UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UMA ABORDAGEM PARA AUTO IDENTIFICAÇÃO VOLUNTÁRIA E VERIFICÁVEL DE PARTICIPANTES DE APLICAÇÕES BASEADAS EM LIVROS-RAZÃO DISTRIBUÍDOS

MARCELO HÉRCULES CUNHA SOARES

JOÃO PESSOA - PB

Abril - 2020

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UMA ABORDAGEM PARA AUTO IDENTIFICAÇÃO VOLUNTÁRIA E VERIFICÁVEL DE PARTICIPANTES DE APLICAÇÕES BASEADAS EM LIVROS-RAZÃO DISTRIBUÍDOS

MARCELO HÉRCULES CUNHA SOARES

Dissertação submetida ao Centro de Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Orientador: Prof. Dr. Rostand Edson Oliveira Costa

João Pessoa
2020

Catalogação na publicação
Seção de Catalogação e Classificação

S676a Soares, Marcelo Hercules Cunha.
Uma abordagem para auto identificação voluntária e verificável de participantes de aplicações baseadas em livros-razão distribuídos / Marcelo Hercules Cunha Soares. - João Pessoa, 2020.
101 f. : il.

Orientação: Rostand Edson Oliveira Costa.
Dissertação (Mestrado) - UFPB/CI.

1. Tecnologia de livro-Razão distribuído. 2. DLT. 3. blockchain. 4. certificação digital. 5. assinatura digital. 6. associação entidade-endereço. I. Costa, Rostand Edson Oliveira. II. Título.

UFPB/BC

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Ata da Sessão Pública de Defesa de Dissertação de Mestrado de Marcelo Hercules Cunha Soares, candidato ao título de Mestre em Informática na Área de Sistemas de Computação, realizada em 05 de maio de 2020.

Aos cinco dias do mês de maio do ano de dois mil e vinte, às dez horas, reuniram-se os membros da Banca Examinadora constituída para julgar o Marcelo Hercules Cunha Soares, vinculado a esta Universidade sob a matrícula nº 20181000798, candidato ao grau de Mestre em Informática, na área de "Sistemas de Computação", na linha de pesquisa "Computação Distribuída", do Programa de Pós-Graduação em Informática, da Universidade Federal da Paraíba. A comissão examinadora foi composta pelos professores: Rostand Edson Oliveira Costa (PPGI-UFPB) Orientador e Presidente da Banca, Guido Lemos de Souza Filho (PPGI-UFPB), Examinador Interno, Daniel Faustino L de Souza (UFERSA), Examinador Externo à Instituição, Denio Mariz Timoteo de Sousa (IFPB), Examinador Externo à Instituição. Dando início aos trabalhos, a Presidente da Banca cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo fizesse a exposição oral do trabalho de dissertação intitulado: "Uma Abordagem para Auto Identificação Voluntária e Verificável de Participantes de Aplicações Baseadas em Livros-Razão Distribuídos". Concluída a exposição, o candidato foi arguido pela Banca Examinadora que emitiu o seguinte parecer: "***aprovado***". Do ocorrido, eu, Ruy Alberto Pisani Altafim, Coordenador do Programa de Pós-Graduação em Informática, lavrei a presente ata que vai assinada por mim e pelos membros da banca examinadora. João Pessoa, 05 de maio de 2020.

Prof. Dr. Ruy Alberto Pisani Altafim

Prof. Dr. Rostand Edson Oliveira Costa
Orientador (PPGI-UFPB)

Prof. Dr. Guido Lemos de Souza Filho
Examinador Interno (PPGI-UFPB)

Prof. Dr. Daniel Faustino L de Souza
Examinador Externo à Instituição (UFERSA)

Prof. Dr. Denio Mariz Timoteo de Sousa
Examinador Externo à Instituição (IFPB)

Dedico esta conquista ao meu avô Antônio Soares Nuto (in memorian) que foi exemplo de caráter e honestidade.

Agradecimentos

Agradeço à minha família por ser minha principal motivação para que eu consiga enfrentar os desafios que a vida me traz. Aos meus pais Marcílio e Marta por sempre estarem ao meu lado e confiarem em minhas decisões. À minha esposa Gizely e a Henrique por me fortalecerem indiretamente a cada dia. O amor de vocês foi o combustível essencial para que eu conseguisse chegar até aqui.

Agradeço aos meus amigos da Superintendência de Tecnologia da Informação que passaram de apenas colegas de trabalho para se tornarem grandes amigos, dos quais posso citar: Danilo Alexandre, Arthur Santos, Raphael Patrício, Jansen Cruz, Juracyr Lucena, José Augusto, Sandro Lopes, Alan Bonifácio e Jeysibel Dantas. Agradeço ao atual superintendente da STI, Hermes Pessoa Filho pelo apoio para o desenvolvimento deste trabalho.

Ao meu falecido avô Antônio Soares Nuto, a quem dedico este trabalho, por ter sido um homem exemplar e ter me passado valores que contribuíram para a minha formação pessoal e intelectual.

Ao meu orientador, professor Dr. Rostand Edson Oliveira Costa por ter confiado em minha pessoa e dedicado o seu tempo e esforço para a construção deste trabalho, sempre me motivando e extraindo o melhor de mim.

Agradeço também a toda a equipe de professores do PPGI que lecionaram as disciplinas cursadas por mim neste programa, assim como ao coordenador Caluirton e a secretária Maria Alice por todo o profissionalismo com que sempre me atenderam.

Resumo

As tecnologias de livro-razão distribuído popularizaram-se com o advento das criptomoedas, sobretudo a Bitcoin, surgindo novas aplicabilidades e novos desafios. Embora tenham sido concebidas em um contexto financeiro, associado à criptomoeda, atualmente é possível encontrar tais tecnologias sendo aplicadas em diversos contextos não-financeiros. Com a evolução da tecnologia, surgiram customizações com suporte a controle de permissão de acesso em redes privadas, onde os participantes da rede são previamente conhecidos. No entanto, as DLTs públicas podem ser consideradas mais seguras por possuírem um grande número de nós compondo as redes, o que torna ainda mais difícil a ocorrência de fraudes. No ecossistema das DLTs públicas, a privacidade e o anonimato são atributos desejáveis por parte dos utilizadores. Porém, há diversas evidências de que, em alguns casos, tais atributos podem ser dispensáveis, sendo necessário que a propriedade de um endereço de carteira digital seja conhecida para garantir a legitimidade de uma transação. Este trabalho atua na investigação de mecanismos de identificação de participantes em DLTs públicas, com a propositura de um serviço público de associação de endereço para entidade, alimentado voluntariamente pelo portador do endereço, chamado de Address Name System (ANS). O ANS combina o uso de tecnologias consolidadas, a exemplo de assinatura e certificação digital, para a geração de um artefato, denominado ANS Certificate, que associa criptograficamente o detentor de um endereço de carteira digital com uma entidade do mundo real, e fornece uma arquitetura para a consulta e validação de tais certificados. Como prova de conceito da proposta, foi implementado um protótipo do serviço e foram feitos experimentos de integração para a validação funcional das suas operações com aplicações reais.

Palavras-chave: Tecnologia de livro-Razão distribuído, DLT, blockchain, certificação digital, assinatura digital, associação entidade-endereço.

Abstract

Distributed ledger technologies have become popular through the advent of cryptocurrencies, especially Bitcoin, bringing new applicabilities and new challenges. Although they have been conceived in a financial context, associated to cryptocurrencies, it is currently easy to find such technologies being applied in several non-financial contexts. With the evolution of technology, there have been customizations with permission access control in private networks, where network participants are previously known. However, public DLTs can be considered safer because they have a large number of nodes composing networks, which makes fraud even more difficult. In the ecosystem of public DLTs, privacy and anonymity are desirable attributes for users. However, there is ample evidence that, in some cases, such attributes may be dispensable and the ownership of a digital wallet address must be known to ensure the legitimacy of a transaction. This work investigates the mechanisms of identification of participants in public DLTs, with the provision of a public address association service for entity, voluntarily fed by the address holder, called the Address Name System (ANS). The ANS combines the use of consolidated technologies, such as digital signature and certification, for the generation of an artifact, called ANS Certificate, which encryptively associates the holder of a digital wallet address with a real-world entity, and provides a architecture for the searching and validation of such certificates. As proof of the proposal's concept, a prototype of the service was implemented and integration experiments were performed for the functional validation of its operations with real applications.

Keywords: Distributed Ledger Technology, DLT, blockchain, digital certificate, digital signature, entity-address association.

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

É crescente a inserção das tecnologias de informação e comunicação em nosso cotidiano. Tais tecnologias vêm se disseminando e atuando em diferentes contextos sociais. Na Engenharia de Software, um sistema evolui para atender a novos requisitos, corrigir falhas, ou mesmo se adaptar a novas tendências de mercado (Sommerville, 2011). Analogamente ao que ocorre com os sistemas de informação, a evolução das tecnologias em geral, é contínua e se adapta às tendências do ecossistema em que estão contidas, surgindo novos conceitos e novas aplicabilidades. Um exemplo disso é a verdadeira revolução que as tecnologias de livro-razão distribuído (ou DLTs, do inglês *Distributed Ledger Technologies*) estão provocando em diversos segmentos produtivos, sobretudo o financeiro. Esta revolução pode ser vista como um movimento global que se iniciou com o advento das criptomoedas, cuja primeira concepção foi publicada em 2008 no artigo "Bitcoin: A Peer-to-Peer Electronic Cash System", por uma figura anônima que assinava com o pseudônimo Satoshi Nakamoto (Nakamoto, 2008a). Em sua proposta, Nakamoto especificou o que viria a posteriormente ser implementado como a *blockchain* da criptomoeda **Bitcoin**[1].

