Public Ledger for Auctions

Segurança de Sistemas e Dados 2024/2025

Eduardo Luís Fernandes Roçadas
Faculdade de Ciências
Universidade do Porto
Porto, Portugal
up202108758@up.pt

Leonardo Araújo Freitas
Faculdade de Ciências
Universidade do Porto
Porto, Portugal
up202400832@up.pt

Manuel Ramos Leite Carvalho Neto
Faculdade de Ciências
Universidade do Porto
Porto, Portugal
up202108744@up.pt

***Abstract*—Tradicionalmente, um sistema de leilões segue uma lógica centralizada, assente numa plataforma ou entidade central que é responsável por garantir a integridade do funcionamento de todo o processo, em particular ao assegurar a validade de todas as ações e transações efetuadas pelos utilizadores/clientes. No entanto, recorrendo a uma *blockchain* e a um protocolo de comunicação segura *Peer-to-Peer* como o S/Kademlia, é possível implementar um sistema semelhante em termos de funcionamento e de garantias de segurança, mas de forma descentralizada, logo, mais democrático e eficaz.**

Assim, explora-se a arquitetura subjacente a este sistema descentralizado para leilões, dando particular destaque ao racional associado a cada componente e às decisões tomadas para a sua conceção, juntamente com as assunções implícitas ao funcionamento correto e seguro do sistema como um todo.

***Index Terms*—Blockchain, Proof-of-Work, Proof-of-Reputation, S/Kademlia, Sybil, Eclipse, Leilões**

I. Introdução

No âmbito da Unidade Curricular Segurança de Sistemas e Dados, propõe-se o desenvolvimento de um projeto que consiste na implementação de uma *blockchain* pública cujo propósito passa por ser capaz de armazenar, de forma descentralizada, transações de leilões. Nesse sentido, o trabalho divide-se em três partes: (1) a *blockchain* distribuída propriamente dita, (2) o protocolo de comunicação segura *Peer-to-Peer* (P2P) e (3) o mecanismo de leilões.

Este relatório visa documentar as decisões de *design* e de arquitetura derivadas dos requisitos funcionais do sistema, assim como as assunções resultantes das limitações teóricas e práticas impostas ao desenvolvimento do mesmo.

II. Arquitetura

Tal como referido, o sistema assenta numa arquitetura tripartida, constituída pela *blockchain*, o S/Kademlia - enquanto protocolo de comunicação segura P2P - e o mecanismo de leilões. O diagrama da Figura 1 mostra, numa visão de alto-nível, como é que os três elementos se relacionam entre si.

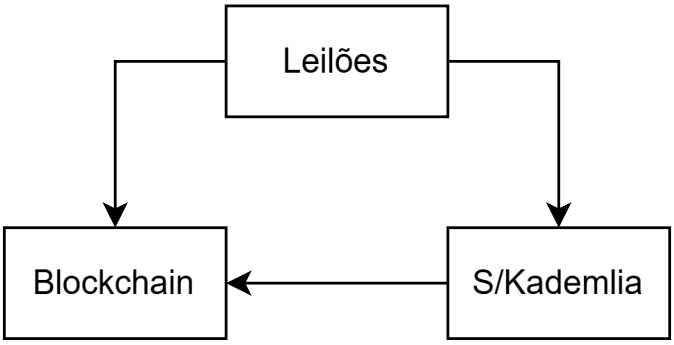

Fig. 1. Arquitetura

Tal como demonstra a Figura 1, a *blockchain* é utilizada pelo S/Kademlia que, por sua vez, suporta o mecanismo de leilões. Assim, a *blockchain* funciona como o armazenamento de cada nó no protocolo S/Kademlia, para representar a informação sobre os leilões e as propostas.

As subsecções seguintes explicam o funcionamento de cada componente de forma mais detalhada.

A. Blockchain

A *blockchain* é o elemento basilar do sistema proposto, na qual devem ser armazenadas as transações associadas aos leilões, agrupadas em blocos.

A Figura 2 permite visualizar a estrutura base da *blockchain*.

Fig. 2. Blockchain

Efetivamente, como se evidencia na Figura 2, a *blockchain* é constituída por uma lista de blocos minerados que, por sua vez, contêm um dado número de transações, limitado a 10 pela implementação.

