Grupo 8

TÉCNICO LISBOA

3ª Entrega: Sistema Binas

SISTEMAS DISTRIBUÍDOS

2º SEMESTRE – 2017/2018

Grupo 8

https://github.com/tecnico-distsys/T08-SD18Proj.git

André Fonseca
84698

Diogo D'Andrade
84709

Leonor Loureiro
84736

PROTOCOLO KERBEROS (VERSÃO V5) – VERSÃO SIMPLIFICADA

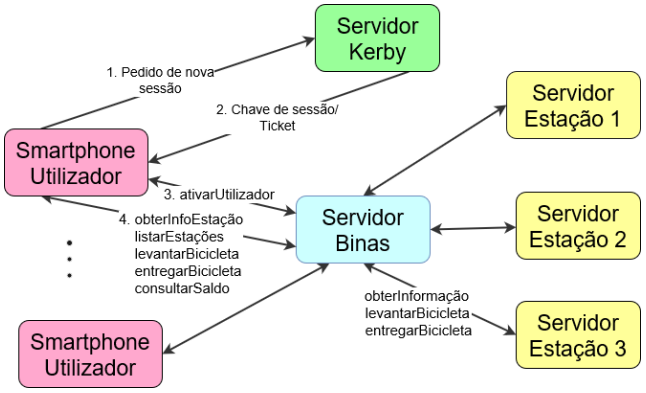

Figura 1: Diagrama da Solução

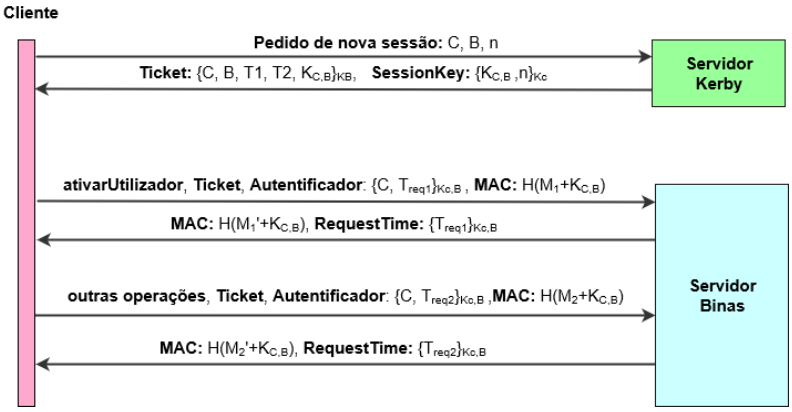

Figura 2: Trocas de Mensagens

Login

1. Cliente envia pedido de nova sessão ao servidor Kerby, incluindo no pedido um nonce, n
2. O Kerby retorna
   - a chave de sessão, $\{K_{C,B}, n\}_{Kc}$, e
   - o ticket respetivo, $T = \{C, B, T1, T2, K_{C,B}\}_{KB}$

Acesso ao Binas

3. Cliente decifra a chave de sessão, obtendo $K_{C,B}$
4. Cliente gera um novo autenticador, $A = \{T_{req}\}_{Kc,B}$
5. Cliente gera o resumo do pedido, $S_1 = H(M+K_{C,B})$
6. Cliente invoca a operação ativarUtilizador do servidor Binas, incluindo no pedido:
   - o ticket T,
   - o autenticador A, e
   - o resumo S_1
7. Binas decifra o ticket e verifica a sua frescura.
8. Se o ticket ainda estiver no período de validade,
   i. computa o resumo S_1', e
   ii. verifica se $S_1=S_1'$
9. Binas decifra o autenticador A, obtendo o request time, T_{req}
10. Verifica se o T_{req} coincide com o *timestamp* atual, com uma margem mínima de erro, e não coincide com o *request time* de nenhum pedido anterior.
11. Binas executa o pedido

Autenticação do Binas

12. Binas gera o resumo da mensagem de resposta, $S_2 = H(M'+K_{C,B})$
13. Binas retorna, incluindo na resposta:
    - o request time, T_{req}, encriptado com a chave de sessão $K_{C,B}$, e
    - o resumo S_2
14. Cliente computa o resumo $S_2'=H(M'+K_{C,B})$
15. Verifica se $S_2=S_2'$
16. Decifra T_{req} e verifica se coincide com o *timestamp* atual, com uma margem mínima de erro

SOAP HANDLERS

Cliente:

- Mensagens *OUTBOUND*:
  - **Autenticação**
    1. Verificar se tem um ticket válido
    2. Caso não tenha, efetuar pedido de nova sessão ao Kerby
    3. Adicionar à mensagem um *header* com o ticket
    4. Decifrar a chave de sessão, usando a chave do cliente
    5. Gerar um novo autenticador com *timestamp* atual
    6. Encriptar o autenticador, usando a chave de sessão
    7. Adicionar à mensagem um *header* com o autenticador
    8. Guardar a chave de sessão e o timestamp usada no autenticador, para posteriormente tratar a mensagem de resposta
  - **Assinatura Digital (MAC)**
    9. Computar o *digest* (MAC) da chave de sessão + conteúdo da mensagem
    10. Adicionar à mensagem um *header* com o *digest* (MAC)

- Mensagens *INBOUND*:
  - **Autenticação**
    11. Extrair o *request time* do *header*
    12. Decifrar o *request time*
    13. Verificar que o *request time* decifrado coincide com o *request time* do pedido
  - **Assinatura Digital (MAC)**
    14. Extrair o *digest* (MAC) do *header*
    15. Computar o *digest* da chave de sessão + conteúdo da mensagem
    16. Verificar que coincide o *digest* extraído coincide com o computado

Binas:

- Mensagens *INBOUND*:
  - **Autenticação**
    1. Extrair o ticket e decifra-lo, obtendo a chave de sessão
    2. Verificar que o nome do servidor no ticket está correto
    3. Verificar que o ticket ainda não expirou
    4. Extrair o autenticador e decifra-lo
    5. Verificar que o *username* no ticket coincide com o *username* no autenticador
    6. Guardar o *request time* e o username no contexto da mensagem
  - **Autorização**
    7. Verificar que o *request time* decifrado coincide com o *timestamp* atual, com uma margem mínima de erro e que não corresponde ao *request time* de nenhum pedido anterior.
    8. Verificar que o cliente com o *username* dado tem permissões para executar a operação pedida
  - **Assinatura Digital (MAC)**

- Mensagens *OUTBOUND*:
  - **Autenticação**
    9. Encriptar o *request time* do pedido, usando a chave de sessão
    10. Adicionar um *header* à mensagem com o *request time* do pedido
  - **Assinatura Digital (MAC)**