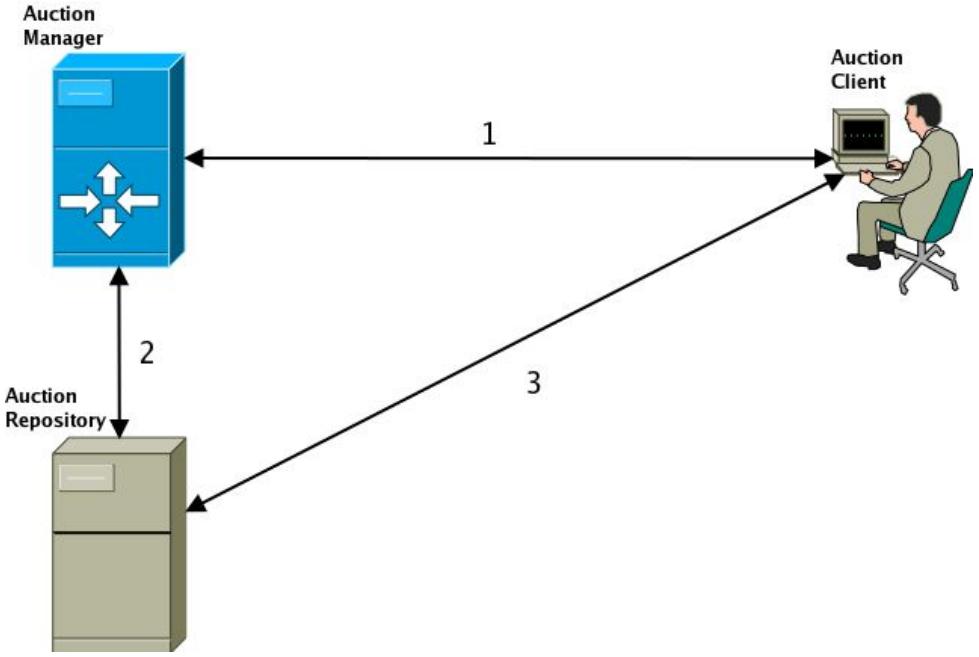
Milestone Segurança

Canais de Comunicação

Serão necessários três canais de comunicação:



1. **Canal Auction Client <-> Auction Manager**
    a. Criação e remoção de leilões ativos.

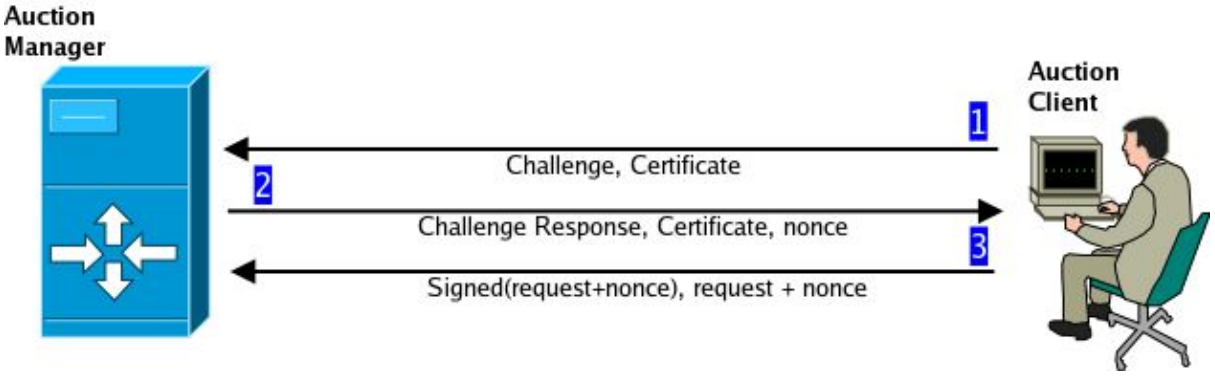
2. **Canal Auction Manager <-> Auction Repository**
    a. Criação e remoção de instâncias de leilões após pedido de utilizador.
    b. Validação dinâmica de ofertas.

3. **Canal Auction Repository <-> Auction Client**
    a. Leitura de leilões ativos.
    b. Criação de ofertas.

Autenticação

Canal Auction Client <-> Auction Manager

Este canal é apenas usado para criação e remoção de leilões. Isto significa que não existe a necessidade de uma sessão entre os dois. Contudo, é necessário certificar e identificar a origem das mensagens. Para existir comunicação neste canal é necessário que o Auction Manager e o Auction Client possuam pares de chaves em que a chave pública é certificada. No caso do Auction Client usamos, obviamente, o cartão de cidadão.

