# Relatório da 3ª entrega do Projeto de Sistemas Distribuídos

18 de Maio de 2018


80832
**Margarida Ferreira**

81805
**Duarte David**

83557
**Ricardo Brancas**


Repositório com o projeto:
https://github.com/tecnico-distsys/A60-SD18Proj.git

O protocolo **Kerberos** consiste numa forma de autenticação que utiliza *tickets* e que permite que diferentes nós comuniquem sobre uma rede insegura, provando a sua entidade aos outros interlocutores.

Na versão simplificada utilizada na implementação do projeto, descarta-se a utilização de *Ticket Granting Servers*, sendo apenas necessário um pedido-resposta para obter um *ticket*.

Para implementar esta camada de segurança utilizámos quatro *SOAP handlers* (Figura 1): o `KerberosClientHandler`, o `KerberosServerHandler`, o `AuthorizationHandler` e o `MACHandler`; cada um responsável por uma fase do processo de autenticação.

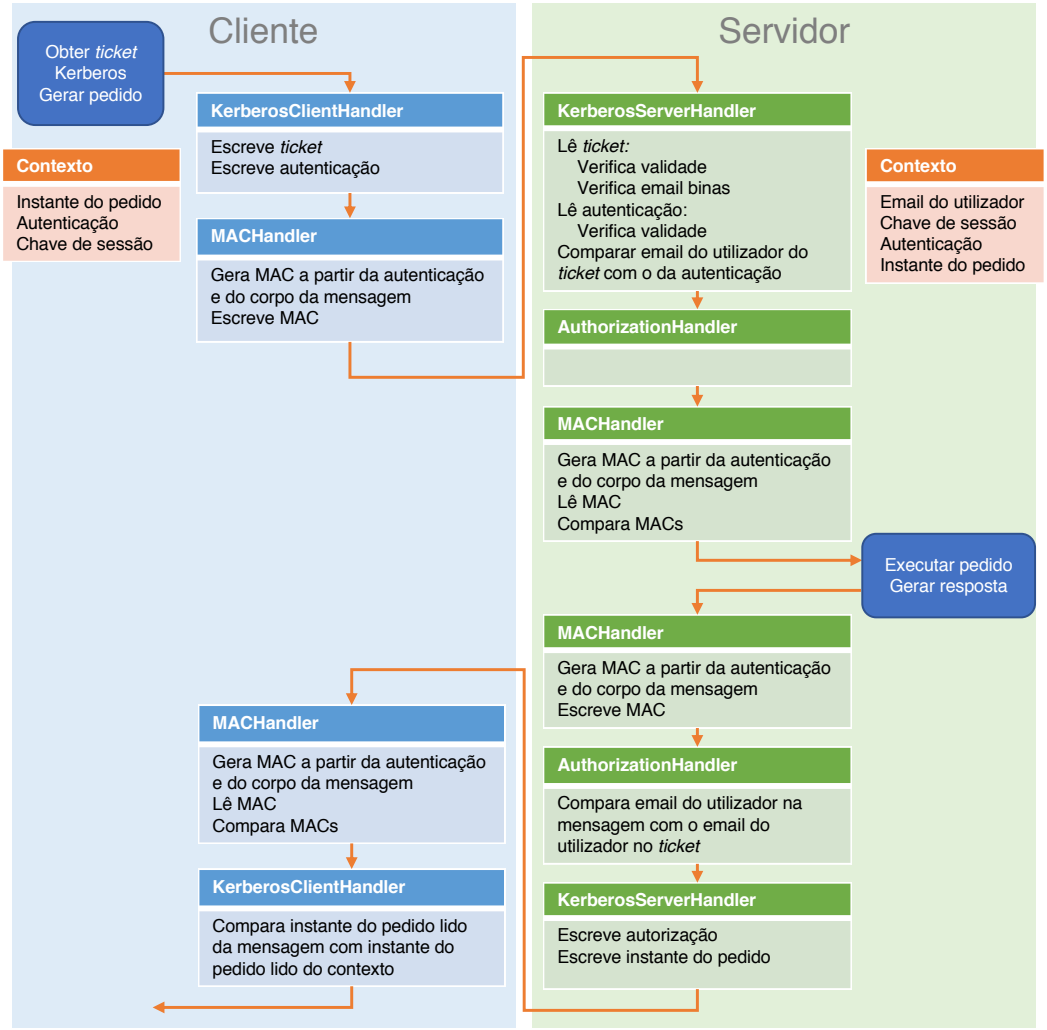


Figura 1

Os *handlers* `KerberosClientHandler` e `KerberosServerHandler` implementam o protocolo propriamente dito, adicionando os elementos necessários aos *headers* das mensagens (*ticket*, *auth*, *request time*) e validando os elementos recebidos. No `AuthorizationHandler`, o servidor Binas valida se utilizador tem permissão para efetuar operação pedida, validando se o email do pedido é igual ao email contido no *ticket* da sessão Por fim no `MACHandler` é verificado se a mensagem não foi adulterada, i. e. o MAC presente na mensagem corresponde aos outros elementos.

Como particularidades da nossa implementação apontamos:

1. Para calcular o MAC utilizamos não só o corpo da mensagem mas também a autenticação, para garantir que não é possível pegar numa mensagem que esteja a passar na rede e trocar-lhe o corpo (juntamente como o MAC) por um capturado antigamente (efetuando assim uma espécie de *replay attack*).

2. Quando o Binas deteta que um *ticket* está expirado, manda uma `Fault` ao cliente, altura em que este tenta obter um novo *ticket* e depois re-enviar o pedido anterior.