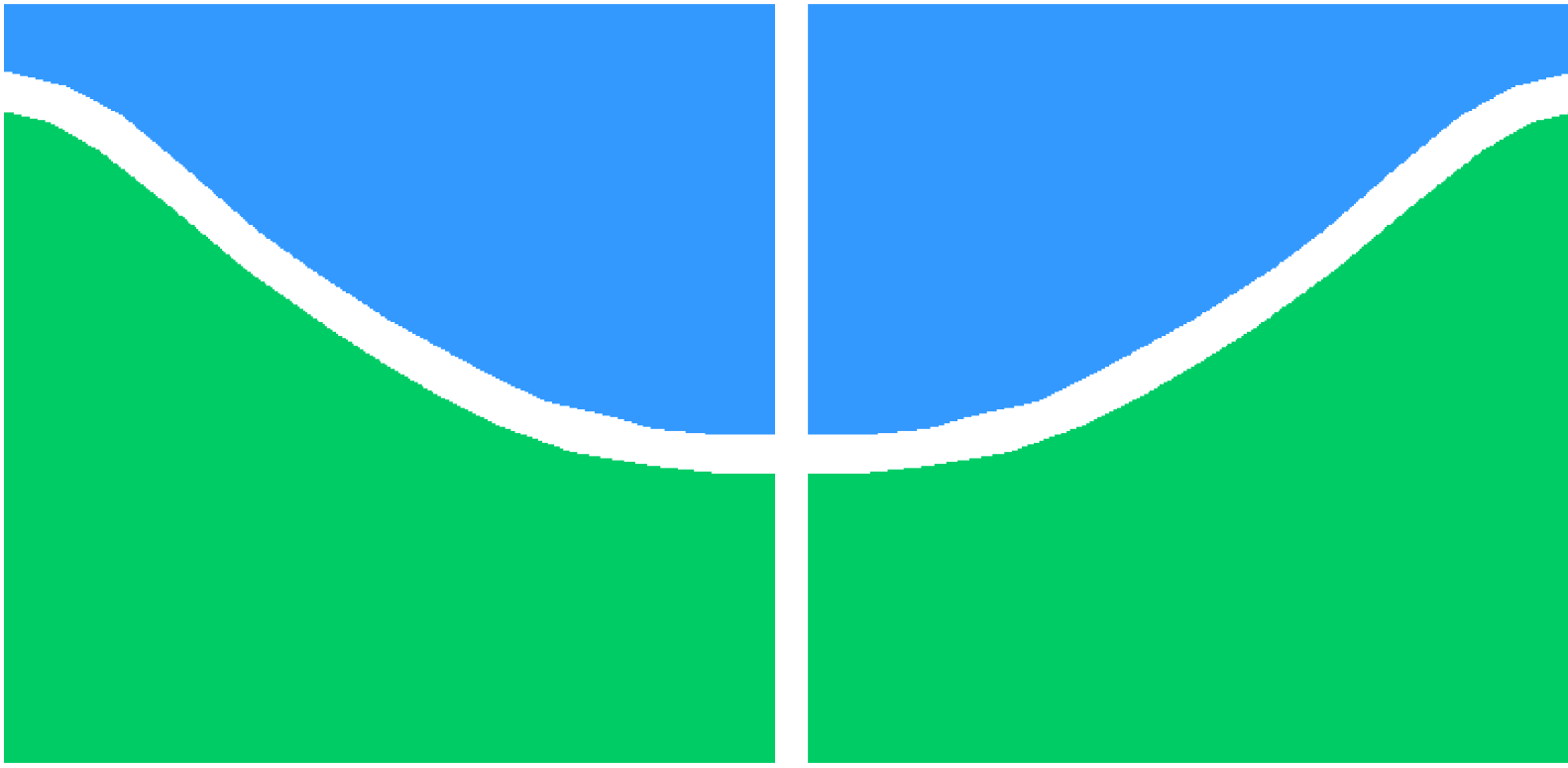

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Estruturas Matemáticas para Computação - 195405

