RSA Public Key Cryptosystem provides an adequate level of security for most communication requirements

Originator of a particular RSA coding algorithm

Creation of the Public Key K_R

a. construct p & q such that
   i. p=3•j + 2
   ii. q=3•k + 2
   where j & k are sufficiently large so that the value K_R constructed below cannot be factored in any reasonable amount of time
b. compute $K_R = p \cdot q$
c. publish K_R but keep p & q private, i.e., secret
d. compute s = (1/3)(2•(p-1)•(q-1)+1) and keep it private, i.e., secret

Sender of coded message

Using the ASCII table, encode the alphabetic ClearText into binary numbers, e.g., the ClearText "CSUN IT" encodes into ASCII as 0100 0011 0101 0011 0101 0101 0100 1110 0010 0000 0100 1001 0101 0100

Group the ASCII code into bundles of predetermined length, e.g.,
(0100 0011) (0101 0011) (0101 0101) (0100 1110) (0010 0000) (0100 1001) (0101 0100)

For the following discussion, assign T_1 = (0100 0011), T_2 = (0101 0011), …, T_7 = (0101 0100)

For each T_i compute the remainder of the whole division $\left({T_i^3}/{K_R} \right)$; for purposes of the following discussion, assign R_i to the said remainder. For each T_i transmit the remainder R_i to the Receiver.

Receiver of coded message
(also the Originator of the particular RSA coding algorithm being used by the Sender)

Using s = (1/3)(2•(p-1)•(q-1)+1), compute the remainder of the whole division $\left({R_i^s}/{K_R} \right)$; for purposes of the following discussion, assign M_i to the said remainder. In our example, M_1 = (0100 0011) = T_1

For all i, we have that $M_i = T_i$; hence all we have to do is ungroup the M_i and convert them from the ASCII code to ClearText.

RSA encryption provides adequate security for most transactions since the time required to factor K_R into p & q exceeds the life time of any person currently alive, <u>if p and q are of sufficient size</u>, e.g., p & q both exceed 127 digits in length.

Relative Frequencies of Letters in English & Other Selected Languages

http://en.wikipedia.org/wiki/Letter_frequency#Relative_frequencies_of_letters_in_the_English_language

Essential Characteristics

a. **Worm – an independent program that makes copies of itself and transmits them over the internet to infect other computer systems.**

b. **Virus – a fragment of a program that embeds itself into another program, thus effectively hiding itself; when the infected program is executed, the virus make copies of itself which are released to infect other programs on the same machine. When infected files are transferred to other machines, the virus is transferred as well, and thus infects the new machine.**

c. **Trojan Horse – a virus that hides in another useful program that performs operations unbeknownst to the user such as recording keystrokes, providing a security hole for system access or illicit access to the communication protocols.**

d. **Phishing –**
    i. **Email – solicitation to participate in a joint venture required the victim to provide confidential financial &/or personal information that will be used to the detriment of the victim.**
    ii. **Web-based – spoofing of a website to make it appear to be a valid commercial website, but designed to acquire confidential financial &/or personal information that will be used to the detriment of the victim.**
    iii. **Face-to-Face – personal solicitation designed to acquire confidential financial &/or personal information that will be used to the detriment of the victim.**

e. **Cookies** http://en.wikipedia.org/wiki/HTTP_cookie

Number Systems

Decimal	Octal	Binary	Hexadecimal
0	0	0000	0
1	1	0001	1
2	2	0010	2
3	3	0011	3
4	4	0100	4
5	5	0101	5
6	6	0110	6
7	7	0111	7
8	10	1000	8
9	11	1001	9
10	12	1010	A
11	13	1011	B
12	14	1100	C
13	15	1101	D
14	16	1110	E
15	17	1111	F

Conversions

Octal ↔ Binary ↔ Hexadecimal

765_8 ↔ 111 110 101 ↔ 0001 1111 0101 ↔ $1F5_{16}$

groups of three — groups of four

Conversions

Decimal → Base n

Successive Divisions of the Decimal Number by n, preserving the remainders

65_{10} → X_5

5 | 65

5 | 13 0

2 → 3

65_{10} → 230_5

Base n ➔ Decimal

Polynomial Expansion

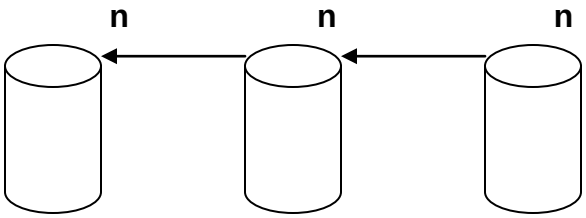
230_5 ➔ 2 3 0 $_5$ ➔ $2*5^2 + 3*5^1 + 0*5^0$ ➔ 50 + 15 + 0 ➔ 65_{10}

Base
Index 2 1 0

Coefficient * Index^{Base} + Coefficient * Index^{Base} + …

Addition

Base n ➔ (1) dump the bucket when it has n stones in it;
(2) add one stone to the bucket on the left



Subtraction

“Take Away”

When bucket is empty for Base n ➔

(1) remove one stone from the bucket on the left
(2) place n stones in the bucket that was empty