Apesar do seu surgimento ter se dado no contexto financeiro, associada à criptomoedas, as DLTs estão sendo exploradas para a utilização em diferentes tipos de aplicações. Como, basicamente, uma DLT é projetada para armazenar, de forma imutável, dados de transações, é possível expandir a sua utilização para além do contexto financeiro por meio do armazenamento de dados com outras semânticas. Campos reservados para conteúdos arbitrários em diversas implementações de criptomoedas, como a Bitcoin, permitiram que usuários inici-

[1] https://bitcoin.org/

assem experimentos utilizando tais DLTs públicas para um propósito diferente do que foi inicialmente proposto (Sward et al., 2018). Atualmente, podemos encontrar aplicações apoiadas nos livros-razão distribuídos em diferentes áreas, das quais podemos citar: aplicações notariais, saúde, indústria, agricultura, telecomunicações, etc (Lemieux et al., 2018; Taylor et al., 2016). A Figura 1.1 representa uma abstração arquitetural de alto nível sobre aplicações apoiadas em livros-razão distribuídos. O potencial das DLTs aliado a sua flexibilização na utilização em diversos segmentos, vêm atraindo o interesse de governos de vários países para a exploração dessa nova tecnologia em direção ao provimento de serviços à sociedade (Deshpande et al., 2017; Dhar e Bose, 2016; Maltese, 2015; Taylor et al., 2016). Na América do Sul, o Brasil se tornou o primeiro país a realizar uma prova de conceito bem sucedida na utilização de DLT pelo governo, através do desenvolvimento de um sistema de verificação de documentos de identidade (Bakker, 2018).

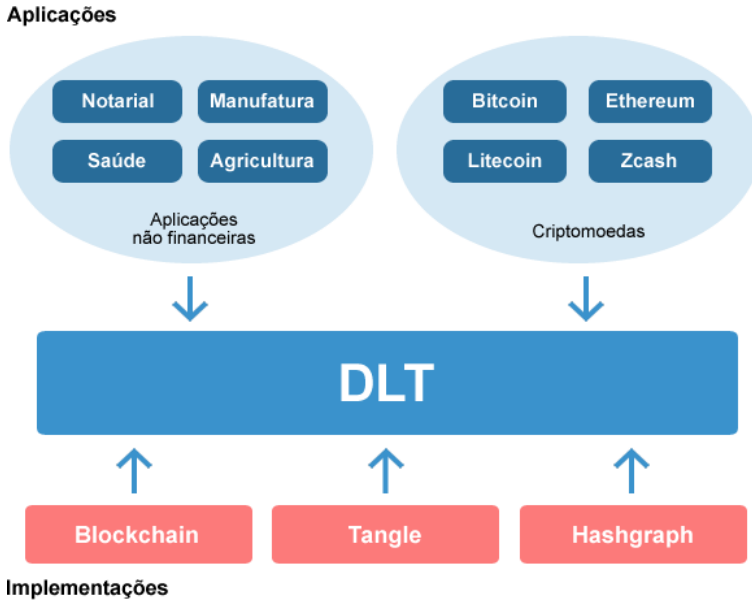

Figura 1.1: Aplicações em DLTs

Fonte: Próprio autor

As partes envolvidas em transações são associadas a um par de chaves: chave pública e chave privada. A chave privada é utilizada para a autenticação e assinatura de transações. A partir da chave pública é derivado um identificador que representa os participantes de uma DLT. Como as principais DLTs em operação surgiram de forma associada com alguma criptomoeda, os identificadores são comumente referenciados como “endereço de carteira

digital", numa alusão ao tradicional receptáculo usado para guardar dinheiro.

Como ocorre com diversas categorias de tecnologias e sistemas de informação, a evolução e o amadurecimento são dependentes de diversos aspectos, incluindo segurança. No contexto em pauta, os termos privacidade e anonimato são comumente encontrados na literatura que envolve as tecnologias de livro-razão distribuído, sobretudo nas DLTs públicas, onde não há qualquer controle de gestão dos participantes que compõem a rede. Para uma parte da comunidade de usuários de DLTs, o anonimato e a privacidade são atributos imprescindíveis. No entanto, diversas evidências apontam que, em certos cenários envolvendo aplicações baseadas em DLTs, a privacidade e o anonimato das partes envolvidas em transações podem ser dispensáveis, havendo ainda em certos casos, a identificação segura dos atores por trás dos endereços de carteiras, como pré-requisito para a legitimação das transações (Costa et al., 2018; Júnior et al., 2018). Na web tradicional, a confiança de que um usuário acessa um serviço autêntico pode ser atendida através de uma conexão segura e certificados digitais emitidos por autoridades responsáveis. É necessário que tal confiança também exista no escopo das redes de DLTs. Esse cenário aponta para uma questão em aberto no atual contexto de utilização das DLTs públicas: como identificar atores envolvidos em transações de forma segura, legítima, sem a intervenção no desejo por anonimato e privacidade por parte de outros atores?

1.1 Justificativa

As tecnologias de livro-razão distribuído estão em constante amadurecimento e crescente utilização, seja no setor privado, ou mesmo por governos e instituições públicas. Projeções apontam para uma expectativa de continuidade no crescimento da utilização de tais tecnologias. Segundo a *International Data Corporation* (IDC), espera-se que o gasto anual com DLTs chegue a 9,7 bilhões de dólares em 2021 (IDC, 2018).

Desde as sua primeiras implementações até os dias atuais, as DLTs são exploradas por uma gama de aplicações e comunidades, onde a privacidade e anonimato são atributos desejáveis por parte dos usuários. No entanto, embora estejam surgindo evidências de que o anonimato pode, em alguns casos, ser dispensável, como também há cenários em que a identificação se faz necessária, ainda há uma lacuna na literatura acerca de propostas de soluções

para a identificação das entidades por trás de endereços de carteiras. O requisito de identificação dos atores pode não ser um problema para as DLTs privadas, uma vez que apenas atores selecionados podem compor a rede (Deshpande et al., 2017).

Entretanto, um dos princípios das DLTs é a confiança provida pela rede aos seus usuários. Em certos casos, decisões de projetos apontam para uma preferência por utilização de redes públicas, uma vez que os registros de tais DLTs são públicos e qualquer interessado pode auditar as transações (BNDES, 2018). Redes públicas com muitos nós são consideradas mais seguras já que estão menos vulneráveis aos ataques de 51%, que ocorre quando um participante detém mais de 50% de poder de processamento da rede, podendo comprometer a integridade dos dados do livro-razão (Watanabe et al., 2016). Por outro lado, as redes públicas são abertas para a participação por qualquer interessado, sem mecanismos para a identificação dos usuários que as compõem.

Provas de conceito utilizando DLTs foram realizadas pelo governo brasileiro. Em projetos desenvolvidos por instituições públicas, como é o caso do BNDES (Júnior et al., 2018) e da Universidade Federal da Paraíba (Costa et al., 2018), a necessidade de identificação das entidades por trás das chaves foi explícita. Percebe-se, claramente, que o cenário exposto justifica a propositura e desenvolvimento deste projeto de pesquisa. Estes projetos são abordados em trabalhos relacionados, na seção 3.

Neste sentido, através da investigação realizada para o entendimento da atual situação das tecnologias de livro-razão distribuído e dos conceitos relacionados, este trabalho apresenta os resultados de um projeto de pesquisa focado na investigação da utilização de tecnologias consolidadas, a exemplo de certificação e assinatura digital (Stallings, 2015), para o provimento de um modelo que garanta a associação confiável entre endereços de carteiras e entidades do mundo real[2], contribuindo assim para a evolução e amadurecimento das DLTs, permitindo a sua exploração por uma gama maior de aplicações.

[2]Os termos "entidade do mundo real" ou "pessoa/entidade" serão usados para fazer referência à uma pessoa física ou jurídica que tenha a posse de algum certificado digital válido.

1.2 Objetivos

1.2.1 Objetivo Geral

Considerando o cenário exposto anteriormente, o objetivo geral deste trabalho é atuar na resolução do problema de identificação de atores por trás de endereços de carteiras, com o desenvolvimento de um mecanismo de identificação que possibilite que essa categoria de DLTs possa ser utilizadas em cenários onde a identificação das partes envolvidas em transações seja necessária para prover mais segurança às aplicações, contribuindo assim com a evolução das tecnologias de livro-razão distribuído públicas. É importante também mencionar que não é objetivo deste trabalho sugerir a quebra permanente de anonimato no ecossistema das DLTs, mas sim, propor mecanismos que permitam a convivência pacífica dos usuários que desejam privacidade com os atores de aplicações que precisam da identificação das partes de uma transação.

1.2.2 Objetivos Específicos

- Definir uma infraestrutura consolidada a ser utilizada que provenha a ligação entre chave pública e entidade.
- Definir um modelo que permita a identificação dos atores por trás de endereços de carteiras, isto é, a associação entre entidades do mundo real e endereços de carteiras digitais.
- Desenvolver um modelo arquitetural de um sistema de referência que seja agnóstico em relação à implementação de DLT.
- Implementar o sistema de referência de acordo com o modelo especificado.
- Validar a abordagem proposta através de experimentos e testes em cenários reais.
- Apresentar e avaliar os resultados obtidos.

1.3 Estrutura da Dissertação

O restante deste trabalho está estruturado da seguinte forma:

Capítulo 2 No capítulo 2 é feita uma revisão da literatura acerca dos conceitos e tecnologias necessários para o compreendimento deste trabalho. Este capítulo traz o resultado de uma investigação da literatura, com a compilação dos principais tópicos envolvidos na temática tratada. Este capítulo foi escrito de modo que conduza o leitor por uma sequência coesa de conceitos considerando a dependência de que alguns tópicos possuem com relação a outros. Nesta seção também é feita uma discussão sobre anonimato e privacidade em DLTs com foco na problemática a ser tratada neste trabalho.

Capítulo 3 O capítulo 3 elenca alguns trabalhos relacionados ao tema em questão. Foram identificadas abordagens que atuam na identificação de atores em algumas DLTs públicas como também trabalhos que se relacionam por compartilharem da mesma problemática.