1) Proof-of-Work: O funcionamento da *blockchain* assenta num mecanismo de consenso, implementado através de *Proof-of-Work* (PoW). Ora, os nós da rede registam transações em blocos, que devem ser minerados para serem validados. Cada bloco contém uma lista de transações, juntamente com a *hash* do bloco anterior - para garantir integridade e consistência da sequência - e um número de uso único (*nonce*). Deste modo, de maneira a minerar um bloco, cada nó deve computar o *nonce* que, combinado com os restantes elementos do bloco (transações e *hash* anterior), resulta numa *hash* que se inicia por um dado número de zeros, definido através do parâmetro de dificuldade. Assim, quando este valor é determinado, o nó adiciona o bloco à *blockchain* e este bloco pode ser validado pelos restantes membros da rede, através da computação da respetiva *hash*. Com isto, garante-se a implementação correta e segura de uma *blockchain* descentralizada, com *Proof-of-Work* como mecanismo de consenso.

2) Proof-of-Reputation: Outra possibilidade considerada para mecanismo de consenso é o conceito de *Proof-of-Reputation* (PoR), que consiste em atribuir o direito de validar transações e minerar blocos com base na reputação de cada nó, dependendo do seu histórico de comportamentos e de um determinado mecanismo de confiança para determinar a sua fiabilidade. No entanto, apesar de proposto e ponderado, opta-se por não se implementar *Proof-of-Reputation* como mecanismo de consenso. Esta decisão deve-se ao facto de que, numa rede de tamanho tão diminuto como a que se concretiza para efeitos de demonstração, o mecanismo de *Proof-of-Reputation* poderia rapidamente levar à centralização do poder num número reduzido de nós, contrariando a justiça e a democracia pretendidas. Além disto, também se torna difícil determinar eficazmente a reputação de um determinado nó, o que limita a implementação deste mecanismo. Como tal, a *blockchain* implementada tem *Proof-of-Work* como único mecanismo de consenso.

B. S/Kademlia

A implementação do S/Kademlia procura ser tão fiel quanto possível à especificação do protocolo original [1], incluindo resistência a ataques Sybil e Eclipse [2].

Para tal, o S/Kademlia considera três entidades fundamentais e implementa cinco *Remote Procedure Calls* (RPCs). Os elementos constituintes do S/Kademlia são os nós, as *routing tables* e os *k-buckets*. A Figura 3 demonstra a interação entre estes componentes.

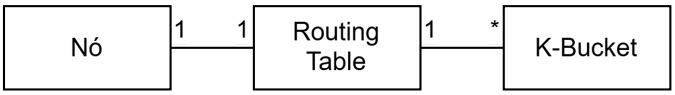

Fig. 3. S/Kademlia

Ora, conforme mostra a Figura 3, cada nó contém uma *routing table*, que é constituída por um número limitado de *k-buckets*. Na implementação, este limite é definido para 160. De acordo com o protocolo original, o propósito dos *k-buckets* passa por armazenar nós agrupados por distância XOR do seu ID relativamente ao ID do nó original. Deste modo, consegue-se um mecanismo eficiente de organização/estruturação da visão parcial de cada nó da rede, para uma comunicação eficaz.

De maneira a possibilitar esta mesma comunicação, implementam-se os quatro RPCs do protocolo original: **PING**, **STORE**, **FIND_NODE** e **FIND_VALUE**. O **PING** verifica se outro nó se encontra ativo, o **STORE** solicita o armazenamento de um par chave-valor noutro nó, o **FIND_NODE** procura por um nó na rede e o **FIND_VALUE** procura por um determinado valor. O quinto RPC é o **JOIN**, que se explica posteriormente.

A rede tem ainda tem três mecanismos essenciais para o seu funcionamento. Em primeiro lugar, a pesquisa iterativa por nós ou valores - usando o respetivo RPC - pretende aumentar a eficiência da comunicação, através do envio do mesmo pedido em paralelo para diferentes nós, sendo este número definido de acordo com um parâmetro de paralelismo α, configurado para três. Em segundo lugar, quando um *k-bucket* fica cheio, isto é, com $k = 20$ nós, é aplicado um mecanismo de substituição que segue uma lógica *Least Recently Used* (LRU), substituindo o nó há mais tempo não utilizado/contactado. Por último, sempre que um novo nó entra na rede, deve comunicar com determinados nós, denominados *bootstrap*, enviando-lhes um RPC **JOIN** - e não **FIND_NODE** como na especificação original, por razões a explicar posteriormente - para obter como resposta toda a informação necessária sobre a rede.

A implementação dos RPCs é feita seguindo a framework gRPC da Google usando *protocol buffers* (**protobuf**) como formato de serialização. Os RPCs especificam-se no ficheiro **protoc/kademlia.proto**.

