



universidade
de aveiro

Blockchain-based auction management

PROJETO DE SIO

RUI COELHO - 86182

DANIEL TEIXEIRA - 84710

Índice

Introdução	2
Dependências	2
Processos	3
Mensagens	3
Protocolos de comunicação	3
Negociação de chaves	4
Fases de negociação.....	4
Fase 1.....	4
Fase 2.....	4
Fase 3.....	4
Funcionalidades implementadas	4
Encriptação das mensagens trocadas	5
English Auction.....	5
Blind Auction.....	5
Identificação do autor das licitações com o Cartão de Cidadão	5
Proteção das licitações até ao final do leilão.....	6
Expor a informação necessária sobre uma licitação no final de um leilão	6
Construção de uma blockchain por leilão	6
Verificar o resultado de um leilão em que o cliente participou	6
Validar recibos	6
Chaves	7
Servidores.....	7
Clientes.....	7
Cartão de Cidadão	7

Introdução

Neste relatório descrevemos a estrutura de transmissão de mensagens que implementamos bem como todos os métodos de segurança implementados-

O objetivo é providenciar um sistema de blockchain seguro e confiável em que todos os utilizadores conseguem participar em leilões já existentes bem como criar novos leilões e listar aspetos importantes relativos a este.

Todas as comunicações usam cifras simétricas ou assimétricas e a sua implementação é descrita posteriormente neste relatório.

Destaca-se neste projeto que todas as informações dos utilizadores são obtidas via Cartão de Cidadão.

Dependências

Para a realização deste projeto foi necessário importar algumas livrarias e, deste modo, é necessário descrever as mesmas para um melhor entendimento do projeto. Foram usadas 7 livrarias no total para facilitar a comunicação e desenvolvimento de segurança neste sistema de blockchain.

Livraria	Descrição	pip
cryptography	Interfaces para operações criptográficas	cryptography
OpenSSL	Validação de cadeia x509 e testes de revocação	pyOpenSSL
PyKCS11	Interface para operações com smartcards	python-pkcs11
pem	Separação de múltiplos PEMs num único ficheiro	Pem
sockets	Providenciar comunicação entre os servidores e cliente	sockets
diffiehellman	Troca de chaves entre servidores e cliente	diffiehellman
jsonschema	Validação do esquema de JSON de cada mensagem	jsonschema
pyCrypto	Interfaces para operações criptográficas	Crypto

Processos

Tendo em conta o código desenvolvido e, tal como era base do projeto, o nosso código suporta:

1. Criar e terminar leilões
2. Listar leilões abertos e fechados
3. Mostrar todas as licitações de um leilão
4. Mostrar todas as licitações de um cliente
5. Verificar o resultado de um leilão em que o cliente participou
6. Validar recibos

Tal como era pedido foram implementados dois tipos de leilões:

1. English auction – No qual a licitação é feita baseando-se no valor da anterior aumentando-a
2. Blind auction – Leilão em que as licitações são feitas às cegas, isto é, o cliente não conhece as bids feitas por outros utilizadores e, como tal, licita sem qualquer dependência das anteriores licitações.

No decorrer do presente relatório iremos explicar o funcionamento de cada um destes processos.

Mensagens

No que concerne a troca de mensagens optamos por fazer o intercâmbio das mesmas sempre com objetos JSON sendo que o esquema é validado por recurso a uma livraria chamada `jsonschema`.

De um modo prático definimos um formato das mensagens que queremos, se o esquema dessas mensagens não corresponder ao que é expectável o servidor retorna uma mensagem a dizer que o mesmo é inválido e por isso o utilizador nada consegue fazer.

Os campos sensíveis das mensagens foram contudo encriptados de modo a garantir a integridade e segurança do sistema, no que diz respeito a este ponto será explicado com mais detalhe nos tópicos seguintes.

Protocolos de comunicação

Para estabelecer as conexões entre os servidores e clientes optamos por usar sockets sendo este o canal de comunicação usado para passar as mensagens json.



Cada parte integrante do sistema tem a sua porta específica. O Auction Repository encontra-se na porta 8081, o Auction Manager encontra-se na porta 8080.

Deste modo para que seja possível iniciar as comunicações o cliente deve-se ligar na porta 8080 depois do Auction Repository se ter ligado na porta 8081 e o Auction se ter ligado na porta 8080 e conectado na 8081.

Quando todos se encontram ligados da forma acima descrita é possível começar comunicações seguras entre todos e, como tal, é possível começar a criação leilões e realizar licitações de forma segura.

Negociação de chaves

Primeiramente são carregadas as chaves publicas que são distribuídas da forma:

-  Auction Repository <-> Auction Manager
-  Auction Repository <-> Auction Client

Não é problemático distribuir a chave publica, contudo tem de ser verificada para nos certificarmos que a chave pertence mesmo a quem achamos que é. Isto é também possível fazer utilizando a fingerprint, deixando assim de ser necessário realizar a verificação da cadeia de certificação.

Fases de negociação

Fase 1

Troca de chaves entre as primeiras entidades a serem abertas, no caso o Auction Repository e o Auction Manager.

Ambos procedem à verificação que cada um diz ser quem é, verificando a certification chain de cada certificado trocado e validando as chaves publicas.

Fase 2

Troca de chaves entre o Auction Manager e o Auction Client.

Ambos procedem à verificação que cada um diz ser quem é, verificando a certification chain de cada certificado trocado e validando as chaves publicas.

