A Criptografia RSA e o Algoritmo Chinês do Resto

WELLERSON LOPES DA SILVA[1]
LUCAS MONTEIRO CHAVES[2]

[1]DCC – Departamento de Ciência da Computação
[2]DEX – Departamento de Ciências Exatas
UFLA – Universidade Federal de Lavras
Cx. Postal 37 – CEP 37.200-000 Lavras (MG)
popowls@comp.ufla.br
lucas@ufla.br

Resumo: A criptografia tem com objetivo codificar toda e qualquer mensagem em que se deseja privacidade. Atualmente quase todo sistema que utiliza envio e recebimento de informações utiliza algum método de criptografia. O uso desta é imprescindível em qualquer sistema de envio e recebimento de informações onde se deseja privacidade e segurança. Nesse artigo são descritos alguns aspectos básicos sobre a criptografia RSA e o aumento da eficácia desta com o uso do Algoritmo Chinês do Resto.

Palavras Chaves: criptografia RSA, algoritmo chinês do resto, chave pública, chave privada, números primos.

1 Introdução

A Internet é o grande paradigma de nosso tempo. Apesar de seu crescimento extraordinário ainda apenas vislumbramos o que ela representará em nossa vida nos próximos anos. Sua grande característica foi seu caráter democrático. As informações fluem de maneira pública e o acesso a elas é aberto a todos. Surge então a necessidade, para manter a privacidade de certas informações, do uso de uma ciência tão antiga como a ciência da escrita, a ciência da escrita codificada, a criptografia. Só assim a Internet se torna também um veículo para informações que não podem ser públicas como por exemplo o comércio eletrônico e as transações financeiras.

A criptografia RSA é um sistema de criptografia onde a chave de codificação é pública, permitindo então que qualquer pessoa codifique mensagens, e a chave de decodificação é privada. Este tipo de criptografia é extremamente adequado para, por exemplo, comércio eletrônico na Internet. A impossibilidade de se quebrar o sistema de criptografia RSA ocorre em razão da não existência de algoritmos eficientes para o processo de divisão de inteiros. Atualmente são utilizados números com 150 algarismos, para os quais com a capacidade de computação atual o processo de fatoração levaria milhares de anos.

O tempo de codificação de uma mensagem é praticamente desprezível mas o tempo de decodificação pode tornar o processo inviável. O objetivo deste trabalho é o de mostrar que o uso do algoritmo chinês do resto, um resultado clássico da Teoria dos Números, conhecido a mais de 1500 anos, diminui significativamente o tempo de decodificação.

2 A Criptografia RSA

O mais conhecido dos métodos de criptografia de chave pública é o RSA. Este código foi inventado em 1978 por R. L. Rivest, A. Shamir e L. Adleman, que na época trabalhavam no Massachussets Institute of Technology (M.I.T). As letras RSA correspondem às iniciais dos inventores do código. Há vários outros códigos de chave pública, mas o RSA é, atualmente, o mais usado em aplicações comerciais [Coutinho (1997), p.3].

Para implementar o RSA precisamos de dois números primos p e q. Para codificar a mensagem basta utilizar o produto destes dois números que chamaremos de n, que é a *chave pública de codificação.* Cada usuário do RSA possui sua própria chave. Esta chave é dita pública pois todos podem codificar uma mensagem. A *chave de decodificação* é constituída pelos primos e deve ser mantida em segredo pois a segurança do RSA depende disto. O fato de n ser conhecido para os casos

em que n (150 algarismos) torna praticamente impossível se conhecer os primos p e q.

2.1 Pré-codificação

A primeira coisa que temos que fazer para utilizar o RSA é transformar a mensagem em uma sequência de números. Vamos utilizar o código ASCII para converter cada caracter da mensagem em seu respectivo valor numérico na tabela ASCII transformando o texto então em um número gigantesco. Este número é quebrado em blocos de números menores do que o valor da *chave pública n*.

2.2 Codificação e decodificação

Para codificar a mensagem precisamos de n e de um inteiro positivo que seja inversível módulo $\phi(n)$, onde $\phi(n) = (p-1)(q-1)$. Em outras palavras, $mdc(e, \phi(n)) = 1$. Chamaremos de *chave de* codificação o par (n, e). Agora com a mensagem dividida em blocos, codificaremos cada um destes separadamente. Vamos chamar de $\boldsymbol{C}(b)$ o bloco codificado.

$\boldsymbol{C}(b)$ = resto da divisão de b^e por n

Para decodificar a mensagem precisamos também de dois números: n e o inverso de e em $Z_{\phi(n)}$, que denotaremos por d. O par (n, d) é a *chave de decodificação* do sistema RSA. Seja a o bloco codificado e $\boldsymbol{D}(a)$ o processo de decodificação:

(1) $\boldsymbol{D}(a)$ = resto da divisão de a^d por n

É fácil verificar que $\boldsymbol{D}(\boldsymbol{C}(b)) = b$.