1) Resistência a Ataques Sybil: Um ataque Sybil caracteriza-se pelo comportamento malicioso de um atacante que cria diversas identidades/nós, de maneira a conseguir controlar a rede e influenciar o seu funcionamento. Na implementação, este ataque é prevenido/mitigado ao obrigar a que cada nó que se junte à rede tenha de cumprir um desafio, numa lógica de *Proof-of-Work*. Assim, para entrar na rede, o novo nó deve demonstrar ao nó *bootstrap* que computou corretamente a *hash* de um determinado valor para se iniciar por um dado número de zeros - definido pelo parâmetro de dificuldade -, o que implica o gasto de recursos computacionais e inviabiliza ou dificulta este tipo de ataques. Para este efeito, utiliza-se o RPC **JOIN**.

2) Resistência a Ataques Eclipse: Um ataque Eclipse consiste no comportamento malicioso de um nó com o intuito de isolar outro nó da rede, limitando ou manipulando a informação que lhe é transmitida. A prevenção deste tipo de ataques assenta não só na natureza distribuída do S/Kademlia e na lógica de *Proof-of-Work* anteriormente explicada, mas também, essencialmente, na fiabilidade dos nós *bootstrap*. Como estes nós são predefinidos - logo, confiáveis - e é com eles que cada novo nó estabelece a comunicação inicial, são sempre conhecidos por todos os membros da rede, pelo que são responsáveis por transferir a informação correta de forma fidedigna.

C. Leilões

O mecanismo de leilões é a interface de mais alto-nível do sistema desenvolvido, com a qual os utilizadores podem interagir diretamente, através do menu.

1) Ações sobre Leilões e Propostas: A lógica do sistema de leilões segue o modelo inglês, assente em leilões com propostas crescentes. Nesse sentido, existem três tipos de ações passíveis de serem realizadas pelos utilizadores: criar, iniciar e terminar leilões. A criação de um leilão permite definir as suas propriedades/características iniciais para ser, posteriormente, iniciado. O término de um leilão implica a atribuição do bem leiloado ao utilizador com a licitação mais elevada, em troca do respetivo montante. Assim, cada utilizador tem, evidentemente, a possibilidade de licitar cada leilão, definindo uma quantia que está disposto a pagar para obter o ativo leiloado. Todas estas ações estão disponíveis através do menu.

2) Publisher/Subscriber: A partilha/transferência de informação sobre os leilões na rede é feita segundo um mecanismo *publisher/subscriber*. O funcionamento deste mecanismo baseia-se em dois elementos fundamentais: o *publisher* - que publica uma nova informação (como um leilão ou uma proposta) na rede - e o *subscriber* - que subscreve determinados leilões com o intuito de receber novas informações (propostas) sobre o mesmo. Efetivamente, a implementação segue este princípio, na medida em que quando um utilizador inicia um leilão, publica essa informação na rede, para que esse leilão possa ser subscrito por outros utilizadores da rede, de maneira a serem atualizados quando ocorrer uma nova proposta sobre o mesmo, bem como quando terminar. Além disto, quando um nó licita um leilão, torna-se automaticamente subscritor do mesmo, para receber informação sobre novas licitações e sobre a conclusão do leilão. Assim, segue-se a lógica *publisher/subscriber* para a divulgação da informação sobre os leilões.

D. Integração

Ora, estes três componentes são essenciais ao desempenho do sistema como um todo, mas devem estar devidamente integrados para o seu correto funcionamento. Assim sendo, descreve-se sucintamente a integração do sistema desenvolvido.

A *blockchain*, enquanto estrutura de dados basilar do sistema, possibilita o armazenamento de todas as informações relacionadas com os leilões e as respetivas transações. Em particular, a *blockchain* regista todas as ações de criação, início e término de leilões, bem como as propostas efetuadas pelos utilizadores.

Quando um nó entra na rede, é-lhe atribuído um par de chaves criptográficas assimétricas, juntamente com uma cópia do estado atual da *blockchain*. Com isto, cada utilizador pode participar ativamente em todo o processo de leilões (na sua criação, início e término), assim como em licitações. Para tal, o utilizador deve assinar todas as suas transações com a própria chave privada, para que possam ser verificadas pelos outros utilizadores através da chave pública correspondente. Cada nó pode ainda minerar blocos de modo a contribuir para a *blockchain*, sendo esta mineração efetuada numa *thread* separada da *thread* principal, que lida com os leilões.

Note-se que, para efeitos de demonstração, existe um mecanismo de injeção de falhas que permite terminar um ou mais nós de forma simultânea, para mostrar a robustez do sistema. Este mecanismo funciona através do um novo RPC - **SHUTDOWN** -, que solicita que um nó abandone a rede imediatamente.

III. Assunções

Apesar de todos os esforços no sentido de garantir a máxima segurança, robustez, eficácia e viabilidade do sistema, existem algumas assunções que devem ser estabelecidas, das quais se podem destacar quatro.