Trabalho Final: Aplicação da Cifra RSA

Alunos:
Jonathan Moraes & Pedro de Lyra & Phelipe Wener

Orientador:
Vinicius de Carvalho Rispoli

Brasília, DF

13 de Dezembro de 2014

Jonathan Moraes & Pedro de Lyra & Phelipe Wener

Trabalho Final: Aplicação da Cifra RSA

Trabalho submetido durante o curso de graduação em Engenharia de Software da Universidade de Brasília como requisito parcial para obtenção curricular da disciplina de Estruturas Matemáticas para Computação.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Vinicius de Carvalho Rispoli

Brasília, DF

13 de Dezembro de 2014

Sumário

Lista de abreviaturas e siglas

RSA	Algoritmo de criptografia de dados, seu nome é referência aos três criadores: Ronald **R**ivest, Adi **S**hamir e Leonard **A**dleman
MIT	*Massachusetts Institute of Technology*, Instituto de Tecnologia de Massachusetts
MSR	Algoritmo de Teste de Primalidade Estocástico

1 Introdução

A criptografia é uma área que se dedica ao estudo de princípios e técnicas que convertam informaões de seu formato original à um formato ilegível, com o principal objetivo de garantir que o acesso àquela informação seja concedido apenas à entidades autorizadas (YASCHENKO, 2002).

Uma chave, que é um conjunto de dados específicos, é utilizada para realizar essa conversão, o que implica que apenas quem a detêm possui acesso à informação em questão.

Neste relatório procuramos esclarecer ao leitor o funcionamento de uma das técnicas de criptografia mais bem sucedidas até hoje, o algoritmo RSA.

2 O Algoritmo RSA

O RSA é um algoritmo de criptografia de dados amplamente utilizado na área de segurança da informação por garantir a transmissão segura de dados. Esta técnica de encriptação de dados foi desenvolvida por três professores do MIT, Ronald Rivest, Adi Shamir e Leonard Adleman, seu nome é constituído pelas iniciais dos sobrenomes dos seus autores. Foram realizadas diversas tentativas para quebrar o RSA, e o mesmo é um sucesso por ter resistido a todas elas.

A ideia é relativamente simples, o algoritmo implementa um sistema de chaves assimétricas e se baseia na teoria clássica dos números. Antes de explicá-lo, farei uma rápida introdução sobre sistemas de chaves assimétricas.

Em contraste aos métodos de chaves simétricas, onde existe uma única chave compartilhada entre o emissor e o receptor (Ex: Cifra de César, cifra de Vigenère, cifra afim, etc), um sistema de chaves assimétricas utiliza um par de chaves: uma chave pública e uma chave privada.

Neste sistema, cada entidade da rede de comunicações possui um par de chaves pública e privada, sendo que a chave pública é distribuída livremente a todos da rede, enquanto a chave privada deve ser conhecida apenas pelo seu dono. Por exemplo, em uma rede formada pelos personagens João e Maria, cada um deles possuiria seu par de chaves. Caso o João quisesse enviar uma mensagem encriptada à Maria, ele utilizaria a chave pública da Maria para encriptar esta mensagem, e apenas a Maria poderia decifrar a mensagem encriptada para conhecer seu real conteúdo. O processo inverso é análogo.

Como já foi dito, o sistema RSA se apoia nos fundamentos da teoria dos números, portanto, serão apresentados alguns conteúdos da área citada, como aritmética modular (SHOKRANIAN; SOARES; GODINHO, 1998). O primeiro passo do algoritmo consiste em gerar o par de chaves pública e privada.

