


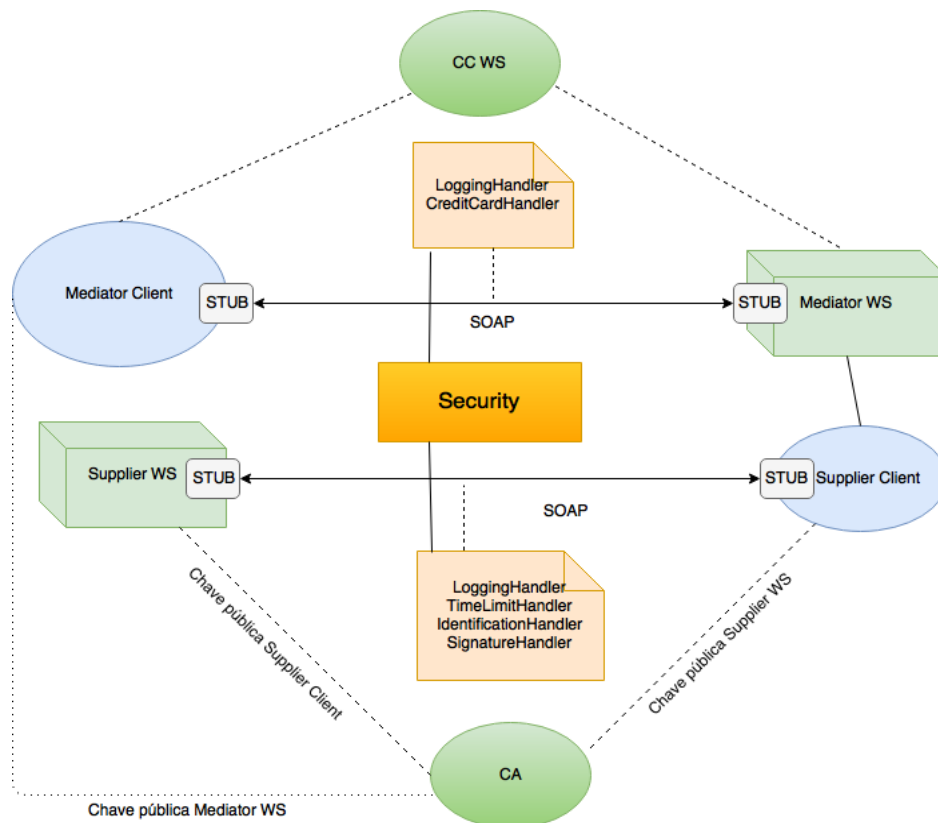


Instituto Superior Técnico
2016/2017
Sistemas Distribuídos
3ªParte

| | | |
|---|---|---|
|  |  |  |
| David Calhas | Francisco Cristóvão | José Mota |
| 80980 | 81505 | 81726 |

A06

<https://github.com/tecnico-distsys/A06-Komparator>



Autenticidade, Integridade e Não-repúdio

Para garantir a autenticidade, integridade e não repúdio (por parte do emissor) da comunicação no canal mediator - supplier recorreu-se à assinatura digital de todas as mensagens que circulam no canal. Esta assinatura permite não só identificar inequivocamente o autor de um documento (autenticidade) bem como impedir alterações do documento (integridade) e impedir que o autor repudie o conteúdo *a posteriori*. O modelo implementado neste canal é da responsabilidade do SignatureHandler e funciona da seguinte maneira (exemplo de comunicação supplier - > supplierClient, ver Imagem 1 do Anexo):

1. Supplier envia SOAPMessage com $T, \{D(T)\}_{K_{privada\ Supplier}}$, onde D é uma função de digest=SHA256withRSA conhecida por todos os intervenientes e T é a concatenação do conteúdo do SOAPBody com o conteúdo da tag Timestamp do SOAPHeader (data em que a mensagem foi enviada)
2. supplierClient recebe a SOAPMessage enviada pelo Supplier e vai:
 - a. Pedir à CA o certificado do remetente para decifrar a mensagem recebida (chave pública do remetente obtida do certificado)
 - b. Garantir a autenticidade do certificado obtido (obter a chave pública da CA e verificar se o certificado foi assinado por aquela entidade)
 - c. Decifra a $\{Assinatura\ de\ T\}_{K_{publica\ Supplier}}$, obtendo D(T)
 - d. Calcular D(T) sobre T recebido na SOAPMessage e verificar se o valor resultante é igual a D(T)

Nota: pode ser o supplierClient a enviar e o Supplier a receber, o processo é o mesmo (é isso que está ilustrado na Imagem 1).

