INE5680 – SEGURANÇA DA INFORMAÇÃO E DE REDES – Prova 1 – 04/10/2013 – Turmas A e B – Prova A

NOME : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ MATRÍCULA : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Indique (Verdade/Falso), sublinhando no texto e comentando, brevemente, o porquê de sua resposta: (0,10 cada certa = 0,5)

1. (Verdade/Falso) Para um algoritmo de criptografia simétrica ser computacionalmente seguro, um

dos seguintes critérios é atendido: (a) o custo para quebrar a cifra é muito maior do que valor da informação cifrada; (b) o tempo exigido para quebrar a cifra é muito maior ao tempo de vida útil da informação. Estas são boas razões para aceitarmos/confiarmos em algoritmos de criptografia simétrica E por isso, utilizamos, até hoje, algoritmos antigos que já foram quebrados, como o caso do uso do DES.

1. (Verdade/Falso) Várias técnicas tem sido propostas para distribuição de chaves públicas.

Praticamente todas essas propostas podem ser agrupadas nos seguintes esquemas gerais: (a) anúncio público de chaves públicas, (b) diretórios distribuídos disponíveis publicamente, (c) autoridade de chave pública e (d) certificados de chave pública. O GnuPG gerencia a distribuição de chaves públicas através do anúncio de chaves públicas. GnuPG usa diretórios distribuídos disponíveis publicamente, que são inclusive, replicados em servidores na Internet.

c. (Verdade/Falso) Qualquer mensagem pode ser criptografada com o algoritmo do acordo de Diffie-Hellman. Diffie-Hellman não criptografa, apenas resolve o acordo de uso, por ambas as partes se comunicando, de uma chave compartilhada.

d. (Verdade/Falso) Um email recebido portando um link suspeito, mas que é não é aberto, constitui uma ameaça, mas se aberto, dependendo do que existe no link, pode ser uma invasão por software, via algum programa de vírus, cavalo de tróia ou um *worm*. Se você não clicou no link, nada irá acontecer. O ataque só se concretiza caso você clique.

e. (Verdade/Falso) O modo de cifra EBC (cifra um-a-um), para determinadas aplicações, é mais seguro que o modo de cifra CBC (cifra de encadeamento de blocos). CBC é mais seguro do que EBC, porque CBC faz o encadeamento usando os blocos e EBC faz a criptografia de blocos, um-a-um. Mas, EBC serve para aplicações com pequena informação a ser protegida.

2. Alice *(A)* assina digitalmente uma mensagem *M* para Bob, usando sua chave privada *PRA* sobre o *H(M)* com o algoritmo de assinatura RSA. Bob obtém a chave pública *PUA* relacionada a *PRA*. (1,0)

1. Como o requisito de segurança de ***não-repúdio***, por parte de Alice, é garantido pela verificação da assinatura ?   
     
   Uma assinatura digital garante não-repúdio, porque uma vez que Alice criptografe o hash da mensagem M com sua chave privada KRA, somente a chave pública KUA correspondente da KRA pode verificar a assinatura, o que garante que foi Alice quem enviou a mensagem M.
2. Suponha que Alice revele acidentalmente sua chave privada *PRA*para um terceiro. Como o ***não-repúdio*** pode ser garantido neste caso ?

A revelação da chave privada de Alice, KRA, acidentalmente ou maliciosamente, acaba impedindo a garantia de se saber quem enviou, de fato, a mensagem.

1. Três amigos, A, B e C, residentes em cidades distantes desejam trocar informações pela Internet de forma segura, usando uma chave de sessão KS. Um deles propôs o seguinte protocolo para troca da chave simétrica. Somente um deles, C, possui um par de chaves pública/privada (KU/KR). (1,5)

A : Gera N pseudo-randômico, N é um nonce

A 🡪 B : N , N não está sendo criptografado ....

A 🡪 C : N , N não está sendo criptografado ....

C : Gera Rpseudo-randômico, calcula X = N ⊕ R, X é um *nonce* gerado por C e   
 usado no cálculo de X.

C 🡪 B : EN ( X)

C 🡪 A : [ EN ( X) || EKR ( N ) ]

A : DN ( X ) N é usado como se fosse uma chave ???? Na realidade N não é para   
 ser usado assim, pois N é um *nonce*.

A : N ==DKU [EKR ( N)]

A : Gera KS**,** calculaR = N ⊕X R deve ser um *nonce* gerado por A, em função   
 de X, que pode ser decifrado.

A 🡪 B : ER ( KS )

A 🡪 C : ER ( KS )

Pergunta-se:

1. A, B e C podem, ou não, trocar mensagens seguras, cifradas com KS  ?

Resposta: NÃO PODEM.

O protocolo somente utiliza chaves simétricas para troca da chave Ks.

1. Caso não possam, indicar os erros no protocolo.

Alguns erros identificados, claramente, no protocolo proposto:   
  
(a) N trafega em texto plano, não sendo criptografado; neste caso, ocasionando insegurança no protocolo, pois, se N for capturado, um atacante pode usá-lo, como se fosse o usuário real, praticando um ataque de repetição.

