	Departamento de Informática **Faculdade de Ciências e Tecnologia** **UNIVERSIDADE NOVA DE LISBOA**

Licenciatura em Engenharia Informática
Sistemas Distribuídos I – 1ª chamada, 26 de Junho de 2006
2º Semestre, 2005/2006

Exame sem consulta com a duração de 2h30min

Aluno nº _____Nome: _____

1. Considere o serviço de sistema distribuído de ficheiros fornecido pela máquina *phoenix* nos laboratórios do departamento. Descreva brevemente como poderia implementar o mesmo serviço com uma arquitectura cliente/servidor particionado e quais as vantagens e desvantagens dessa aproximação face à solução actual (baseada num único servidor).

2. No contexto do sistema DNS, assumindo que não existem problemas de comunicação, indique duas situações que possam levar a que os dados obtidos não sejam os mais actuais. Indique ainda qual das situações é mais fácil de corrigir de forma a que o sistema mantenha um bom desempenho.

3. Como sabe, num ataque de "denial of service", um (ou vários) atacante impedem um servidor de prestar serviço útil a cliente legítimos. Comente a seguinte afirmação: a utilização de canais seguros previne todos os ataques do tipo "denial of service".

Verdadeiro, porque... / **Falso**, porque...

4. Suponha que pretendia criar um serviço de registo de objectos remotos baseado num arquitectura cliente/servidor replicado com funcionalidade semelhante ao do sistema RMI registry, mas que permitisse registar todos os objectos remotos disponíveis numa rede local. Ignorando as falhas de comunicação, descreva brevemente como poderia desenhar este serviço usando comunicação multi-ponto de modo a que os servidores não necessitassem de comunicar entre si (com excepção do momento do arranque). NOTA: Para cada operação (lookup, bind, rebind, list), indique o tipo de comunicação (broadcast, multicast, anycast, unicast) e ordenação de mensagens a usar de forma a que o sistema seja o mais eficiente possível.

5. Como sabe, o sistema Java RMI usa um mecanismo de *garbage collection distribuído* para recolher (remover) os objectos remotos que já não são necessário. Explique o que acontece a um objecto remoto no caso de existir uma falha temporária da rede que impeça a comunicação entre a máquina virtual em que executa um objecto remoto e a máquina virtual do único programa que mantém uma referência para esse objecto remoto.

6. Considere um sistema distribuído cliente/servidor. Suponha que:

   - Existe uma entidade de certificação (CA) em que todos os elementos do sistema confiam. Todos os elementos do sistema conhecem a chave pública da entidade de certificação (KpubCA);
   - O servidor (S) tem um par de chaves assimétricas (KpubS/KprivS) e um certificado assinado pela entidade de certificação (CertS);
   - Cada cliente (C) tem um par de chaves assimétricas (KpubC/KprivC) e um certificado assinado pela entidade de certificação (CertC);
   - No início, tanto o cliente como o servidor apenas conhecem as suas próprias chaves e a chave pública da entidade de certificação. O cliente conhece ainda a chave pública do servidor (KpubS).

   Apresente um protocolo que permita a um cliente executar uma operação no servidor de forma segura, i.e., garantindo o secretismo da operação efectuada (op) e do resultado (res) devolvido pelo servidor, assim como a autenticação dos parceiros (i.e., S deve ter a certeza qual o cliente que o contacta e vice-versa). Na sua solução tenha em atenção possíveis ataques por "replaying" na resposta do servidor (comunicação S->C), apresentando uma solução que minimize a informação que cada entidade necessita de manter. Adicionalmente, procure usar as primitivas criptográficas mais apropriadas tendo em atenção que a operação e o resultado a enviar podem ter uma grande dimensão. O protocolo deve ter, no máximo, duas mensagens, efectuando a seguinte interacção: C->S->C.

   NOTA: Caso não consiga resolver o problema assumindo que os elementos do sistema apenas conhecem as chaves indicadas, indique explicitamente as chaves adicionais que assume serem conhecidas por cada um dos elementos do sistema.

C -> S:

O que garante o secretismo?

Como se garante qual o cliente que enviou a operação *op*?

S -> C:

O que garante o secretismo?

Como se garante que foi S que gerou o resultado *res*?

Como se evita o replaying?

Qual a informação que C e S devem manter, se alguma, para evitar o replaying?

AMBOS: Porque é que a aproximação usada é apropriada para propagar informação de grande dimensão?

Definição dos símbolos usados (complete, se necessário)
op – mensagem codificando a operação a efectuar por C em S
res – mensagem codificando o resultado da operação (a enviar por S a C)
KpubCA – chave pública da entidade de certificação
KpubS/KprivS – par chave pública/privada de S
CertS – certificado de S
KpubC/KprivC – par chave pública/privada de C
CertC – certificado de C

7. Considere que um cliente e um servidor estabelecem o seguinte protocolo para negociarem um par de chaves de sessão – antes do início do protocolo, o cliente apenas conhece o par de chaves assimétricas e o certificado do cliente e o servidor apenas conhece o par de chaves assimétricas e o certificado do servidor.

