PHƯƠNG PHÁP SINH KHÓA VÀ MÃ HÓA RSA

Thangdn – 18.7.2015

I. PHƯƠNG PHÁP SINH KHÓA

Theo định lý về số nguyên tố, thì khoảng cách trung bình của hai số nguyên tố lớn cỡ nbit khoảng n*ln2. Vậy trong khoảng này có khản năng cao sẽ tồn tại số nguyên tố.

1. Sinh số ngẫu nhiên

Ta cần tạo ra số giả ngẫu nhiên n kích thước đủ lớn cơ 3072bit. Vì được sử dụng trong mật mã nên số giả ngẫu nhiên phải đảm bảo không đoán được, không trùng nhau và cách xa nhau.

Trong chương trình này sử dụng hàm RandomBits()(thư viện NTL) để sinh một chuỗi ngẫu nhiên n bit đảm bảo an toàn.

2. Tiền xử lý

Ta sẽ xét trong khoảng hơn 2*n*ln2 số bằng thuật toán Rabin-miller, ta có thể loại nhanh hơn các hợp số với 1000 số nguyên tố nhỏ. Qua bước xử lý này có thể loại bỏ 93% hợp số trong khoảng [n,n+2*n*ln2).

Các bước thực hiện:

- Tạo danh sách 1000 số nguyên tố đầu tiên.
- Với mỗi số ngẫu nhiên n, tạo một mảng bít S kích thước 2*n*ln2, khởi tạo bằng 0.
- Với mỗi số nguyên tố p khởi tạo ban đầu, tính r=n%p. Gán tất cả các phần tử S[p*k-r]=1. Ta thấy rằng n+p*k-r luôn chia hết cho p. Như vậy những phần tử S[i]=1, ta luôn có n+i chia hết cho p.

Sau khi kết thúc ta sẽ chỉ cần kiểm tra các phần tử n+i với S[i]=0.

3. Kiểm tra số giả nguyên tố Fermat cơ sở 2

Việc kiểm tra số giả nguyên tố fermat cơ sở 2 sẽ giúp loại được nhiều hợp số.

4. Kiểm tra Rabin-miller

Với sai số của số giả nguyên tố là 2^{-128}, vậy nên ở đây ta sẽ thực hiện kiểm tra Rabin-miller với 128/2 cơ số khác nhau cho n, bao gồm 20 số nguyên tố đầu tiên và 44 số sinh ngẫy nhiên <= căn bậc 2 của n.

5. Sinh số nguyên tố mạnh

Với việc sinh số nguyên tố mạnh đảm bảo tính khó tấn công bằng việc phân tích nguyên tố n=p*q đối với p,q là số nguyên tố yếu. Số nguyên tố mạnh phải đảm bảo những yêu cầu sau:

- Số p là một số cực lớn
- Thừa số nguyên tố lớn nhất của p-1 là p^- một số lớn.

sao cho $p=a^-*p^- +1$ là số nguyên tố nhỏ nhất với a^- tự chọn

- Thừa số nguyên tố lớn nhất của $p^- - 1$ là p^{--} một số lớn.

sao cho $p^-=a^{--}*p^{--} + 1$ là số nguyên tố nhỏ nhất với a^{--} tự chọn

Thừa số nguyên tố lớn nhất của p + 1 là p^+ là số nguyên tố

sao cho $p=a^+*p^+ -1$ là số nguyên tố với a^+ là tự chọn

Ta sử dụng thuật toán của Gordon's cho việc tìm số nguyên tố mạnh

1. Tìm hai số p^{--} và p^+ là hai số ngẫu nhiên lớn cỡ n/2bit (nbit là độ dài số nguyên tố mạnh cần tìm). Chú ý hai số này phải khác nhau hoàn toàn.
2. Tìm p^- là số nguyên tố nhỏ nhất theo công thức $p^-=a^-*p^{--}+1$
3. Tính $po=((p^+)^{p^- -1}- (p^-)^{p^+ -1})\bmod (p^-*p^+)$.
4. Tìm p là số nguyên tố nhỏ nhất theo công thức $p=po+a*p^-*p^+$

6. Thời gian chạy

Qua quá trình cài đặt thuật toán bằng ngôn ngữ C++ trên hệ điều thành Ubuntu thấy tố độ như sau.

Sinh số nguyên tố độ dài 3072bit trong khoảng thời gian là 0-2s.

Sinh số nguyên tố mạnh độ dài 3072bit trong khoảng thời gian là 1-4s

Yêu cầu sử dụng thư viện NTL để sinh số ngẫu nhiên an toàn, và tính toán bằng thư viện GMP

II. MÃ HÓA RSA

1. Sinh khóa

Sinh hai ngẫu nhiên hai số p và q cách xa nhau là hai số nguyên tố mạnh cơ 3072bit

Tính n=p*q

Tính phi(n)=(p-1)*(q-1).

Chọn e sao cho gcd(e, phi(n))=1

Tính $d = e^{-1} \bmod phi(n)$

Private key (n,d) Public key (n,e)

2. Mã hóa

1. Mã hóa một chuỗi số có độ dài thuộc KEY{128,192,256}
2. Sinh chuỗi 2048 bit trong đó 16bit đầu là bit mặc định KEY là số bit đặt cuối, 2032-KEY là số bit giữa sẽ random.
3. Mã hóa chuỗi số 2048bit trên bởi công thức $C=M^e \bmod n$
4. Lưu C dưới dạng base64 tức gom 6bit một để tránh các ký tự điều khiển.

3. Giải mã

Quy trình giải mã làm ngược lại quá trình mã hóa với $M=C^d$ mode n

TÀI LIỆU THAM KHẢO:

[1]. *Are `Strong' Primes Needed for RSA?*,Ronald L. Rivest #,Robert D. Silverman y November 22, 1999

[2]. *Strong primes are easy to find*, John Gordon, Cybermation L t d