Capítulo 4: Este capítulo apresenta o trabalho proposto, com a descrição das metodologias utilizadas, a apresentação da estratégia proposta para atender aos objetivos descritos, como também apresenta a solução desenvolvida chamada de *Address Name System* ou ANS. O objetivo do ANS é permitir que o detentor de um endereço de carteira digital declare a sua posse voluntariamente em um padrão verificável de forma autônoma por qualquer interessado. Para a validação da abordagem proposta, foi desenvolvido um protótipo funcional do ANS, que é apresentado e detalhado neste capítulo.

Capítulo 5 No capítulo 5 são apresentados os resultados de experimentos realizados com o ANS. Integrações foram realizadas com aplicações existentes para a validação de viabilidade do sistema.

Capítulo 6 Por fim, este capítulo apresenta as considerações finais, discussões e possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Para a completa compreensão da estratégia adotada, faz-se necessário uma breve revisão sobre alguns dos conceitos e tecnologias utilizadas na sua implementação. Nesta seção faremos uma revisão sobre conceitos relacionados à criptografia, especialmente a criptografia de chave pública (assimétrica), uma vez que as tecnologias envolvidas nesta pesquisa, tal como a solução proposta, lidam com criptografia e assinatura digital. Em seguida, serão explanados conceitos acerca de infraestrutura de chave pública e seus componentes, a exemplo de certificação digital, como também acerca de suas arquiteturas para a distribuição de chaves, tais como: Autoridade Certificadora (AC) e PGP (do inglês *Pretty Good Privacy*). Partindo para o ponto da descentralização, abordaremos também o conceito de infraestrutura de chave pública descentralizada (DPKI do inglês *Decentralized Public Key Infrastructure*), explanando seus componentes.

Mais adiante será feita uma abordagem sobre conceitos relacionados à identidade digital e elementos relacionados, tais como credenciais verificáveis. Em seguida será discutido o funcionamento das tecnologias de livro-razão distribuído com uma breve discussão sobre anonimato e privacidade no escopo das DLTs. Também será mostrada a utilização das DLTs como componentes de infraestruturas de chave pública descentralizadas como também sua utilização com identificadores descentralizados (DID do inglês *Decentralized Identifier*). Por fim, é apresentado um estudo acerca da presença de elementos da criptografia e criptografia de chave pública dentro das DLTs. A ordem dos itens abordados nesta seção foi concebida no intuito de facilitar a leitura, como também tornar o texto mais coeso, levando-se em consideração a dependência de compreendimento entre os tópicos abordados. Objetiva-se com a

leitura desta seção, que os principais conceitos e tecnologias abordadas nesta pesquisa sejam compreendidos, como base para o compreendimento da solução proposta neste trabalho.

2.1 Criptografia

Uma vez que estamos na era da informação digital, a discussão acerca da segurança da informação torna-se mais acentuada. Durante a história da computação e evolução da internet, diversas técnicas foram estudadas a fim de prover segurança aos dados digitais, a exemplo da criptografia. A palavra criptografia tem sua origem do grego, que significa "escrita secreta". Para entender a criptografia, podemos pensar na técnica de encriptar/decriptar mensagens utilizando chaves secretas, com o intuito de tornar tais mensagens seguras e imunes à ataques. O termo encriptar refere-se à transformação de um texto plano em uma mensagem codificada, também chamada de texto cifrado e o termo decriptar remete ao processo inverso, isto é, a transformação de um texto cifrado em um texto plano (Stallings, 2013).

De acordo com Abood e Guirguis (Abood e Guirguis, 2018), a criptografia pode ser utilizada para atingir certos objetivos, dos quais alguns são listados a seguir:

- **Autenticação**: Mecanismo para estabelecer prova de identidade. Garante que a origem da mensagem está corretamente identificada.
- **Confidencialidade**: Princípio que especifica que apenas o emissor e o receptor possam acessar o conteúdo de uma mensagem.
- **Não repúdio**: Garante que emissor e receptor reconheçam a entrega da informação.
- **Integridade**: Mecanismo para garantir que o conteúdo da mensagem está íntegro ao chegar no receptor.
- **Controle de acesso**: Mecanismo para especificar quem pode acessar determinado conteúdo.

Na literatura envolvida no estudo da criptografia, é possível encontrar dois tipos de criptografia. Uma categoria, referida como criptografia de chave secreta ou criptografia simétrica consiste em um modelo de criptografia que utiliza uma mesma chave para a encriptação e

decriptação de uma mensagem, isto é, a mesma chave deve ser conhecida pelo emissor e pelo receptor da mensagem, conforme ilustrado na figura 2.1. A outra categoria é, referida como criptografia de chave pública, ou por vezes citada como criptografia assimétrica (Stallings, 2013).

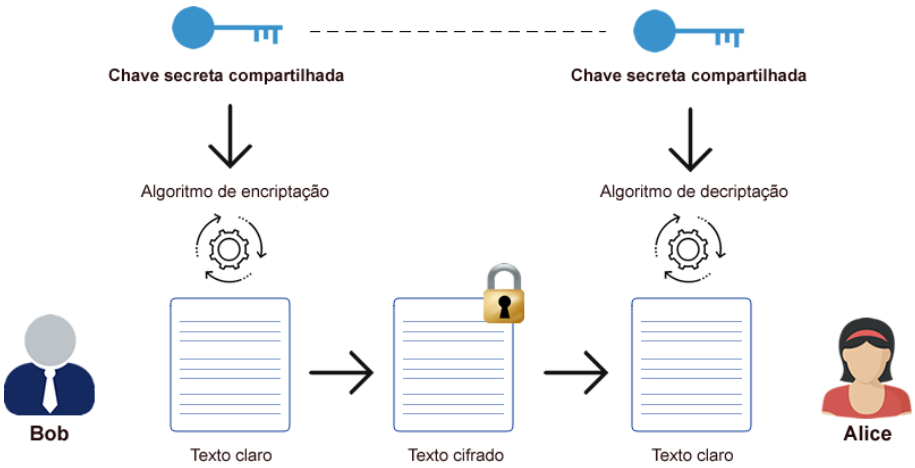

Figura 2.1: Criptografia simétrica - encriptação com chave secreta

Fonte: Adaptado de Stallings (2013)

2.1.1 Criptografia de Chave Pública

Considerada a maior revolução na história da criptografia, a criptografia de chave pública[1], também conhecida como criptografia assimétrica, é um modelo de criptografia que envolve um par de chaves associadas, uma pública, que pode ser amplamente divulgada, e uma privada, que é conhecida apenas pelo seu proprietário (Diffie, 1988). Uma mensagem encriptada com a chave pública só pode ser decriptada com a respectiva chave privada. Em alguns algoritmos, como o RSA (acrônimo de Rivest-Shamir-Adleman)[2], o oposto também é verdadeiro isto é, uma mensagem encriptada com uma chave privada, só pode ser decriptada

[1] Ainda que não haja um consenso com relação à invenção da criptografia de chave pública, a primeira aparição documentada de conceitos relacionados foi em 1970 em um relatório confidencial do *Communications-Electronics Security Group* por James Ellis. A primeira aparição em um documento não confidencial foi em 1974 em uma proposta de projeto por Merkle (Stallings, 2015). O documento pode ser visto em http://merkle.com/1974

[2] Um dos primeiros algoritmos de chave pública. Amplamente utilizado como ténica para a encriptação de chave pública (Stallings, 2015).

com a respectiva chave pública, garantindo assim o princípio da integridade, como também a autenticidade da mensagem, conforme mostrado na figura 2.2 (Stallings, 2015).

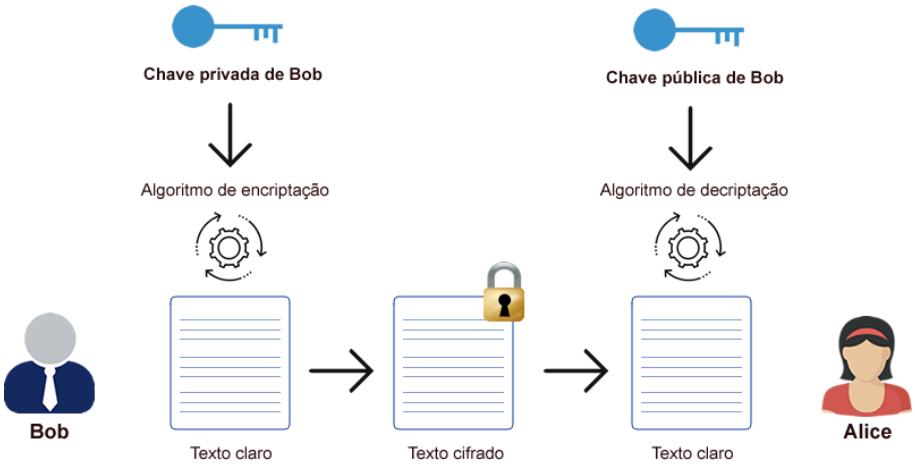

Figura 2.2: Encriptação com chave privada

Fonte: Adaptado de Stallings (2015)

A criptografia de chave pública fornece base tecnológica para uma de suas aplicações mais importantes, a assinatura digital (Stallings, 2015). Diffie trouxe para a discussão que se a criptografia fosse popularmente utilizada, para fins comerciais ou particulares, haveria a necessidade de assinaturas com a mesma legitimidade de assinaturas usadas em papel, para documentos eletrônicos (Diffie, 1988). Uma assinatura digital pode ser entendida como um código associado a uma mensagem ou documento eletrônico que permite de forma única a comprovação da autoria do artefato (ITI, 2018). Para assinar digitalmente um documento, uma técnica comum é gerar uma representação simplificada do conteúdo através da aplicação de uma função de *hash*. O *hash* é uma função que recebe um parâmetro como entrada e após a aplicação de um algoritmo, retorna uma saída em um tamanho pré-definido de bits (Thomsen e Knudsen, 2009). Para verificar a autenticidade de uma mensagem ou documento assinado, é necessário decriptar a mensagem utilizando a chave pública de quem fez a assinatura. A mesma função de *hash* deve ser aplicada no documento original para que o resultado seja comparado e a autenticidade do documento possa ser comprovada (Stallings, 2015). O procedimento de assinatura e verificação é ilustrado na figura 2.3

