TÉCNICO LISBOA

# Sistemas Distribuídos 2015-2016

Licenciatura em Engenharia Telecomunicações e Informática

# Relatório

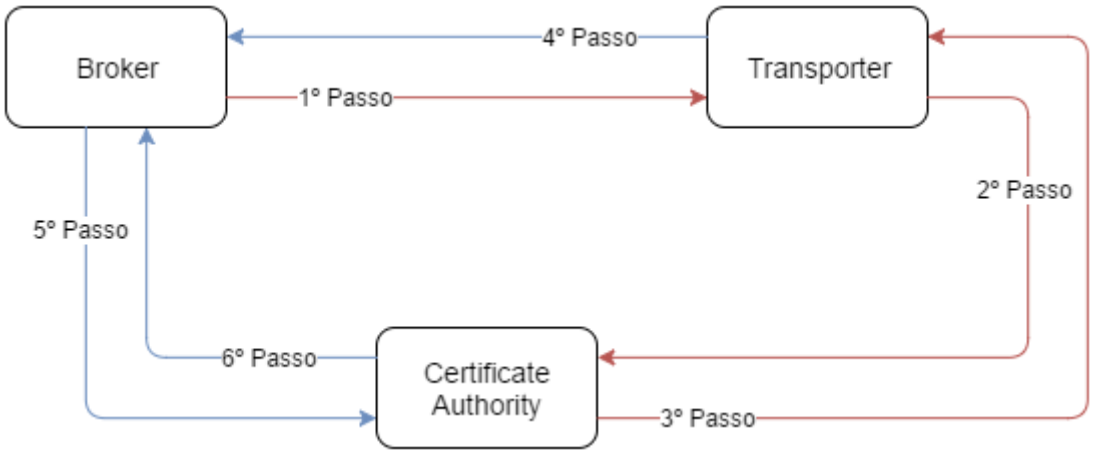


sexta-feira, 13 de maio de 2016


Grupo 46
SDis151113L08

João Freitas - 81950
Hugo Gaspar - 81977
João Carlos Costa - 82528

# Segurança




**1º Passo** – Broker envia uma mensagem para a Transporter. Nessa mensagem, vai na header uma assinatura do body, o ID da mensagem, a assinatura do ID da mensagem e o nome do Broker.

**2º Passo** – Transporter pede à Certificate Authority a chave pública do Broker.

**3º Passo** – Certificate Authority devolve à Transporter a chave pública do Broker.

**4º Passo** – Transporter decifra as assinaturas digitais e caso esteja tudo em ordem, envia uma resposta ao Broker. Na header desta nova mensagem vai uma assinatura do body, o ID da nova mensagem, a assinatura dessa nova mensagem e o nome da Transporter.

**5º Passo** – Broker pede à Certificate Authority a chave pública da Transporter.

**6º Passo** – Certificate Authority envia ao Broker a chave pública da Transporter, a qual servirá para decifrar as assinaturas digitais.

