* 1. i)

RV

yi

1.2

As “*Stream Ciphers”* ao contrário das “*Block Ciphers”* não têm um tamanho fixo e invariável de bits que devem ser processados, a cifra é feita bit a bit usando uma “*key stream*” permitindo assim que blocos mais pequenos, do que os valores habituais nas “*block ciphers”,* sejam processados.

1.3

Para não permitir que sejam identificados padrões que possam levar a alguma previsão dos dados que se encontram cifrados.

2

2.1

O esquema MAC cifra toda a informação das chaves e certificados usando uma chave que estará protegida por palavra-chave, isto permite que a informação não seja lida permitindo ainda que seja verificada a integridade do *keystore*.

2.2

Caso os bits fornecidos ao método doFinal não sejam de um tamanho múltiplo ao tamanho de um bloco este irá inserir padding no final, sendo que o padding apenas deve ser inserido no último bloco. O método update não irá inserir padding, irá apenas guardar em buffer os bits que não atingiram um bloco até ser chamado o método doFinal ou novamente o update.

3

3.1 São gerados tickets para autenticar o cliente perante o serviço, estes tickets são cifrados usando a chave do cliente ou serviço, consoante o seu destinatário, e contêm um *timestamp* que irá impossibilitar ataques de replay.

3.2 Não, pois quando o *ticket* é gerado este é gerado usando informação de B. Logo, se B tentar usar para outro serviço este não irá ser válido.

3.3 Não, pois teria que ser enviado em claro o *ticket* e este contém a chave de sessão. Caso alguém escuta-se a comunicação e obtivesse a chave de sessão poderia identificar-se como sendo A e pedir acesso a serviços.

3.4 Não, pois caso o *authenticator A* fosse em claro, dado a ser um esquema MAC, o atacante poderia obter TA e usando este poderia replicar as mensagens para B e quando B pedisse confirmação de posse da chave de sessão, enviando TA cifrado com a chave de sessão, este poderia simplesmente enviar o que obteve na mensagem em claro, sem ter que proceder a qualquer decifra, validando assim a conversa.

4

4.1 Tem que se obter o certificado do emissor e verificar, usando a chave pública, se o certificado que temos foi realmente emitido usando a chave privada daquele certficiado.

4.2 É usado o número de sequência das mensagens para evitar ataques de replay, esse número é usado na geração de uma marca MAC, o que torna impossível replicar mensagens fora de ordem.

4.3 No protocolo handshake são geradas as chaves usadas para garantir a segurança do canal, essas chaves são geradas usando como base uma outra chave. Esta é gerada aleatoriamente pelo cliente e protegida usando a chave pública do servidor, o facto de esta chave ser aleatória torna impossível que seja usado um ataque de replay pois duas instâncias do protocolo handshake iriam originar chaves distintas.

4.4 A última mensagem do cliente e do servidor irá conter uma marca MAC gerada usando toda a conversa entre os dois sendo a chave o master secret para que o outro possa verificar se ambos viram o mesmo, caso algum hacker modifica-se as mensagens e dado que este não sabe o master secret ele não irá conseguir alterar esta mensagem de forma a que o servidor e/ou o cliente validem a conversa.