Frequency Examination

```
    ABCDEFGHIJKLMNOPQRSTUVWXYZ
1   31004011301001300112000000
2   10022210013010000010404000
3   12000000201140004013021000
4   21102201000010431000000211
5   10500021200000500030020000
6   01110022311012100000030101
```

Letter frequencies are (H high, M medium, L low):

```
    HMMMHMMHHMMMMHHMLHHHMLLLLL
```

Begin Decryption

- First matches characteristics of unshifted alphabet
- Third matches if I shifted to A
- Sixth matches if V shifted to A
- Substitute into ciphertext (bold are substitutions)

AD**I**YS **RI**U**K**B O**CK**K**L** MI**GH**K **A**ZO**TO**
E**I**OO**L** **I**F**T**AG **PA**U**E**F V**AT**A**S** CI**IT**W
EOC**NO** E**I**OO**L** **B**M**T**FV **EG**G**O**P C**NE**K**I**
HS**SE**W **N**EC**SE** D**D**AA**A** **R**W**C**XS **AN**S**N**P
H**HE**U**L** QO**NO**F **E**EG**OS** W**L**PC**M** **A**J**E**OC
MIU**A**X

Look For Clues

- **AJE** in last line suggests "are", meaning second alphabet maps A into S:

ALIYS **RICK**B O**CKSL** MI**GHS** **A**ZO**TO**

MIOO**L** **INT**AG **PACE**F V**ATIS** CI**ITE**

EOC**NO** **MI**OO**L** **BUT**FV **EGOO**P C**NESI**

HS**SEE** **N**EC**SE** **LD**AA**A** **REC**XS **ANAN**P

H**HECL** QO**NON** **E**EG**OS** **EL**PC**M** **ARE**OC

MICAX

Next Alphabet

- **MICA**X in last line suggests "mical" (a common ending for an adjective), meaning fourth alphabet maps O into A:

ALIMS **RICKP** O**CKSL** **A**I**GHS** **AN**O**TO**
MICO**L** **INTO**G **PACET** V**ATIS** **Q**I**ITE**
ECC**NO** **MIC**O**L** **BUTT**V **EGOOD** C**NESI**
VS**SEE** **NS**C**SE** **LDO**A**A** **RECL**S **ANAND**
H**HECL** **E**O**NON** **ES**G**OS** **ELD**C**M** **AREC**C
MICAL

Got It!

- QI means that U maps into I, as Q is always followed by U:

```
ALIME RICKP ACKSL AUGHS ANATO
MICAL INTOS PACET HATIS QUITE
ECONO MICAL BUTTH EGOOD ONESI
VESEE NSOSE LDOMA RECLE ANAND
THECL EANON ESSOS ELDOM ARECO
MICAL
```