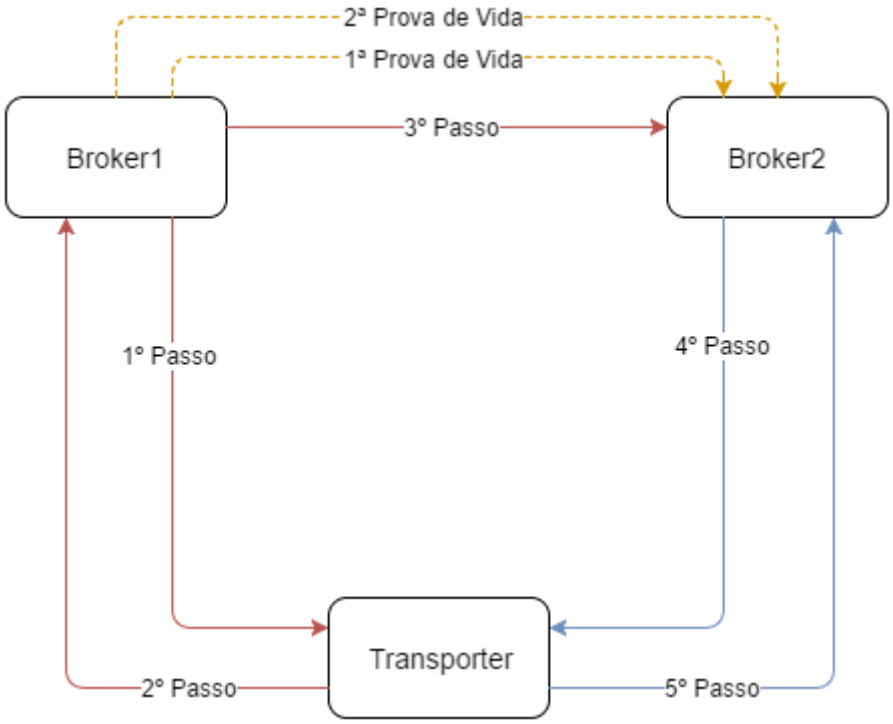
Em termos de segurança, foi-nos pedido para garantirmos os garantirmos os princípios de autenticidade, integridade, não-repúdio e frescura.

Para garantir estes princípios criamos um novo handler (SignatureHandler). Os três primeiros princípios (autenticidade, integridade e não-repúdio) são garantidos ao adicionar-se a assinatura digital do body da OutboundMessage . Do lado da InboundMessage obtém-se a assinatura digital e decifra-se a mesma com a chave pública do emissor. Caso este não a tenha, comunica-se com a CA para a obter.

Se o body e o texto decifrado forem iguais é garantida a integridade. A autenticidade é garantida ao adicionar-se à header da mensagem o nome da entidade emissora. Se este nome for alterado, no lado da InboundMessage, a decifra da assinatura digital torna-se impossível uma vez que se está a usar um certificado diferente. O Não-Repúdio é garantido através da assinatura digital do body, pois a entidade faz a assinatura com a sua chave privada, não podendo mais tarde afirmar que não foi ela que enviou essa mensagem.

Para garantir a frescura foi gerado um ID aleatório diferente (através do UUID), que é associado a cada mensagem, e de seguida foi feita a assinatura digital do mesmo. Este ID é enviado na header da OutboundMessage, assim como a sua assinatura digital. Do lado da InboundMessage obtém-se o ID e a assinatura digital deste, e verifica-se se este já se encontra na lista de ids. Se estiver significa que a mensagem já foi enviada (deste modo deve-se ignorá-la – garante-se que a mensagem já foi enviada uma vez). Por fim, é feita a verificação da assinatura digital do id para ver se alguém alterou o id na mensagem.

# Replicação




**1ª Prova de Vida** – Broker1 vai informando a Broker2 que está viva.
**1º Passo** – Broker1 envia uma mensagem para a Transporter.
**2º Passo** – Transporter responde à mensagem do Broker1.
**3º Passo** – Broker1 envia informação nova para a Broker2.
**2ª Prova de Vida** – Broker1 é fechada e não chega qualquer prova de vida à Broker2. Broker2 toma o lugar de Broker1, ou seja, passa a ser o servidor principal.
**4º Passo** – Broker2 envia uma mensagem para a Transporter.
**5º Passo** – Transporter responde à mensagem do Broker2.

Em relação à replicação, era pedido que fossem lançados dois servidores Broker, um primário (Broker1) que faz o procedimento normal, e um secundário (Broker2) num porto diferente, que se mantém em espera (sendo sempre actualizado). A criação do Broker2 é semelhante à criação das transportadoras.

O Broker1 envia, de 5 em 5 segundos, uma prova de vida ao Broker2. Se este deixar de receber tal prova de vida, assume o seu lugar, passando a ser o Broker2 o servidor primário. Uma vez que se quer ver o servidor secundário sempre atualizado para continuar a função do primário, sempre que há troca de informação vital entre o Broker1 e as Transporters, o Broker1 envia a mesma informação para o Broker2.