A assinatura digital é amplamente aplicada em documentos eletrônicos, que podem ser utilizados em serviços públicos ou mesmo transitados entre pessoas físicas. A assinatura digital também é frequentemente utilizada para fins de autenticação, uma vez que a posse da

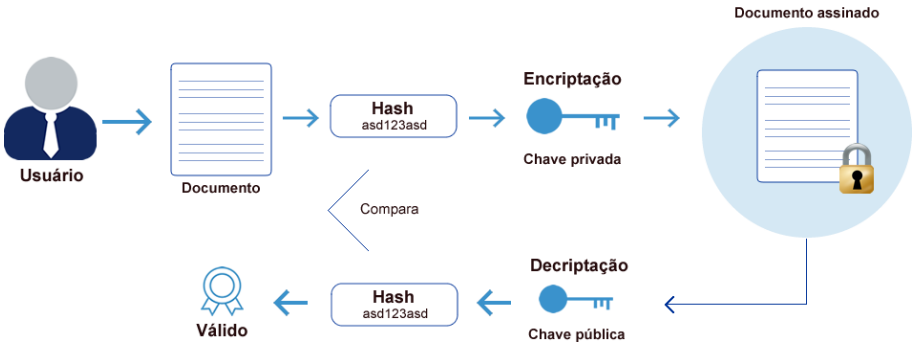

Figura 2.3: Assinatura Digital de Documento Eletrônico

Fonte: Adaptado de Stallings (2015)

chave privada pode ser verificada com a respectiva chave pública, provendo a autenticação em determinado serviço como alternativa à tradicional inserção de dados como login e senha.

2.2 Infraestrutura de Chave Pública

Embora parecidos, os termos criptografia de chave pública e infraestrutura de chave pública possuem significados distintos. Enquanto que o primeiro se refere a um modelo de criptografia que utiliza um par de chaves, o termo infraestrutura de chave pública pode ser entendido como um conjunto de tecnologias (servidores, *softwares* e estações de trabalho), políticas e processos empenhados na administração de certificados e no gerenciamento de pares de chave pública/privada, incluindo a capacidade de emitir, manter e revogar certificados de chave pública (Kuhn et al., 2001).

Em uma visão geral, uma PKI pode ser classificada em duas categorias de acordo com o modelo de confiança adotado para a gestão de chaves. Algumas PKIs são baseadas em terceiros confiáveis como Autoridades Certificadoras para a emissão segura de chaves, enquanto que outras são baseadas no conceito *"Web of Trust"*, onde usuários geram suas próprias chaves e criam uma relação de confiança através da assinatura de suas chaves por outros usuários, a exemplo do PGP (Fromknecht et al., 2014) (Albarqi et al., 2015).

Outra definição encontrada menciona uma infraestrutura de chave pública como um *framework* composto por *hardware*, *software*, políticas e procedimentos para gerenciar chaves e certificados digitais (Choudhury et al., 2002).

Percebe-se que em ambas as definições, o termo certificado está presente como um ele-

mento de uma PKI, sendo por vezes mencionado certificado de chave pública ou mesmo certificado digital. Neste trabalho será utilizado o termo certificado digital. Um certificado digital é um documento eletrônico usado para identificar um indivíduo, uma empresa ou uma entidade e associá-lo à uma chave pública (Stallings, 2013). Em uma PKI baseada em Autoridade Certificadora, certificados digitais são emitidos e assinados digitalmente por ACs, que realizam os procedimentos necessários para a confirmação de autenticidade de acordo com a localização e tipo de certificado, podendo ainda delegar responsabilidades nos processos de verificação de documentação a Autoridades de Registro (Stallings, 2015) (Boeyen et al., 2008). Cada autoridade certificadora possui o seu próprio certificado assinado digitalmente por uma outra autoridade certificadora em uma hierarquia superior, formando uma espécie de cadeia de entidades, até a autoridade raiz, como mostra a Figura 2.4 (Lackey, 2012).

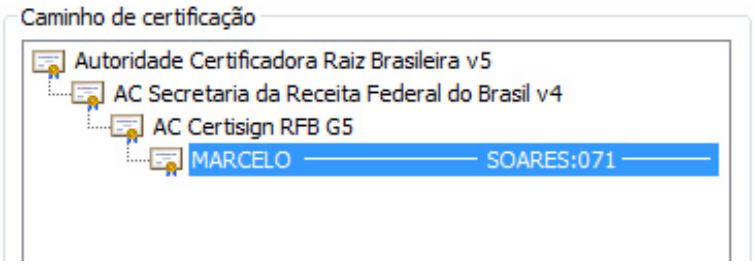

Figura 2.4: Caminho de certificação de uma AC Raiz da ICP-Brasil

Fonte: Printscreen de visualizador de certificado do Windows 10

Desde a introdução da criptografia de chave pública, o gerenciamento das chaves passou por um considerável amadurecimento. Uma primeira concepção originou um diretório global para o depósito de chaves, seguida por uma padronização de um diretório global criada pelo *International Telecommunication Union* (ITU), chamada de X.500. O padrão X.509 foi proposto para fins de autenticação, uma vez que define um formato de certificado e liga a identidade do detentor de uma chave à sua respectiva chave pública (Albarqi et al., 2015). Desde 1995, um grupo de trabalho[3] denominado PKIX trabalha na infraestrutura de chave pública baseada na utilização de certificados X.509 (Boeyen et al., 2008). A **Infraestrutura de Chaves Públicas Brasileira** (ICP-Brasil) é uma implementação de uma PKI que utiliza o padrão de certificado X.509 na versão 3 de acordo com a padronização internacional para

[3]https://datatracker.ietf.org/wg/pkix/charter/

uso na internet descrita na RFC 5280 (acrônimo de *Request for Comments*[4]) (Boeyen et al., 2008) (ITI, 2018). A estrutura de um certificado, em uma visão mais ampla, pode ser dividida em 3 seções principais. Uma seção descrita como *tbsCertificate* (*to be signed*), que corresponde ao conjunto de campos do certificado que são assinados com a chave privada da Autoridade Certificadora. Uma outra seção com o campo *signatureAlgorithm* que corresponde ao identificador do algoritmo criptográfico usado pela AC para assinar o certificado. Por fim, o certificado também contém o valor da assinatura digital calculada com base no *tbsCertificate* e assinada pela Autoridade Certificadora com a sua respectiva chave privada (Boeyen et al., 2008).

Alguns atributos importantes contidos em um certificado, de acordo com a RFC 5280 (Boeyen et al., 2008) são:

- **Versão**: Indica a versão do certificado. Caso haja extensões deverá ser a versão 3.
- **Serial Number**: Um inteiro positivo emitido pela AC. Identifica unicamente o certificado apenas no escopo da AC.
- **Validade**: Data de início e fim da validade do certificado.
- **Subject**: *Distinguished names*[5] da entidade associada à chave pública armazenada no campo de chave pública.
- **Issuer**: *Distinguished names* da entidade que assinou o certificado.

No Brasil, assinaturas digitais em documentos eletrônicos assinados por chaves emitidas por uma AC da cadeia ICP-Brasil são juridicamente reconhecidas (Brasil, 2001). Para a validação de uma mensagem/documento assinado, é necessária a posse da mensagem/documento em sua versão não original (não assinada), a assinatura digital e, neste caso, o certificado digital que contém a respectiva chave pública e a relaciona com uma pessoa/entidade. Há diversas abordagens para o agrupamento de tais artefatos de modo a garantir a interoperabilidade em diferentes sistemas. A W3C, principal entidade de padronização da *World Wide*

[4]RFCs são documentos técnicos com padronizações desenvolvidos por indivíduos e organizações.

[5]*Distinguished names* são estruturas compostas por atributos. Alguns atributos que podem ser encontrados dependendo da implementação da infraestrutura são: *country, organization, organizational-unit, distinguished name qualifier, state or province name, common name*.

Web, disponibiliza um documento técnico que descreve uma sintaxe e padronização para a assinatura digital em documentos XML, chamada de XML-DSig (Bartel et al., 2015). Na União Europeia, a assinatura digital é regulamentada pelo seu parlamento europeu, sendo mencionada como assinatura eletrônica avançada, ou AdES, do inglês *advanced electronic signature* (Parlamento Europeu, 2014). A *European Telecommunications Standards Institute* (ETSI) define perfis de assinatura para formatos de documentos comuns, a exemplo do PAdES para PDF[6], como também o XAdES para XML[7], uma extensão do padrão XML-DSig, e por fim o CAdES[8] para qualquer tipo de documento. Tais perfis são utilizados como base para políticas de assinatura da ICP-Brasil.

2.2.1 Infraestrutura de Chave Pública Descentralizada

Apesar da nossa sociedade estar baseada na centralização, com hierarquias institucionais para governar as nossas atividades socioeconômicas, várias tecnologias disruptivas apontam para uma tendência de descentralização, inclusive no âmbito das identidades digitais (Aste et al., 2017). Não está no escopo deste trabalho a discussão acerca de deficiências da PKI que possam justificar o surgimento de uma infraestrutura descentralizada (DPKI do inglês *Decentralized Public Key Infrastructure*), uma vez que a propositura descrita neste trabalho considera a coexistência de ambas infraestruturas (centralizadas e descentralizadas).

Ao contrário das PKIs baseadas em terceiros confiáveis, como autoridades certificadoras, a DPKI parte da premissa de que nenhum terceiro pode comprometer a integridade e a segurança da rede como um todo. Ao contrário das PKIs que detém a sua confiança baseadas em terceiros confiáveis, como autoridades certificadoras, na DPKI a confiança é descentralizada (Durand et al., 2017). Tal característica pode ser obtida através do uso de tecnologias descentralizadas, a exemplo de DLTs, que permitem que entidades distribuídas cheguem a um consenso sobre o estado de um banco de dados compartilhado. A DPKI está apoiada em tecnologias de *datastore* descentralizados de chave valor, como a *blockchain* ou mesmo outras tecnologias que possuam características de segurança semelhantes (Christopher Allen, 2015).