Confidencialidade

Para garantir a confidencialidade do número do cartão de crédito nas comunicações no canal mediatorClient - mediator recorreu-se à cifra(RSA/ECB/PKCS1Padding). O modelo de cifra implementado neste canal é da responsabilidade do CreditCardHandler e funciona da seguinte maneira:

1. mediatorClient (message outbound)
 - a. Pedir à CA o certificado do destinatário para cifrar a mensagem recebida (chave pública do destinatário obtida do certificado)
 - b. Garantir a autenticidade do certificado obtido (obter a chave pública da CA e verificar se o certificado foi assinado por aquela entidade)
 - c. Cifrar {creditCardNr}_{K pública Mediator}, adicionar à tag creditCardNr o resultado depois de cifrar e enviar a SOAPMessage
2. Mediator (message inbound)
 - a. Decifrar {creditCardNr}_{K privada Mediator}

Frescura

Para garantir a frescura das SOAPMessages nas comunicações no canal mediatorClient - mediator foi implementado um handler (TimeLimitHandler) que:

1. Adiciona ao SOAPHeader das outbound messages a tag timestamp com o valor da data em que a SOAPMessage foi enviada
2. Verifica, nas inbound messages, se a mensagem recebida foi enviada há mais de 3 segundos (comparando a data que se encontra no SOAPHeader com a data atual)

Esta implementação é suscetível de ser alvo de ataques do tipo *Replay Attack*, que consistem na introdução de repetição ou atraso na transmissão de dados (no envio de SOAPMessages). Se esta repetição/atraso se encontrar dentro do intervalo definido (3 segundos), o recetor não consegue detetar o ataque e considera os dados recebidos válidos. Facilmente se verifica um exemplo concreto deste ataque no nosso projeto: se um atacante interceptar uma SOAPMessage que contenha uma operação do tipo buyCart, poderá facilmente repeti-la n vezes (dentro do intervalo de 3 segundos), comprando assim n carrinhos, quando o cliente apenas queria comprar um.

No entanto a alternativa seria gerar um nonce (número arbitrário que só pode ser utilizado uma vez) para servir como identificador único de cada SOAPMessage, e apenas a aceitar se aquele identificador não tivesse sido utilizado previamente. Porém esta alternativa é bastante cara e complexa pois é necessário validar esse identificador cada vez que uma SOAPMessage é recebida, o que implica guardar todos os identificadores recebidos numa estrutura de dados apropriada para o efeito.

Optámos ainda por implementar o IdentificationHandler, que adiciona ao Header do SOAPEnvelope o wsName do Web Service remetente da mensagem. Este é particularmente útil para ser possível saber qual a chave pública que é necessário ir buscar para decifrar as mensagens recebidas pelo destinatário, nas comunicações no canal mediator – supplier.

Nota: As chaves e certificados encontram-se todas na pasta /resources do módulo security.

Anexo

```

[2017-05-05T02:59:58.364] intercepted Inbound SOAP message:
<?Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <?Timestamp xmlns="http://ws.supplier.komparator.org/">2017-05-05T02:59:58.046</?Timestamp>
    <?SenderName xmlns="http://ws.security.komparator.org/">A06_Mediator</?SenderName>
    <?signature xmlns="http://ws.supplier.komparator.org/">?Jeg5jWmz3Wk1IrnhoE=ObKTwKsdSF020Z8JNTEC1uCY6wrg1h9HgLS80
WjMF14uZ2MRV6kLbVHEhyA7wZN34vF8RcMQ374/4m3VpA5gfkIXsAp3wdkucJ9k+1lVri/2UzdVxzvgcPMW0FT4uTBqLY2GL68E11DTJ/0V9YeZe
W1d1xgQ4u19L42nq5wFZAH0S05FKKcn7bH6tB8fgbKALHRS1In5SpKATYc0uOIBcdE0Y30Yvgc+piAUNHJ3h88z52gBa7GNx3MduYfUR2d
H3074wRg2qCnULFa1tcc6/BgQ0uTmny04w=</?signature>
  <SOAP-ENV:Body>
    <ns2:ping xmlns:ns2="http://ws.supplier.komparator.org/">
      <arg>client</arg>
    </ns2:ping>
  </SOAP-ENV:Body>
</?Envelope>
[2017-05-05T02:59:58.377] intercepted Outbound SOAP message:
<?Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <?Timestamp xmlns="http://ws.supplier.komparator.org/">2017-05-05T02:59:58.046</?Timestamp>
    <?SenderName xmlns="http://ws.security.komparator.org/">A06_Mediator</?SenderName>
    <?signature xmlns="http://ws.supplier.komparator.org/">?Jeg5jWmz3Wk1IrnhoE=ObKTwKsdSF020Z8JNTEC1uCY6wrg1h9HgLS80
WjMF14uZ2MRV6kLbVHEhyA7wZN34vF8RcMQ374/4m3VpA5gfkIXsAp3wdkucJ9k+1lVri/2UzdVxzvgcPMW0FT4uTBqLY2GL68E11DTJ/0V9YeZe
W1d1xgQ4u19L42nq5wFZAH0S05FKKcn7bH6tB8fgbKALHRS1In5SpKATYc0uOIBcdE0Y30Yvgc+piAUNHJ3h88z52gBa7GNx3MduYfUR2d
H3074wRg2qCnULFa1tcc6/BgQ0uTmny04w=</?signature>
  <SOAP-ENV:Body>
    <ns2:pingResponse xmlns:ns2="http://ws.supplier.komparator.org/">
      <return>Hello client from Supplier</return>
    </ns2:pingResponse>
  </SOAP-ENV:Body>
</?Envelope>