Para isso, é realizada uma escolha aleatória de dois números primos, p e q, significativamente grandes. Em seguida, é calculado o produto n entre esses primos (a título de ilustração, em aplicações reais este produto é da ordem de 1024 a $4096 bits$. Por exemplo, um número inteiro em um computador geralmente se encontra na ordem de $32 bits$, o que permite uma representação de inteiros no intervalo de -2^{31} à 2^{31} aproximadamente! Imagine o que é possível representar com 1024 bits!).

O próximo passo computa a função totiente de Euler em n:

$$\varphi(n) = \varphi(p) * \varphi(q) = (p-1) * (q-1) \tag{2.1}$$

. Em seguida, é escolhido um inteiro e tal que se encontre no intervalo de números entre 1 e $\varphi(n)$ que seja relativamente primo de $\varphi(n)$. Este número e, junto com o número n constitui a chave pública!

Para calcular a chave privada, basta calcular o inverso multiplicativo d de e:

$$e^{-1} \equiv d(mod\varphi(n)) \tag{2.2}$$

Isto é:

$$e * d(mod\varphi(n)) = 1 \tag{2.3}$$

O número d encontrado, juntamente com o número n constitui a chave privada!

Após a geração do par de chaves, é possível realizar a encriptação e decifração de mensagens! O processo de encriptação funciona da seguinte forma:

$$c(m) = m^e mod(n) \tag{2.4}$$

Onde m é o dado que será cifrado, e o par e e n são os valores que constituem a chave pública. O processo de decifração é muito similar, como segue:

$$m(c) = c^d mod(n) \tag{2.5}$$

Onde c é o dado cifrado, e o par d e n são os valores que constituem a chave privada! Como foi visto, a encriptação depende apenas da chave pública e a decifração, apenas da privada, o que condiz com o sistema de criptografia de chaves assimétricas.

A encriptação produzida pelo algoritmo RSA, rigorosamente falando, não é inquebrável (na verdade, nenhuma encriptação é), o problema é que para realizar as operações necessárias para encontrar a chave privada, o esforço computacional necessário vai muito além da tecnologia existente hoje, o que permite garantir que um computador executando um algoritmo que tente quebrar a encriptação (ou seja, encontrar a chave privada) levaria anos para encontrá-la! É por isso que o RSA é até hoje um dos algoritmos mais seguros existentes na área de segurança da informação. Consequentemente, ele é amplamente utilizado em diversas operações computacionais que envolvam informações sigilosas.

Agora que foi explicado a teoria por trás do algoritmo RSA, demonstraremos na prática na Seção 3 - Funcionamento do Sistema, como ele pode ser implementado em linguagem C, desde a geração das chaves, passando pela encriptação e decifração, até um algoritmo que executa uma tentativa de quebra da mensagem cifrada.

3 Funcionamento do Sistema

O sistema requisita um texto, que pode ser inserido via console, através do endereço de um arquivo ou de uma lista de exemplos pré-definidos. Tal informação será armazenada em um arquivo (*text/exported.txt*).

Depois disso, o sistema oferece a opção de inserir dois números primos para gerar as chaves públicas e privadas, que ao serem inseridos irão passar pelo teste de primalidade de Miller-Selfridge-Rabin (MSR). Esse por sua vez é um teste de natureza probabilística, que basicamente procura testemunhas contra a condição de primalidade. Se não encontrado, ele julga o número um possível primo. Tem uma margem de erro de 25 porcento, pode ser reduzida repetindo o passo o qual o algoritmo gera números aleatórios para um mesmo número. As chaves serão armazenadas em dois arquivos na pasta *keys*.

Após gerado as chaves, o usuário poderá pedir para que o texto informado seja encriptado. Esse texto codificado será gravado em um arquivo (*messages/encrypted.txt*). Para descriptar o texto, o usuário pode selecionar um arquivo externo ou deixar a opção padrão que é a referente ao arquivo que acabou de ser dito. O texto descriptado será gravado em um arquivo (*messages/decrypted.txt*).