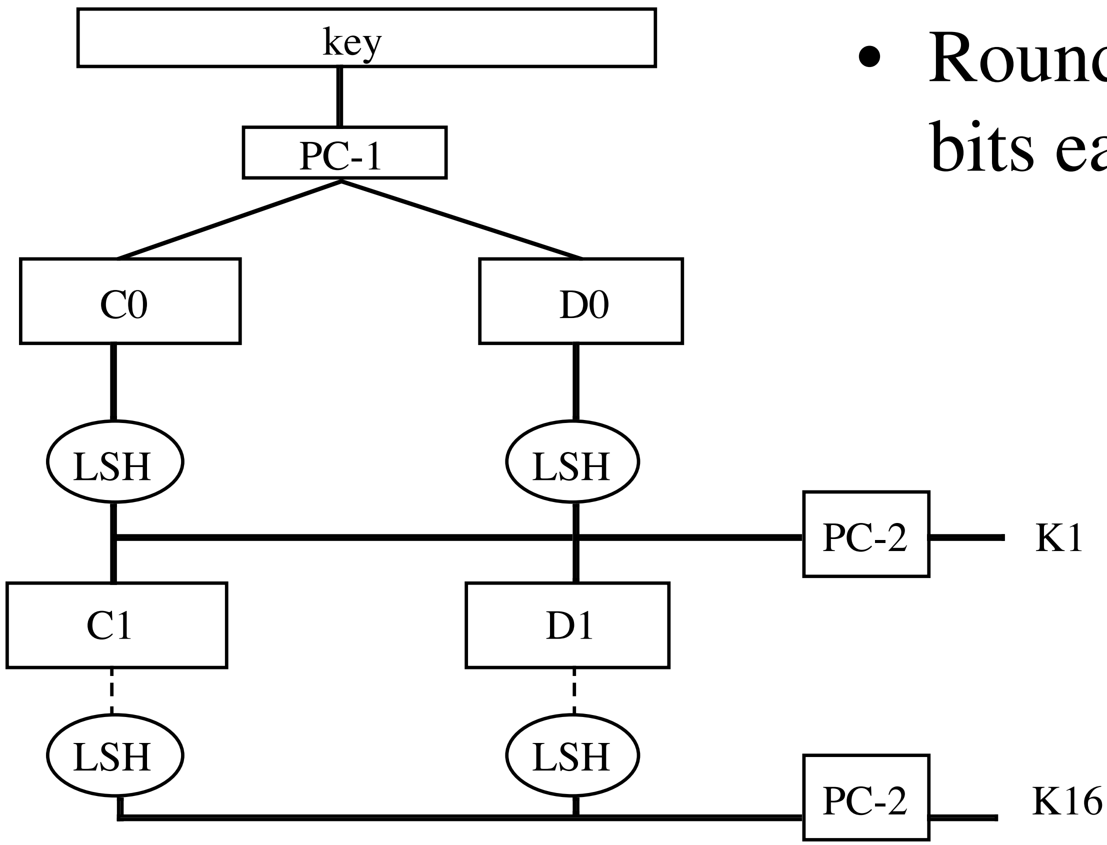
One-Time Pad

- A Vigenère cipher with a random key at least as long as the message
  - Provably unbreakable
  - Why? Look at ciphertext `DXQR`. Equally likely to correspond to plaintext `DOIT` (key `AJIY`) and to plaintext `DONT` (key `AJDY`) and any other 4 letters
  - Warning: keys *must* be random, or you can attack the cipher by trying to regenerate the key
