RELATÓRIO DE ARQUITETURA
PROJETO DE SIO
2018/2019

Proteção das mensagens trocadas:

- ➔ A comunicação é feita de uma forma segura, de acordo com os seguintes passos:
    a. O Auction Manager gera um par chave pública-chave privada com o algoritmo RSA, partilhando a sua chave pública com os clientes.
    b. O Cliente gera 1 chave simétrica/privada através do algoritmo AES.
    c. O Cliente cifra, com a chave pública do Auction Manager, o conjunto da sua mensagem com a chave privada auto-gerada (desta forma, apenas o Auction Manager poderá desencriptar esta mensagem) - chamemos ao bloco cifrado "Bloco A".
    d. O "Bloco A", cifrado com a chave pública do Auction Manager, chega ao Auction Manager e é decifrado pela chave privada do Auction Manager.
    e. O Auction Manager cria um receipt e cifra-o com a chave simétrica/privada que recebeu do cliente. Desta forma, apenas o cliente poderá decifrar esta mensagem - denominamos este bloco por "Bloco B".
    f. **Opcional :** Podemos, também, cifrar o "Bloco B" com a chave privada do Auction Manager e enviar este novo bloco ("Bloco C" para o Cliente)).

- ➔ A cada nova mensagem será gerada uma nova chave assimétrica e esta será posteriormente partilhada, mais uma vez, com o Auction Manager.

- ➔ As mensagens descritas representam apenas as mensagens trocadas com o Auction Repository. Todas as mensagens que envolvam bids serão descritas mais abaixo.

Daniel Nunes nº 84793
Rafael Direito nº 84921

Proteção das bids:

- ➔ O cliente envia uma mensagem com a bid para o Auction Repository
- ➔ O Auction Repository manda a bid para o Auction Manager, que irá validar a mesma.
- ➔ Posteriormente, o Auction Manager, envia uma resposta a dizer se a bid é válida ou não, e, caso o seja, encripta alguns campos da bid com a sua chave simétrica. A Bid encriptada seguirá para o Auction Repository onde ficará guardada.

Identificação do autor da bid:

- ➔ Todas as bids são assinadas digitalmente através do uso do Cartão de Cidadão. O método utilizado é o PKCS#11.

Exposição das bids necessárias no final do leilão:

- ➔ O cliente envia uma mensagem para Auction Manager dizendo o que quer ver do leilão. (por exemplo, quer ver toda a cadeia de bids)
- ➔ O Auction Manager envia uma mensagem ao Auction Repository dizendo que bids quer ter acesso.
- ➔ O Auction Repository devolve por sua vez as bids pedidas (estando estas encriptadas)
- ➔ O Auction Manager decripta todos os campos encriptados e envia ao cliente.

Validação de bids usando código dinâmico:

O Auction Manager terá uma API de funcionamento definida por diversas funções com uma assinatura bem definida (on_message(), validate(), …). Aquando a criação do leilão, o seu criador pode dar override de algum método da API, importando o seu código.

Após o fim do leilão, todos os utilizadores poderão ver qual o código importado, garantindo a validade do leilão.

Modificação de bids validadas por código dinâmico:

- ➔ Bids (em formato XML) serão cifradas pelo Auction Manager. Este poderá encriptar alguns campos destas, de forma a que o repositório não consiga ter acesso a estas informações aquando o seu armazenamento na blockchain.

Daniel Nunes nº 84793
Rafael Direito nº 84921

- ➔ Esta encriptação poderá ser efetuada com recurso a uma chave simétrica que nunca sai do servidor do Auction Manager.
- ➔ Caso seja necessário, o Auction Manager pode carregar uma bid do Repository e desincriptá-la, com recurso à sua chave simétrica.

Construção da Blockchain do leilão:

- ➔ Cada bloco terá um header (gerado por uma hash function). Este header será gerado, tendo em conta o header do bloco anterior, sendo que a hash do header do novo bloco irá ser gerada tendo, também, em consideração a hash contida no header do bloco anterior.
- ➔ Este blocos serão guardados com recurso a uma linked list, sendo que cada linked list estará associada a um leilão.

