Panel 1

Public Key Encryption

Caesar's Cypher: Replace each letter by another one!

Clear text: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Cipher text: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Example: Encode "HELLO": KHOOR
Decode "ZKDW XS" WHAT UP

Symmetric (Key) Encryption: security depends on key to be secret, and on how often/long text it is used.

1

Panel 2

Cracking Caesar's Cypher

① Brute force: 26! = 403 291 461 126 605 635 584 000 000

② Steal the key!

③ Frequency analysis: letter that occurs most often: e
common one-letter words: a, I
2-letter phrases: no, on, up, ...

Works best on long text without changing key!

2

Panel 3

Prevent cracking the key:

① Brute Force: IBM Blue Gene : 360 Tera Flops

360 000 000 000 000 FLOPS

26! / 360 TFLOPS = 37 000 years

② Keep key secret.

3

Panel 4

③ Frequency Analysis: change Key often!

So: Symmetric key encryption is as good as the key is secret!

Need to have secure way to exchange keys!

Problem!

4

Panel 5

Public Key Encryption

Public + Private key.

Encrypt with public key

Decrypt with private key

5

Panel 6

Background:

$f: X \to$ is invertible if $\exists\ g : Y \to X$ s.t. $g \circ f = id$

Opposite is called One-way function: $f(x)$ is easy to find, but $f^{-1}(x)$ is hard

Trap door One-Way Function: one-way function s.t. it is easy to invert it with extra info

6

Panel 7

Public Key Encryption is based on the existence of a

RSA Crypto Algorithm

(e, d) pair
e = public key
d = private key

Let R be encryption function. Need for M = message
R(e, M) = C and R(d, C) = M

7

Panel 8

RSA Algorithm

Pick two large prime numbers p and q

① Compute n = p · q one-way: mult. is easy
fact. is hard

② Select (small) e relatively prime to (p-1)(q-1)
→ (e, n) public key!

③ Mult. inv. of e mod (p-1)(q-1) called d
→ (d, n) private key

8

Panel 9

I have (e,n) = public key
(d,n) = private key
me to Alice!

Want to exchange message with Alice:
Alice generates (e,n) and (d,n).
Gives me (e,n), her public key!
I encode message M:
$M^e \bmod n = C$
Alice: $C^d \bmod n = M$

9

Panel 10

Example: $p = 11,\ q = 13$

$n = p \cdot q = 143$ $\quad (p-1)(q-1) = 10 \cdot 12 = 120$

e rel. prime with $120 = 2^3 \cdot 3 \cdot 5$, $e = 7$
$(e,n) = (7, 143)$ is public key!

d s.t. $d \cdot e \bmod 120 = 1$
$d \cdot 7 \bmod 120 = 1$
$d \cdot 7 = 121, 241, 361, 481, 601, 721$
$d = 103$ ✓

$(103, 143)$ is private key!

10

Panel 11

We have: p=11, q=13, n=143, e=7, d=103

Encrypt, say M=42.

$42^7 \bmod 143 = 81$

$81^{103} \bmod 143 = 42$

Decrypt:

11

Panel 12

Example: I want to send secret message to Danielle.