O sistema também é capaz de buscar a chave privada através do método de força bruta, o qual tenta todas as chaves possíveis, comparando o resultado de cada chave com um dicionário de palavras pré-definidas, contidas no arquivo (*dicionario.pt.txt*). Tal algoritmo possui um tempo de verificação muito elevado, se tornando um método muito ineficiente para chaves de grandes valores. Entretanto, um curioso fato foi verificado que um mesmo código pode ser quebrado por diferentes chaves, logo se concluiu a não unicidade de uma chave.

4 Análise do Sistema

Através do uso do sistema em um computador com o sistema operacional Debian, processador dual-*core* 3Ghz e com 4GB de memória RAM, foi realizado uma análise de eficiência das atividades de encriptação, decriptação e quebra de chave por força bruta. A Tabela 4 trata sobre os valores gerados por arquivos de textos distintos, enquanto a Tabela 4 relaciona os valores gerados pelo mesmo arquivo de texto.

Arquivo	**Primos**	**Chave Pública**	**Chave Privada**	**T. de Encriptação**	**T. de Decriptação**	**T. de Quebra**
Dostoiévski	61,13	793 7	793 103	0s	0s	5s
JK Rowling	197,607	119579 5	119579 95021	0s	0.5s	26s
Spotted	1021,3037	3100777 7	3100777 2211943	0s	32s	93s
Dostoiévski	10141,16547	167803127 7	167803127 23968063	0s	180s	300s+

Tabela 1 – Análise de Eficiência do Sistema (Genérico)

Primos	**Chave Pública**	**Chave Privada**	**T. de Encriptação**	**T. de Decriptação**	**T. de Quebra**
13,61	793 7	793 103	0s	0s	0.3s
347,743	257821 3	257821 171155	0s	0.1s	124s
1109,2741	3039769 3	3039769 2023947	0s	15s	330s
10589,15377	162827053 3	162827053 108534059	0s	13m26s	1200s+

Tabela 2 – Análise de Eficiência do Sistema (Arquivo Dostoiévski)

5 Considerações Finais

Neste trabalho procuramos esclarecer ao leitor os fundamentos da criptografia, exibindo uma explicação teórica e prática do algoritmo RSA, que constitui uma das técnicas mais confiáveis da área e que é amplamente utilizada em operações computacionais que lidem com informações sigilosas

A importância da criptografia na atualidade é incontestável, por estar presente em diversas situações do nosso cotidiano, mesmo que não percebamos. Desde uma simples troca de *e-mails* até operações bancárias, são necessárias técnicas que garantam a integridade das informações contidas no processo em questão. Mostramos também uma análise do algoritmo que tenta descobrir a chave secreta e porque o RSA é considerado tão seguro.

Procuramos demonstrar ao leitor a aplicação da teoria dos números descrevendo seu papel no algoritmo RSA. O algoritmo RSA conta com diversos artifícios e propriedades disponibilizados pela teoria dos números. A ironia é que, Leonhard Euler formalizou a teoria dos números há mais de 200 anos, demonstrou e introduziu diversas propriedades que são aplicadas atualmente nas diversas tecnologias da informação. A base do algoritmo RSA, que por sua vez forma o alicerce de toda a comunicação digital atual, foi fundamentada há mais de 200 anos. O próprio Euler jamais imaginaria que sua contribuição poderia gerar um impacto tão grande, mesmo que de forma discreta, em toda a sociedade contemporânea. Isso só nos faz refletir a importância da matemática e os frutos que esta disciplina milenar produz quando aplicada no desenvolvimento de tecnologias para o homem.

Referências

SHOKRANIAN, S.; SOARES, M.; GODINHO, H. *Teoria dos Números*. 2. ed. [S.l.]: Editora UnB, 1998. Citado na página 5.

YASCHENKO, V. V. *Cryptography: An Introduction*. 18. ed. [S.l.]: Library of Congress Cataloging-in-Publication Data, 2002. Citado na página 4.