Cryptopuzzles:

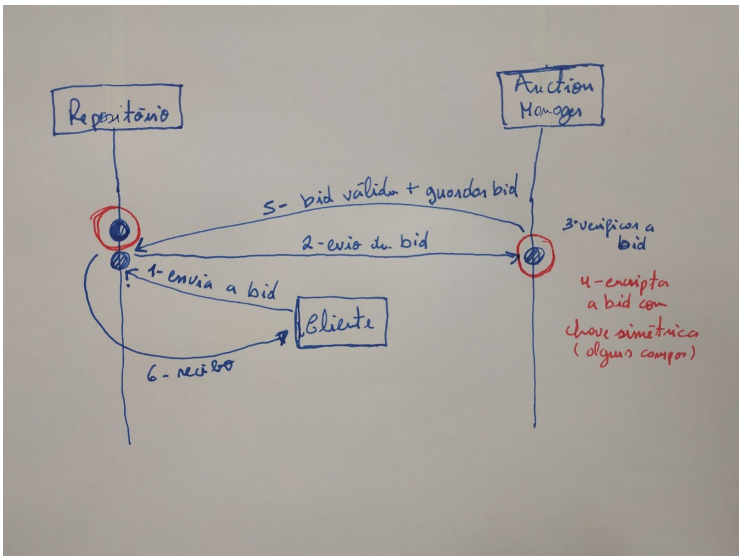
- ➔ Será definido um valor e o Cliente terá de calcular (isto é, o CPU) um determinado número. Para isto, este utilizará N hashes que lhe permita obter este valor.
- ➔ Ao obter o valor, o Cliente poderá prosseguir no request feito, demonstrando, assim, um proof of work.

Produção e validação do recibo:

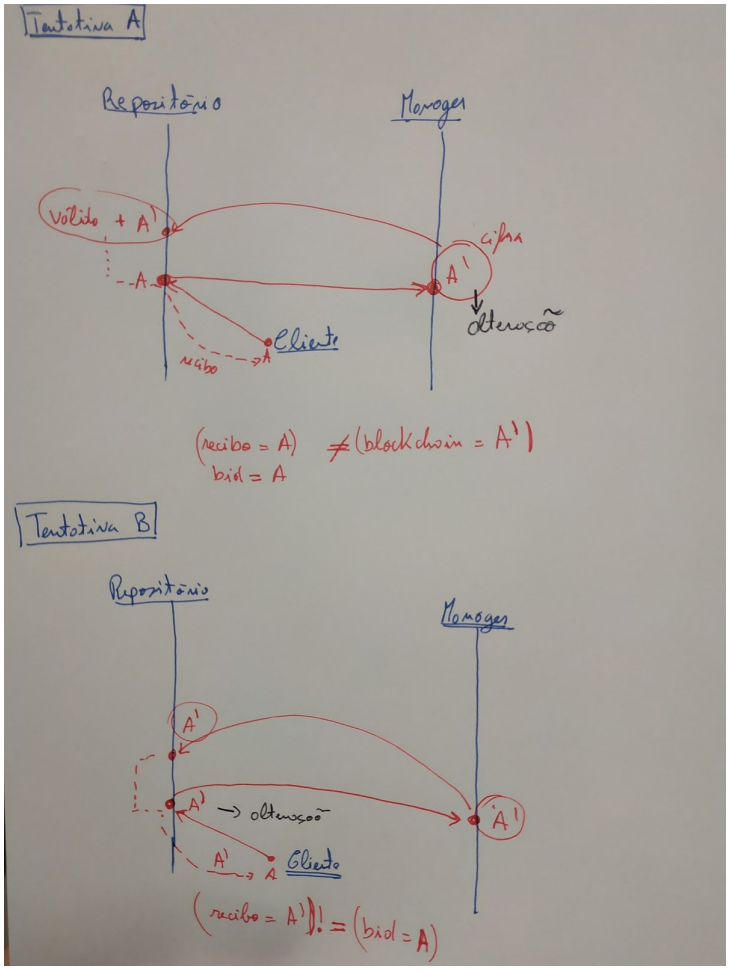
1. O Cliente envia a sua bid para o Auction Repository (cifrada com a public key deste), que cria um pré-recibo para a bid em questão;
2. Posteriormente, o Auction Repository envia a bid para ser validada pela Auction Manager (cifrada pela chave privada do repositório e posteriormente pela chave pública do manager);
3. Após o Auction Manager validar a bid em questão, este irá encriptar alguns dados da mesma, enviando-a posteriormente para o Repository, que guardará esta bid cifrada numa blockchain. O Auction repository receberá, também, a informação de que a bid é válida.
4. Após verificar a validade da bid, o repositório acrescenta o campo <valid>True</valid> ao recibo que construiu anteriormente e envia-o para o cliente.

Daniel Nunes nº 84793
Rafael Direito nº 84921

5. Por fim, o cliente, caso o recibo recebido coincida com a bid que este efetuou, guarda-o em memória não volátil.



Exemplos de tentativas de fraude:



Daniel Nunes nº 84793
Rafael Direito nº 84921

Validação de um leilão terminado:

- ➔ O Cliente pode, no final do leilão, enviar uma mensagem ao Auction Manager dizendo que quer verificar a cadeia final de bids.
- ➔ Através do recibos, o Cliente poderá validar todo o leilão feito, comparando-o com os seus recibos.

Daniel Nunes nº 84793
Rafael Direito nº 84921