Universidade Federal do Ceará -UFC
Campus de Crateús
Ciência da Computação

Relatório

Alunos: Bruno Teixeira de Sousa e Francisco Antonio F. de Almeida
Professor: Roberto Cabral Rabelo Filho

03 de Dezembro de 2017

Universidade Federal do Ceará -UFC
Campus de Crateús
Ciência da Computação

Relatório

Relatório sobre a implementação do algoritmo RSA e assinatura digital.

Alunos: Bruno Teixeira de Sousa e Francisco Antonio F. Almeida

Professor: Roberto Cabral Rabelo Filho

03 de Dezembro de 2017

Conteúdo

1 Introdução

Partindo da necessidade de colocar em pratica os conceitos visto no decorrer da disciplina de criptografia principalmente em relação ao algoritmo de RSA e assinatura digital, foi idealizada a problemática de implementar o algoritmo RSA e um protocolo que forneça assinatura digital usando o mesmo. Basicamente o RSA é um algoritmo de encriptar e descriptar dados e o protocolo que forneça assinatura enfatizando quem é o remetende real daquele determinado dado.

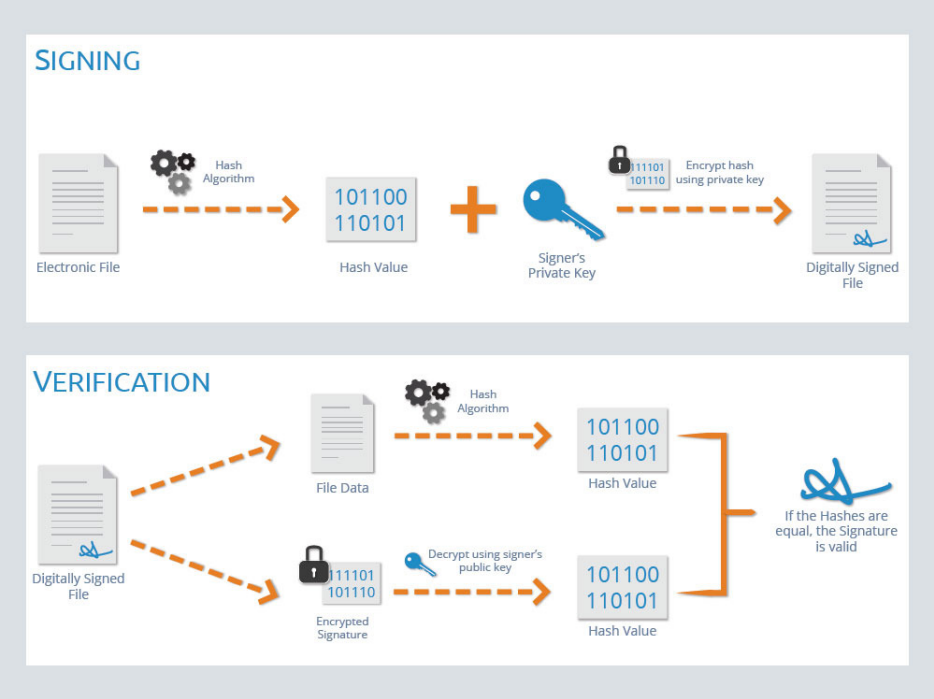
2 Desenvolvimento

Objetivo central é implementar algoritmo RSA e usando o mesmo implementar a assinatura digital. Tivemos as seguinte sub- rotinas:

- modInverse que encontra o inverso multiplicativo de a mod m
- mulmod calcula (a * b) % mod
- Miller realiza o teste de primalidade de um número usando teorema Miller-Rabin.
- gcd calcula o mdc de dois números
- squareMultiply calcula a^b % n