    - Approximations, such as using pseudorandom number generators to generate keys, are *not* random

Overview of the DES

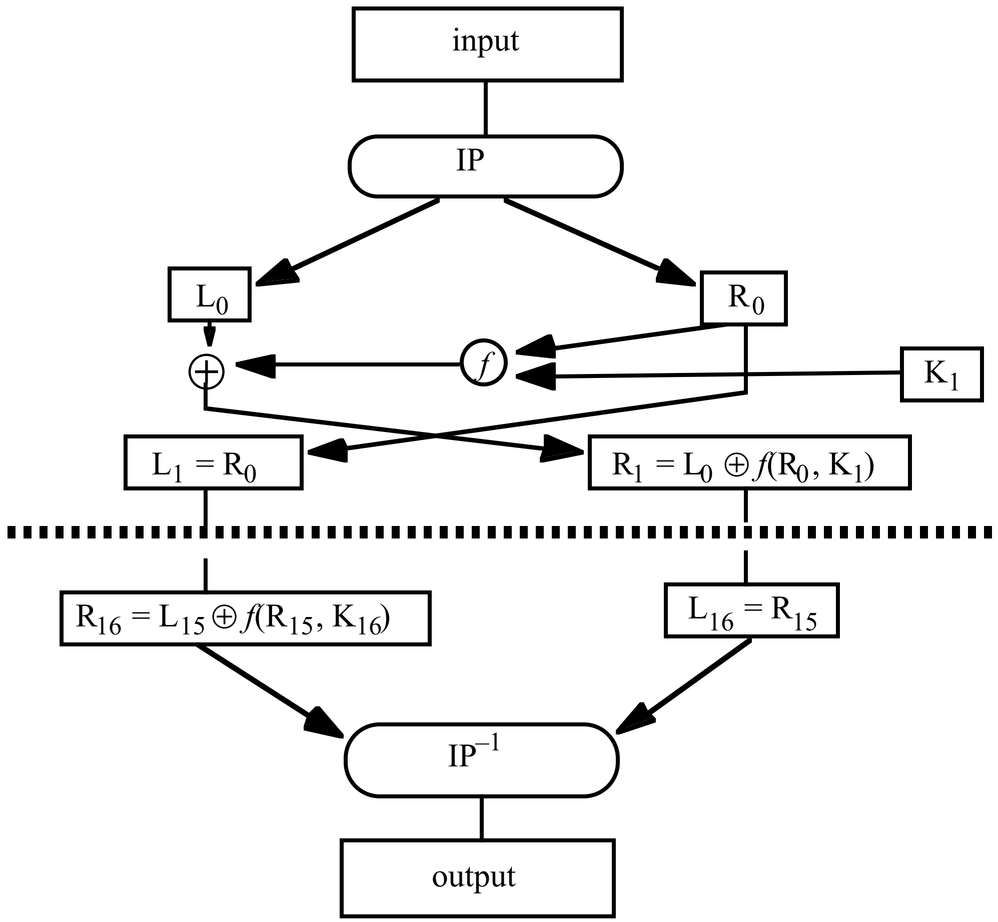
- A block cipher:
  - encrypts blocks of 64 bits using a 64 bit key
  - outputs 64 bits of ciphertext
- A product cipher
  - basic unit is the bit
  - performs both substitution and transposition (permutation) on the bits
- Cipher consists of 16 rounds (iterations) each with a round key generated from the user-supplied key