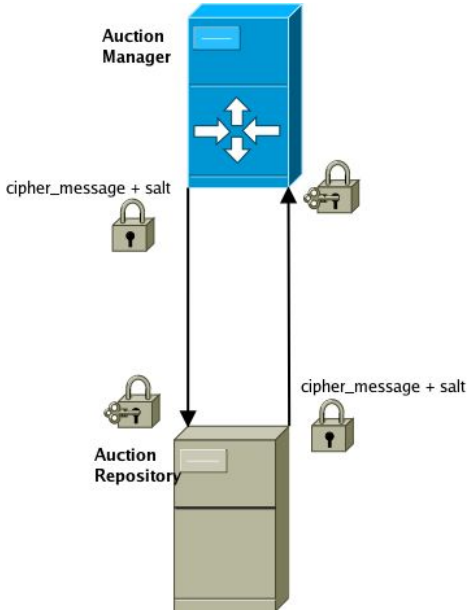


1. A comunicação é iniciada com o envio de um challenge e o certificado do utilizador para o servidor. O challenge trata-se de 16 bits random e o certificado corresponde ao do cartão de cidadão do utilizador.
2. O servidor responde ao challenge usando a sua chave pública e enviando o seu certificado para que o utilizador o possa validar. Na mesma mensagem o servidor envia também um nonce (random de 16 bits) ao utilizador de maneira a tornar o request único.
3. O utilizador termina por enviar o pedido a realizar (criação / remoção de leilão) assinado junto com o nonce.

Dúvidas ainda existentes:

- A criação do leilão contém informação sensível? Por exemplo, o código dinâmico? Se o código dinâmico puder efectivamente cifrar dados significa que existirá uma KEY que deverá ser privada entre o utilizador e o servidor…?

Canal Auction Manager <-> Auction Repository



De maneira a garantir a confidencialidade na comunicação entre os servidores, as mensagens serão cifradas com uma chave simétrica previamente instaladas fisicamente nos servidores. É adicionado salt na cifragem de maneira a prevenir a duplicação de textos cifrados e dificultar ataques de dicionário.

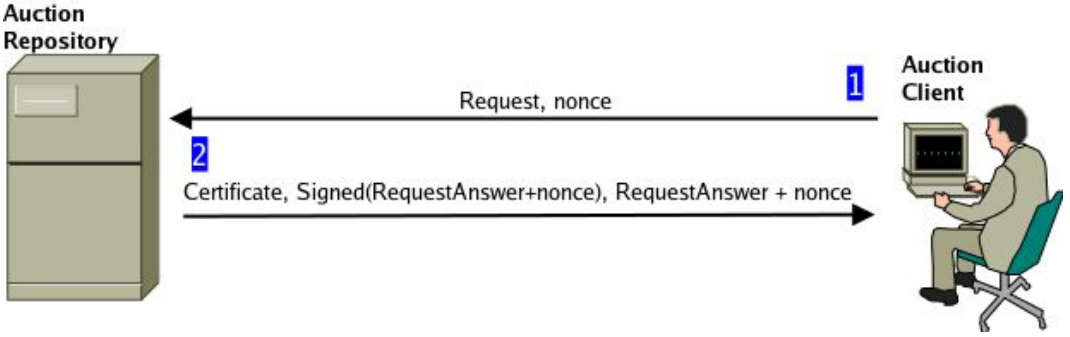
Dúvidas ainda existentes:
- Existe a necessidade de uso de MAC?
- Suficientemente resistente a ataque de dicionário?
- Talvez fosse mais prático cifrar com a chave pública de cada um e adicionar salt...

Canal Auction Repository <-> Auction Client

Este canal é usado para a criação de ofertas e leitura de informação pública aos utilizadores. As duas acções necessitam de comportamentos diferentes de autenticação.

Leitura de Informação:

Leilões ativos, informações de um leilão particular, etc. É assumido que não existe dados sensíveis na informação disponibilizada. Com isso, não existe a necessidade de autenticidade do utilizador nem cifragem de dados. O único requisito nesta comunicação passasse pela autenticidade do servidor, que é feita da seguinte forma:

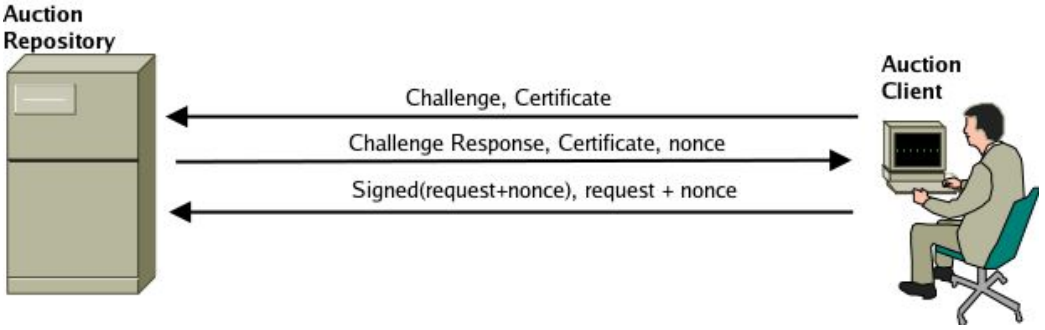


1. O Auction Client envia uma descrição do que pretende junto com um nonce. O nonce é usado para impedir que, por exemplo, alguém impeça o utilizador de participar no leilão enviando-lhe uma lista desatualizada de leilões ativos já assinada pelo servidor.
2. O servidor responde ao pedido com o seu certificado e a resposta ao pedido do utilizador assinada junto com o nonce.

