October 29th, 2016

Shubham Kr. Bharti
Varun Kr. Khare
B.Tech. II nd Year

CRYPTOGRAPHY

Basics Of Cryptography

Symmetric and Asymmetric Crypto

Diffie-Hellman Key Exchange

RSA Cryptosystem

Elgamal Cryptosystem

Cryptography

The study of techniques for secure communication in presence of eavesdroppers

<u>Key Terms:</u>

- Plain Text
- Cipher Text
- Keys
  - Symmetric Key
  - Asymmetric Key
  - Public Key
  - Private Key

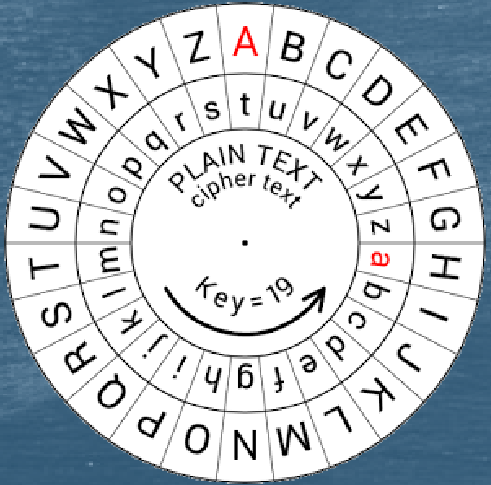
Symmetric Key Cryptosystem

use the same cryptographic keys for both encryption of plaintext and decryption of cipher text

- Both the parties should have the access to same secret key
- Any third person, if obtains the key can decode the cipher text even without letting the communicators know that their key is exposed.
- Are exposed to various attacks such as known-plaintext attacks, chosen-plaintext attacks, differential cryptanalysis and linear cryptanalysis
- Eg. Caser Cipher, Vigenere Cipher, XOR Cipher

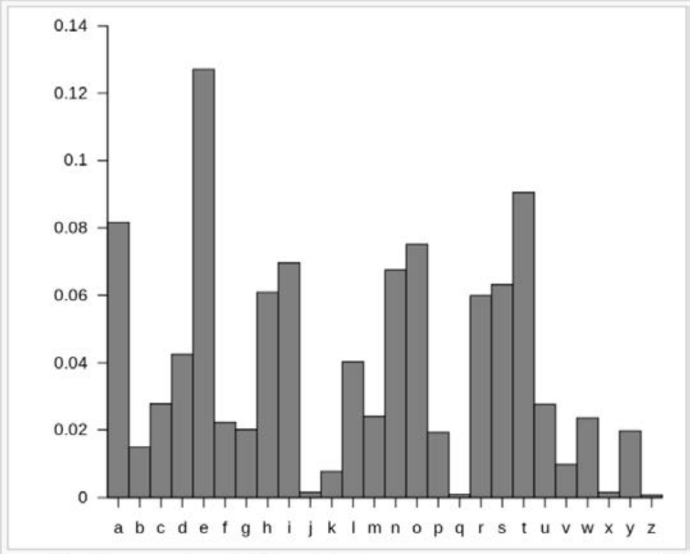
Shift Cipher

- Based on Shift parameter (s)
- Method can be improved by randomly choosing distinct keys for distinct position.
- Eg. Caeser Cipher,
  Vigenere Cipher (Polyalphabetic Cipher)
  One Time Pad
- Random Numbers and Pseudorandom key generators are used to generate private key space.



Reason for failure

- Easy to compute all possible cipher text with modern computational power.
- Frequency Fingerprint provides a backdoor for decrypting cipher text. Also certainty of presence of a message makes it easy to decrypt the cipher text.



Relative frequencies of letters in text.

Asymmetric Key Cryptosystem

use different cryptographic keys for encryption of plaintext and decryption of cipher text

- Also called Public Key Cryptography
- Method uses both public key and private keys for encryption and decryption. Private keys are kept with individuals.
- Any person can encrypt the message by using public key of the receiver but decryption can be done only by private keys.
- Strength of such system lies on degree of difficulty to determine private keys from corresponding public keys.
- Eg. Diffie Hellman, Digital Signature algorithm, RSA, ElGamal etc;

RSA Cryptosystem

- A practical public-key cryptosystem where the encryption key is public while the decryption key is kept secret.
- Described by Rivest, Shamir and Adleman in 1977.
- Algorithm requires to big prime numbers (over 250 digit) long and is secure as long as factorization remains a computationally expensive task with most modern available computers.
- Theoretical Quantum computers with exceptionally high computation power have proven to break RSA using the famous Shor’s algorithm.

The Algorithm

- Euler's Theorem $a^{\varphi(n)+1} \equiv a \mod n$
- Euler totient function is multiplicative. $\varphi(n)=\varphi(p1)*\varphi(p2)$
- For prime p, $\varphi(p)=p-1$

where $n=p1*p2$

- Take two large prime numbers, p1 and p2. Let $n=p1*p2$, then
  $\varphi(n)=\varphi(p1)*\varphi(p2)=(p1-1)(p2-1)$

- Select base e such that $1<e<n$ and $(e,n)=1$; e,n are co-prime and a random int k.

❖ Note that finding $\varphi(n)$ is very costly if primes p1 and p2 are very big unless we know p1 or p2 or both.

- $a^{k*\varphi(n)+1} \equiv a \mod n$ Find $d=\frac{k*\varphi(n)+1}{e}=\frac{k*(p1-1)*(p2-1)+1}{e}=\frac{k*(n-p1-p2+1)+1}{e}$

The RSA Algorithm

1. Generate prime p1,p2 and random base e. p1 and p2 are private keys.

2. Find n=p1*p2. and share public keys e and n.

3. Calculates $d=\frac{k*(p1-1)*(p2-1)+1}{e}$, and find $m=g^d$ mod n which gives him message m.

e, n is now public

g is now public

Decrypting the message.

