

# Primeiro Trabalho Prático

Neste trabalho prático de Introdução à Criptografia, cada aluno (individualmente) deverá apresentar um programa completo de sua autoria para implementar um conjunto de tarefas solicitadas, relacionadas ao protocolo de prova de identidade descrito.

## O Problema

Nós vimos em aula o protocolo de conhecimento zero para prova de identidade proposto por Feige-Fiat-Shamir. Neste protocolo, um participante demonstra para outro que conhece a *raiz quadrada* de um número em algum módulo sem que o outro consiga obter nenhuma informação sobre o valor da raiz.

Teodoro é conhecido e confiado por todos. Ele escolheu  $p$  e  $q$ , dois primos mantidos em segredo e publica  $n = p \cdot q$ . Teodoro fornece para Fábio dois números  $s$  e  $v$  tais que  $s^2 \cdot v \equiv 1 \pmod{n}$ . Ou seja, o inverso de  $v$  é resíduo quadrático de  $s$ . Teodoro publica  $v$  de Fábio.

Agora, Fábio quer convencer Patrícia que conhece  $s$ , mas não quer revelar o seu valor (caso contrário Patrícia poderia fingir ser Fábio). E aqui começa o protocolo de identificação.

Neste trabalho implementaremos o protocolo e também uma tentativa de fraude. As tarefas essenciais são:

1. Fábio escolhe aleatoriamente  $r$ , um número entre 1 e  $n$  tal que  $p \nmid r$  e  $q \nmid r$ , calcula  $x = r^2 \pmod{n}$  e envia  $x$  para Patrícia;
2. Patrícia escolhe um bit  $b$  aleatório e envia para Fábio;
3. Se  $b = 0$ , Fábio envia  $r$  para Patrícia. Se  $b = 1$ , Fábio envia  $y \equiv r \cdot s \pmod{n}$ ;
4. Se  $b = 0$ , Patrícia verifica que  $r^2 \equiv x \pmod{n}$ . Se  $b = 1$ , Patrícia verifica que  $v \cdot y^2 \equiv x \pmod{n}$ . Se o teste falhar, Patrícia rejeita a identificação de Fábio. Se o teste passar, Patrícia pode pedir para Fábio recommear.

Após uma certa quantidade de passos, Patrícia se convence da identidade de Fábio e eles podem começar a trocar segredos. A quantidade de passos é escolhida por Patrícia a ponto de se sentir segura que Fábio é quem diz ser, mas será um número não maior que 50, pois Fábio também pode cansar de gerar números aleatórios e fazer contas.

Todos os valores fornecidos serão inteiros entre 0 e  $2^{64} - 1$ , compatíveis com uma variável inteira sem sinal de 64 bits. Para efetuar multiplicações modulares será preciso usar uma variável temporária de 128 bits antes de computar o resto.

Neste trabalho você implementará um algoritmo capaz de comportar-se ora como Fábio, ora como Patrícia, ora como Teodoro e ainda, como Ester (descubra sobre ela mais a frente).

## Passos

O seu maior desafio neste trabalho é computar o inverso multiplicativo (para calcular  $v$  a partir de  $s$  e fazer todas as contas modulares sempre computando o resto para que o resultado não estoure a capacidade da variável).

Para calcular o inverso de um inteiro  $a$  no módulo  $n$ , você deve computar o mínimo divisor comum de  $a$  e  $n$  usando o Algoritmo Estendido de Euclides, que também computa os inteiros  $s$  e  $t$  tais que  $as + tb = \text{mdc}(a, n)$ . O inverso de  $a$  será o inteiro  $s$ . Note que, se  $s$  for negativo, para este trabalho, ele deverá ser convertido em um equivalente positivo.

## Solução

Seu código deve ser escrito em Linguagem C para compilar com o gcc e executar em Linux num sistema de 64 bits. Ele deverá se chamar `soueu.c` e será compilado pelo comando

```
gcc -o soueu soueu.c -std=c99 -pedantic
```

Ao ser executado, deverá ler da entrada padrão as tarefas e seus parâmetros, executá-las e escrever na saída padrão os resultados ou indicativos de erro.

Cada tarefa será composta de uma linha cujo primeiro elemento é um caractere que identifica a tarefa e os seguintes são seus argumentos, conforme tabela abaixo:

Para executar em cada um dos modos (como um participante), seu programa será invocado da seguinte forma:

```
./soueu F|P|T|E
```

A letra de argumento escolhe a pessoa.

No primeiro caso (Fábio) os comandos desejados serão:

Tarefa	Identificador	Parâmetros	Saída esperada	Saída erro
Identificar	I	$n s v$	C	E
Iniciar	X		C $x$	E
Responder	R	$b$	C $x_b$	E
Terminar	T		C	

Tabela 1: Comandos da entrada padrão para Fábio.

Na tarefa Identificar, o programa deve armazenar os valores de  $n$ ,  $s$  e  $v$ , verificando se  $s^2 v \equiv 1 \pmod{n}$ . Caso não esteja correto, um erro deve ser reportado.

Na tarefa Iniciar, gere o número aleatório  $r$  e compute  $x$ , escrevendo-o na saída. Isto só falha se não foi identificado ainda.

Na tarefa Responder, leia o bit fornecido (sempre 0 ou 1), calcule e imprima o valor  $x_b$  adequadamente. Isto é,  $x_b = r$  quando  $b = 0$  e  $x_b = r \cdot s \bmod n$  se  $b = 1$ . Neste momento, o valor de  $r$  não é mais necessário e o mesmo pode ser descartado. Você não deve responder a mais de uma tentativa de validação. Erros devem ser reportados caso já tenha executado uma resposta, caso o bit fornecido seja inválido ou caso não foi iniciado ainda.

