Número: Nome:

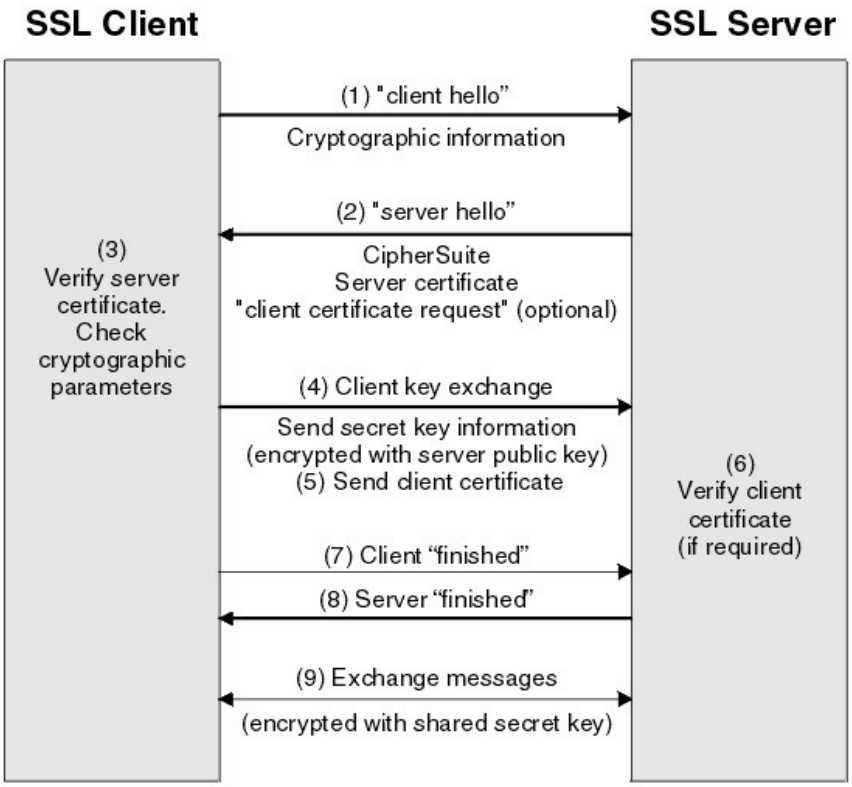
LETI/LEIC 2016/17, 2º Teste de Sistemas Distribuídos

20 de junho de 2017

Responda no enunciado, usando apenas o espaço fornecido. Identifique todas as folhas.
Uma resposta errada numa escolha múltipla desconta 1/(N-1) do valor da pergunta em N alternativas.
Duração da prova: 1h30m

Grupo I [8 valores]

O gráfico seguinte representa uma versão simplificada do protocolo SSL/TLS amplamente utilizado e que recorre a diversas tecnologias aprendidas em Sistemas Distribuídos:



1) [0,5] Na etapa 2 é referido o *Server certificate*. Esse certificado:
   A. Tem a chave pública do *Server*, validade e data assinadas por uma autoridade.
   B. Tem a chave pública e privada do *Server*, este decifra o certificado para obter a chave privada.
   C. Tem a chave pública do *Server*, validade e data assinadas com a chave privada do *Server.*
   D. Tem um prazo de validade durante o qual não pode ser revogado.

   [ ]

2) [1] O certificado é verificado na etapa 3. Descreva genericamente quais os principais passos necessários para efetuar uma verificação o mais rigorosa possível.

Verificar que o certificado corresponde ao nome e domínio do Server
Validar a assinatura usando a chave pública da CA
Validar a validade temporal do certificado através das datas presentes no certificado
Consultar a CA para saber se o certificado não está na lista dos certificados revogados

3) Um dos elementos de segurança presentes no certificado da etapa 3 é uma assinatura digital, cujo esquema abstrato está representado abaixo. Nas alíneas seguintes, escolha entre as opções: Ks_{ervpub} $Ks_{ervpriv}$, Kca_{pub}, Kca_{priv} onde CA é uma autoridade de certificação.

{ Hash(X + outros elementos) }Y

a) [0,5] No contexto das perguntas anteriores qual o valor de X. Justifique.

X =

b) [0,5] Qual o valor de Y. Justifique.

Y =

c) [0,6] Indique e justifique qual o principal objetivo da introdução da função *Hash* referenciada na fórmula.

O principal objetivo da função de Hash é reduzir a dimensão do texto a cifrar na assinatura, uma vez que a cifra assimétrica consume significativos recursos computacionais

d) [0,6] A introdução do *Hash* cria uma vulnerabilidade. Explique qual.