Generation of Round Keys

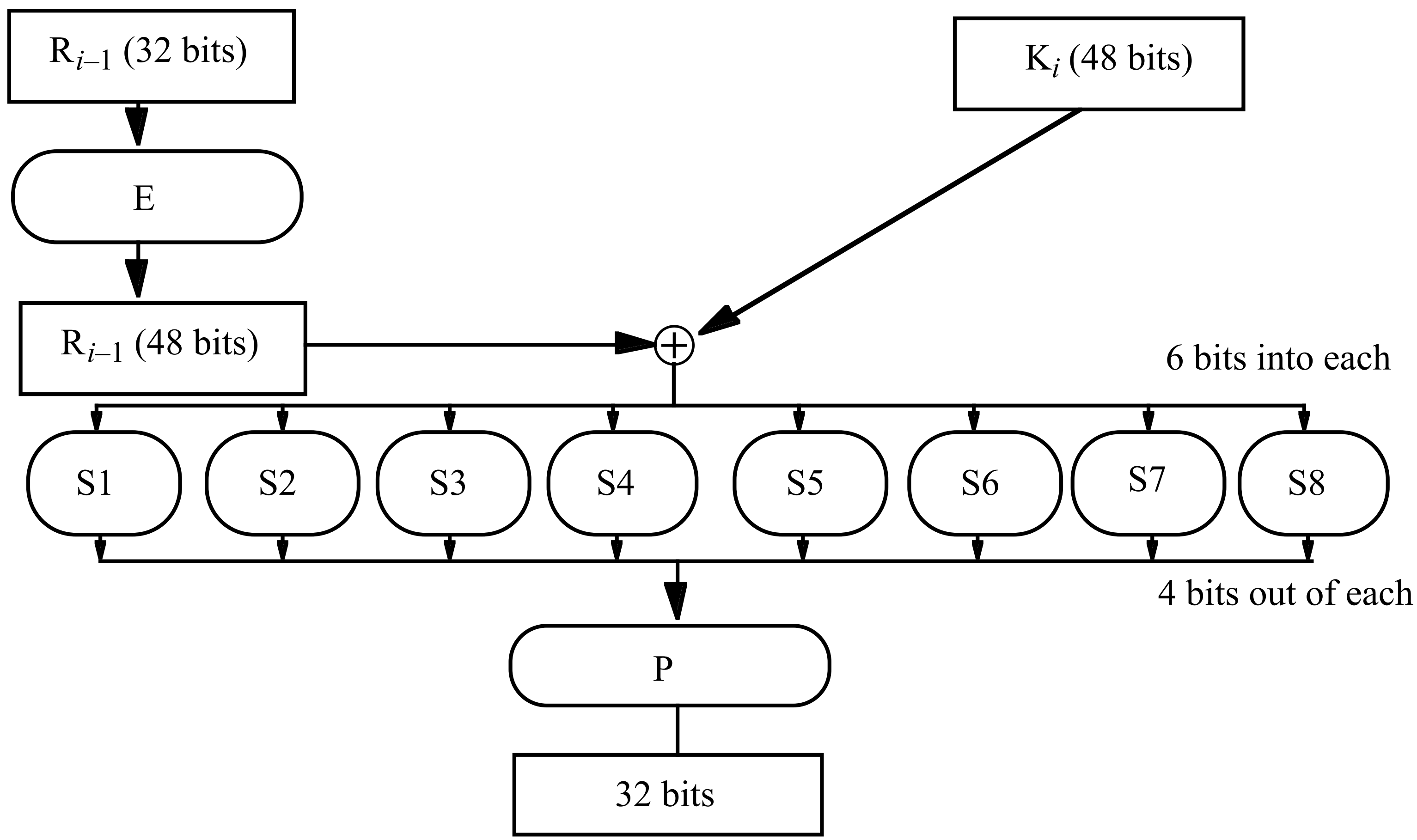


- Round keys are 48 bits each

Encipherment



The *f* Function



Controversy

- Considered too weak
  - Diffie, Hellman said in a few years technology would allow DES to be broken in days
    - Design using 1999 technology published
  - Design decisions not public
    - S-boxes may have backdoors

Undesirable Properties

- 4 weak keys
  - They are their own inverses
- 12 semi-weak keys
  - Each has another semi-weak key as inverse
- Complementation property
  - $\mathrm{DES}_k(m) = c \Rightarrow \mathrm{DES}_{k'}(m') = c'$
- S-boxes exhibit irregular properties
  - Distribution of odd, even numbers non-random
  - Outputs of fourth box depends on input to third box