CLT->SRV: CertCLT
SRV->CLT: CertSRV
CLT->SRV: {Ks1}KpubSRV
SRV->CLT: {Ks2}KpubCLT

CLT->SRV: {m1}Ks2
CLT->SRV: {m2}Ks2
CLT->SRV: {m...}Ks2

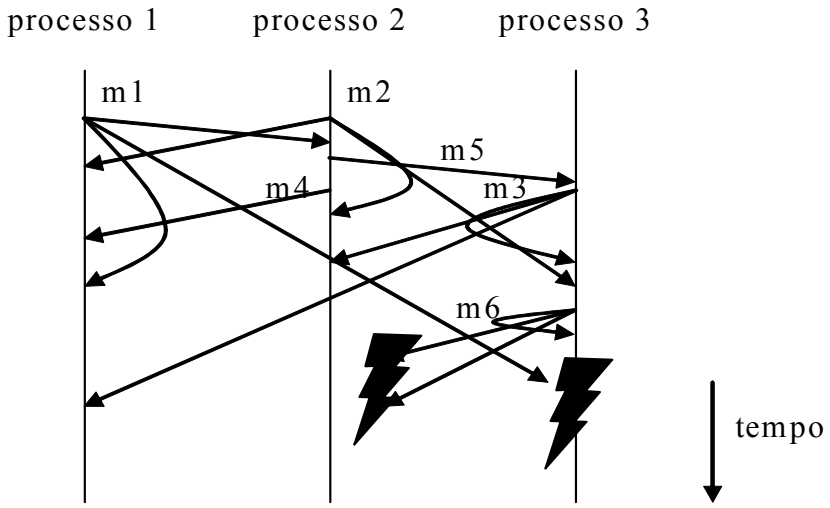
Após o protocolo inicial, a comunicação é apenas unidireccional, sendo as mensagens enviadas do cliente para o servidor cifradas apenas com uma chave (a chave Ks2), como indicado. Assumindo que os certificados trocados são válidos e podem ser verificados por ambos os parceiros da comunicação, responda às seguintes perguntas:

a. Indique, justificadamente, se este protocolo garante o secretismo das mensagens (m1, m2, ...) trocadas e a autenticação dos parceiros da comunicação?

SIM, porque... / **NÃO**, porque...

b. Sabendo que a comunicação será apenas unidireccional, e portanto não sendo necessário negociar um par de chaves de sessão, apresente um protocolo alternativo, baseado no anterior, que garanta o secretismo, autenticação e evite o replaying.

8. Considere o seguinte esquema temporal representando a propagação de mensagens num sistema composto por três processos. As setas indicam a propagação de mensagens entre os vários processos. O início de uma seta indica o envio de uma mensagem. O fim de uma seta representa a recepção de uma mensagem num processo. Suponha que o processo 3 falha antes de receber a mensagem m1 e que a mensagem m5 não é entregue nos processos 1 e 2. Suponha ainda que os processos comunicam usando um sistema de *middleware* capaz de atrasar a entrega das mensagens recebidas para garantir a sua entrega por uma dada ordem.



a. Com base na informação anterior, indique se cada uma das seguintes afirmações é **[V]erdadeira** ou **[F]alsa**. **Nota: as respostas erradas descontam.**

i. Supondo que no processo 2, m1 é entregue antes da emissão de m4 e m5, a seguinte ordem de entrega no processo 1 respeita a ordem FIFO: m2, m4, m1, m3.

ii. Supondo que no processo 2, m1 é entregue antes da emissão de m4 e m5, e no processo 3 m5 é entregue antes da emissão de m3, a seguinte ordem de entrega no processo 1 respeita a ordem causal: m1, m4, m2, m3.

iii. Se se pretende respeitar a ordem causal não é possível entregar a mensagem m5 no processo 3.

iv. É possível respeitar a ordem causal entregando m4 antes de m3 no processo 1.

v. O sistema de comunicação em grupo pode implementar multicast fiável se entregar m1 apenas nos processos 1 e 2.

vi. O sistema de comunicação em grupo pode implementar multicast fiável se entregar m6 apenas no processo 3.

b. Assumindo que cada processo mantém um relógio lógico iniciado a 0 e um relógio vectorial iniciado a [0 0 0], os quais são actualizados no momento da emissão e recepção de cada mensagem, indique se cada uma das seguintes afirmações é **[V]erdadeira** ou **[F]alsa**. **Nota: as respostas erradas descontam**.

i. O evento da emissão de m1 pode ser identificado com o valor 2.

ii. O evento da emissão de m4 pode ser identificado com o valor 3.

iii. O evento da recepção de m1 no processo 1 pode ser identificado com o valor 6.

iv. A mensagem m3 pode ser estampilhada com o relógio vectorial [1 3 1].

9. Suponha que tem disponível um algoritmo de consenso, acessível por uma operação: res = propose(v), em que v é o valor proposto pelo processo e res o resultado do consenso. Com base neste algoritmo, seria possível implementar um algoritmo de eleição? Explique detalhadamente como ou porque não.

10. Considere um sistema Coda, que inclui três servidores e um número elevado de clientes. Como sabe, o mecanismo de replicação do sistema Coda mantém, em cada réplica, para cada ficheiro um vector versão (semelhante a um relógio vectorial) que permite identificar a versão do ficheiro. Para um dado ficheiro, f, o valor dos vectores versão nos três servidores é o seguinte: v1[f]=[3,2,3], v2[f]=[2,3,4], v3=[2,3,4].
    a. Apresente um cenário que permita justificar a divergência dos vários vectores-versão.

    b. É possível que o conteúdo das réplicas seja resultado da mesma operação de escrita? Justifique a resposta.

    c. O sistema de caching do sistema CODA é baseado na técnica conhecida como *callback promise*. Para acesso a ficheiros que não são normalmente modificados, indique duas vantagens desta aproximação face à aproximação do sistema NFS. Justifique a resposta.