Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 8, n. 1, 2021.

Trabalho apresentado no XL CNMAC, Evento Virtual - Co-organizado pela Universidade do Mato Grosso do Sul (UFMS).

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Criptografia RSA: Uma abordagem alternativa utilizando Inteiros de Gauss e Inteiros de Eisenstein

Manoela D Lourdes Alves Barbosa Pessoa [1]
Escola Politécnica da Universidade de Pernambuco
Guilherme Pontes de Oliveira Lima[2]
Escola Politécnica da Universidade de Pernambuco
Emerson Alexandre de Oliveira Lima, DSc[3]
Escola Politécnica da Universidade de Pernambuco

A rápida evolução da comunicação em redes, sobretudo com a introdução de tecnologias, tais como a 5G, e a crescente migração de serviços para plataformas digitais implicam em desafios cada vez maiores na manutenção da privacidade e confiabilidade dos dados. Neste contexto, os algoritmos criptográficos desempenham um papel de crescente importância na segurança de dados contra ataques maliciosos. Este artigo descreve uma abordagem de criptografia assimétrica mediante uma modificação do algoritmo clássico RSA proposto por [5], introduzindo chaves nos Domínios de Fatoração Única conhecidos como Inteiros de Gauss e Inteiros de Eisenstein. Esta modificação parece reduzir os cálculos complexos envolvidos no algoritmo RSA fornecendo o mesmo nível de segurança de chaves inteiras com uso de uma menor quantidade de bits na chave.

Define-se *inteiros de Gauss* como o subconjunto $\mathbb{Z}[i] = \{a + bi | a, b \in \mathbb{Z}\} \subset \mathbb{C}$ dos números complexos. Em outras palavras, um inteiro de Gauss é um número complexo de partes real e imaginária, ambas inteiras. Define-se, ainda, os *inteiros de Eisenstein*[4] como o subconjunto $\mathbb{Z}[\omega] = \{a + b\omega | a, b \in \mathbb{Z}\} \subset \mathbb{C}$ dos números complexos da forma $z = a + b\omega$ nos quais a e b são inteiros e $\omega = \frac{-1+i\sqrt{3}}{2} = e^{\frac{2\pi i}{3}}$ é raiz cúbica primitiva da identidade. A caracterização dos primos e unidades em $\mathbb{Z}$, $\mathbb{Z}[i]$ e $\mathbb{Z}[\omega]$ podem ser encontradas em [4] e [2].

Tanto os inteiros de Gauss, quanto os inteiros de Euler, assim como os próprios inteiros formam anéis que são, cada um, Domínios de Fatoração Única, ou seja, qualquer inteiro, inteiro de Gauss ou inteiro de Eisenstein é produto unicamente definido, a menos da ordem de elementos e do produto por unidades de elemenetos do anel que são nele irredutíveis, o que no caso de Domínios de Fatoração Única, são equivalentes ao conceito de primos do anel [4].

O *RSA*[5] é um algoritmo computacional de criptografia baseado em duas chaves, uma mantida privada e outra divulgada como chave pública, sendo então algoritmo criptográfico assimétrico ou criptografia de chave pública [6]. O algoritmo baseia-se na dificuldade computacional de encontrar os fatores de um número composto com grande quantidade de bits em sua representação binária, sendo comum pares de chaves de 256, 512 e até 1024 bits. Toda codificação do RSA pode ser feita em qualquer domínio de integridade [3].

Para os testes de segurança, utilizou-se a abordagem proposta pelo RSA Laboratories, o chamado The RSA Challenge Numbers[6], que consiste em fatorar o produto de primos de grande quantidade de bits cada. O RSA-100, por exemplo, é um número inteiro composto de 100 dígitos (330

[1]mpessoa.@fis.mat.br
[2]gpol@poli.br
[3]eal@poli.br
[4]também conhecido por inteiros de Euler
[5]Acrônimo Rivest - Shamir - Adleman
[6]http://www.ontko.com/pub/rayo/primes/rsa fact.html

bits) cuja fatoração é conhecida e cujo o intento é testar os algoritmos de fatorações propostos. Atualmente a lista consta com desafios de RSA-100 (cuja solução demanda cerca de uma hora em um computador comum) até RSA-2048 cuja solução não parece viável em um futuro próximo. A implementação do teste proposto neste trabalho consistiu em criar números RSA, i.e., produto de dois primos de um dado anel e fatorá-los utilizando o algoritmo *Quadratic Sieve* (QS) conforme [1]. O tamanho de um número inteiro é sua quantidade de bits. Dos inteiros de Gauss e de Eisenstein o tamanho é a soma da quantidade de bits de suas partes real e imaginária. Foram medidos tempos de CPU em função da quantidade de bits (média da execução de 10 rodadas do algoritmo para casa uma das 10 instâncias de números a serem fatorados). A configuração do computador utilizada foi um processador Intel Core *i*77700 com placa de vídeo dedicada GeForce GTX 1060 de 6GB alocada como GPU e memória RAM de 32GB DDR4 rodando Sistema Operacional Linux. Todos os programas foram implementados em linguagem *C*. Os testes realizados indicam que a complexidade adicional dada no QS para inteiros de Gauss e de Eisenstein parece indicar a segurança do uso de chaves com menores quantidades de bits advindas desses dois domínios de fatoração única em comparação com as chaves classicamente tomadas no domínio dos inteiros. Não houve diferença significativa do tempo necessário ao produto de inteiros, inteiros de Gauss ou inteiros de Eisenstein indicando que a complexidade adicional refere-se, apenas, ao tempo necessário a sua fatoração, mas não às operações convencionais de soma ou produto.
Proposta de trabalhos futuros incluem melhorias na implementação do QS para inteiros de Gauss e de Eisenstein permitindo uma melhor comparação da complexidade em função do tamanho dos números envolvidos. Também é proposta futura a implementação do RSA nesses domínios e a comparação qualitativa da codificação obtida em diferentes domínios via testes de criptoanálise em textos e imagens.

Agradecimentos

O presente trabalho foi realizado com recursos do Edital PIBIC-POLI-CNPq 2019.

Referências

[1] Bressoud, D. and Wagon, S. *A Course in Computational Number Theory*. Key College Pub, New York, 2000.

[2] Conway, J. H. and Guy, R. K. *The Book of Numbers*. Springer-Verlag, New York, 1996.

[3] Das, S. B., Mishra, S. K. and Sahu, A. K. A New Modified Version of Standard RSA Cryptography Algorithm, *Smart Computing Paradigms: New Progresses and Challenges*, Springer Advances in Intelligent Systems and Computing, volume 767, pages 281–287, 2020. DOI: 10.1007/978-981-1309680-924.

[4] Ireland, K. and Rosen, M. *A Classical Introduction to Modern Number Theory, 2nd edition*. Springer-Verlag, New York, 1998.

[5] Jung, A. Implementing the RSA cryptosystem, *Computers & Security*, volume 6, pages 342–350, 1987. DOI: 10.1016/0167-4048(87)90070-8.

[6] Rivest, R., Shamir, A. and Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, volume 21, pages 120–126, 1978. DOI: 10.1145/359340.359342.