Criação de uma oferta:

A criação de uma oferta é, sem dúvida, a comunicação que terá de ser aplicada a máxima segurança.

Consideramos usar o mesmo método que a de criação de leilão:



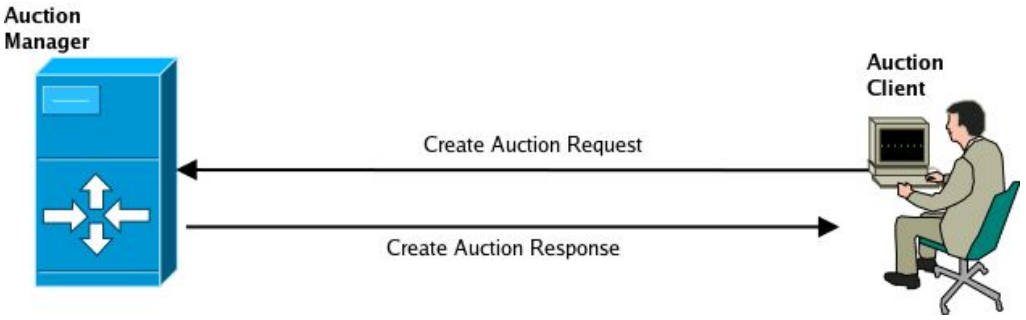
Dúvidas ainda existentes:

- **O método parece eficaz… mas haverá melhor?**
- **Existem dados sensíveis na criação de uma oferta? Talvez o valor no caso da blind auction…… Isto pode ser resolvido cifrando o request com a chave pública do servidor?**
- **Supostamente o Auction Repository nao sabe a identidade do Utilizador até ao fim do leilão… Mas o certificado do utilizador já revela a sua identidade...**

Comunicação

Canal Auction Client <-> Auction Manager

Criação e remoção de leilões ativos:

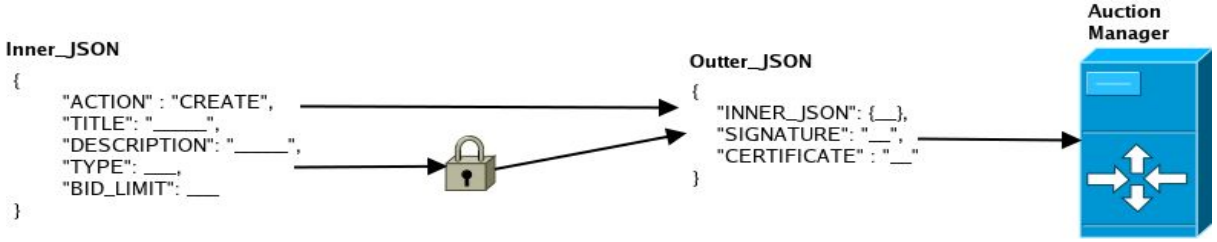


O pedido de criação de um leilão será realizado por o Auction Client após pedido dos parâmetros necessários ao utilizador.

Um leilão é definido por:

(Required) Nome
(Required) Descrição
(Required) Tipo de Leilão
(Required) Tempo limite para novas ofertas
(Optional) Código para validação de ofertas.

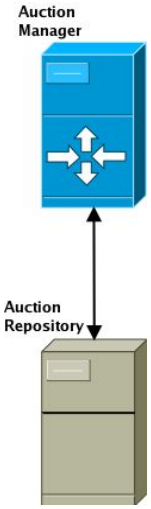
Aspectos de segurança: Tendo em conta que a criação de um leilão não apresenta nenhum dado sensível, não existe a necessidade cifragem de dados. Contudo, é necessária a autenticação do utilizador. Isto será realizado com a assinatura digital do cartão de cidadão do utilizador.



A remoção de um leilão será feita da mesma forma mas com diferenças ao nível do conteúdo do JSON.

Canal Auction Manager <-> Auction Repository

A comunicação entre estes dois servidores irá realizar-se sobre UDP/IP requests/responses.



Criação e remoção de instâncias de leilões após pedido de utilizador:

Após o pedido de criação do leilão por parte do utilizador, o Auction Manager irá verificar se os parâmetros são válidos. E, se for o caso, requisitará ao Auction Repository a criação deste leilão e do blockchain associado.

