Sistemas Distribuidos, 2014-2015

Grupo 39, LEIC

Ricardo Abreu
76370

Pedro Oliveira
64057

Alexandre Ferreira
76465

SD-ID A:

Para o primeiro round-trip do protocolo Kerberos decidimos que o Cliente iria enviar o nome do Serviço e o N (identificador do pedido) num vector de bytes em que o primeiro e segundo elementos estão separados por um “\n”. Definimos que o N seria a data e a hora nesse dado momento (dd-MM-yyyy HH:mm:ss);
No SD-ID a resposta segue o mesmo critério. Assim, é enviado um vector de bytes com o ticket cifrado e os dados do cliente (também cifrados), separados por um “\n”.

Para a segunda round-trip foi necessária a implementação de Handlers de modo a enviar os elementos do protocolo Kerberos nos cabeçalhos de mensagens SOAP.
Decidimos implementar uma classe BDclient que usa tanto o cliente do Store como o do ID de forma a partilhar informação entre estes.

Do lado do Cliente (store), o Handler intercepta a mensagem de saída e coloca no header os elementos do protocolo Kerberos.;
No Handler do Servidor, a mensagem que chega é desencriptada.Também é verificado se o autenticador está consistente com o ticket (tempo e identificador) e se o MAC está válido.

O MAC é feito a partir de um hashing do corpo da mensagem enviada, cifrado com a chave de sessão. Quando o servidor recebe a mensagem decifra o MAC que está no header e faz um hashing do body da mensagem para verificar se é igual ao recebido.

Em termos de criptografia escolhemos usar o algoritmo SHA-256 para gerar a chave do Cliente a partir da sua password. Usamos o algoritmo AES para cifrar uma mensagem com base numa determinada chave.

SD-STORE B:

Visto que para este requisito só era necessária a garantia relaxada de consistência eventual, não era necessária a implementação do protocolo quorum na totalidade, mais precisamente o mecanismo que envolve a gestão das tags.

Desta forma, na operação de leitura “listDocs”, após receber Q respostas - também referidas como Acknowledges - o FrontEnd apenas as junta numa lista e elimina duplicados.

O mesmo raciocínio se aplica à operação de escrita “createDoc”. Após o envio da chamada de escrita, o FrontEnd espera por Q respostas e quando este número é atingido assume como sucesso.

Após termos uma implementação básica alterámos as nossas chamadas de escrita e leitura para serem feitas de forma assíncrona pois só assim conseguiríamos ter a espera por respostas pretendida.

Foi então necessário criar um Handler no lado do Client para contabilizar o número de respostas que chegam após um determinado pedido. Assim, o nosso Handler incrementa em 1 o numero de Acknowledges(AQ) sempre que uma repsosta assíncrona chega.