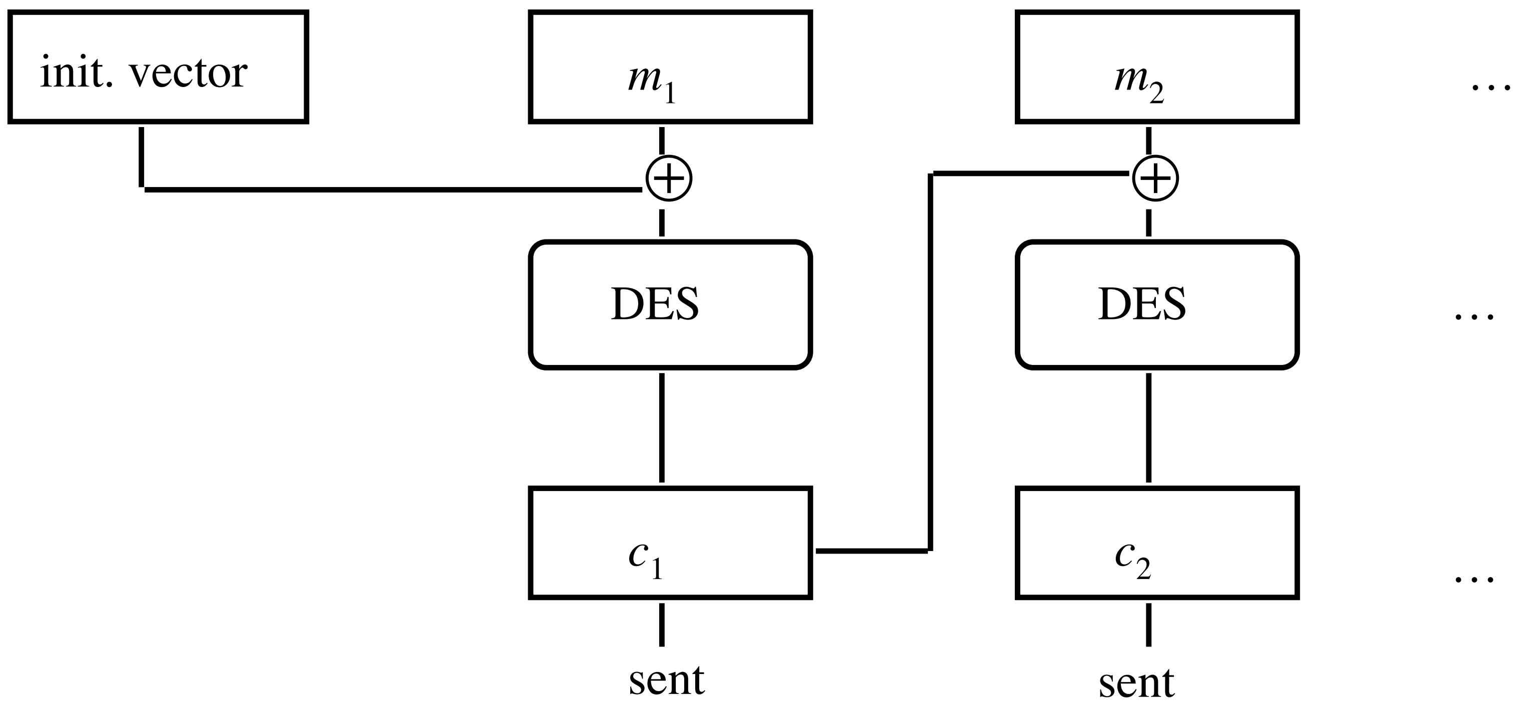
Differential Cryptanalysis

- A chosen ciphertext attack
  - Requires 2^{47} plaintext, ciphertext pairs
- Revealed several properties
  - Small changes in S-boxes reduce the number of pairs needed
  - Making every bit of the round keys independent does not impede attack
- Linear cryptanalysis improves result
  - Requires 2^{43} plaintext, ciphertext pairs