[6] *PDF Advanced Electronic Signatures*
[7] *XML Advanced Electronic Signatures*
[8] *CMS Advanced Electronic Signatures*

Enquanto que em um modelo centralizado, sistemas de gestão de identidade (IdM do inglês *Identity Management*) convencionais são controladas por uma autoridade central, sistemas de gestão de identidade descentralizada permitem que qualquer pessoa possa criar e gerenciar seus próprios identificadores. Em uma infraestrutura de chave pública descentralizada, as identidades pertencem às entidades que elas representam (Christopher Allen, 2015). Redes descentralizadas, tais como redes formadas por DLTs vêm sendo utilizadas como registro de dados verificável[9] (*Verifiable Data Registry*), um componente responsável por mediar a criação e verificação de identificadores e chaves, dentre outros atributos. No contexto de identidades descentralizadas, identificadores são chamados de **DID** (acrônimo de *decentralized identifier*) (Sporny et al., 2019).

2.3 Identidade Digital

Ainda que este trabalho não trate exclusivamente de identidade digital, há uma associação de tal conceito com o tema da pesquisa. Portanto, é importante uma breve revisão sobre conceitos relacionados à identificação, sobretudo em um ambiente digital. O termo identidade pode ser estudado no campo da filosofia, psicologia e sociologia. Na filosofia, a discussão acerca da identidade pessoal está relacionada à seguinte questão: "Quem sou eu?". Basicamente, o termo identidade pode descrever um conjunto de propriedades que garantem uma unicidade, ou seja, tornam um indivíduo diferente de outro (Olson, 2019). Com a adição do termo "digital", o termo identidade digital é comumente encontrado no campo da ciência da computação. Em uma visão geral, uma identidade digital pode ser entendida como um conjunto de revindicações feitas por um sujeito/entidade digital sobre ele próprio ou sobre outro sujeito/entidade. Tais reivindicações, correspondem portanto, a atributos cujo o sujeito/entidade reivindica (Cameron, 2005).

Um sujeito digital pode ser entendido como algo com quem possamos tratar e está sendo descrito em um mundo digital, no entanto, neste trabalho, será utilizado o termo entidade para referenciar algo do mundo real, como um sujeito digital, uma vez que a própria especificação de certificados PKI para a internet menciona que uma entidade final é o sujeito de um

[9]Uma lista de tecnologias distribuídas utilizadas como registros pode ser acessada em https://w3c-ccg.github.io/did-method-registry/

certificado. Portanto, uma identidade digital pode ser entendida como um conjunto de atributos associados a uma entidade em um ambiente digital (Boeyen et al., 2008) (Sporny et al., 2019) (Cameron, 2005). Uma identidade digital pode ser construída com a participação das entidades interessadas, e possuem natureza de multiplicidade, isto é, entidades podem ter diferentes personas de acordo com o contexto em que estão interagindo (Clauundefined e Köhntopp, 2001).

Em nosso ambiente físico, identidades podem ser provadas por meio de credenciais, a exemplo de documento de identidade, passaporte, certidão de óbito etc. Em um ambiente digital, credenciais também são adotadas para a comprovação de identidade em determinado contexto, no entanto, com a adição de tecnologias como assinatura digital, que tornam as credenciais mais confiáveis. Um exemplo disso é a padronização de Credenciais Verificáveis que vem sendo desenvolvida pela W3C. De acordo com a W3C (Longley et al., 2019), uma credencial possui as características listadas a seguir:

- Informações relacionadas ao sujeito (nome, foto, número de identificação);
- Informações relacionadas à autoridade emissora;
- Informações relacionadas ao tipo de credencial;
- Informações sobre atributos que o emissor declara sobre o sujeito (nacionalidade, data de nascimento);
- Evidências relacionadas a como a credencial foi derivada;
- Informações sobre restrições da credencial (data de validade).

Em nosso meio físico, Joyce e Gupta citam 4 maneiras de verificar a identidade de um indivíduo: através de objetos (chaves, credenciais, etc.), conhecimento sobre algum dado (senha, PIN, etc.), ações (assinatura) ou padrões fisiológicos (impressão digital, voz, etc.) (Joyce e Gupta, 1990). No ambiente digital, conforme visto anteriormente, o análogo a uma assinatura em punho é uma assinatura digital. Assim como ocorre no mundo físico, em um ambiente digital, uma assinatura digital também pode ser utilizada em um processo de autenticação para comprovar uma identidade. No Brasil, diversas plataformas governamentais[10]

[10]https://cav.receita.fazenda.gov.br/autenticacao/login

permitem a autenticação através de assinaturas digitais associado a certificados emitidos pela ICP Brasil.

2.4 Tecnologias de Livro-Razão Distribuído

O estudo das tecnologias livro-razão distribuído se inicia com a primeira concepção publicada em 2018 no artigo ”Bitcoin: A Peer-to-Peer Electronic Cash System”, por uma figura anônima que assinava como Satoshi Nakamoto. Nakamoto levantou a necessidade de um sistema de pagamento baseado em prova criptográfica através de uma rede *peer-to-peer* ao invés de uma entidade intermediadora (Nakamoto, 2008a). A tecnologia por trás dessa abordagem descrita por Nakamoto foi denominada *blockchain*. Posteriormente, outras tecnologias foram implementadas, surgindo o termo *"Distributed Ledger Technology"* (DLT) (Natarajan et al., 2017). A atualidade do tema em questão reflete na dificuldade de clareza na terminologia. Muitas vezes os termos *"Distributed Ledger Technology"* e *"Blockchain"* se conflitam, como também é possível encontrar diferentes abordagens na implementação de DLTs (Deshpande et al., 2017). Embora os termos DLT e *blockchain* sejam comumente confundidos, é importante o entendimento de que a *blockchain* é um modelo de implementação de uma DLT que usa um determinado tipo de estrutura de dados para o armazenamento de transações (Taylor et al., 2016) (Natarajan et al., 2017).

A palavra *ledger* pode ser entendida como um livro ou banco de dados no qual as transações contábeis de dupla entrada são armazenadas e resumidas. Em um ambiente informatizado, podemos entender um *ledger* como sendo uma estrutura de dados contendo as mesmas informações que seriam encontradas em uma página de contabilidade de um sistema manual (Bragg, 2017). Tais informações referem-se a registros de transações não só de natureza monetária (Deshpande et al., 2017). O termo dupla entrada, em um contexto financeiro, quer dizer que cada transação de uma empresa resultará em um valor registrado em pelo menos duas contas do sistema contábil (Averkamp, 2017). A tradução mais presente em dicionários para a palavra *ledger* na língua portuguesa é: livro-razão. Um livro-razão, na contabilidade, é um livro usado para controlar separadamente o movimento de todas as contas de uma empresa. Um livro razão deve conter dados referentes às transações efetuadas.

No campo da ciência da computação, é possível encontrar o termo "distribuído"associado

à diversas áreas, tais como: Banco de Dados Distribuído, Computação Distribuída, Sistemas Distribuídos, Engenharia de Software Distribuída, etc. Sistemas distribuídos são descritos por Tanenbaum e Van Steen (2007) como uma coleção de computadores independentes que aparecem para o usuário como um único sistema (Tanenbaum e Steen, 2006).

De acordo com Bech (2017), um livro-razão distribuído refere-se aos protocolos e infraestrutura de suporte que permitem que computadores em diferentes locais proponham e validem transações e atualizem registros de maneira sincronizada em uma rede (Bech e Garratt, 2017). O termo compartilhado é frequentemente encontrado ao invés de distribuído, formando o termo *shared ledger* (Taylor et al., 2016) (Natarajan et al., 2017).

Uma vez que transações podem ser feitas entre os participantes da rede sem a intermediação de uma autoridade central a descentralização se faz um atributo presente no escopo das DLTs. Desde a sua primeira concepção, Nakamoto já descrevia a tecnologia que viria a ser implementada e posteriormente denominada DLT, como sendo um modelo de rede descentralizada *peer-to-peer* (Nakamoto, 2008a). Uma rede *peer-to-peer* distribui informações entre os nós ao invés de concentrá-las em um único servidor. O termo *peers* (pares) refere-se aos computadores da rede que podem atuar como clientes ou como servidores (Parameswaran et al., 2001) (Tanenbaum et al., 2011). Na Engenharia de Software, *peer-to-peer* pode ser categorizado como um padrão arquitetural para sistemas distribuídos, enquanto que sistemas *peer-to-peer* são sistemas descentralizados onde o processo computacional pode ser realizado por qualquer nó da rede (Sommerville, 2011).

A ideia inicial de uma rede *peer-to-peer* era o compartilhamento de arquivos entre os participantes, formando uma rede de distribuição de conteúdo (Parameswaran et al., 2001) (Tanenbaum et al., 2011). Cada sistema *peer-to-peer* implementa os seus protocolos para a comunicação e transferência de arquivos entre os nós (Sommerville, 2011). Tal como as DLTs foram exploradas fora do contexto financeiro, o paradigma *peer-to-peer* se aprimorou para ser aplicado às DLTs, resultando em redes *peer-to-peer* compartilhando *ledgers* ao invés de músicas ou vídeos.

Em uma DLT, os *ledgers* não são apenas compartilhados, mas replicados em cada nó da rede. Em sistemas distribuídos, dados são replicados para garantir confiabilidade no sistema ou mesmo para prover uma melhor performance (Tanenbaum e Steen, 2006). A redundância provida pela replicação dos dados, provê um modelo descentralizado mais resistente a falhas

e, consequentemente, mais seguro (Parameswaran et al., 2001).

