DI/FCT/UNL
Mestrado Integrado em Engenharia Informática

Segurança de Redes e Sistemas de Computadores
2º Semestre, 2015/2016
(14/Abril/2016 TP1B)

T1-P: Teste Prático
Enquadramento e Desenvolvimento do Trabalho Prático nº 1
Duração (Ref.): 45 minutos

Questão 1 (Fase 1)

Considere a sua implementação e a possibilidade de suportar por configuração (ou parametrização) diferentes suites criptográficas e os respetivos parâmetros, na configuração de uma sessão.

Nota: os nomes nas alíneas a seguir (RC6, DESede, HMACSHA3-384 ou DESMAC) são exatamente os nomes usados para os respetivos algoritmos nas concretizações dos *crypto-providers* (ex., Bouncy Castle) para utilização com Java JCE.

a) Diga, de acordo com processa as configurações criptográficas, como seria a configuração na sua implementação (suite criptográfica e respetivos parâmetros necessários e adequados na configuração) se quiser ter uma sessão que vai usar:

Como cifra simétrica: o algoritmo Blowfish, com chave de 448 bits (como cifra simétrica), com *padding* PKCS7 e operado em modo CFB (Nota, este algoritmo, algoritmo operando sobre blocos-base de 64 bits.

Como função MAC: o algoritmo HmacSHA256 para ser usado como função HMAC.

b) Idem, como em a) - mas agora a configuração a usar é a seguinte:

Como cifra simétrica: o algoritmo Triple DES – DESede, usando uma chave de 168 bits, com *padding* PKCS5 e usando o modo ECB.
Como função MAC: um função CMAC usando o algoritmo AESCMAC

Nota: os nomes DESede e AESCMAC são exatamente os nomes usados para os respetivos algoritmos nas concretizações dos *crypto-providers* (ex., *Bouncy Castle*) para utilização com JAVA-JCE. Para além dos dados que acima são fornecidos, deve indicar como proporia os outros valores a colocar na sua configuração.

Questão 2 (Fase 1)

Na implementação de referencia do enunciado do trabalho, indicava-se que deveria ser implementado um formato universal de segurança para as mensagens na aplicação (*Chat/Messaging*). Usando uma notação formal este formato pode ser representado da seguinte forma:

HEADER || DADOS || MAC

Em que:
HEADER = PHASE || VERS|| LEN // informação enviada em claro (ou *plaintext*)

DADOS: dados da mensagem, parte enviada cifrada (para garantir confidencialidade) e com prova de autenticidade e integridade, na seguinte forma (duas opções possíveis):

Formato A) DADOS = { TYPE || uID || r1 || r2 || Dados da mensagem || MAC_{Kh} }$_{Ks}$
ou

Formato B) DADOS = { TYPE || uID || r1 || r2 || Dados da mensagem }$_{Ks}$ || MAC_{Kh}

a) Para que servem na sua implementação os valores r1 e r2 e que propriedade de segurança asseguram (de acordo com a tipologia de ataques da *framework* X.800 e modelo de adversário considerado) ?

b) Sobre a utilização dos valores r1 e r2 na sua implementação da fase 1: use o seu código para mostrar onde e como são estes valores utilizados e processados, no envio e na verificação da recepção da mensagem, de modo a garantirem a propriedade de segurança que referiu em a) de forma completa e correta.

Questão 3 (Fase 1)

Na especificação do formato de segurança da mensagem e tendo em conta os dois formatos alternativos indicados (conforme indicado na anterior **questão 2**)

a) Qual dos formatos (**A ou B)** está suportado na sua implementação ? Justifique mostrando e anotando no seu código que esse é o formato suportado.

b) Apresente uma modificação que permitisse suportar o outro formato alternativo.
(**Nota:** *mostre especificamente que linhas do seu programa mudaria e como mudaria)*

c) Vai usar na sua implementação uma configuração que envolve a utilização de uma simples função de síntese (*secure hash*), ex., SHA-256 ou outra função de síntese considerada segura, em vez de uma função MAC (HMAC ou CMAC). De acordo com a sua implementação concreta do trabalho, isso altera as propriedades de segurança e as garantais da mesma ? Sim ou Não ? Justifique.
Nota: *pode justificar com uma argumentação que tenha em conta o código da sua implementação e deve utilizar o próprio código como suporte da sua argumentação).*

Questão 4 (Fase 2).

Não tendo feito a fase 2, pode ainda assim tentar responder.

a) Imagine que nas configurações criptográficas subjacentes à sua implementação dos formatos de segurança das mensagens trocadas entre os cliente e o SSOServer e entre os clientes (quando já estão na sessão) se decide usar apenas uma função de síntese (secure *hash*, ex., SHA-256 ou SHA-512) em vez de uma função MAC propriamente dita (ex., HMAC ou CMAC). Isso mudaria alguma coisa ao nível das garantais e propriedades de segurança da comunicação ? Justifique.

b) Indique no seu código, onde é que o SSOServer (após verificar o controlo de acesso) processa e envia a um cliente que vai entrar numa sessão a suite criptográfica a usar na sessão e, nomeadamente, a chave de sessão a usar.

c) Indique no código do SSOServer onde e como é que este está a implementar o controlo de detecção e defesa contra *message-replaying* durante o protocolo de autenticação e distribuição de chaves aos clientes quando estes pretendem entrar numa sessão.