Universidade Federal
do Espírito Santo

CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA

Sistemas Distribuídos – Turmas 01 e 02 – 2022/1
Prof. Rodolfo da Silva Villaça – rodolfo.villaca@ufes.br
Monitor: Eduardo M. Moraes Sarmento – eduardo.sarmento@ufes.br
Laboratório VII – Assinaturas Digitais

1. Objetivos

- Reimplementar o protótipo desenvolvido no Laboratório VI, considerando aspectos de Assinatura Digital na troca de mensagem entre os pares;
- Explorar bibliotecas na linguagem de programação Python para geração de chaves criptográficas assimétricas, criptografia e descriptografia, assinatura e geração de certificados digitais;
- Experimentar a verificação de assinaturas de mensagens por meio de chaves públicas/ privadas;

2. Mineração de Criptomoedas

Init

O sistema deverá inicializar sempre no estado *Init*, onde cada participante terá o seu identificador próprio na rede (*NodeID),* gerado aleatoriamente como um inteiro de 32 *bits*. Após gerar o seu *NodeID*, o participante deverá enviar seu *NodeID* para os demais por meio da publicação de uma mensagem texto (*InitMsg*) no *broker* de comunicação *Pub/Sub*. Considere a existência de um *broker* único e todos os participantes (nós) deverão publicar e assinar a fila "*sd/init*" neste *broker.*

Importante:

- A mensagem *InitMsg* não deverá ser assinada, e seu conteúdo deve, obrigatoriamente, ser formatado como um arquivo JSON contendo somente o *NodeID* em formato texto em representação hexadecimal;
- Faça com que o número de participantes no sistema seja um valor configurável, que pode ser alterado como um parâmetro de configuração dos nós durante sua inicialização;
- Teste com no mínimo 4 nós;
- Desconsidere falhas de indisponibilidade ou desconexão por parte dos participantes do sistema distribuído.

Após o envio da sua própria mensagem de inicialização, o nó deverá aguardar a chegada de um total de *n-1* mensagens de inicialização distintas, provenientes dos demais participantes. Ao receber mensagens de todos os participantes, encerra-se a fase de inicialização (*Init*).

Importante:

- O nó deverá reenviar continuamente sua própria mensagem de inicialização com seu *NodeID* enquanto as *n* demais mensagens não forem recebidas. A definição do intervalo e periodicidade do reenvio é prerrogativa de cada grupo;
- Entretanto, trate as situações onde poderá haver a recepção de mensagens duplicadas, provenientes do mesmo nó, decorrentes do reenvio das *InitMsg*.

PubKey

Após concluir a fase de inicialização (*Init*), tem-se inicio a fase de troca das chaves públicas entre os participantes do sistema (*PubKeys*). Todos os nós participantes deverão possuir um par de chaves pública e privada, de criptografia RSA e comprimento 1024 bits. A chave privada (*PrivKey*) de cada participante é de uso local e não pode ser compartilhada entre os demais participantes. A chave pública deverá ser trocada entre todos os participantes.

a) Graduação: a troca das chaves públicas entre os nós deverá ser feita por meio de uma mensagem *PubKeyMsg*, na fila "sd/pubkey". A mensagem deverá estar obrigatoriamente no formato JSON e conter, exatamente, o *NodeID* e a chave pública (*PubKey*) em formato hexadecimal.

Mensagem	**Conteúdo JSON**	**Tipos Internos**	**Fila no Rabbit**
InitMsg	*NodeId*	*int*	*sd/init*
PubKeyMsg	*NodeId* *PubKey*	*int* *str (hexadecimal)*	*sd/pubkey*

Importante:

- Todos os nós deverão aguardar o recebimento das chaves públicas de todos os demais *n-1* participantes, e armazenar localmente essas chaves (pode ser em memória ou disco);
- Para evitar que o sistema trave por inanição, reenvie sua própria chave regularmente até que esta fase do sistema seja concluída – da mesma forma como foi feito o reenvio das mensagens de inicialização.

b) Pós-Graduação: cada nó deverá enviar o seu *NodeID* e a sua *PubKey* na fila *sd/pubkey* por meio da *PubKeyMsg*. Essa fila deverá ser assinada somente por um processo que representará uma espécie de "autoridade certificadora" (CA). Ao receber a mensagem *PubKeyMsg*, a "CA" irá gerar um certificado digital contendo a PubKey do nó em questão. Ao final desta fase, a "CA" terá *n* certificados, um para cada participante do sistema.

Cada nó, por sua vez, deverá assinar a fila *sd/cert*. Após os *n* certificados digitais terem sido gerados, a "CA" deverá publicar *n* mensagens do tipo *CertMsg* na fila *sd/cert*, onde cada mensagem *CertMsg* contém o *NodeID* e o certificado *Cert* de cada um dos *n* nós. Cada nó deverá extrair a *PubKey* dos demais n-1 participantes do sistema a partir dos seus certificados.

Mensagem	**Conteúdo JSON**	**Tipos Internos**	**Fila no Rabbit**
InitMsg	*NodeId*	*int*	*sd/init*
PubKeyMsg	*NodeId* *PubKey*	*int* *str (hexadecimal)*	*sd/pubkey*
CertMsg	*NodeId* *Cert*	*int* *str (PEM)*	*sd/cert*

Election e Challenge

Após concluir a fase *PubKeys*, quando todas as *n* chaves públicas dos participantes do sistema estarão compartilhadas entre eles, segue-se para a fase de eleição do coordenador (ou líder) do grupo (*Election*) e, posteriormente, para o estado *Challenge* (desafio), exatamente conforme

especificado no Laboratório VI. A partir deste estado do sistema, as mensagens *SolutionMsg*, *ChallengeMsg*, *ResultMsg*, *ElectionMsg* deverão estar assinadas e, ao receber qualquer mensagem, é necessário certificar-se da autenticidade da origem por meio da sua assinatura digital.

Como premissas, em seu projeto assuma que:

a) Não haverá entrada/saída dinâmica de nós no sistema (*churn*);
b) O *broker* Pub/Sub é único, infalível, e de conhecimento prévio por todos os participantes;
c) Os nós participantes do sistema podem não ser bem-comportados, havendo possibilidade comportamento malicioso para atrapalhar o funcionamento do sistema – falhas bizantinas. Teste essa possibilidade em sua apresentação, pois essa condição será testada na avaliação do trabalho.

4. Instruções Gerais

- O trabalho pode ser feito em grupos de até 3 alunos. Não serão aceitos trabalhos em grupos de mais de 3 alunos. Se for necessário, o professor reserva-se no direito de ter que subdividir grupos já existentes;
- Data de Entrega: 13/07/2023, por meio da Sala de Aula Virtual da disciplina no *Google Classroom*, na atividade correspondente ao Laboratório VII. 1 (uma) submissão por grupo é suficiente;
- Deve-se submeter apenas o *link* para o repositório virtual da atividade (*Github*, *Bitbucket*, *Google Colaboratory* ou similares) contendo: i) códigos-fonte; ii) instruções para compilação e execução; iii) relatório técnico (.pdf ou README); e iv) vídeo curto (máx 5 min) mostrando uma execução, resultado e análise dos resultados encontrados – considere o caso de falhas bizantinas em seus testes;
- O relatório técnico deverá conter: a descrição da implementação, testes e resultados encontrados;

Bom trabalho!