Fase 3

A partir do momento que todas as cadeiras de certificação foram verificadas e todas as chaves trocadas os servidores e cliente assumem que podem confiar um no outro e, como tal, está estabelecido uma comunicação segura entre ambos.

Funcionalidades implementadas

Para alcançar uma comunicação segura entre os intervenientes deste projeto foram implementadas as seguintes features:

1. Encriptação das mensagens trocadas
2. Proteção das licitações até ao final do leilão
3. Identificação do autor das licitações com o Cartão de Cidadão (uuid)
4. Expor a informação necessária sobre uma licitação no final de um leilão
5. Construção de uma blockchain por leilão
6. Lançamento de cryptopuzzels
7. Produção e validação de recibos

8. Validação de um leilão fechado

Apesar de terem sido pedidos mais features no enunciado não conseguimos implementar algumas entre as quais:

1. Validação de bids usando código dinâmico
2. Modificação de bids usando código dinâmico

Ficando, assim, deste modo, tudo o que estava relacionado com código dinâmico por implementar.

Encriptação das mensagens trocadas

Para garantir as comunicações seguras entre todos optamos por encriptar os campos sensíveis das mensagens, o que é considerado sensível depende de leilão para leilão, consideremos que os campos são encriptados de diferentes formas quando o leilão é do tipo “english auction” e quando é “blind auction”.

English Auction

Neste tipo de leilão não são encriptados campos específicos da mensagem, a mensagem em si é que é encriptada pelo cliente e é enviada pelo canal de comunicação, assim que que chega ao Manager este descripta a mensagem para verificar a sua autenticidade e novamente encripta-a enviando todos os pedidos para o Auction Repository onde toda a informação é guardada.

Blind Auction

No caso da blind auction que é, como o próprio nome indica um leilão às cegas, optamos por encriptar o campo do valor deste modo garantimos que ninguém, de facto, sabe e como tal não pode viciar o leilão.

Para além disto o conteúdo das mensagens é, obviamente, sempre encriptado de modo a que a comunicação seja segura.

Identificação do autor das licitações com o Cartão de Cidadão

A partir do cartão é gerado o identificador da pessoa, utilizador o cartão de cidadão para obter o UUID. Para este efeito não é necessário qualquer pin de autenticação por parte do cliente, apenas usamos uma função muito simples que nos permite gerar isso.

```
def generate_uuid(self):  
    pem = self.get_certificate_pem()  
    return hashlib.sha224(pem).hexdigest()
```

Figura 1 - Geração UUID

Proteção das licitações até ao final do leilão

As licitações encontram-se guardadas na blockchain e devidamente encriptadas de modo a que, apenas no final a informação sensível sobre as mesmas seja revelada.

Contudo sempre que o seu criador achar necessário pode pedir para ver as suas licitações e, como tal, o servidor irá retornar as mesmas.

Expor a informação necessária sobre uma licitação no final de um leilão

Como todo o conteúdo se encontra encriptado, apenas quem tem o correspondente par de chaves pode fazer a sua descriptação, deste modo, só é possível que os utilizadores vejam esse mesmo conteúdo no final.

Construção de uma blockchain por leilão

Cada novo leilão tem direito à criação de uma nova blockchain, essa blockchain é armazenada em disco possibilitando que, caso o servidor pare, seja mesmo assim possível recuperar tudo o que já se fez antes.

O conteúdo de toda a blockchain está encriptado e foram criados mecanismos de controlo de integridade da mesma garantindo assim que o leilão não teve qualquer adulteração. Caso a blockchain seja de alguma forma alterada o servidor informará que a mesma é inválida e tratará de a terminar.

Verificar o resultado de um leilão em que o cliente participou

Como numa primeira fase é gerado o UUID do utilizador via cartão de cidadão é assim possível a qualquer momento fazer tracking de toda a sua atividade desde que as suas chaves sejam válidas e o mesmo se consiga autenticar no servidor através do seu cartão de cidadão.

Validar recibos

Como forma de provar que nada de errado se passou o servidor envia um recibo ao cliente a provar que o mesmo fez determinada ação, com isso o cliente, caso detete alguma

irregularidade poderá garantir o que se passou e num outro sentido, não poderá repudiar algo que tenha feito visto que os documentos se encontram assinados.

Chaves

Servidores

Para gerar as chaves privadas e publicas dos servidores, isto é, Auction Manager e Auction Repo foi criado um programa em python que se baseia na livreria cryptography do python. Estas chaves privadas possuem um tamanho de 2048 bits e as correspondentes chaves publicas são baseadas nestas.

Clientes

Apesar dos clientes se autenticarem com o CC não nos é possível utilizar encriptações baseadas no cartão de cidadão nem as consequentes encriptações por isso mesmo cada uuid retirado do cartão de cidadão tem um conjunto de pares de chaves gerado e que é trocado com o servidor de modo a que este possa garantir a integridade.

Cartão de Cidadão

Utilizamos o cartão de cidadão para autenticar o utilizador, de um modo geral, se o cliente colocar o seu pin de forma correta e a sua certification chain for válida este tem acesso e está habilitado para usar o sistema. Em termos práticos o que acontece é:

- ✓ Se a certification chain é valida o utilizador cumpre os requisitos e, como tal, pode se autenticar e participar em leilões ou mesmo criá-los.
- ✓ Se a certification chain é inválida isto implica que o utilizador não cumpre os requisitos e, como tal, nenhum acesso lhe é fornecido não conseguindo assim entrar no sistema.