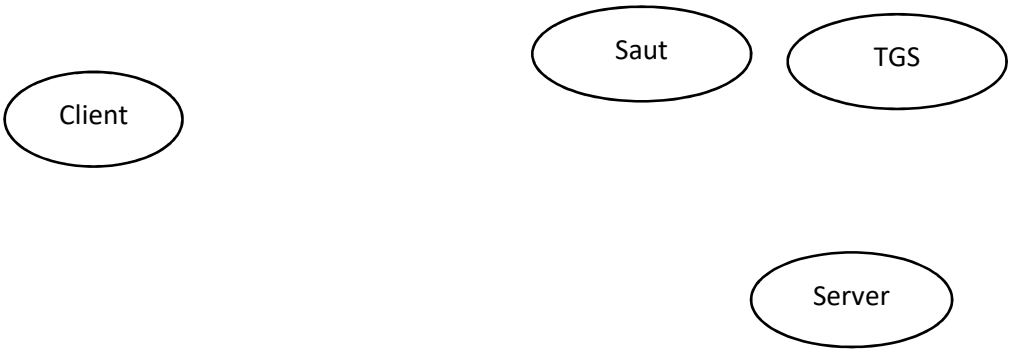
Ao efetuar o resumo para uma informação de dimensão reduzida existe a possibilidade de conseguir criar colisões resultantes de dois textos diferentes que produzam o mesmo hash. A probabilidade de sucesso deste ataque é superior à de uma ataque brute force e é calculada no birthday problem em teoria das probabilidades pelo que o ataque se designa por birthday attack

4) [1] O certificado do servidor recebido na etapa 2 pode ser ter sido emitido por uma CA desconhecida pelo cliente. Caso o cliente opte mesmo assim por aceitar o certificado, corre o risco de ser vítima de um ataque *man-in-the-middle*. Explique em detalhe como este ataque pode ser posto em prática.

O atacante(I) escuta as comunicações e pode inserir ou retirar mensagens. Quando vê a mensagem 2 do protocolo captura-a e construi uma outra em que mantem a identificação do server (nome, domínio), junta uma chave publica sua KpubI e cifra o certificado com uma chave privada que ele também controla KprivCAI. O cliente recebe o certificado e não tendo o certificado de CAI vai tentar obtê-lo, por exemplo pedindo-o à CAI. O atacante tinha previamente produzido este certificado assinado por uma CA legitima e envia-o. O cliente aceita a chave e estabelece com o atacante o canal pretensamente confidencial. O Atacante estabelece com o Server o canal, que por ser opcional a identidade do cliente, será aceite, ficando a intermediar todas as comunicações

Número: Nome:

5) A utilização de certificados no caso habitual do SSL permite estabelecer uma autenticação que não é mútua, uma vez que os clientes podem não dispor de certificado (por exemplo, é o que acontece tipicamente nos *web browsers*). Se o sistema necessitasse de autenticação dos clientes poderia usar Kerberos ou um protocolo semelhante.



Considere os seguintes passos:

1: C, TGS, n1
2: {Kc,tgs, n1 }Kc , ticketc,tgs
3: ticketc,tgs, authc,tgs, S, n2
4: {Kc,s, n2 }Kc,tgs, ticketc,s
5: ticketc,s, authc,s, {pedido} Kc,s, n3
6: $\{n3\}_{Kc,s,}$ $\{Resposta\}_{Kc,s}$

a) [1] Indique no diagrama a que correspondem os 6 passos através de setas, numeração e identificação de qual a operação que realizam.

b) [1] Preencha os elementos na tabela acima com os valores referentes ao protocolo trocados em cada mensagem.

c) [0,8] No protocolo aparece uma estrutura de dados chamada *Authenticator*. Descreva o seu conteúdo e indique qual a sua função.

O authenticator tem o formato auth x,y = {x, Treq} Kx,y. em que x é a identidade de quem envia e Kx,y a chave de sessão entre x e y. Treq é o tempo atual quando o authenticator é produzido. A principal função é evitar replay attacks garantindo a frescura da mensagem, uma vez que os relógios dos sistemas devem estar sincronizados