```

Imagem 1 – Pedido da operação ping recebido pelo Supplier (enviado pelo supplierClient) e respectiva resposta do Supplier.

Nesta mensagem podemos verificar a tag <w:signature> no SOAPHeader, que contém cifrada a concatenação do conteúdo do SOAPBody com o conteúdo da tag <w:timestamp>. Ao receber esta mensagem o Supplier vai executar os passos descritos acima e devolver uma resposta à operação executada.

A tag <w:timestamp> é adicionada e verificada pelo TimeLimitHandler. Sendo que a operação teve sucesso, a frescura da mensagem foi garantida.

```

[2017-05-05T14:24:20.831] intercepted Outbound SOAP message:
<?Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-Header>
    <S:Body>
      <ns2:buyCart xmlns:ns2="http://ws.mediator.komparator.org/">
        <id>3041</id>
        <redirection>cdCpdyqYvaeJDSuHwB6261X3C/wfAr9Gsf0r/91mKYLq16hpceelLh5y835081C/K4WvWMrAPcyB58/
          EN2A/348Pvmbnb7880Gb1b9TxsFbwGdA1fUK1YONzhcMczT2pYkSdaSx6fr3CV3EY67F7k3Z1905+y42AnGjB96AdP82QhJqUgSXWBA6c4H2k61bdNvMoydVotE1ZtE/
          07X4d2SCQvz1J5fjPyY01vmuqeJaeJhfM61HbQacCP67KkrYba1/Gx0ffr5W71L/Y3mAtZcyfybJLJ1CtCh8Nf1R1bC391RSU/6vpt35cSBV1Z25u10n7bW==
        </redirection>
      </ns2:buyCart>
    </S:Body>
  </SOAP-Header>
</?Envelope>

[2017-05-05T14:24:21.497] intercepted Inbound SOAP message:
<?Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-Header>
    <S:Body>
      <ns2:buyCartResponse xmlns:ns2="http://ws.mediator.komparator.org/">
        <shopResult>
          <id>A861</id>
          <result>COMPLETE</result>
          <purchasedItem>
            <item>
              <product>X1</product>
              <supplier>A86_Supplier</supplier>
            </item>
            <item>Basketball</des>
            <price>18</price>
          </item>
          <quantity>10</quantity>
          <purchase>
            <totalPrice>180</totalPrice>
          </shopResult>
        </ns2:buyCartResponse>
      </S:Body>
    </SOAP-Header>
  </?Envelope>

```

Imagem 2 – Pedido da operação buyCart enviado pelo mediatorClient e respectiva resposta do Mediator.