Rohan

1. Receives e, n

2. Say he has encoded message m. Calculate $g=m^e$ mod n and send it back to Shyam.

Encrypting the message.

Key Generation

- Usually e is chosen small, where (e,n)=1
- Generating Big Prime Numbers p1 and p2.

  Using Probabilistic Primality test,

  1. Rabin Miller Test
  2. Baillie-PSW primality test

  Deterministic Primality Test like AKS can also be used but it's a bit computationally slower.

Diffie-Hellman Key Exchange

Method to securely exchange private keys over a public channel

Discrete Logarithm Problem

given prime p, and numbers a, g its computationally very costly to find k such that

$$g=a^k \bmod p$$

for large values of p (say 100 digits long)

- Earlier exchange of private keys involved some secure physical channel such as courier, paper etc.
- DH made it possible to securely exchange private keys between parties over the internet.
- Key Factor:
  - Computation Expense of discrete logarithm problem is very high for large primes.
  - Involves high communication overhead in case of sharing keys to multiple parties.

Diffie-Hellman Key Exchange

Shyam

1. Generate prime p and random base a<p

2. Generate Private key k1 and find $g1=a^{k1} \bmod p$

3. Share g1, a, p publicly.

g1, a, p is now public →

4.Recieves g2 from Rohan.

5. Find $g=g2^{k1} \bmod p$

$=a^{k1*k2} \bmod p$

Rohan

1. Receives a,p, g1

2. Generate Private key k2 and find $g2=a^{k2} \bmod p$

3. Send g2 to Shyam publicly

← g2 is now public

4. Find $g=g1^{k2} \bmod p$

$=a^{k1*k2} \bmod p$

Even if someone has a,p,g1,g2 its currently impossible for them to get key g. Thanks to computational complexity of discrete logarithm problem!

Discrete Logarithm Problem

Elgamal (1984): discrete logarithm problem (DLP).

- Group G is set with operation (·) and each element has inverse.
- Main idea: very easy to compute $h = g^x$ for given x, but very hard to find x given h and g.
- Popular choices: finite fields and elliptic curves.

Problems Related to DLP

Given an abelian group $(G, \cdot)$ and $g \in G$ of order n.

- Discrete Logarithm Problem (DLP) : Given $h \in G$ such that $h = g^x$ find x. ($DLP(g, h) \rightarrow x$)
- Computational Diffie-Hellman Problem (CDH) : Given $a = g^x$ and $b = g^y$ find $c = g^{xy}$ ($CDH(g, a, b) \rightarrow c$).
- Decisional Diffie-Hellman Problem (DDH) :Given $a = g^x$, $b = g^y$ and $c = g^z$, determine if

  $g^{xy} = g^z$ or equivalently $xy \equiv z \bmod n$

  ($DDH(g, a, b, c) \rightarrow$ true/false)

Breaking the discrete logarithm problem

For example: In case of Elliptic Curves

$Q=P^x$

$Q*P^{-am}=P^b$

Baby-step, giant-step

- Calculate m= ⌈√n⌉
- For every b in 0,…,m, calculate bP and store the results in a hash table.
- For every a in 0,…,m:
  - calculate P^{am};
  - calculate $Q*P^{-am}$;
  - check the hash table and look if there exist a point P^b such that
    $Q*P^{-am}$=P^b;
  - if such point exists, then we have found x=am+b.
  - time and space complexity O(√n) or $O(2^{bk/2})$ {VERY LARGE TO BREAK}

Discrete Logarithm Problem

- Let (G, *) be an abelian group.
- Discrete Logarithm Problem Given g, h ∈ G, find an x (if it exists) such that

$$g^x = h.$$

- The difficulty of this problem depends on the group G:
  - Very easy: polynomial time algorithm, e.g. (ZN , +)
  - Hard: sub-exponential time algorithm, e.g. (Fp, ×)
  - Very hard: exponential time algorithm, e.g. elliptic curve groups.

ElGamal Encryption - Key Generation

ElGamal (1985): A public key cryptosystem and a signature scheme based on discrete logarithms.

Domain Parameter Generation:

- Generate a "large prime" p (≥ 1024 bits) such that p−1 is divisible by another "large prime" q (> 160 bits).
- Compute a generator g of the multiplicative group of order q in GF(p)*, via (for some random r)

$g \equiv r^{(p-1)/q} \bmod p$

until g!=1.

ElGamal Encryption - Key Generation

Key Generation:

- Select a random integer a, $1 \leq a \leq q - 1$ and compute

  $h \equiv g^{a \bmod p}$

- Public key = (p, g, h) which can be published.
- Private key = a which needs to be kept secret.

ElGamal - Encryption / Decryption

Shyam encrypts a message for Rohan as follows:

- Obtain Rohan's authentic public key (p, g, h).
- Generate random k

  $(1 < k < q - 1)$

  with $\gcd(k, p - 1) = 1$
- $r \equiv g^{k \bmod p}$ (k and r are ephemeral key pair)
- $s \equiv h^k.m \bmod p$ $(0 \leq m \leq p - 1)$
- Ciphertext: c = (r, s)

To recover the message, Rohan does the following:

$m \equiv s \cdot r^{-a} \bmod p$

Indeed:

$r^{-a} \equiv g^{-ka} \equiv h^{-k} \bmod p$

Questions & Discussion