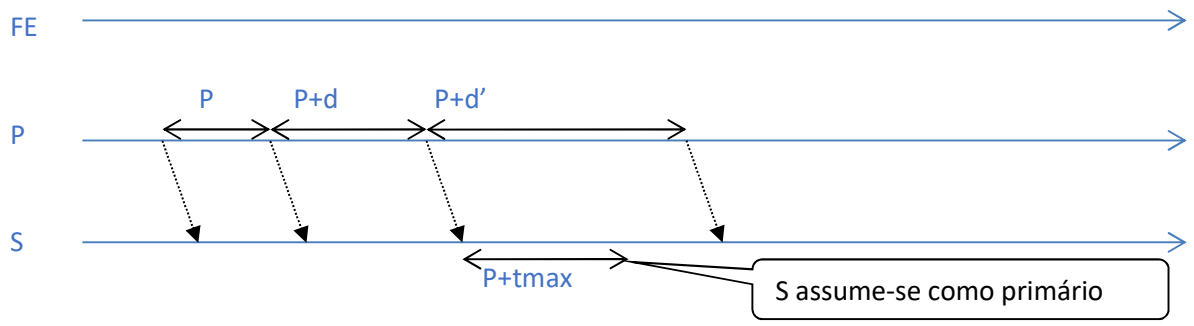
6) [0,5] Vantagens do Kerberos em relação a uma autenticação com base em certificados:

A. Uso de cifra assimétrica.
B. Pode revogar a identidade de forma mais rápida.
C. Não necessita de uma base de dados com todos os clientes.
D. Não depende da sincronização dos relógios.

Grupo II [6 valores]

1. Considere o protocolo "*primary backup*" estudado nas teóricas e aplicado no projeto.
   a. Para funcionar corretamente, o protocolo parte de um conjunto de pressupostos.
      Para cada condição apresentada de seguida, indique se é um dos pressupostos do protocolo ou não. Em caso afirmativo, apresente no diagrama uma execução em que, devido à condição não se ter verificado, ocorreu uma falha; em caso negativo, justifique.

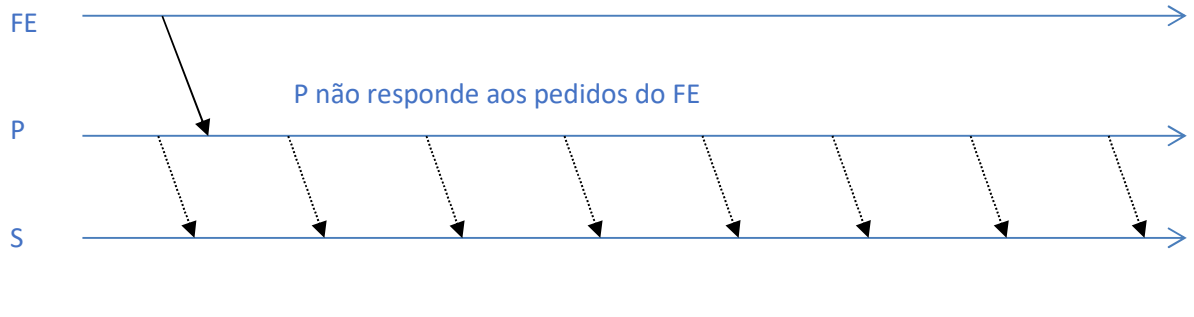
      i. [0,7] É conhecido o desvio máximo permitido entre o relógio local do servidor primário e o relógio local do servidor secundário.
      É pressuposto / ~~não é pressuposto~~ do "*primary-backup*".



Justificação ou explicação do exemplo acima:

O desvio entre os relógios locais de ambos os servidores leva-os a medir o período "P" (que determina o envio das provas de vida do primário) com durações diferentes.
Consequentemente, tal como no exemplo, as provas de vida podem sair do primário com intervalos de tempo que, do ponto de vista do secundário, excedem o tempo limite para este se assumir como novo primário.

      ii. [0,7] Não ocorrem faltas bizantinas dos servidores.
      É pressuposto / ~~não é pressuposto~~ do "*primary-backup*".



Justificação ou explicação do exemplo acima:

Um exemplo possível (entre vários): Primário deixa de responder aos pedidos do front-end mas continua a enviar provas de vida regulares ao secundário.

Número: Nome:

b. Observou-se a seguinte situação:
O atual servidor primário recebe um pedido enviado pelo *front-end* de um cliente.
Ao analisar o identificador único associado ao pedido, o servidor constata que já tinha executado esse pedido antes. A comunicação entre o cliente e o servidor é feita por TCP/IP.
   i. [0,6] O que deve o servidor primário fazer perante este pedido?

   ii. [1] Explique os acontecimentos que levaram à situação descrita.