3 O Algoritmo Chinês do Resto

Este algoritmo, utilizado para resolver sistemas de congruências lineares, é muito antigo e foi inventado, independentemente, pelos chineses e pelos gregos, para resolver problemas de astronomia.

O algoritmo chinês do resto tem este nome porque um dos primeiros lugares em que aparece é o livro *Manual de aritmética do mestre Sun*, escrito entre 287 d.C. e 473 d.C [Coutinho (1997), p.120].

De acordo com o Teorema Chinês do Resto, temos:

Sejam m e n inteiros positivos, primos entre si.

O sistema

$x \equiv a \pmod m$

$x \equiv b \pmod n$

sempre tem uma única solução em Z_{mn}.

A idéia é utilizar este teorema para resolver a equação modular (1). O algoritmo foi implementado no programa de computação algébrica *Mathematica 2.2*.

```
chines[primo1_, primo2_, base_, potencia_] :=
    Module[{valor, pot1, pot2, resto1, resto2, a,
     inversivel, b},

     pot1 = Mod[potencia, primo1-1];
     pot2 = Mod[potencia, primo2-1];

     resto1 = PowerMod[base, pot1, primo1];
     resto2 = PowerMod[base, pot2, primo2];

     if[resto1 > resto2,
        a = resto2 - resto1 + primo2,
        a = resto2 - resto1];

     inversivel = PowerMod[primo1, -1, primo2];

     b = Mod[inversivel*a, primo2];

     valor = resto1 + primo1 * b;

     Return[valor]
]
```

4 Análise do Tempo

Nesta seção é mostrado o tempo gasto para codificar e decodificar várias mensagens diferentes, utilizando o sistema RSA com e sem o algoritmo chinês do resto. Os números primos utilizados foram *113* e *89*.

A análise do tempo é realizada utilizando mensagens de diferentes tamanhos usando o sistema RSA sem o algoritmo chinês do resto e o sistema RSA com o uso do algoritmo.

5 Análise dos Resultados

Observamos que aumentando linearmente o número de caracteres da mensagem, o tempo gasto pelo sistema RSA sem o algoritmo chinês do resto aumenta linearmente enquanto que o sistema que utiliza o algoritmo mantém um tempo quase constante, como na Figura 1. É importante salientar que foram usados números com poucos algarismos. Visto que se aumentarmos muito tais números, o sistema sem o algoritmo torna-se totalmente inviável de implementação.

6 Conclusões

Observamos que a nossa implementação do sistema RSA sem o algoritmo chinês do resto é totalmente inviável em termos práticos. A segurança do RSA depende de uma chave com muitos dígitos. Já o sisetma

RSA com o uso do algoritmo chinês do resto mostrou-se viável, pois mesmo aumentando o número de caracteres da mensagem como o número de dígitos da chave, este demonstrou grande eficácia.

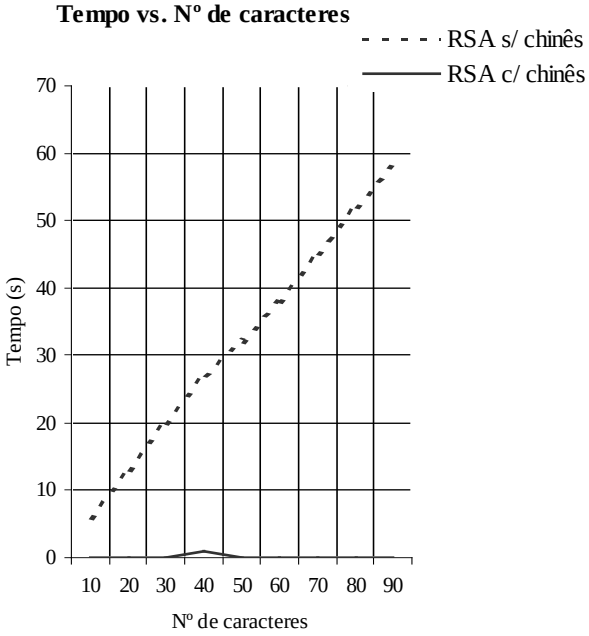

Figura 1: Tempo(s) vs. Nº de caracteres

7 References

Coutinho, S. C. "*Números inteiros e criptografia RSA*", Série de Computação e Matemática, IMPA, Rio de Janeiro, 1997.

Giblin, P. "*Primes and programming*", Cambridge University Press, Cambridge, 1993.

Knuth, D. E. "*The Art of Computer Programming*", vol. 2, Seminumerical algorithms, Segunda edição, Addison Wesley Publishing Company, Reading, 1981.

Lemos, M. "*Criptografia, números primos e algoritmos*", 17º Colóquio Brasileiro de Matemática, IMPA\CNPq, 1989.

Voloch, J. F. "*A distribuição dos números primos*", Matemática Universitária, número 06, 71-82, 1987.

Wolfram, S. "*Mathematica – A System for Doing Mathematics by Computer"*, Addison – Wesley Publishing Company, Second Edition, 1991.