O grande alavancamento da popularidade das DLTs está diretamente relacionado à sua utilização pelas criptomoedas, que propõem uma abordagem diferente ao modelo centralizado até então utilizado na nossa sociedade. De acorcdo com Benkler (2006), descentralização descreve as condições sob as quais as ações de muitos agentes são coerentes e eficazes apesar do fato de não dependerem da redução do número de pessoas cuja vontade conta para direcionar uma ação eficaz. Ainda conforme Benkler, estamos vivendo em uma nova etapa da economia da informação, que denomina economia da informação em rede, onde as ações individuais ou ações cooperativas coordenadas por mecanismos distribuídos e independentes desempenham um papel maior do que em uma economia de informação industrial (Benkler, 2006).

2.4.1 DLTs Permissionadas e Não-permissionadas / Públicas e Privadas

Uma DLT pode ser classificada como permissionadas ou não-permissionadas (do inglês *permissioned* ou *unpermissioned/permissionless*), de acordo com o nível de permissão dos participantes na rede. É possível encontrar diferentes traduções para os termos citados. Neste trabalho serão usados os termos permissionadas e não-permissionadas, embora também seja possível encontrar os termos permissionárias e não-permissionárias na literatura. Uma DLT não-permissionada é aberta para que qualquer participante possa inserir dados no livro-razão. Neste tipo de DLT, não há uma autoridade central que controle quem possa, ou não, se juntar à rede, assim como o procedimento de garantia de integridade do conteúdo do livro-razão pode ser feita por qualquer participante interessado. O exemplo mais popupar de uma DLT não-permissionada é a *blockchain* da Bitcoin (Natarajan et al., 2017) (Dhar e Bose, 2016). O processo de garantia de integridade da rede através do algoritmo usado para o consenso na alteração do legder, na *blockchain* chamado de mineração, é feito por nós aleatórios e anônimos, sem qualquer intervenção da rede na escolha (Taylor et al., 2016) (Swanson, 2015).

Em uma DLT classificada como permissionada, a rede é restrita a uma quantidade de participantes que são conhecidos uns pelos outros, e pré-selecionados por uma entidade administradora do sistema, quem controla o acesso e as regras da rede (Natarajan et al., 2017). Neste caso, a integredidade do *legder* é garantida por um processo envolvendo participantes confiáveis definidos pelo proprietário do sistema. As DLTs permissionadas podem perten-

cer a organizações ou comunidades que compartilham interesses em comum (Taylor et al., 2016). Conforme mencionado anteriormente, há uma certa dificuldade na terminologia deste campo de pesquisa. As DLTs permissionadas e não-permissionadas são comumente chamadas de privadas e públicas respectivamente (Dhar e Bose, 2016) (Narayanan e Clark, 2017) (Davidson et al., 2016). No entanto, alguns autores ainda classificam as DLTs como públicas ou privadas como sendo um atributo a mais, de acordo com a restrição de acesso à leitura e escrita no *ledger* (Brennan e Lunn, 2016) (Natarajan et al., 2017).

2.4.2 Consenso Distribuído

As DLTs, por serem distribuídas, carregam consigo desafios inerentes aos sistemas distribuídos. O problema de como chegar a um consenso sobre os dados produzidos pelos nós remotos da rede é um dos problemas fundamentais da computação distribuída (Fischer et al., 1985). Tanenbaum e Van Steen (2006) apontam a problemática da consistência dos dados replicados em sistemas distribuídos (Tanenbaum e Steen, 2006).

Outro problema foi descrito por Lamport (1978) e está relacionado a um consenso sobre a ordem cronológica dos eventos em um sistema distribuído (Lamport, 1978). Os algoritmos utilizados pelas DLTs, além de resolverem os problemas típicos de sistemas distribuídos, devem atender a um requisito específico de um *ledger*, descrito no campo da contabilidade: a imutabilidade. Um livro-razão não pode ter um registro apagado. Correções devem ser feitas apenas inserindo novos registros ao livro (Helland, 2015). Quando se tratando do uso de DLTs em criptomoedas, ainda é preciso considerar o problema do duplo gasto. O termo duplo gasto foi usado pela primeira vez em 1992 por David Chaum (Chaum, 1992).

Cada DLT implementa o seu algorítmo de consenso para lidar com os problemas citados. Proof of work, proof of skate, proof of burn, hashgraph e consenso baseado em peso acumulado são exemplos de métodos utilizados por DLTs para a garantia do consenso. Os termos citados são alternadamente mencionados na literatura como algoritmos de consenso, protocolos, ou mesmo métodos. Para este trabalho, os termos serão mencionados como métodos, já que um método pode ser entendido como um procedimento para chegar a determinado fim (Prodanov e Freitas, 2013).

Embora a preocupação com o gasto de energia para a execução dos algoritmos de consenso venha acarretando no surgimento de novos métodos, o proof of work ainda é o método

utilizado atualmente pelas maiores DLTs públicas, como a Bitcoin e Ethereum. Na prova de trabalho (proof of work), os nós participantes dedicam o seu poder de processamento para incrementar um *nonce* que é concatenado ao conteúdo do bloco com o objetivo de encontrar um resultado de hash que comece com uma determinada quantidade de bits 0, definida pela rede. O participante da rede que conseguir encontrar primeiro o valor do *nonce* correto para resolver o desafio, irá adicionar o bloco à cadeia e receber uma recompensa pelo trabalho.

2.4.3 Principais Tecnologias

Conforme visto na seção anterior, diferentes abordagens e implementações de livros distribuídos vêm sendo propostas e implementadas. Nesta seção, serão apresentados os principais modelos de DLTs, como também, as suas principais características de funcionamento.

Blockchain

A sua implementação é baseada na concepção de cadeia de blocos. Descrita pelo pseudônimo Satoshi Nakamoto, publicada no ano de 2008 em uma lista de e-mail (Nakamoto, 2008b) do que viria a ser a primeira tecnologia livro-razão distribuída, e posteriormente seria a tecnologia base da primeira criptomoeda descentralizada, a Bitcoin (Nakamoto, 2008a).

Em uma *blockchain*, transações são agrupadas em blocos. Cada bloco é "carimbado"com um *hash* que é gerado com base no conteúdo do bloco. O processo de geração do *hash* do bloco se dá através do método de consenso distribuído utilizado pela DLT. No contexto de criptomoedas, novas moedas são colocadas em circulação através deste processo. Ocorre que, uma vez que o nó vencedor do desafio matemático, no proof of work, por exemplo, recebe uma recompensa em criptomoedas que podem ser criadas nesse momento, em concordância com a comunidade, o termo "minerar criptomoeda"é comumente utilizado para esse processo análogo à mineração. O conteúdo de cada bloco contém o hash do bloco anterior, com exceção do primeiro bloco (*genesis block*), formando uma espécie de cadeia, como mostra a Figura 2.5. Isso implica que, a alteração de um bloco, acarretará na necessidade de recalcular todos os valores *hash* dos blocos seguintes (Nakamoto, 2008a).

O grande esforço computacional necessário para recalcular a cadeia de blocos, aliado à distribuição de cópias do *ledger* na rede, garantem a imutabilidade da *blockchain*. O alto

processamento computacional para realizar uma prova de trabalho custa energia, o que torna inviável uma tentativa de quebrar a rede (Nakamoto, 2008a).

Outras customizações da *blockchain* foram implementadas, extendendo ainda mais a sua possibilidade de utilização. A Ethereum é uma *blockchain* que fornece uma linguagem de programação que pode ser usada para a construção de aplicações descentralizadas, que por sua vez podem controlar ativos digitais através de contratos inteligentes (Buterin, 2014).

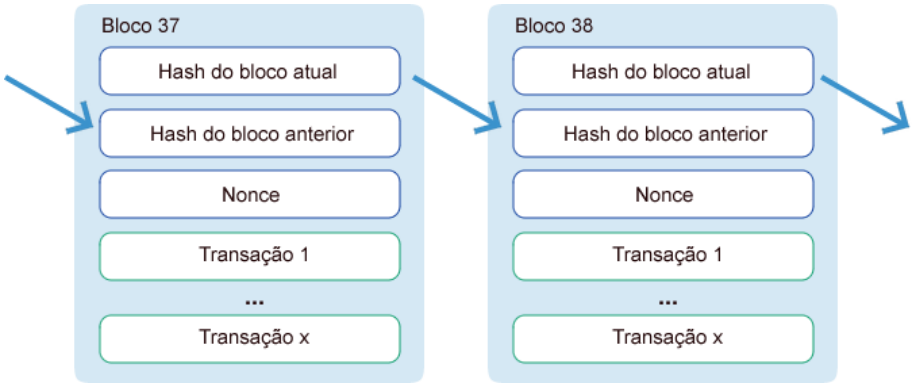

Figura 2.5: Encadeamento de blocos

Fonte: Adaptado de Nakamoto (2008a).

Tangle

Assim como o surgimento da *blockchain* ocorreu com a sua utilização na Bitcoin, o Tangle está associado ao desenvolvimento da IOTA, uma criptomoeda desenvolvida para a indústria da *Internet of Things* (IOT)[11]. A ideia do Tangle é prover uma plataforma onde os seus participantes não precisem pagar taxas para transacionarem no *ledger* (Popov, 2018).

Criada em 2015, o Tangle surge como uma alternativa à *blockchain*. No modelo mais tradicional, há dois tipos de participantes: os que fazem as transações e os que aprovam as transações. Diferentemente da *blockchain*, onde as transações são armazenadas em blocos, o funcionamento do Tangle é baseado em um grafo acíclico dirigido (DAG). Para um usuário lançar uma transação na rede, ele precisa contribuir para o funcionamento e segurança da rede aprovando outras duas transações anteriores (Popov, 2018). O participante precisa resolver um desafio de prova de trabalho para validar a transação. No entanto, não há uma

[11] https://www.iota.org/

competição para o desafio, já que não há recompensa para o vencedor. A prova de trabalho é realizada para a proteção de spam na rede.

