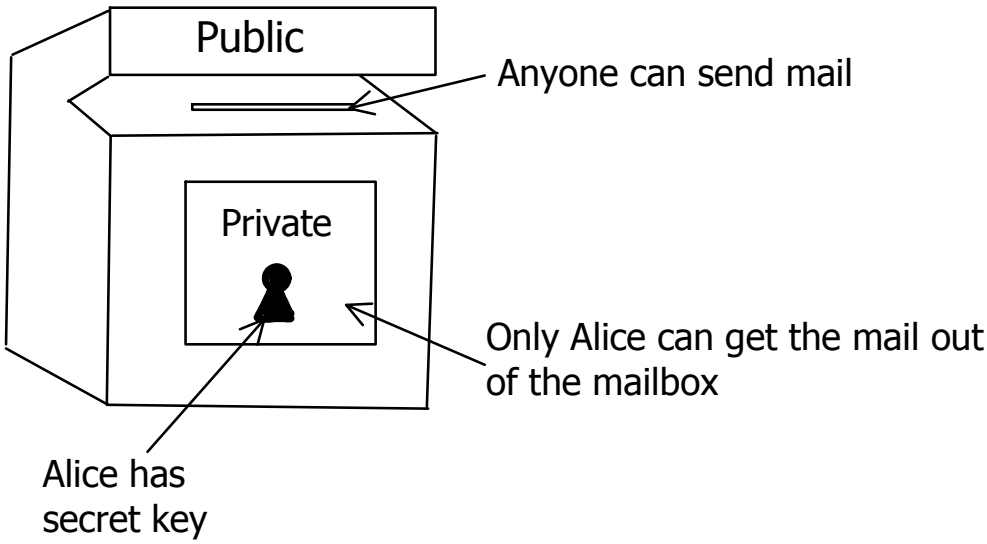
Problem: Secure key distribution. Vernam cipher is secure as long as the number of key bits is at least as large as the size of the message encoded as keys can not be reused! Key bits must be delivered in advance, guarded, and then destroyed.
Solution: quantum key distribution!

Public key distribution

RSA cryptosystems

Basic idea of public key cryptosystems (much like a mailbox)

Alice sets a mailbox.
Public key is available
to the public



Result: anyone in the world can communicate with Alice privately.
Note: there are two distinct keys; a public key and a private key (which only Alice has).

How does it work?

Suppose Bob wishes to send private message to Alice.

(1) Alice generates two keys, a public key (P) and a secret (private) key (S).

(2) Bob gets a copy of a public key (P).

(3) Bob encrypts the message using P. Encryption stage is very difficult to reverse!
Like a trap door for the mail: if you put in your mail you can not get it out.
Bob sends the encrypted message.

(4) Alice uses a secret key to decrypt the message.

Problem: There is no known scheme which is proven to be secure. It is just widely believed that it is!

In order to discuss how RSA encryption actual works, we need more modular arithmetic.

More on modular arithmetic

Multiplicative inverse

$$ab = 1 \pmod{n}$$

When does has a multiplicative inverse mod n?

Example: a

$$2 \pmod 5 = 2$$
$$3 \pmod 5 = 3$$
$$\boxed{2 \cdot 3 = 6 \pmod 5 = 1}$$

Let n be an integer greater than 1. An integer a has a multiplicative inverse modulo n if and only if **gcd (a, n)=1**, that is a and n are co-prime (or relatively prime). They must have no prime factors in common.

Let a and b be integers and let r be the remainder when a is divided by b. Then, provided that $r \neq 0$

$$\gcd(a, b) = \gcd(b, r) \quad \text{if} \quad a = n \cdot b + r$$

Example: find $\gcd(6825, 1430)$

Euclid's algorithm

$$\underset{a}{6825} = 4 \times \underset{b}{1430} + \underset{r}{1105}$$
$$1430 = 1 \times 1105 + 325$$
$$1105 = 3 \times 325 + 130$$
$$130 = 2 \times 65$$

$$\gcd(6825, 1430)$$
$$= \gcd(1430, 1105)$$
$$= \gcd(1105, 325)$$
$$= \gcd(325, 130)$$
$$= \gcd(130, 65) = 65$$

Therefore, $\boxed{\gcd(6825, 1430) = 65}$

Note: in our factoring 15 notes we needed $\gcd(48,15)$

$\gcd(48,15) = \gcd(15,3) = 3 \quad (48 = 3\times 15 + 3)$

$\gcd(50,15) = \gcd(15,5) = 5 \quad (50 = 3\times 15 + 5)$

An adaptation of Euclid's algorithm may be used to find x and y such as

$$ax + by = \gcd(a,b)$$

How is it relevant to our problem of multiplicative inverse?

Suppose we want to find multiplicative inverse of 17 mod 24:

$$17 \times a = 1 \bmod 24$$

multiplicative inverse

What we want to find is $17 \times a = 24 \times b + 1$

$a = 17, \quad b = 12$

$17 \times 17 = 24 \times 12 + 1 \Rightarrow$

multiplicative inverse of 17 mod 24 is 17

$65 = 325 - 2\times 130 = 325 - 2\times(1105 - 3\times 325)$

$= -2\times 1105 + 7\times 325 = -2\times 1105 + 7(1430 - 1105)$

$= 7\times 1430 - 9\times 1105 = 7\cdot 1430 - 9(6825 - 4\times 1430)$

$= -9\cdot \underset{a}{6825} + 43\times \underset{b}{1430}$

The Mathematical Guts of RSA Encryption

(http://fringe.davesource.com/Fringe/Crypt/RSA/Algorithm.html)

The RSA algorithm was invented in 1978 by Ron Rivest, Adi Shamir, and Leonard Adleman.

1. Find *P* and *Q*, two large (e.g., 1024-bit) prime numbers.

   Example:
   P = 61 <= first prime number (destroy this after computing E and D)
   Q = 53 <= second prime number (destroy this after computing E and D)
   PQ = 3233 <= modulus (give this to others)

2. Choose *E* such that *E* is less than *PQ*, and such that *E* and *(P-1)(Q-1)* are *relatively prime*, which means they have no prime factors in common. *E* does not have to be prime, but it must be odd. *(P-1)(Q-1)* can't be prime because it's an even number.

   E = 17 <= public exponent
   Your ***public key*** is the pair *(PQ, E)* (give it to others).

3. Compute *D* that is *multiplicative inverse* of *E modulo (P-1)(Q-1)*.
   DE = 1 mod (P-1)(Q-1)

   D = 2753 <= secret (private) exponent.
   Your ***private key*** is the number *D* (reveal it to no one).

4. The encryption function is *encrypt(T) = (T^E) mod PQ*, where *T* is the plaintext (a positive integer) and '^' indicates exponentiation.

   In our example the encryption function is:
   encrypt(T) = (T^E) mod PQ= (T^17) mod 3233
   To encrypt the plaintext value **123**, do this:
   encrypt(123) = (123^{17}) mod 3233= 337587917446653715596592958817679803 mod 3233= 855

5. The decryption function is *decrypt(C) = (C^D) mod PQ*, where *C* is the ciphertext (a positive integer) and '^' indicates exponentiation.

   In our example the decryption function is:
   decrypt(C) = (C^D) mod PQ= (C^2753) mod 3233

   To decrypt the ciphertext value 855, do this: decrypt(855) = (855^2753) mod 3233= 123

You can publish your public key freely, because there are no known easy methods of calculating *D*, *P*, or *Q* given only *(PQ, E)* (your public key).