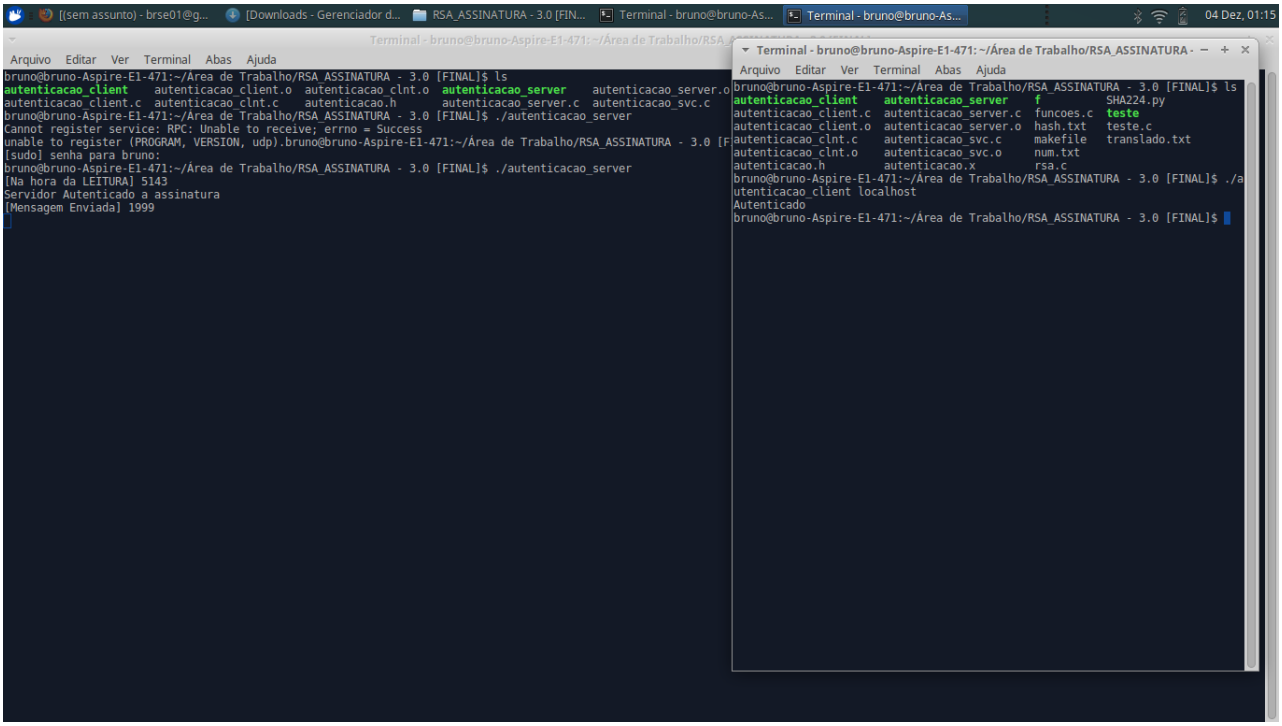
Os passos seguintes são estrutural:

- Geramos dois primos p e q
- Calculamos o
$$\phi(n) = (p-1) * (q-1)$$
- Escolha um
$$e, 1 < e < \phi(n), talque, ee\phi(n)sejaprimos$$
- calcula "d"seja inverso multiplicativo de "e"
- Usando o algoritmo em python de sha224 gera a hash da mensagem
- Encripta tanto a mensagem quando a hash da mensagem
- Após isso envia-se os dados para o endereço desenjado é la usando a chave privada vai ter a forma de descriptar tanto a assinatura quanto a mensagem é confirmar se realmente foi enviada por quem dizer ser enviada.