Hashgraph

As DLTs categorizadas como Hashgraph, são baseadas em um modelo de consenso onde as transações são espalhadas pela rede através de um protocolo conhecido como *"gossip protocol"*, ou protocolo de fofocas. Na prática, um usuário escolhe outro usuário da rede aleatoriamente e envia todas as informações que ele conhece até o momento. Este procedimento é chamado de evento. O usuário que recebeu as informações repetirá o procedimento assim como todos os outros usuários da rede. Um evento contém os valores *hash* de outros dois eventos: o último evento do próprio participante e o último evento do participante que está enviando as informações. É possível ver um exemplo através da Figura 2.6, onde o evento em vermelho (Alice) recebe um hash do seu último evento, representado pelo círculo azul escuro, e também recebe um hash de Bob representado pelo círculo azul claro. No momento da criação de um evento, o participante poderá assiná-lo digitalmente, assim como armazenar novas transações (Baird, 2016).

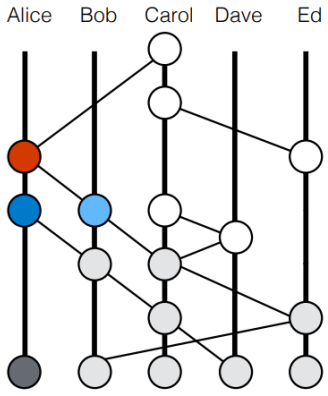

Figura 2.6: Hashgraph

Fonte: Baird (2016)

2.5 Anonimato e Privacidade em DLTs

Em geral, os termos privacidade e anonimato estão automaticamente incluídos dentro do escopo da utilização de DLTs. Normalmente, o modelo de funcionamento das criptomoedas

tenta prover o anonimato dos atores envolvidos, e por consequência, a privacidade em suas ações. Ainda que Nakamoto não tenha mencionado a motivação para a necessidade de privacidade em uma DLT, é possível encontrar na literatura colocações que sugerem a importância do anonimato. Tennant menciona que, sem a privacidade, informações confidenciais poderiam ser vazadas, assim como transações para IOT poderiam ser monitoradas para o planejamento de roubos (Tennant, 2017). Conforme citado por Nakamoto, em um modelo bancário tradicional, um certo nível de privacidade é alcançada mantendo a informação entre as partes envolvidas em uma transação e uma entidade terceira confiável (Nakamoto, 2008a). Para um cenário onde as transações devem ser públicas, a privacidade pode ser alcançada mantendo anônima a propriedade de uma chave pública. Nesse contexto, o termo pseudônimo digital é frequentemente usado para designar uma chave pública associada a uma chave privada de propriedade desconhecida (Narayanan e Clark, 2017) (Clauundefined e Köhntopp, 2001).

Mas será que o modelo de utilização de criptografia de chave pública para que os usuários possam transacionar em uma DLT realmente provê um anonimato completo?

Desde a primeira concepção de DLT, já era sugerida a utilização de vários pares de chaves em transações distintas para evitar a quebra do anonimato (Nakamoto, 2008a). É comum encontrar trabalhos que investigam o anonimato e privacidade em DLTs. Reid e Harrigan concluem que, por meio de ferramentas apropriadas, é possível associar muitas chaves públicas umas às outras e monitorar a atividade de usuários da rede (Reid e Harrigan, 2011). Meiklejohn et al. apresenta uma heurística de *clustering* baseada em endereços de mudança, que permite agrupar endereços pertencentes ao mesmo usuário (Meiklejohn et al., 2013). Ainda no ambiente de criptomoedas, o detentor de ativos digitais está sujeito a trocar os seus valores por moedas reais, como o dólar ou euro. Para tal, o usuário deverá interagir com um sistema de câmbio, que poderá aplicar as suas regras de conhecimento do cliente (Taylor et al., 2016). Uma simples compra de produtos em um e-commerce já seria o suficiente para que o anonimato de quem está por trás do endereço da carteira seja quebrado e sua identidade seja revelada. Tais estudos nos levam a concluir que o anonimato em DLTs é, na verdade, um pseudo-anonimato. A própria comunidade Bitcoin alerta em sua página que a criptomoeda não é anônima, já que a privacidade é mantida apenas no escopo da rede descentralizada (Bitcoin, 2018a).

Muitos estudos e pesquisas direcionam os seus esforços na busca por métodos para prover

o anonimato pleno, e, consequente, a privacidade plena. É importante salientar que, em um contexto financeiro, o anonimato pleno dos atores envolvidos em transações pode facilitar o uso para atividades criminosas (Möser e Böhme, 2017). Em 2013, o FBI fechou um website que funcionava como um comércio de drogas ilegais e já havia movimentado cerca de 1,3 bilhão de dólares com pagamentos feitos em Bitcoin (Pagliery, 2013). Mais recentemente, em 2017, um ataque em escala mundial de um ransomware conhecido como *WannaCry* sequestrou dados de mais de 45 mil computadores em 74 países, de acordo com a empresa russa de segurança Kaspersky (Perekalin, 2017). Mais uma vez, a criptomoeda Bitcoin esteve associada aos ataques por ser o meio indicado pelos invasores para o pagamento de um "resgate" dos dados sequestrados. Por sua associação constante à ataques e a sites da rede dark/deep web, a Bitcoin é alvo de desconfiança por parte da sociedade (Taylor et al., 2016).

Claramente há inúmeros desafios associados com a garantia de anonimato e privacidade, sobretudo nas DLTs públicas. Mas, e quando o anonimato é explicitamente dispensado?

Uma das principais criptomoedas em operação, a Bitcoin, disponibiliza em seu site o endereço da carteira digital de propriedade da comunidade mantenedora da criptomoeda (Figura 2.7). Também é comum encontrar páginas de comércio eletrônico exibindo os seus endereços de carteiras com representação em QR Code. Uma simples divulgação de endereço com fins de arrecadar fundos nos leva a refletir sobre a necessidade de identificação das entidades por trás dos endereços de carteiras digitais. Nota-se que, mesmo em um contexto financeiro envolvendo uma criptomoeda, não há preocupação com o anonimato de uma das partes envolvidas. Uma invasão ao sistema gerenciador de conteúdo do website oficial da Bitcoin com a alteração do endereço de carteira exibido resultaria em milhares de pessoas enviando fundos, sem conhecimento, para um terceiro malicioso.

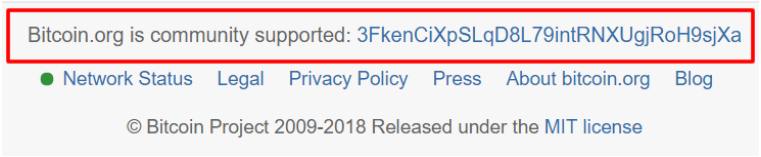

Figura 2.7: Endereço de carteira

Fonte: Printscreen do site bitcoin.org

Também é possível encontrar ocorrências de programas maliciosos que atuam de forma a manipular a área de transferência do sistema operacional, com o intuito de alterar o endereço

de destino de uma transação que foi copiado pelo usuário do sistema. Como os softwares de carteira[12] não apresentam qualquer informação sobre o proprietário do endereço de destino, os fundos são transferidos erradamente para um outro receptor, sem que o remetente perceba (Rashid, 2014).

Neste sentido, tanto a garantia do anonimato de quem deseja quanto a legitimidade da declaração de posse de um endereço de carteira digital são relevantes e desejáveis no contexto de DLTs. Este trabalho foca no segundo caso.

2.6 Criptografia em DLTs

Ao detalhar o funcionamento de uma DLT, é possível perceber que diversos elementos da criptografia se fazem presentes. Funções de dispersão criptográfica ou funções *hash* são amplamente utilizadas para a representação de identificadores de transações e blocos. Outro elemento da criptografia presente em DLTs é a árvore de Merkle, que corresponde a um tipo de estrutura de dados que utiliza uma informação resumida para garantir a integridade de um conjunto de dados de uma árvore binária. A técnica consiste basicamente em concatenar valores *hash* de folhas vizinhas recursivamente até que se chegue ao *hash* de uma folha raíz, chamado de *hash* raiz (Merkle, 1982). Na Bitcoin, a árvore de Merkle é utilizada para gerar uma representação resumida sobre todos os valores *hash* de transações armazenadas em um bloco. O valor do *hash* raiz é então armazenado na área de cabeçalho de um bloco e cada participante da rede pode reconstruir a sua árvore para a verificação da árvore e, consequentemente, validação das transações contidas no bloco (Nakamoto, 2008a). Em caso de alteração em alguma transação da árvore, outro *hash* raiz seria gerado, e o bloco poderia ser invalidado. Na Ethereum, o cabeçalho de um bloco contém 3 árvores de Merkle que representam 3 diferentes conjuntos de dados: *state root*, *transaction root* e *receipt root* que representam dados de estado das contas, dados das transações e dados de recibos de transações respectivamente (Wood, 2017).

O processo de autenticação na rede para a escrita no *ledger* se dá através de outro método criptográfico: a assinatura digital. A posse de uma chave privada permite ao participante

[12]Softwares clientes responsáveis por gerenciar o acesso a endereços e auxiliar na realização de transações em DLTs.

transacionar em uma DLT ou mesmo participar do processo de consenso distribuído (Nakamoto, 2008a). Na Bitcoin, a forma mais comum de enviar uma transação e escrevê-la na *blockchain* é a *P2PKH* (acrônimo de *Pay To Public Key Hash*). Neste método de criação de transações, um emissor envia em um campo de saída da transação chamado *Pubkey Script* uma instrução que deverá ser atendida pelo receptor, representado por um *hash* de chave pública, para que este possa reivindicar o montante no momento de encadear uma nova transação. No momento em que o receptor deseja transacionar o montante recebido e se tornar um emissor de uma nova transação, o mesmo deverá inserir uma entrada na transação chamada *Signature Script*, que contém uma assinatura digital realizada com a sua chave privada e a respectiva chave pública cujo o seu *hash* é o mesmo inserido no *Pubkey Script*. Concatenando o *Pubkey Script* e o *Signature Script*, cada minerador pode executar o procedimento para a validação da transação antes de tentar inseri-la em um bloco (Bitcoin, 2018b). Em geral, o mecanismo para a realização de transações utilizando de criptografia de chave pública e assinatura digital sobre uma estrutura de dados está presente nas DLTs.

