UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Tarefa Prática – Protocolo STS (*Station-to-Station*) e comunicação cifrada com sockets

Suponha que você está implementando comunicação segura em um ambiente interno de uma empresa. Os usuários da empresa possuem certificados digitais e chaves privadas guardadas no keystore Java da empresa (ou então guardados em arquivos cifrados em disco).

Você deve:

1. Desenvolver a implementação do protocolo STS (*Station-to-Station*) que é uma versão modificada do Diffie-Hellmann que usa assinatura digital para fornecer autenticação. O protocolo que deve ser implementado é o seguinte:

$$A \rightarrow B:\ g^a, A$$
$$B \rightarrow A:\ g^b, cert_B, E_K\{sig_B\{g^a, g^b, A\}\}$$
$$A \rightarrow B:\ cert_A, E_K\{sig_A\{g^a, g^b, B\}\}$$

E_K - Cifragem usando a chave K (**a chave K é a chave de sessão criada com o cálculo do Diffie-Hellmann: g^{ab}**)

sig_A ou sig_B – assinatura usando a chave privada de A ou a chave privada de B

$cert_A$ ou $cert_B$ – certificado digital X.509 de A ou B

A ou B – identificadores das partes (nome de A e nome de B). O identificador de B, a parte A consegue "deduzir" do certificado digital de B

2. Para implementar este protocolo talvez seja necessário que A e B tenham que trocar/conhecer os parâmetros: g, certificado de A, certificado de B. Os certificados podem ser armazenados no keystore Java ou então em arquivos no formato PEM que podem ser lidos do disco.
3. Depois de estabelecer a chave de sessão K com o uso do protocolo STS, o programa deve enviar as mensagens cifradas entre cliente e servidor usando os sockets. O cliente envia msg cifrada usando a chave de sessão e o servidor decifra a mensagem e escreve na tela a mensagem decifrada. Toda a comunicação entre Alice e Bob deve acontecer usando sockets.
4. Nenhuma chave estática e nem IV devem estar escritos no código. Se precisar guardar algum parâmetro, use o keystore ou algum arquivo cifrado.
5. Para cifragem simétrica deve ser usado o modo AES/GCM.
6. Existem vários exemplos de códigos que devem ser usados:
    a. O projeto Sockets (classes Cliente e Servidor) tem exemplo de uso de Sockets em Java.
    b. Também existe o exemplo de código do projeto testeOAEPRSA para ver como funciona o uso do RSA em Java. NÃO use o BaseRSAExample.java. USE APENAS o OAEPPaddedRSAExample.java como base para usar o RSA.
    c. O projeto testeSignature (PKCS1SignatureExample.java) tem exemplo de assinatura digital usando o RSA com SHA256.
    d. O projeto testeDH (BasicDHExample.java) tem exemplo de funcionamento do DH.

Referências:

- STS Protocol - https://en.wikipedia.org/wiki/Station-to-Station_protocol
- Key Exchange Protocols - https://web.stanford.edu/class/cs259/WWW08/slides/04-Key%20Exchange.pdf
- Sessão 2.3.2 do livro *Protocols for authentication and key establishment*, dos autores *Colin A. Boyd e Anish Mathuria* (livro disponível no moodle).