$n = 7 \cdot 11 = 77$ $\quad 6 \cdot 10 = 60 = 2^2 \cdot 3 \cdot 5$

$e = 7 \Rightarrow (7, 77)$ public key

$d \cdot e \bmod 60 = 1$

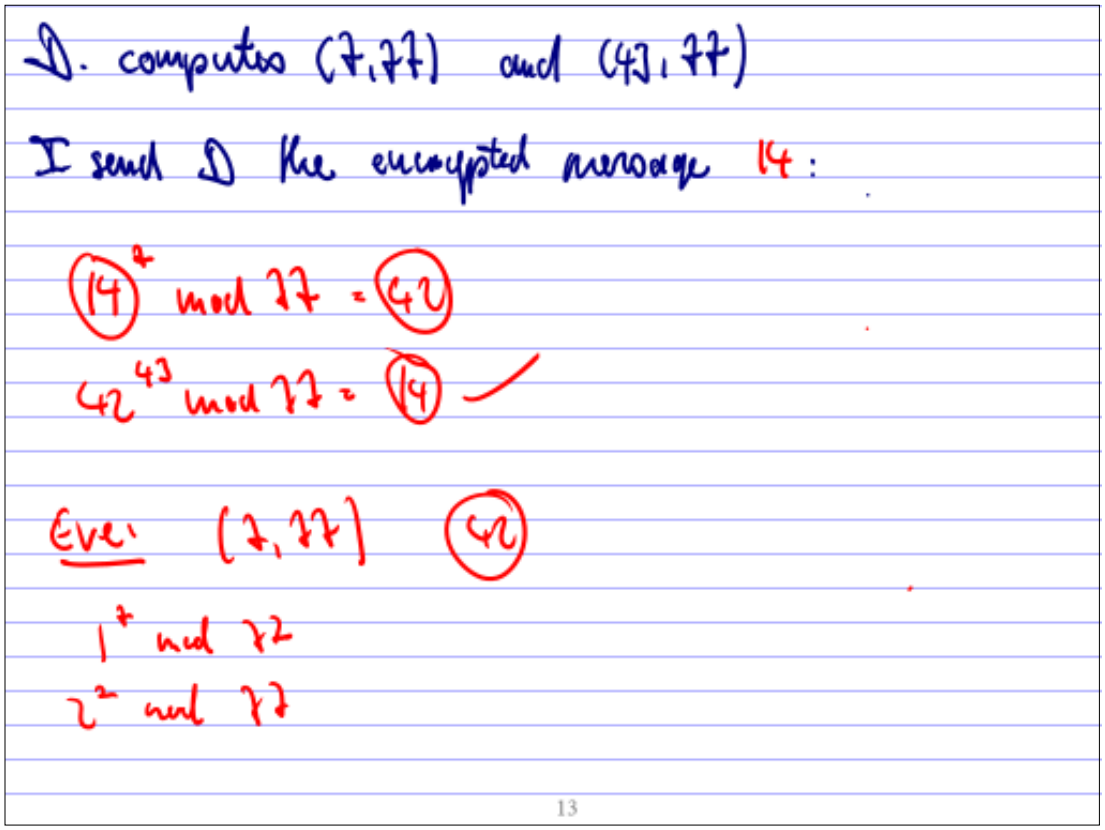
$d \cdot 7 \bmod 60 = 1$: 61, 121, 181, 241, 301

$d = 301/7 = 43$

(43, 77) private key.

12

Panel 13



Panel 14

Here is another example of RSA encryption and decryption. The parameters used here are artificially small

Choose two distinct prime numbers, such as p = 61 and q = 53. Compute n = pq giving

n = 3233

Compute the product (p − 1)(q − 1) = 3120

Choose any number 1 < e < 3120 that is coprime to 3120. Choosing a prime number for e leaves us only to check that e is not a divisor of 3120.

e = 17

Compute d, the modular multiplicative inverse of e mod 3120 yielding

d = 2753

The public key is (n = 3233, e = 17).
The private key is (n = 3233, d = 2753).

For instance, in order to encrypt m = 65, we calculate

65^17 mod 3233 = 2790

To decrypt c = 2790, we calculate

2790^2753 mod 3233 = 65

In real-life situations the primes selected would be much larger; in our example it would be trivial to factor n, 3233 (obtained from the freely available public key) back to the primes p and q. Given e, also from the public key, we could then compute d and so acquire the private key.

14

Panel 15

How do I get someone's public key?

Deposit public keys with

"Certification Authority"

15

Panel 16

Thus behind RSA Algorithm

$((a \bmod n) + (b \bmod n)) \bmod n = (a+b) \bmod n$

$((a \bmod n) \cdot (b \bmod n)) \bmod n = (a \cdot b) \bmod n$

Fermat's Little Thm: If p is prime,

$a^{p-1} = 1 \bmod p$

Chinese Remainder Thm: If p, q are prime then

$x = a \bmod p$ and $x = a \bmod q$

$\Rightarrow$ $x = a \bmod pq$

16

Panel 17

Why does this work? Have (e,n) and (d,n) s.t.
$R(e,M) = C$ and $R(d,C) = M$

$\Rightarrow R(d, R(e,M)) = M$

$(M^e)^d = M^{ed} \bmod n = M$ to prove!

$ed = k(p-1)(q-1) + 1$

$M^{ed} = M^{k(p-1)(q-1)+1} = M \cdot (M^{p-1})^{k(q-1)}$

$= M(1 \bmod p)^{k(q-1)} = M \bmod p$

Similarly $M^{ed} = M \bmod q \Rightarrow M^{ed} = M \bmod pq$

$= M \bmod n$

17

Panel 18

Weaknesses of RSA Algorithm

need large key n (= 1024 bits)

slow

Why is it called RSA alg.:

Ron Rivest, Adi Shamir, Leonard Adleman (MIT)

18

Panel 19

Security of RSA Algorithm

Depends on the fact that:

p·q is easy to do

but hard to factor

unless you know trap-door (private key)

Factoring is believed to be comp. difficult.

Not proven!

19

Panel 20

Wed + Mon: Review

Mon: Take-home final. 1 Def + 1 Q from each category

20