Em primeiro lugar, o correto funcionamento da *blockchain* implica que os nós minerem os blocos. Num contexto real, tende a existir um incentivo para que os nós estejam dispostos a fazê-lo, visto que minerar implica gastar recursos computacionais para resolver um problema difícil. Contudo, no âmbito deste projeto, esse incentivo não existe, pelo que se assume que os nós mineram os blocos por sua opção voluntária, contribuindo para a manutenção da rede num espírito colaborativo.

Em segundo lugar, também quanto à *blockchain*, surge a necessidade de decidir como é que cada nó deve lidar com blocos concorrentes, isto é, com ramificações da mesma cadeia que seguem caminhos divergentes. Perante este problema, a decisão tomada passa por preservar o caminho com o maior número de blocos válidos, assumindo, implicitamente, que este caminho é o correto. Efetivamente, considera-se razoável este raciocínio, visto que um maior número de blocos implica mais recursos computacionais gastos, pelo que é mais provável que provenha de uma maioria de nós honestos.

Em terceiro lugar, assume-se que os nós *bootstrap* são confiáveis, legítimos e honestos. Esta assunção deriva do facto de ser necessário estabelecer uma raiz/âncora de confiança para inicializar o sistema distribuído, que, neste caso, acaba por recair sobre os nós *bootstrap*. Tendo em conta que estes nós são definidos pelo utilizador aquando do início da execução do programa, esta assunção parece razoável.

Por último, a atribuição do par de chaves pública e privada a cada nó aquando da sua entrada na rede segue um processo determinístico, com base na origem (endereço IP e porto) do nó. Com isto, não só se pressupõe que a cada origem só pode estar associado um único nó, isto é, uma única entidade, mas também que a mesma origem corresponde sempre ao mesmo nó, de forma permanente. Efetivamente, esta suposição resulta da necessidade de permitir que o mesmo utilizador saia da rede num dado momento, com a possibilidade de se voltar a conectar, caso em que deve assumir a mesma identidade e ter acesso à mesma informação que detinha anteriormente. Assim, por simplificação, opta-se por derivar as chaves criptográficas a partir da origem do nó, em vez de obrigar a um processo de autenticação aquando da entrada de cada nó na rede.

Em síntese, existem quatro assunções fundamentais que sustentam a correção e a segurança do sistema desenvolvido, derivadas diretamente das necessidades do mesmo e das respetivas limitações.

IV. Conclusão

Em suma, implementou-se um sistema distribuído descentralizado capaz de suportar um mecanismo de leilões, assente numa *blockchain* e num protocolo de comunicação segura *Peer-to-Peer*.

Para tal, definiu-se a arquitetura que o sistema como um todo deve seguir. Nesta arquitetura, o armazenamento de todas as informações relativas às transações reside na *blockchain*, regida por *Proof-of-Work* como mecanismo de consenso. Assim, todos os nós são responsáveis por minerar blocos, garantindo a integridade da cadeia de transações. O protocolo de comunicação P2P é o S/Kademlia, construido no topo do Kademlia original de maneira a resistir a ataques Sybil e Eclipse. A este protocolo, além dos quatro RPCs especificados originalmente, acrescentou-se ainda um RPC adicional para a entrada de cada novo nó na rede, encarregue de estabelecer a comunicação com um nó *bootstrap* e realizar *Proof-of-Work*. Finalmente, o mecanismo de leilões segue o modelo convencional inglês, com licitações ascendentes e com a possibilidade de criar, iniciar e terminar leilões, atribuindo o bem/ativo leiloado à proposta vencedora, ou seja, à de montante mais elevado.

Em todo o sistema desenvolvido, procurou-se tratar a segurança como um cidadão de primeira classe, nomeadamente ao endereçar questões de identidade, autorização, autenticação, controlo de acessos e domínios de confiança. Além disto, a criptografia utilizada vai ao encontro dos objetivos pretendidos para o mesmo, em particular para a assinatura e verificação de transações.

Em último lugar, implementou-se também um mecanismo de injeção de falhas, que visa demonstrar a robustez do sistema em caso de falha simultânea de um ou mais nós da rede.

Assim sendo, os objetivos do projeto consideram-se cumpridos, pelo que o projeto se considera bem-sucedido.

References

[1] Petar Maymounkov, David Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric", International Workshop on Peer-to-Peer Systems, pp. 53–65, Springer, 2002.

[2] Ingmar Baumgart, Sebastian Mies, "S/kademlia: A practicable approach towards secure key- based routing", Parallel and Distributed Systems, 2007 International Conference, pp. 1–8, IEEE, 2007.