Ilustrativa da assinatura digital

Utilizando Chamada remota de procedimento (RPC, acrônimo de Remote Procedure Call) que é uma tecnologia popular para a implementação do modelo cliente-servidor de computação distribuída. Uma chamada de procedimento remoto é iniciada pelo cliente enviando uma mensagem para um servidor remoto para executar um procedimento específico. Uma resposta é retornada ao cliente. Uma diferença importante entre chamadas de procedimento remotas e chamadas de procedimento locais é que, no primeiro caso, a chamada pode falhar por problemas da rede. Nesse caso, não há nem mesmo garantia de que o procedimento foi invocado. Com intuito de virtualizar uma conversa real entre duas pessoas ou apenas uma troca de dados entre duas pessoas para ver os algoritmos implementados atuando de forma mais fiel ao cenário da atualidade. Segue uma imagem do resultado do algoritmo executando a parte do servidor é cliente


[(sem assunto) - brse01@g...
[Downloads - Gerenciador d...
RSA_ASSINATURA - 3.0 [FIN...
Terminal - bruno@bruno-As...
Terminal - bruno@bruno-As...
04 Dez, 01:15
Arquivo Editar Ver Terminal Abas Ajuda
bruno@bruno-Aspire-E1-471:~/Área de Trabalho/RSA_ASSINATURA - 3.0 [FINAL]$ ls
autenticacao_client autenticacao_client.o autenticacao_clnt.o autenticacao_server autenticacao_server.o
autenticacao_client.c autenticacao_clnt.c autenticacao.h autenticacao_server.c autenticacao_svc.c
bruno@bruno-Aspire-E1-471:~/Área de Trabalho/RSA_ASSINATURA - 3.0 [FINAL]$ ./autenticacao_server
Cannot register service: RPC: Unable to receive; errno = Success
unable to register (PROGRAM, VERSION, udp).bruno@bruno-Aspire-E1-471:~/Área de Trabalho/RSA_ASSINATURA - 3.0 [F
[sudo] senha para bruno:
bruno@bruno-Aspire-E1-471:~/Área de Trabalho/RSA_ASSINATURA - 3.0 [FINAL]$ ./autenticacao_server
[Na hora da LEITURA] 5143
Servidor Autenticado a assinatura
[Mensagem Enviada] 1999
Terminal - bruno@bruno-Aspire-E1-471:~/Área de Trabalho/RSA_ASSINATURA
Arquivo Editar Ver Terminal Abas Ajuda
bruno@bruno-Aspire-E1-471:~/Área de Trabalho/RSA_ASSINATURA - 3.0 [FINAL]$ ls
autenticacao_client autenticacao_server f SHA224.py
autenticacao_client.c autenticacao_server.c funcoes.c teste
autenticacao_client.o autenticacao_server.o hash.txt teste.c
autenticacao_clnt.c autenticacao_svc.c makefile translado.txt
autenticacao_clnt.o autenticacao_svc.o num.txt
autenticacao.h autenticacao.x rsa.c
bruno@bruno-Aspire-E1-471:~/Área de Trabalho/RSA_ASSINATURA - 3.0 [FINAL]$ ./a
utenticacao_client localhost
Autenticado
bruno@bruno-Aspire-E1-471:~/Área de Trabalho/RSA_ASSINATURA - 3.0 [FINAL]$

Resultado

3 Modo de usar

Modo de usar é bem simples. Segue os passos de forma a poder utilizar o algoritmo de forma comoda é rápida:

1. execute o comando make
2. execute o seguinte comando no terminal .\ autenticacao_server
3. execute o seguinte comando no terminal .\ autenticacao_cliente localhost

4 Dificuldades Encontradas

No decorrer do desenvolvimento tivemos alguns problema na geração dos primos, tivemos exito na implementação do algoritmo RSA e assinatura Digital. Com resalva que não conseguimos implementar de modo que aceita-se strings, contudo aceita números na dimensão de long long um tamanho até razoável.

5 Conclusão

Consideramos, portanto, que o trabalho foi concluido com sucesso, mesmo contendo algumas resalvas, mesmo assim asseguramos a integridade e a essência do algoritimo RSA e da assinatura digital. Com isso porporcionando um maior aprendizado sobre o conteúdo de forma a fixa mais nitidamente o mesmo.