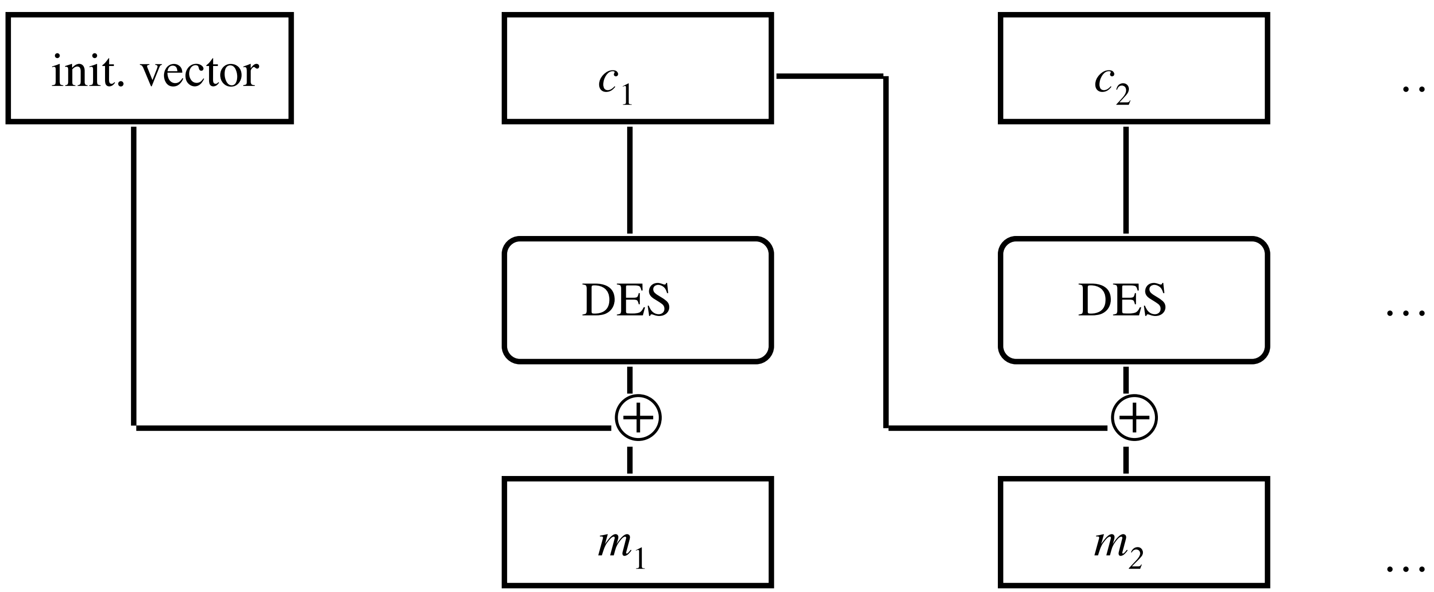
DES Modes

- Electronic Code Book Mode (ECB)
  - Encipher each block independently
- Cipher Block Chaining Mode (CBC)
  - Xor each block with previous ciphertext block
  - Requires an initialization vector for the first one
- Encrypt-Decrypt-Encrypt Mode (2 keys: k, k')
  - $c = \mathrm{DES}_k(\mathrm{DES}_{k'}^{-1}(\mathrm{DES}_k(m)))$
- Encrypt-Encrypt-Encrypt Mode (3 keys: k, k', k'')
  - $c = \mathrm{DES}_k(\mathrm{DES}_{k'}(\mathrm{DES}_{k''}(m)))$

CBC Mode Encryption



CBC Mode Decryption



Self-Healing Property

- Initial message
  - 3231343336353837 3231343336353837 3231343336353837 3231343336353837
- Received as (underlined 4c should be 4b)
  - ef7c4cb2b4ce6f3b f6266e3a97af0e2c 746ab9a6308f4256 33e60b451b09603d
- Which decrypts to
  - efca61e19f4836f1 3231333336353837 3231343336353837 3231343336353837
  - Incorrect bytes underlined
  - Plaintext "heals" after 2 blocks

Current Status of DES

- Design for computer system, associated software that could break any DES-enciphered message in a few days published in 1998
- Several challenges to break DES messages solved using distributed computing
- NIST selected Rijndael as Advanced Encryption Standard, successor to DES
  - Designed to withstand attacks that were successful on DES

Public Key Cryptography

- Two keys
  - *Private key* known only to individual
  - *Public key* available to anyone
    - Public key, private key inverses
- Idea
  - Confidentiality: encipher using public key, decipher using private key
  - Integrity/authentication: encipher using private key, decipher using public one

Requirements

1. It must be computationally easy to encipher or decipher a message given the appropriate key
2. It must be computationally infeasible to derive the private key from the public key
3. It must be computationally infeasible to determine the private key from a chosen plaintext attack

Diffie-Hellman

- Compute a common, shared key
  - Called a *symmetric key exchange protocol*
- Based on discrete logarithm problem
  - Given integers n and g and prime number p, compute k such that $n = g^k \bmod p$
  - Solutions known for small p
  - Solutions computationally infeasible as p grows large

Algorithm

- Constants: prime p, integer $g \neq 0, 1, p-1$
  - Known to all participants
- Anne chooses private key $kAnne$, computes public key $KAnne = g^{kAnne} \bmod p$
- To communicate with Bob, Anne computes $Kshared = KBob^{kAnne} \bmod p$
- To communicate with Anne, Bob computes $Kshared = KAnne^{kBob} \bmod p$
  - It can be shown these keys are equal

Example

- Assume p = 53 and g = 17
- Alice choses $kAlice$ = 5
  - Then $KAlice = 17^5 \bmod 53 = 40$
- Bob chooses $kBob$ = 7
  - Then $KBob = 17^7 \bmod 53 = 6$
- Shared key:
  - $KBob^{kAlice} \bmod p = 6^5 \bmod 53 = 38$
  - $KAlice^{kBob} \bmod p = 40^7 \bmod 53 = 38$

RSA

- Exponentiation cipher
- Relies on the difficulty of determining the number of numbers relatively prime to a large integer *n*

Background

- Totient function $\phi(n)$
  - Number of positive integers less than *n* and relatively prime to *n*
    - *Relatively prime* means with no factors in common with *n*
- Example: $\phi(10) = 4$
  - 1, 3, 7, 9 are relatively prime to 10
- Example: $\phi(21) = 12$
  - 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20 are relatively prime to 21

Algorithm

- Choose two large prime numbers p, q
  - Let $n = pq$; then $\phi(n) = (p-1)(q-1)$
  - Choose $e < n$ such that e is relatively prime to $\phi(n)$.
  - Compute d such that $ed \bmod \phi(n) = 1$
- Public key: (e, n); private key: d
- Encipher: $c = m^e \bmod n$
- Decipher: $m = c^d \bmod n$

