O Método Diffie-Hellman

Mário Leite

...

Estou sempre defendendo a importância da Lógica na programação; e observem o que aconteceu comigo quando estava criando mais um programa para o meu novo livro.

A ideia era criar um programa que gerasse uma chave secreta para dois usuários (**A** e **B**) na troca de mensagens, encriptando-as (como faz o *WatsApp*) de modo que só os dois tivessem acesso aos seus conteúdos. Para fazer isto existem muitos algoritmos que implementam vários métodos; e um desses métodos é o "Diffie-Hellman", onde os dois usuários concordam com um divisor modular **D** (normalmente um número primo), e cada um escolhe seus próprios números secretos (N**A** e **NB**), para uma base **B** comum (também um número primo). Estes números são utilizados em dois cálculos modulares: **Xa=(B^{NA}) Mod D** e **Xb=(B^{NB}) Mod D**, (onde **Xa** é o valor calculado do usuário **A** para ser enviando ao usuário **B** e **xb** o valor de **B** a ser enviado ao usuário **A**) para depois calcular a chave final de encriptação da mensagem, com cálculos modulares intensos. Observem que a expressão B^{NA} ou B^{NB}, pode gerar valores extremamente altos e inviabilizar todo o processo. Por exemplo, vamos supor que a base **B** acordada entre os dois interlocutores fosse **389**, o número secreto do interlocutor **A** fosse **235** e o divisor modular comum fosse **391**. Então, o valor a ser enviado de **A** para **B** seria calculado do seguinte modo: **Xa = 389^{235} Mod 391** (resto da divisão de 389 elevado a 235 por 391). E vocês sabem quanto dá 389235!? Um número muito grande! Este número é: **4.346666415102061535075914826674 5E+608**. É, basicamente, o número 4 seguido de seiscentos e oito zeros; um número assustador, não é!? Isto inviabilizaria qualquer cálculo "normal" dentro de um programa! E como resolver esta situação!?

Usando APENAS a Lógica criei um pseudocódigo para encontrar a solução; depois testei o algoritmo no Visualg e, finalmente, codifiquei o programa em Python.

A **figura 1** mostra o resultado da expressão acima: **145**; este seria o valor a ser enviado do interlocutor **A** para o interlocutor **B**. Assim, pessoal, mais uma vez venho aqui desfazer aquela ideia de que para programar tem que saber uma linguagem de programação "poderosa", "orientada a isto", "orientada à aquilo", "na moda", "na plataforma X", "com *framework* Y", etc, etc; não é bem assim! Observem que solução inicial que apresentei está quase em Português (em Portugol). O IMPORTANTE é saber que a solução tem que vir da aplicação da Lógica pois nem o C, nem o C++, nem o C#, nem o Java, nem o Python, nem qualquer linguagem de programação resolve o problema; eu só codifiquei em Python, DEPOIS que encontrei a solução em *algoritmo/pseudocódigo*. NENHUMA linguagem de programação resolve o problema; apenas a LÓGICA, BOM SENSO e, em casos como este, conhecimentos básicos de Matemática!

```
"D:\Postagens no Face\Códigos\Diffie-Hellman\venv\Scripts\python.exe"
"D:/Postagens no Face/Códigos/Diffie-Hellman/Diffie-Hellman.py"

Digite a base: 389
Digite a potência: 235
Digite o módulo divisor: 391
Chave enviada:  145

Process finished with exit code 0
```

Figura 1 - Saída do programa em Python

```
Programa "PotenciaModular"
//Calcula o resto da divisão entre números muito grandes utilizando potência //modular.
//Em Pseudocódigo
//Autor: Mário Leite-
//-------------------------------------------------------------------------------------
   Declare R, j: inteiro
      a, x, n: inteiro
Início
    Escreva("Entre com a base: ")
    Leia(Base)
    Escreva("Entre com a potência: ")
    Leia(Expo)
    Escreva("Entre com o divisor modular: ")
    Leia(Divi)
    Chave ← 1
    Para j De 1 Até Expo Faça
       Chave ← (Chave*Base) Resto Divi
    FimPara
    EscrevaLn("")
    EscrevaLn("Resultado de (389^235 Mod 391): ", R)
FimPrograma
```

```
'''
Programa "PoenciatModular"
Calcula o resto da divisão entre números muito grandes
utilizando potência modular.
Em Python
Autor: Mário Leite
'''
endfor = "endfor"
enddef = "nddef"
#-------------------------------------------
def FunCalculaEnvio(NBase,NExpo,NDivi):
    AparaB = 1
    for k in range(1,(Expo+1)):
        AparaB = (AparaB *NBase) % NDivi
    endfor
    return AparaB
enddef


#Rotina principal
print("")
Base = int(input("Digite a base: "))
Expo = int(input("Digite a potência: "))
Divi = int(input("Digite o módulo divisor: "))
Chave = FunCalculaEnvio(Base,Expo,Divi)
print("Chave enviada: ", Chave)
#FimPrograma-----------------------------------
```