Instituto Superior de Engenharia de Lisboa

LEIC, LEIRT e LEIM

Segurança Informática

Época Normal, Semestre de Inverno, 2022/23 – 23 de janeiro de 2023

Duração: 2 horas

- **Respostas às questões 4, 5 e 6 num grupo diferente de folhas das questões 7, 8, 9 e 10.**
- **Todas as respostas às perguntas de desenvolvimento têm de ser justificadas. Valoriza-se a objetividade das respostas.**
- **Nas questões 1, 2 e 3, indique `V`, `F`, ou deixe a caixa em branco se não tiver a certeza que a afirmação é verdadeira ou falsa. A indicação errada desconta 50% da cotação atribuída à afirmação.**

Número: _____ **Nome**: _____

1. (2) No contexto dos esquemas e primitivas criptográficas simétricas:

   ☐ O algoritmo de *padding PKCS#5Padding* é usado nos modos de operação em bloco (ex: ECB, CBC) e nos modos de operação em stream (ex: Counter, GCM)

   ☐ As primitivas simétricas usadas para garantir confidencialidade usam chaves diferentes para cifrar e para decifrar

   ☐ Os modos de operação ECB e CBC são iguais com exceção do CBC precisar de um vetor inicial e o ECB não

   ☐ O esquema MAC não dá garantias de confidencialidade sendo por isso possível ver a mensagem em claro no canal de comunicação

2. (2) No contexto dos certificados X.509, do protocolo TLS e da biblioteca JCA:

   ☐ No certificado fornecido no segundo trabalho, para representar o servidor `www.secure-server.edu`, a assinatura do certificado foi realizada usando a chave privada associada a outro certificado

   ☐ A chave pública num certificado é sempre usada para validar a assinatura desse certificado

   ☐ A classe Keystore pode guardar certificados que são raízes de confiança mas também chaves privadas e os respetivos certificados

   ☐ O *record protocol* usa chaves simétricas diferentes no sentido cliente→servidor e no sentido servidor→cliente

3. (2) No contexto das normas OAuth 2.0 e OpenID Connect:

   ☐ No protocolo OpenID Connect, a estrutura `id_token` é usada pela aplicação cliente para requisitar mais informações sobre o utilizador (ex: foto de perfil) através da API do *userinfo endpoint*

   ☐ Na *framework* OAuth2, em particular no fluxo *authorization code grant*, o `access_token` entregue à aplicação cliente tem informação sobre o dono dos recursos que a aplicação cliente pode consultar

   ☐ A estrutura designada como `access_token` tem de ser usada nos pedidos à API do servidor de recursos

   ☐ Na requisição ao servidor de autorização no fluxo *authorization code grant* do OAuth2, o parâmetro `state` é usado para proteger a aplicação cliente contra ataques de CSRF (*Cross-site request forgery*)

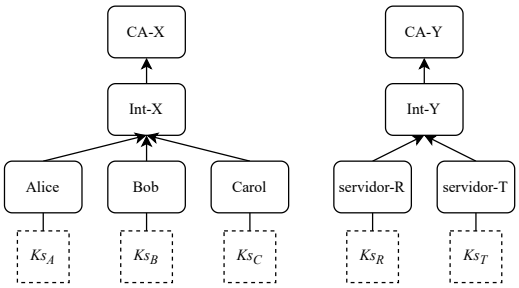
4. (2) Pretende-se desenvolver um novo esquema criptográfico, HCA, para enviar uma mensagem com confidencialidade e autenticidade entre dois participantes (A e B).

   Assume-se que A conhece a chave pública de B (K_eB) e que a chave simétrica k muda em cada comunicação. O novo esquema usa cifra assimétrica (E_a), cifra simétrica (E_s) e MAC (T). O símbolo || representa a concatenação:

   $HCA(K_eB, k, m) = E_a(K_eB)(k) \,||\, E_s(k)(m) \,||\, T(k)(m)$

   Descreva como é feita a decifra e verificação de autenticidade da mensagem 'm', nomeadamente a ordem de operações e as chaves utilizadas (identifique claramente as chaves e indique o seu papel no esquema).

5. (1,5) Considere o diagrama da figura, onde são apresentadas duas hierarquias de certificados semelhantes às usadas no trabalho. CA-X e CA-Y são raízes de confiança, e $Ks_A, Ks_B, Ks_C, Ks_R, Ks_T$ são chaves privadas associadas aos respetivos certificados. Assuma que **Bob** (cliente) irá estabelecer uma sessão TLS com o **servidor-R** com autenticação do cliente. Indique, usando os identificadores da figura, o menor conjunto de certificados e chaves privadas que devem ser instalados no cliente e no servidor.



6. Considere o sub-protocolo *handshake* do protocolo TLS.

   6.1. (1,5) Descreva o mecanismo criptográfico utilizado quando é necessária a autenticação de cliente, nomeadamente as chaves e as mensagens envolvidas?

   6.2. (1,5) No cenário apenas com autenticação de servidor, qual a proteção que existe para detetar ataques de repetição, nos quais o atacante tenta reutilizar as mensagens de cliente de um *handshake* anterior?

7. (1,5) Considere um sistema de armazenamento de palavras-passe as quais são armazenadas na forma $h_u = H(pwd_u)$, sendo H um função de *hash* e pwd_u a palavra-passe do utilizador u. Descreva um ataque a esta forma de armazenamento que **não** implique a utilização da interface de autenticação. Descreva também uma solução para o problema identificado. Admita que a função H é conhecida do atacante.

8. (1,5) Considere uma aplicação *web* que mantém estado de autenticação entre o *browser* e a aplicação servidor usando *cookies*. No cookie é guardado um JSON web token (JWT) com o identificador do utilizador. Como é que a aplicação servidor pode detetar se o conteúdo do cookie foi adulterado no *browser*?

9. Considere uma aplicação web para gestão de projetos de software onde existe a possibilidade de acesso a diferentes recursos (ex: código, documentação, ficheiros de testes).

   9.1. (1,5) Para realizar o controlo de acessos aos recursos foi definida a seguinte política $RBAC_1$ que inclui os papéis *(M)ember*, *(D)eveloper*, *(T)ester* e *(S)upervisor*.

   - $U = \{u_1, u_2, u_3, u_4\}$
   - $RH = \{M \preceq T, M \preceq D, D \preceq S, T \preceq S, T \preceq T_2, D \preceq D_2\}$
   - $UA = \{(u_1, M), (u_2, T_2), (u_3, D_2), (u_4, S)\}$
   - $PA = \{(M, p1), (D, p2), (T, p3), (D_2, p5), (T_2, p4)\}$

   Justifique qual o conjunto total de permissões que podem existir numa sessão com o utilizador u_4?

   9.2. (1) A biblioteca *Casbin* aplica políticas tendo por base dois ficheiros. Explique o objetivo destes dois ficheiros no processo de controlo de acesso, em particular no contexto das regras definidas.

10. (2) No fluxo *Authorization code grant* do protocolo *OAuth2* as mensagens são classificadas como sendo de *front-channel* ou *back-channel*. Explique a diferença entre os dois tipos de mensagens, incluindo a utilização do `client_id` e `client_secret`.