1. N pode ser utilizado para decifrar X.
2. R, sendo uma chave que criptografa a chave de sessão KS, precisa ser enviado por A para B e C, para estas partes poderem decifrar a chave de sessão KS (esta parte não está na descrição fornecida do protocolo), no prosseguimento do protocolo. Mas, se A enviar R, para B e C, a chave R poderá ser capturada no meio de comunicação por um atacante (man-in-the-middle attack) ...

4. Considere a figura seguinte, um protocolo que mostra um procedimento de autenticação de um usuário de um terminal bancário. Suponha que um terminal bancário é uma entidade T e o sistema central (banco) uma entidade B. Considere para o terminal o par de chaves, pública e privada, *(PUT, PRT)* epara o sistema de autenticaçãoem B, respectivamente,o par *(PUB, PRB)* de chave pública e chave privada. No protocolo da figura, a **criptografia simétrica**, com uma chave de sessão *CS* é usada no procedimento de autenticação. (1,0)



Altere o protocolo acima, descrevendo formalmente suas etapas, para funcionar com criptografia de chave pública.

1. Considere que o sistema central (banco B) conheça as chaves públicas dos vários terminais *T* (*PUT*). E que esses terminais T conheçam a chave pública do sistema central.  
     
   B : *PUT  (por construção do sistema de segurança) e (PUB, PRB)*T : *PUB* e *(PUT, PRT)*
2. Alguém está querendo usar o terminal T. O protocolo se inicia quando o terminal envia sua identificação *T* para o sistema central B.

T 🡪 B : IT (aqui, o identificador de T não está criptografado, mas se fóssemos fazer assim, usaríamos *PUB(IT),* assumindo-se como acima que T conhece *PUB .*

1. Pelo protocolo, o sistema central B deve enviar uma chave de sessão *CS* para o terminal *T* poder criptografar (usando criptografia simétrica com uma chave de sessão *CS*), através da chave mestra *CT*.

B 🡪 T : *CT*(*CS*), neste caso, a chave mestra CT do terminal T criptografa a chave de sessão CS enviada pelo banco B para o terminal T.

1. Mas, para se usar criptografia de chave pública, o sistema central, agora, se utilizará da chave pública do terminal T (*PUT*), enviando a chave de sessão *CS* criptografada para *T.* Com suachave privada (*PRT)*, o terminal *T* decifra a chave de sessão *CS*.

Para usar criptografia de chave pública, o banco B, agora substitui a chave CT por uma chave pública *PUT*  
O uso da criptografia de chave pública, se faz aqui na etapa 4 e poderíamos ter:  
  
B 🡪 T : *PUT(CS)* T : *PRT(CS)* T : *CS*

1. Com *CS*, o terminal *T* pode cifrar os números *r* supostamente aleatórios gerados por *T* e enviá-los ao sistema central.   
     
   T : E*CS*(r)onde r é *nonce* gerado por T  
   T 🡪 B : E*CS*(r)B : D*CS*(r)  
    B : r
2. De posso do número *r*, o sistema central modifica esse número *r,* adicionando 1, cifrando-o com *CS* e enviando para *T*. O banco B envia *r+1* para T. Lembrem que os números *r* e *r+1* são usados apenas uma vez, para evitar ataques de repetição no procedimento de autenticação de um usuário do terminal *T*. Daí o termo *nonce*, em inglês, para denominar esses números.

B : r+1  
B: E*CS*(r+1)  
B 🡪 T : E*CS*(r+1)  
T : D*CS*(r+1)   
T : r+1

1. O terminal envia sua identificação *iA* e a senha *sA* para o sistema central poder autenticar usando o arquivo de senhas, contendo os valores *hash* das senhas dos usuários do sistema.

O terminal T recebe o número *r+1* e assim, fica sabendo que o banco B recebeu seu número *r*, envoado anteriormente.  
  
T 🡪 B : E*CS*(IA , SA) ,   
  
onde IA é o identificador de um usuário A e AS é a senha do usuário, Ambos os valores são enviados criptografados para o sistema do banco B.

5. Em um sistema B2B web, com as seguintes características e requisitos: (2,0)

- Empresas vendedoras V acessam o sistema B2B e oferecem seus produtos (ofertas).  
 - Empresas compradoras C acessam o sistema B2B para consulta de preços.  
 - Empresas compradoras C acessam o sistema B2B e realizam pedidos.

Apresente um protocolo criptográfico comentado para solucionar os seguintes ataques.

***Ataques empresas vendedoras:***

1. Autenticação, uma outra quer se fazer passar pela empresa fazendo ofertas falsas.
2. Integridade, uma outra quer modificar a oferta da empresa.

***Ataques empresas compradoras:***

1. Autenticação, uma outra quer se faz passar pela empresa fazendo compras falsas.
2. Integridade, uma outra quer modificar a compra da empresa.