Validação dinâmica de ofertas:

Quando o utilizador apresenta uma oferta ao Auction Repository, este verificará a sua validação a partir da função dinâmica armazenada no Auction Manager.

Canal Auction Repository <-> Auction Client

Leitura de leilões ativos e ofertas realizadas.

Chamada realizada do Auction Client para o Auction Repository. Devolve informação não sensível e totalmente pública pelo que não apresenta nenhum aspecto de segurança a discutir (para além das já discutidas acima).

Criação de ofertas:
MUITAS DÚVIDAS

Uma oferta é caracterizada por:

- ID do leilão a apresentar a oferta
- Valor da oferta

Por cada oferta adicionada ao leilão vai ser enviado um recibo ao cliente e armazenado numa memória não volátil. Este recibo será usado como prova da sua participação e será necessário para reclamar o prêmio (equivalente a uma lotaria).

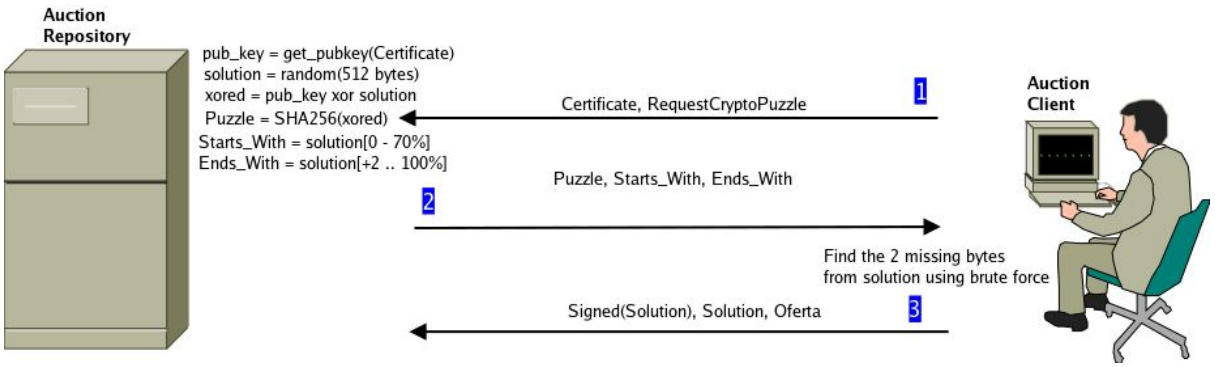
Dúvidas ainda existentes:

- Ainda tenho muitas dúvidas quanto ao que o código dinâmico pode fazer à informação da oferta… o que vai esconder? se esconder o valor da oferta é praticamente transformar numa blind auction...

CryptoPuzzle

Será implementado um mecanismo denominado CryptoPuzzle em que o envio das ofertas pode ser controlado prevenindo a ocorrência de spam, isto é, controla a taxa de transferência com que as ofertas são enviadas.

De maneira a criar uma oferta o utilizador tem primeiro pedir ao Auction Repository por um CryptoPuzzle para validar a sua oferta. A nossa solução funciona da seguinte forma:



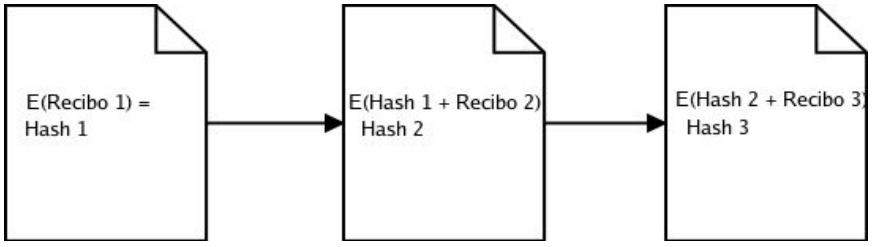
1. O Auction Client envia um pedido de um CryptoPuzzle junto com o certificado do cartão de cidadão do utilizador. A partir do Certificado o servidor obtém a chave pública. A essa chave pública é aplicada um XOR de um random de 512 bytes. O resultado é hashed com SHA256 no que resulta no puzzle. O objectivo do utilizador é obter o random de 512 bytes, como apoio, o servidor envia os primeiros 0% a 70% dos bytes e os restantes excluindo dois bytes entre ambos. Ou seja, o utilizador tem que encontrar os 2 bytes que faltam.

2. O Auction Client terá de encontrar os bytes restantes usando brute force. Testes realizados demonstram que tal é terminado em 0.5 segundos a 4 segundos.

3. O Auction Client ao encontrar a solução, enviará ao Auction Repository assinado junto com a oferta que deseja realizar.

BlockChain

WORK IN PROGRESS



Catarina Silva, 76399
António Sérgio Silva, 76678