Na tarefa Terminar, encerre o programa.

No segundo caso (Patrícia) os comandos desejados serão:

Tarefa	Identificador	Parâmetros	Saída esperada	Saída erro
Inicializar	I	$n v t$	C	E
Receber compromisso	Q	$x$	C $b$	E
Validar resposta	V	$x_b$	C $i$	E $i$
Finalizar	T		C $i f$	

Tabela 2: Comandos da entrada padrão para Patrícia.

O comando Inicializar recebe como argumentos  $n$  e  $v$ , que Teodoro publica, além de  $t$ . Este número indica quantas respostas corretas Patrícia precisa para se sentir segura de que Fábio foi identificado e seu valor esperado é entre 3 e 50.

O comando Receber Compromisso repassa o valor de  $x$  gerado por Fábio (ou alguém que se diz Fábio). Armazene  $x$ , gere um bit aleatório  $b$ , armazene-o e o imprima na saída. Este comando não pode ser repetido antes de validar a resposta.

O comando Validar Resposta repassa o valor de  $x_b$  fornecido por Fábio para Patrícia validar usando os valores de  $x$  e  $b$ . Se tudo estiver certo, exiba quantas repetições ainda faltam para confiar em Fábio ( $i$ ). Quando  $i = 0$ , Fábio está identificado e nenhuma outra tentativa de compromisso deve ser aceita (ou seja, devem retornar erros). O erro também deve indicar quantas iterações faltavam para terminar.

Se estiver executando o papel de Teodoro, estes são seus comandos:

Tarefa	Identificador	Parâmetros	Saída esperada	Saída erro
Inicializar	I	$p q$	C	E
Autenticar	A		C $v s$	E
Terminar	T		C	

Tabela 3: Comandos da entrada padrão para Teodoro.

No comando de Inicializar, são dados  $p$  e  $q$ , os primos secretos. O valor de  $n = pq$  deve ser calculado.

No comando Autenticar, você gerará os valores secreto  $s$  e público  $v$  para um indivíduo, geralmente Fábio. Lembre-se que  $s^2 \cdot v \equiv 1 \pmod{n}$ . Ou seja, você deve antes escolher um  $s$  aleatoriamente e entre 2 e  $n - 1$ . Então, compute  $s^2 \bmod n$  e, por fim, o inverso deste número, que será o valor de  $v$ .

O comando Terminar é autoexplicativo.

Ester vai tentar se passar por Fábio, mas ela não conhece  $s$ . Ela corre seus riscos.

No comando Inicializar, Ester sabe o que todos sabem sobre Fábio,  $v$  e o valor público  $n$ .

Mesmo só com isso, Ester vai gerar uma mensagem para enganar Patrícia com um pouco de sorte. No comando Preparar, Ester gera um  $x$  e  $x_b$  que seriam aceitos por Patrícia se ela escolhesse o bit  $b$  dado.

O comando Sorte é chamado quando Ester esta observando o Fábio responder às perguntas da Patrícia e Ester conseguiu se infiltrar na comunicação. Ester, então, pediu para o desatento Fábio uma segunda

Tarefa	Identificador	Parâmetros	Saída esperada	Saída erro
Inicializar	I	$n \ v$	C	E
Preparar	P	$b$	C $x \ x_b$	E
Terminar	T		C	
Sorte	S	$x_0 \ x_1$	C $s$	

Tabela 4: Comandos da entrada padrão para Ester.

resposta para o mesmo desafio e ela obteve os valores de  $x_b$  tanto para o bit 0 quanto para o bit 1 ( $x_0$  e  $x_1$ ). Calcule e imprima o valor de  $s$  de Fábio, que agora está nas mãos de Ester.

Em C, utilize o tipo `__uint128_t` para armazenar resultados intermediários em 128 bits.

## Entrega e Avaliação

Seu programa será compilado e executado da seguinte forma:

```
./soueu <F|P|T|E> <entrada >saida
```

A saída será comparada com a esperada. Conjuntos de exemplos de entradas e saídas esperadas serão disponibilizados no Moodle (ava.ufms.br).

O programa deve ser entregue pelo Moodle até às 23:55 do dia 25/11/2019. A minha opinião sobre seu código pode ser influenciada pela quantidade e gravidade dos avisos que o compilador emitir e resultar em redução na nota.

A avaliação considerará o código fonte e sua execução correta. Comente claramente o que cada trecho de código faz. Quanto mais claro e simples, melhor. A nota vai de 0,0 a 10,0, distribuídos da forma abaixo (sujeita a alteração até uma semana antes da entrega do trabalho):

Modo	Tarefa	Pontuação
Fábio	Identificar	0,5
Fábio	Iniciar	0,5
Fábio	Responder	1,5
Fábio	Terminar	0,5
Patrícia	Inicializar	0,5
Patrícia	Receber compromisso	0,5
Patrícia	Validar resposta	1,5
Patrícia	Terminar	0,5
Teodoro	Inicializar	0,5
Teodoro	Autenticar	1,5
Teodoro	Terminar	0,5
Ester	Inicializar	0,5
Ester	Preparar	1,5
Ester	Sorte	1,5
Ester	Finalizar	0,5

Tabela 5: Pontuação para avaliação.

Cada tarefa será testada com variadas entradas e sua nota será diretamente proporcional a quantidade de respostas corretas em relação ao número de testes. Se houver um problema com os argumentos, o seu programa deverá indicar com a saída “E”. Se você não implementou uma das tarefas, escreva uma linha apenas, contendo “X”, para que suas demais tarefas sejam consideradas.

Plágio será *severamente* punido. O trabalho é individual!