Disc Math - §17 - Cryptography

Original information → plaintext. Processed by an encryption key → ciphertext. Then transmitted and decoded with decryption key.

Ex: Ceaser cipher: Encryption: Letters are coded by numbers: A→0, B→1, ..., Z→25; Shift the numbers by a positive value k, mod (25). Transmit. Shift the message with $-k$, mod 25. Translate the numbers back to letters.

Ex: These days we have: DES (Data Encryption Standard) which manipulates 64 bit binary strings with a 56-bit key.
AES (Advanced Encryption Standard) - 128-bit key.
DES, AES and Ceaser cipher are examples of symmetric encryption (private key encryption schemes) → the same key is used to encode and to decode the message; both sender and receiver must know the key → so how do we securely transmit the key.

Asymetric encryption (public key encryption): the decryption key cannot be derived efficiently from the encryption key → the encryption key can be made public.

Ex: RSA public key encryption algorithm (Rivest, Shamir, Adleman 1977)

Th: (Euler. Let b, m be relatively prime, i.e. $\gcd(b,m) = 1$. Then

$$b^{\phi(m)} \equiv 1 \pmod{m}.$$

Pr: Let $r_1, r_2, \ldots, r_{\phi(m)}$ be the numbers relatively prime to m and $\leq m$. Then multiplication by b, s.t. $\gcd(b,m) = 1$, and the result considered mod m, only changes the order of these relative primes $r_1, \ldots, r_{\phi(m)}$.

Ex: $m = 10$, $b = 7$; $r = 1, 3, 7, 9$

$$rb = 7, 21, 49, 63 \equiv 7, 1, 9, 3 \pmod{10}.$$ □

Pr Cntd: Then $b^{\phi(m)} r_1 r_2 \cdots r_{\phi(m)} \equiv r_1 r_2 \cdots r_{\phi(m)} \pmod{m}$.
Since $r_1, r_2, \ldots, r_{\phi(m)}$ are relatively prime mod m, they have reciprocals mod m and we can cancel them. $\Rightarrow 1^{\phi(m)} = 1$ (mod m) ∎

Disc Math - §17 - Cryptography

① Select two primes p, q (about 200 digits each) and compute $r = pq$.

② Select an encrypting exponent s such that s and $\phi(r)$ are relatively prime.

③ Everyone who wishes to receive a secret message selects such a triple p, q, s and makes public the pair r, s.

④ The sender then (translates the message into an integer M) and computes $E = M^s \pmod r$; transmits E to the receiver.

rem: Now to decrypt the message the receiver must take s'th root of E; thus she needs the reciprocal of s in some sense.

⑤ Now since $\gcd(s, \phi(r)) = 1$, by Euler: $s^{\phi(\phi(r))} \equiv 1 \pmod{\phi(r}$

i.e. with $t = s^{\phi(\phi(r))-1} \pmod{\phi(r)}$, $ts \equiv 1 \pmod{\phi(r)}$.

rem: Crucially, the receiver knows that $r = pq$ and thus

$$\phi(r) = (p-1)(q-1).$$

⑥ Having obtained t the decryption now could proceed:

$$E^t = M^{st} = M^{k\phi(r)+1} \pmod r.$$

By Euler again $M^{\phi(r)} \equiv 1 \pmod r$ {we must have $\gcd(M, r) = 1$, but the chances of this not being the case $\sim 10^{-100}$}.

$$\Rightarrow E^t = M \cdot (M^{\phi(r)})^k = M \cdot 1 = M \pmod r.$$

Main point: The decrypting exponent t cannot be derived efficiently from the publically known pair s, r, but only from the factors of r, namely p, q. The message remains secret provided r cannot be efficiently factored.

rem: At present, the best algorithm factors a 200-digit number that is the product of two 100-digit factors in 40 trillion years, 2000 times the age of the Universe.

rem: t could also be found with the Euclidean algorithm, since $\gcd(s, \phi(r)) = 1$

$$ts + k\phi(r) = 1$$

Disc Math - §17 - Cryptography

Ex: $r = 187 = 11 \cdot 17$; $s = 7$; The message is $M = 3$.

$$E = M^7 = 3^7 = 2187 = 11 \cdot 187 + 130 = 130 \pmod{187}$$

$$\phi(r) = \phi(11 \cdot 17) = 10 \cdot 16 = 160$$

$160 = 7 \cdot 22 + 6$; $7 = 1 \cdot 6 + 1$; $1 = 7 - 6 = 7 - (160 - 7 \cdot 22) = 7 \cdot 23 - 160$

$\Rightarrow t = 23 \pmod{160}$.

$$E^t = 130^{23} = 130^4 \cdot 130^4 \cdot 130^4 \cdot 130^4 \cdot 130^4 \cdot 130^3 \pmod{187} =$$

$$= 38 \cdot 38 \cdot 38 \cdot 38 \cdot 38 \cdot 124 \pmod{187} = 89 \cdot 124 \pmod{187} =$$

$$= 59 \cdot 187 + 3 \pmod{187} = 3 \pmod{187}.$$

rem: RSA is fairly slow. so it is mostly used for transmitting private key for DES encryption which is much faster.

HW p440, § 5.6

7, 8, 25, 26, 28