2. Assuma que opta por utilizar "*quorum consensus*" em vez de "primary backup".
   a. Das condições abaixo enumeradas (i, ii, iii), quais são pressupostos no protocolo "*quorum consensus*"? Responda apenas "sim" ou "não".
      i. [0,4] É conhecido o desvio máximo permitido entre o relógio local do servidor primário e o relógio local do servidor secundário.
      não
      ii. [0,4] Não ocorrem faltas bizantinas dos servidores.
      sim
      iii. [0,4] A rede assegura que as mensagens são entregues pela mesma ordem pela qual foram emitidas (FIFO).
      não
   b. Num sistema de 3 réplicas e usando quóruns de maioria, o estado das réplicas observado num dado instante durante o funcionamento do sistema é o seguinte:
      Réplica 1: valor = 2; tag = {seq-num=11, client-id=2}
      Réplica 2: valor = 4; tag = { seq-num =12, client-id=4}
      Réplica 3: valor = 5; tag = { seq-num =13, client-id=3}

      i. [0,8] Caso um *front-end* tente ler o valor replicado, que valor(es) retornará ao cliente?

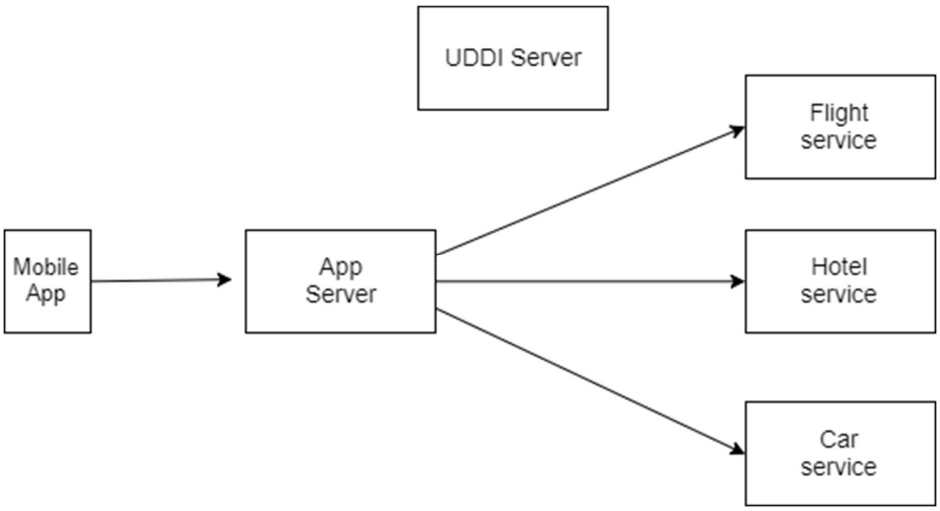
4 ou 5 (consoante o quórum de respostas que receba)

ii. [1] Na situação acima, não há nenhum valor presente em simultâneo numa maioria de réplicas. A situação é possível? Se sim, descreva a sequência de acontecimentos que leva a esta situação. Se não, justifique.

Sim, é possível caso a mensagem com a escrita do valor 5 ainda não tenha chegado às restantes réplicas. Ou seja, caso a operação de escrita ainda esteja em curso.

Grupo III [3,0 valores]

Existe uma App Móvel que pretende oferecer garantias transacionais aos seus utilizadores para reservas de pacotes de férias, que incluem: viagem de avião (Flight), reserva de hotel e aluguer de viatura (Car).
A App comunica com um servidor aplicacional, que contacta três serviços remotos com SOAP/HTTP.



O sistema utiliza uma concretização do protocolo 2PC. Cada serviço oferece localmente as propriedades ACID e está a ser utilizado um controlo de concorrência pessimista. O excerto de código seguinte mostra o que é executado no servidor aplicacional quando o cliente da aplicação móvel pede para comprar o pacote.

Código do Servidor Aplicacional

```
1 void buyTravelPackage(int id) {
2     Tx t = openTransaction();
3     TravelPackage p = getTravelPackage(id);
4     p.flight.book(t);
5     p.hotel.book(t);
6     p.car.book(t);
7     closeTransaction(t);
8 }
```

1) [0,6] No 2PC existem dois papéis a desempenhar: Coordenador (C) e Participante (P).
Assinale na figura com **C** e **P** onde deverão executar-se os respetivos papéis.

