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

CryptMath: Funções Inversas em Algoritmos de Criptografia

Salomão Monteiro[1]
GPTICAM/IFAP, Macapá, AP
Eonay Gurjão[2]
GPTICAM/IFAP, Macapá, AP
Klenilmar Dias[3]
GPTICAM/IFAP, Macapá, AP
André Luiz Ferreira[4]
GPECE/IFAP, Macapá, AP

No âmbito geral se trata de um programa de computador com propósito de criptografar informações com níveis de segurança computacional, baseados em propriedades matemáticas que envolvem cálculos de funções inversas aplicadas a linguagem computacional. Têm-se um formato de Web App (aplicativo mobile) que permite transmitir o seu conteúdo em quaisquer dispositivos, sendo: mobile, tablet, desktop [2], podendo ser utilizado em diversas plataformas para troca de mensagens cifradas, assim baseia-se em algoritmos estruturados em funções matemáticas que garantem os pilares de segurança da informação na troca de mensagens, tornando-o confiável e seguro em referência ao padrão ISO 27001.

Objetiva-se com este trabalho a criação de um modelo de criptografia dinâmica e estruturada, utilizando as rotinas de algoritmos ordenados baseados em definições matemáticas, além de incentivar estudantes e leitores a desenvolver trabalhos que possam relacionar ensino, pesquisas e extensão, também fazer uma aproximação de áreas aplicadas, considerando as possibilidades interdisciplinares.

Historicamente um dos primeiros métodos de substituição conhecidos, utilizado pelo imperador romano Júlio César, que de acordo com [1] afirma que "Um código semelhante a este foi usado, pelo ditador romano Júlio César para comunicar-se com as legiões romanas em combate pela Europa."

Este parece ser o primeiro exemplo de um código secreto de que se tem notícia, o código de Júlio César supracitado é um dos mais antigos, consiste basicamente em substituir uma letra do alfabeto pela terceira letra seguinte. Códigos como o de César, padecem de um grande problema, são muito fáceis de "quebrar", pois caso seja interceptado por outrem (interceptador) que não seja o destinatário legítimo, acaba sendo desvendado facilmente, desta forma a simples troca de letras sem associá-las a operações e ou algoritmos mais complexos torna os códigos vulneráveis.

Generalizando este pensamento, poderemos sugerir que qualquer código envolva a substituição de caracteres do alfabeto de forma sistemática por outro símbolo acaba tendo a mesma fragilidade. Isto ocorre pela frequência média com que cada letra aparece em um texto de uma determinada alfabeto é mais ou menos constante, entretendo, observe a seguinte mensagem "zulu zuou zeze" é nítido que a frequência média com que a letra "z" apareceu nesta mensagem é bem diferente da frequência média com que esta letra apareceria em um texto mais extenso, por isso a análise estatística só funciona bem se a mensagem for longa.

[1]salomaolimamonteiro@gmail.com
[2]eonay.web@gmail.com
[3]klenilmar.dias@ifap.edu.br
[4]andre.ferreira@ifap.edu.br

A ideia do algoritmo desenvolvido melhora o nível de segurança, mas de fato como deve ocorrer este processo? Ao submeter um texto de 100 letras; realiza-se uma divisão a cada 20 letras, assim dividindo o texto em partes menores e independentes, cada parte com 20 letras e em cada uma destas partes que seja aplicado um método diferente até que as 100 letras do texto sejam criptografadas.

Para uma criptografia eficiente, o intervalo de letras referente a um método de cifragem, deve ser pequeno o bastante, para não haver análise suficientemente eficaz para encontrar padrões nas substituições de letras, para um texto extenso deveria ser necessário um grande quantitativo de funções, pois quanto maior o texto, mais letras, consequentemente há necessidade de mais funções para criptografar esse texto. Como forma de aproveitar as funções já definidas, utilizou-se de algoritmos com estrutura de repetição para que os intervalos pudessem ser alternados dentro do texto no momento da cifragem independente do tamanho recebido, uma explicação matemática para isso está explicita no exemplo:

Ao supor um intervalo de 82 letras e um quantitativo de 20 funções, as mesmas seriam capazes de criptografar apenas um texto de 20*82=1640 letras, muito pouco por sinal, para um texto, mas ao aplicar o método desenvolvido, em vez de escolher um quantitativo enorme de funções (chaves) para criptografar um texto grande, pode-se apenas transformar as 20 funções escolhidas em um *loop*, de forma que quando o indivíduo informar o caractere de número 1641, sistematicamente o caractere será criptografado pela primeira função, visto que a capacidade total das 20 funções são apenas 1640 caracteres, e isso seguiria em um ciclo para os múltiplos de 1640 +1, a exemplo o caractere 3281 seria criptografado novamente pela primeira função, assim como os caracteres 4921, 6561, 8201, 9841 e assim sucessivamente.

É importante ressaltar que as 20 funções definidas possuem suas respectivas inversas, que segundo [3], qualifica a função inversa como sendo uma $f : A \to B$. A relação $[f^{-1}]$ é uma função de $B \to A$ se, e somente se, f é bijetora [3], por exemplo, ao usar essas 20 funções (chaves) para criptografar uma mensagem seria necessário usar também as inversas dessas funções para descriptografar a mensagem, por exemplo, dentre as 20 funções definidas, 10 são funções exponenciais e 10 são funções afins, logo na ação de descriptografar é necessário o uso das inversas das exponenciais, que no caso são as funções logarítmicas, assim também será necessário o uso das inversas das funções afins. É imprescindível a existência da inversa, caso contrário a mensagem não poderá ser decodificada, pois o embaralhamento precisa ser desfeito de alguma forma, e essa forma é justamente a ideia da inversa.

Agradecimentos

Agradeço ao GPTICAM e ao Instituto Federal do Amapá.

Referências

[1] Coutinho, S. Criptografia. Rio de Janeiro: IMPA, 2016. 217 p. ISBN: 978-85-244-0340-8.

[2] Goodbarber. Apps nativos e construtor Progressive Web App. Disponível em: `https://pt.goodbarber.com/pwa/create/`. Acesso em: 12 de mar. de 2021.

[3] Iezzi, G.; Murakami, C. Fundamentos de matemática elementar. 8. ed.São Paulo: Atual, 2004. v.1. ISBN: 978-85-357-0455-6.