Use a seguinte notação:  
   
 V : Empresa Vendedora   
 C : Empresa Compradora  
 B2B : sistema  
 KUv : chave pública de empresa vendedora  
 KRv : chave privada de empresa vendedora  
 KUC : chave pública de empresa compradora  
 KRC : chave privada de empresa compradora  
  
Use o verso da página para sua solução.

Solução:

Os dois tipos de ataques que podem acontecer – “***Ataques empresas compradoras” e os “Ataques empresas vendedoras”*** - visam burlar a autenticação de empresas ou a integridade sobre as mensagens que essas comunicam. Assim, a segurança será garantida, se assinaturas digitais foram utilizadas para assinar e verificar as assinaturas. Usando a notação fornecida em folha à parte desta prova, pode-se ter:

Do lado das empresas vendedoras e o sistema B2B, pode-se estabelecer o seguinte protocolo criptográfico:

V : Gera Oferta

V 🡪 B2B : S**KRv** ( Oferta) || KUv // assina oferta e envia sua chave pública concatenada, para o sistema B2B

B2B : KUv ??? // verifica se empresa cadastrada

B2B : V**KRv** ( Oferta ) // verifica a assinatura da empresa que faz a oferta, conseguindo ver a originalidade da oferta.

B2B : Armazena Oferta

Do lado das empresas compradoras e o sistema B2B, pode-se estabelecer o seguinte protocolo criptográfico:

C 🡪 B2B : Consulta Oferta

C : Gera Pedido

C 🡪 B2B : S**KRc** ( Pedido) || KUc // assina pedido e envia sua chave pública concatenada para o sistema B2B.

B2B : KUc ??? // verifica se empresa cadastrada

B2B : V**KRc** ( Pedido ) // verifica a assinatura da empresa compradora que faz o pedido, conseguindo ver a originalidade do pedido.

B2B : Armazena Pedido

A notação S**KRv** ( Oferta) ou S**KRc** ( Pedido) poderia ser consultada na folha à parte sobre a notação formal.

6. Uma técnica de baixo custo baseada sobre uma chave secreta compartilhada, que tem segurança para muitos propósitos é *Message Authentication Code* (MAC). O método usa uma chave compartilhada K secreta que pode ser distribuída, e serve para autenticar comunicação entre partes, baseando-se sobre essa chave. A figura abaixo mostra o funcionamento de um HMAC -- um MAC usando uma função hash H -- que concatena uma mensagem M com uma chave K, H(M+K) ou H(M || K). O documento assinado é enviado para um receptor. A assinatura com MAC, que usa uma função hash, chamada de chamada HMAC, é verificada quando o receptor desconcatenando M e h, recalcula a função hash H e encontra h = h’. (1,0)

Indique a soma das respostas verdadeiras.

(1) O método HMAC autentica assinaturas com chave secreta K, enquanto Hash H apenas verifica a integridade de M.  
(2) Embora, este método apresente desvantagens, ele apresenta uma vantagem na sua performance porque ele não emprega nenhuma criptografia.  
(4) O protocolo SSL v3.1 (o protocolo TLS) suporta uma variedade de MACs, incluindo o HMAC.  
(16) Criptografia simétrica é 3-10 mais rápida que uma função Hash H. (Falsa)  
(32) O método depende da existência de um canal seguro, através do qual a chave compartilhada K pode ser distribuída. Resposta: Na prova A, a soma é 39 (esta) (1+2+4+32=39), pois a única afirmação falsa é a (16), onde, na realidade, é o contrário, “**Uma função Hash H é 3-10 mais rápida que Criptografia simétrica**”. Aproveitando, o assunto, lembrem sempre que “**Criptografia de Simétrica é bem mais rápida que a criptografia de chave pública**”. Por isso, que a criptografia de chave pública para fazer assinaturas digitais clássicas, é usada sobre o H(M) e não sobre M, que pode ser um arquivo muito grande, e neste caso, o processamento da criptografia de chave pública é bem mais duradouro do que o processamento com função hash. Então, no caso em que o sigilo não seja necessário, e é de interesse á a autenticação, a integridade ou o não-repúdio, pode-se aplicar a criptografia de chave pública, somente sobre o resultado da função hash. Ver a figura 11c fornecida em aula.

Na Prova B, a soma é 51 (1+2+16+32)=51.

(1) Embora, este método apresente desvantagens, ele apresenta uma vantagem na sua performance porque ele não emprega nenhuma criptografia.  
(2) O método HMAC autentica assinaturas com chave secreta K, enquanto Hash H apenas verifica a integridade de M.  
(4) Criptografia simétrica é 3-10 mais rápida que uma função Hash H. (Falsa)  
(16) O protocolo SSL v3.1 (o protocolo TLS) suporta uma variedade de MACs, incluindo o HMAC.  
(32) O método depende da existência de um canal seguro, através do qual a chave compartilhada K pode ser distribuída.