C – App Server ou caixa nova à parte
P1 – Flight Service, P2 – Hotel Service, P3 – Car Service

Número: Nome:

2) [0,6] Considere agora o excerto de código do servidor aplicacional. Assinale a(s) linha(s) em que:

Se inicia a transação distribuída	
Onde é atribuído um identificador único à transação distribuída	
Em que o serviço remoto de voos se junta à transação distribuída	
Em que se inicia a votação do 2PC	
Em que se conclui a votação do 2PC	
Em que termina a transação distribuída	

3) [0,4] Considere agora que as linhas 4-6 podem lançar exceções de execução (RuntimeException).
   A. O código deveria incluir um try-catch que engloba as linhas 4-6 e deve fazer AbortTransaction(t).
   B. O código deveria incluir um try-catch que engloba as linhas 4-6 e deve fazer CloseTransaction(t).
   C. O código deveria incluir um throws na linha 1
   D. O código está correto.

   A

4) [0,4] Considere agora que o Coordenador está a fazer a votação 2PC de forma sequencial, pela ordem indicada no código do servidor aplicacional. O serviço de Voo respondeu SIM. O serviço de Hotel não responde. O coordenador deve:
   A. Reenviar o pedido para o serviço de hotel até conseguir resposta.
   B. Consultar outro serviço de hotel em alternativa.
   C. Decidir que o desfecho da transação distribuída é ABORT.
   D. Excluir o serviço de hotel da transação e consultar o serviço de viatura.

   C

5) [0,4] Considere agora que o serviço de voo já respondeu SIM ao Coordenador e está à espera. Após um *timeout* sem que tenha chegado mensagem do Coordenador, o serviço de voo:
   A. Pode decidir que a transação já terminou e foi confirmada.
   B. Pode decidir que a transação já terminou e foi cancelada.
   C. Deve continuar à espera.
   D. Deve contactar o Coordenador e mudar o seu voto para NÃO.

   C

6) [0,6] O 2PC tolera faltas de paragem permanentes de um nó durante a execução do protocolo. Concorda com a afirmação? Justifique com um exemplo (ou contra-exemplo) neste cenário dos pacotes de viagens.

O 2PC não tolera faltas permanentes dos nós. Um contra-exemplo seria
todos os participantes votam SIM, incluindo o Car Service, que depois falha permanentemente.
O coordenador vai depois tentar contactá-lo para concluir a transação e nunca irá obter ACK.
A transação não terminará nem com COMMIT, nem com ABORT.

Grupo IV [3,0 valores]

Considere que está a desenvolver um novo sistema para casas inteligentes (*Smart Home*) com suporte para diversos sensores e atuadores, ilustrados na figura seguinte.

Todos os dispositivos têm um endereço IP na mesma rede doméstica. Os endereços são atribuídos de forma dinâmica através do protocolo DHCP (Dynamic Host Configuration Protocol), que é o mesmo protocolo que é utilizado por um computador pessoal, por exemplo. Isto significa que, ao longo do tempo, a atribuição de IP é variável.

A ligação à Internet é totalmente mediada por um dispositivo *Gateway* que gere também toda a informação sobre o sistema da casa. O sistema *Gateway* implementa uma API com as seguintes operações para ler e escrever valores nos dispositivos que controlam a sua função, por exemplo acender ou apagar uma luz:

```
Value read(Name device, Key key);

void write(Name device, Key key, Value newValue);
```

1) [0,5] O endereço IP pode ser usado como o nome do dispositivo referido nas funções anteriores. Indique uma desvantagem desta opção.

2) [0,5] Proponha um esquema de nomes para os dispositivos domésticos em que cada nome seja **local**. Apresente um exemplo para uma lâmpada inteligente considerando que casa terá algumas dezenas de luzes e que a casa tem várias divisões. Justifique.

Nome da lâmpada:

3) [0,5] A unicidade referencial é uma propriedade base dos sistemas de nomes. Em que consiste? Explique a sua importância no contexto deste sistema.

4) [0,5] Em face das duas respostas anteriores, proponha uma autoridade de nomes e a respetiva implementação. Justifique a sua resposta.

A autoridade de nomes poderá ser o dono da casa, a quem competirá escolher os nomes.
A implementação da autoridade poderá ser feita no sistema do Gateway,
onde se manterá o registo da atribuição de nomes e se garantirá que o esquema de nomes
é respeitado, bem como a unicidade referencial (1 nome refere apenas 1 objeto).

5) [0,5] Pretende-se agora que seja possível aceder à lâmpada inteligente através da Internet. O *gateway* está associado ao nome DNS **myhome.pt**. Proponha um nome **global** à escala da Internet para a lâmpada que permita o controlo remoto. Justifique.

Nome da lâmpada:

6) [0,5] O nome que propôs na alínea anterior é puro ou impuro? Justifique.

Puro/Impuro