Nesta mensagem podemos verificar que a tag <creditCardNr> no SOAPBody contém cifrado o número do cartão de crédito. Ao receber esta mensagem o Mediator vai executar os passos descritos acima e devolver uma resposta à operação executada (sendo que a resposta foi a esperada podemos afirmar que o creditCardNr foi bem

```

[2017-05-05T15:03:39.868] Intercepted Inbound SOAP message:
<?Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-Header>
    <timestamp xmlns="http://ws.supplier.komparator.org/">2017-05-05T15:03:39.801</timestamp>
    <ws:SenderName xmlns="http://ws.security.komparator.org/">A06_SenderName</ws:SenderName>
    <timestamp xmlns="http://ws.supplier.komparator.org/">3188603d0a39f983b5d5C0P12AaK1F+4AHU0j0n[Gm3c7/T140+AfzmedeKPS5eagJw8x0xLS2bame+rs59P
    09580T0dc3kvy1c80+0s55+6A/DQm2Zn93s1b+7qAvQv5Yp73J1w8yJhKs62g561A1XG6Jm8Wbq6e+1E2gHf/XeXwQzG07Yh4w6eVed8Wm0x5H4D16Xl8B/
    V9F33Xy1k6wGvH55VtL/X85XjW0J335WQLE18vntFX+ORe4v1P2zx121718BTd0JmsyT68W/zjNtJelDjKVDWZ2513K36dA==
    </ws:SenderName>
  </SOAP-Header>
  <Body>
    <ns2:getProduct xmlns="http://ws.supplier.komparator.org/">
      <productID>ATTACK</productID>
    </ns2:getProduct>
  </Body>
</?Envelope>
[2017-05-05T15:03:39.869] Intercepted Outbound SOAP message:
<?Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-Header>
    <ns2:getProductResponse xmlns="http://ws.supplier.komparator.org/">
      <product>
        <!--ATTACK-->
        <desc>Padel ball</desc>
        <quantity>30</quantity>
        <price>30</price>
      </product>
    </ns2:getProductResponse>
  </SOAP-Header>
  <Body>
    <ns2:getProductResponse xmlns="http://ws.supplier.komparator.org/">2017-05-05T15:03:39.873</timestamp>
    <ws:SenderName xmlns="http://ws.security.komparator.org/">A06_Supplier1</ws:SenderName>
    <timestamp xmlns="http://ws.supplier.komparator.org/">3188603d0a39f983b5d5C0P12AaK1F+4AHU0j0n[Gm3c7/T140+AfzmedeKPS5eagJw8x0xLS2bame+rs59P
    09580T0dc3kvy1c80+0s55+6A/DQm2Zn93s1b+7qAvQv5Yp73J1w8yJhKs62g561A1XG6Jm8Wbq6e+1E2gHf/XeXwQzG07Yh4w6eVed8Wm0x5H4D16Xl8B/
    V9F33Xy1k6wGvH55VtL/X85XjW0J335WQLE18vntFX+ORe4v1P2zx121718BTd0JmsyT68W/zjNtJelDjKVDWZ2513K36dA==
    </ws:SenderName>
  </SOAP-Header>
  <Body>
    <ns2:getProductResponse xmlns="http://ws.supplier.komparator.org/">
      <product>
        <!--ATTACK-->
        <desc>Padel ball</desc>
        <quantity>30</quantity>
        <price>30</price>
      </product>
    </ns2:getProductResponse>
  </Body>
</?Envelope>
[2017-05-05T15:03:39.883] Intercepted Outbound SOAP message:
<?Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-Header>
    <timestamp xmlns="http://ws.supplier.komparator.org/">2017-05-05T15:03:39.873</timestamp>
    <ws:SenderName xmlns="http://ws.security.komparator.org/">A06_Supplier1</ws:SenderName>
    <timestamp xmlns="http://ws.supplier.komparator.org/">3188603d0a39f983b5d5C0P12AaK1F+4AHU0j0n[Gm3c7/T140+AfzmedeKPS5eagJw8x0xLS2bame+rs59P
    09580T0dc3kvy1c80+0s55+6A/DQm2Zn93s1b+7qAvQv5Yp73J1w8yJhKs62g561A1XG6Jm8Wbq6e+1E2gHf/XeXwQzG07Yh4w6eVed8Wm0x5H4D16Xl8B/
    V9F33Xy1k6wGvH55VtL/X85XjW0J335WQLE18vntFX+ORe4v1P2zx121718BTd0JmsyT68W/zjNtJelDjKVDWZ2513K36dA==
    </ws:SenderName>
  </SOAP-Header>
  <Body>
    <ns2:getProductResponse xmlns="http://ws.supplier.komparator.org/">
      <product>
        <!--ATTACK-->
        <desc>Padel ball</desc>
        <quantity>30</quantity>
        <price>30</price>
      </product>
    </ns2:getProductResponse>
  </Body>
</?Envelope>

```

Imagem 3 – Pedido e resposta da operação getProduct (supplierClient -> Supplier). Quando productId="ATTACK": a resposta do Supplier à operação getProduct é alterada pelo SimulationAttackHandler, que muda o valor da tag price (antes de executar o handler, price=30, depois, price=0).