Example: Confidentiality

- Take $p = 7$, $q = 11$, so $n = 77$ and $\phi(n) = 60$
- Alice chooses $e = 17$, making $d = 53$
- Bob wants to send Alice secret message HELLO (07 04 11 11 14)
  - $07^{17} \bmod 77 = 28$
  - $04^{17} \bmod 77 = 16$
  - $11^{17} \bmod 77 = 44$
  - $11^{17} \bmod 77 = 44$
  - $14^{17} \bmod 77 = 42$
- Bob sends 28 16 44 44 42

Example

- Alice receives 28 16 44 44 42
- Alice uses private key, $d = 53$, to decrypt message:
  - $28^{53} \bmod 77 = 07$
  - $16^{53} \bmod 77 = 04$
  - $44^{53} \bmod 77 = 11$
  - $44^{53} \bmod 77 = 11$
  - $42^{53} \bmod 77 = 14$
- Alice translates message to letters to read HELLO
  - No one else could read it, as only Alice knows her private key and that is needed for decryption

Example: Integrity/Authentication

- Take $p = 7$, $q = 11$, so $n = 77$ and $\phi(n) = 60$
- Alice chooses $e = 17$, making $d = 53$
- Alice wants to send Bob message HELLO (07 04 11 11 14) so Bob knows it is what Alice sent (no changes in transit, and authenticated)
  - $07^{53} \bmod 77 = 35$
  - $04^{53} \bmod 77 = 09$
  - $11^{53} \bmod 77 = 44$
  - $11^{53} \bmod 77 = 44$
  - $14^{53} \bmod 77 = 49$
- Alice sends 35 09 44 44 49

Example

- Bob receives 35 09 44 44 49
- Bob uses Alice's public key, $e = 17$, $n = 77$, to decrypt message:
  - $35^{17} \bmod 77 = 07$
  - $09^{17} \bmod 77 = 04$
  - $44^{17} \bmod 77 = 11$
  - $44^{17} \bmod 77 = 11$
  - $49^{17} \bmod 77 = 14$
- Bob translates message to letters to read HELLO
  - Alice sent it as only she knows her private key, so no one else could have enciphered it
  - If (enciphered) message's blocks (letters) altered in transit, would not decrypt properly

Example: Both

- Alice wants to send Bob message HELLO both enciphered and authenticated (integrity-checked)
  - Alice's keys: public (17, 77); private: 53
  - Bob's keys: public: (37, 77); private: 13
- Alice enciphers HELLO (07 04 11 11 14):
  - $(07^{53} \bmod 77)^{37} \bmod 77 = 07$
  - $(04^{53} \bmod 77)^{37} \bmod 77 = 37$
  - $(11^{53} \bmod 77)^{37} \bmod 77 = 44$
  - $(11^{53} \bmod 77)^{37} \bmod 77 = 44$
  - $(14^{53} \bmod 77)^{37} \bmod 77 = 14$
- Alice sends 07 37 44 44 14

Security Services

- Confidentiality
  - Only the owner of the private key knows it, so text enciphered with public key cannot be read by anyone except the owner of the private key
- Authentication
  - Only the owner of the private key knows it, so text enciphered with private key must have been generated by the owner

More Security Services

- Integrity
  - Enciphered letters cannot be changed undetectably without knowing private key
- Non-Repudiation
  - Message enciphered with private key came from someone who knew it

Warnings

- Encipher message in blocks considerably larger than the examples here
  - If 1 character per block, RSA can be broken using statistical attacks (just like classical cryptosystems)
  - Attacker cannot alter letters, but can rearrange them and alter message meaning
    - Example: reverse enciphered message of text ON to get NO

Cryptographic Checksums

- Mathematical function to generate a set of k bits from a set of n bits (where $k \leq n$).
  - k is smaller then n except in unusual circumstances
- Example: ASCII parity bit
  - ASCII has 7 bits; 8th bit is "parity"
  - Even parity: even number of 1 bits
  - Odd parity: odd number of 1 bits

Example Use

- Bob receives “10111101” as bits.
  - Sender is using even parity; 6 1 bits, so character was received correctly
    - Note: could be garbled, but 2 bits would need to have been changed to preserve parity
  - Sender is using odd parity; even number of 1 bits, so character was not received correctly