2.6.1 Relação entre Endereço e Chave Pública

Conforme mencionado, os atores de sistemas baseados em DLTs são representados por um conjunto de caracteres popularmente referido como endereço de carteira. Também é comum encontrar o termo "conta"fazendo referência a um par de chaves de um determinado usuário. Um endereço é, geralmente, derivado de uma representação mais curta (*hash*) da chave pública. Na Ethereum, o endereço de carteira é formado pelos 160 bits mais a direita do resultado de uma função *hash* Keccak-256 da chave pública (Wood, 2017). Na implementação de DLT utilizada pela Bitcoin, o endereço possui tamanho de 160 bits e é formado a partir do uso combinado de diferentes funções de *hash* (RIPEMD-160 e SHA-256) (Bitcoin-Wiki, 2018). Especificamente na Bitcoin, a formação do endereço de carteira ainda contempla as seguintes etapas:

- Utilização de um byte para indicar se o endereço corresponde à rede principal ou a rede de testes (Bitcoin-Wiki, 2018);
- Codificação em Base58 para remover caracteres que possam parecer iguais em algumas fontes (0OIl por exemplo) (Bitcoin-Wiki, 2017).

Retomando ao contexto de DPKI, endereços de carteiras de DLTs públicas são utilizados como parte na formação de DIDs, em caso da DLT ser utilizada como *Registry*. Por exemplo, o identificador descentralizado `did:ethr:0xE6Fe788d8ca214A080b0f6aC7F48480b2AEfa9a6` possui um endereço como sendo o identificador específico do método **eth**, que faz referência à *blockchain* da Ethereum como *Registry*. Aplicações denominadas *resolvers*[13] precisam implementar a especificação que um *Registry* para que consiga gerenciar chaves.

2.7 Considerações Finais

Esta seção tratou de explanar conceitos e tecnologias envolvidas no desenvolvimento deste trabalho. Percebeu-se, durante a pesquisa bibliográfica realizada para a escrita desta seção, que embora tenham sido concebidos em momentos distintos, as DLTs e os elementos de criptografia estão fortemente relacionados, uma vez que as DLTs utilizam de técnicas criptográficas para o seu funcionamento. Pares de chaves criptográficas são utilizados pelos atores para a realização de transações escritas através de assinatura digital, sobre uma estrutura de dados, cujo muitos estão representados por valores *hash* criptográficos ou ainda árvores de Merkle para uma representação resumida sobre uma árvore de dados. Embora os termos sejam semelhantes, foi visto que criptografia de chave pública e infraestrutura de chave pública são conceitos distintos, ainda que a PKI utilize de criptografia de chave pública. Foi feita uma explanação acerca de infraestrutura de chave pública descentralizada e seus elementos. Neste ponto há uma ligação com as DLTs, conforme visto, que são utilizadas nesse contexto como uma tecnologia distribuída para mediar a criação de identificadores descentralizados e chaves. Percebeu-se durante a pesquisa que os termos **Identidades Centralizadas** e **Identidades Descentralizadas** são comumente encontrados em bases de pesquisa ou fontes mais técnicas. No entanto, percebeu-se também que estes termos na verdade são *alias* para uma generalização que possui em seu *core* os conceitos consolidados na literatura de PKI e DPKI. Em suma, explanações sobre Identidades Centralizadas e Identidades Descentralizadas remetem aos conceitos de Infraestrutura de Chave Pública e Infraestrutura de Chave Pública Descentralizada respectivamente.

[13]Exemplo de *resolver*: `https://uniresolver.io/`

Conceitos relacionados à identidade digital também foram explanados. Foi visto que, basicamente, uma identidade digital pode ser construída pela reivindicação de um conjunto de atributos por um sujeito/entidade. Alguns atributos podem servir como identificadores, pois identificam uma identidade de maneira única dentro de um contexto. Por exemplo: o campo *serial number* de um certificado X.509 identifica um certificado no contexto de uma autoridade certificadora. Um identificador descentralizado (DID) identifica exclusivamente uma identidade em um contexto global, uma vez que utiliza tecnologias distribuídas para a sua criação e gestão.

Ao entrar no escopo das DLTs, percebeu-se que apesar de terem sua primeira concepção divulgada de uma maneira informal, através de um artigo enviado a uma lista de email, tais tecnologias são compostas por outras tecnologias existentes e consolidadas na academia. As implementações existentes amplamente utilizadas comprovam a eficácia da teoria proposta por Satoshi Nakmoto, e rompem a fronteira do que foi inicialmente concebido. As DLTs foram concebidas para serem aplicadas em um contexto financeiro, resolvendo problemas como o *double spending*, no entanto, a exploração e evolução de tais tecnologias permitiu a sua ampla utilização em outros cenários, que não o financeiro. Uma subseção foi reservada para uma investigação e discussão acerca do anonimato e privacidade nas DLTs. Percebeu-se que, apesar de uma parte dos usuários desejarem o anonimato, em certos casos, a identificação dos atores envolvidos em transações é necessária para atender a algum requisito, ou mesmo para a legitimação das transações. Conclui-se o cenário ideal seria um ambiente heterogêneo neste quesito, que permita a coexistência pacífica de ambas categorias de usuários (os que desejam e os que não desejam o anonimato).

Capítulo 3

Trabalhos Relacionados

É comum encontrar trabalhos relacionando tecnologias de livro-razão distribuídos com várias áreas de pesquisa. No contexto deste trabalho, foram pesquisados e selecionados trabalhos que tratam com DLTs e possuem relação com identidade ditital, ou a falta dela, em algum ponto. Dentre tais trabalhos, é possível perceber que há uma categoria que trata da utilização de tecnologias distribuídas, a exemplo de DLTs, para a construção de novos modelos de identidade digital.

Orman traz uma discussão sobre a dificuldade em gerenciar diferentes identidades digitais, propondo que cada pessoa possua uma identidade digital única, que seja padronizada para ser utilizada por diferentes serviços, com um controle de quais dados poderão ser compartilhados a depender do serviço que necessite da identificação (Orman, 2018). O autor faz críticas aos modelos baseados em infraestrutura de chave-pública (PKI) e sugere que tecnologias de livro-razão distribuído, a exemplo de *blockchain*, podem ser utilizadas para a construção de novos modelos de identidades digitais. O autor fortalece seus argumentos referenciando um projeto desenvolvido pelo MIT para a emissão de certificados digitais utilizando *blockchain*.

Lyons et at. consideram que o principal problema com as identidades digitais atualmente, é que elas são, em grande parte, centralizadas (Lyons et al., 2019). Neste sentido, os autores propõem um modelo de identidade descentralizada, também conhecida como identidade auto-soberana, cujo os usuários criam as suas identidades e adicionam informações a partir de outras fontes, em tese, confiáveis. Os autores mencionam que *blockchain* pode ser uma poderosa solução para diferentes aspectos de identidades decentralizadas.

Um outro projeto desenvolvido no laboratório *MIT Media Lab* do Instituto de Tecnologia de Massachusetts[1] deu origem a um padrão para a criação de aplicações para a emissão e verificação de registros sobre entidades (neste caso, pessoas físicas), chamado Blockcerts (Blockcerts, 2016). Na Blockcerts, reivindicações de afirmações podem ser solicitadas e realizadas entre atores. Basicamente, um emissor faz uma afirmação sobre um destinatário, em um artefato chamado de certificado, seguindo a sintaxe do padrão Open Badges[2]. É gerado um *hash* deste certificado que é registrado em uma *blockchain* (Bitcoin ou Ethereum). O certificado então adiciona o *hash* da transação dentro do seu campo **signature** com um identificador da respectiva *blockchain*. O Blockcerts é *open-source* e pode ser utilizado em projetos de pesquisa ou implementações comerciais. Um ponto interessante nessa solução, é que os atores utilizam endereços de uma DLT como seus identificadores. A Blockcerts deixa claro em sua página de FAQ (do inglês *Frequently Asked Question*) que não é capaz de provar a identidade de um indivíduo ou emissor e que não certifica o mapeamento de chaves públicas para indivíduos ou organizações, apontando claramente a problemática tratada neste trabalho.

Durante a leitura de trabalhos relacionados com o tema em questão, percebeu-se que alguns trabalhos apontam a necessidade de identificação de usuários de tecnologias de livro-razão distribuído, sobretudo as instâncias públicas, com reconhecimento legal, isto é, que aplicações baseadas em *blockchain* possam identificar seus usuários valendo-se de modelos consolidados e amparados por legislações vigentes. Os casos abaixo, ambos no Brasil, ilustram essa demanda.

Júnior et al. explícita um cenário onde há a necessidade de associação entre endereços de carteiras e entidades do mundo real (Júnior et al., 2018). O referido trabalho apresenta uma proposta de criação de uma representação de um ativo digital apelidado de BNDESToken, em uma infraestrutura de *blockchain* para rastrear os recursos do Banco Nacional de Desenvolvimento Econômico e Social. Uma das premissas da proposta é que apenas pessoas jurídicas detentoras de um certificado digital e-CNPJ[3] podem receber o BNDESToken. Os autores citam como um pré-requisito para a implementação da proposta, a existência de um

[1] https://learn.media.mit.edu/

[2] https://openbadges.org/

[3] Nome dado a um artefato que contém um certificado digital e suas respectivas chaves emitidas pela ICP-Brasil que representam eletronicamente uma pessoa jurídica do Brasil.

serviço que forneça o mapeamento entre endereços de carteiras em uma *blockchain* e pessoas jurídicas do Brasil.

Costa et al. também apresenta um serviço que demanda que as partes envolvidas sejam identificadas de uma forma segura para dar legitimidade as transações (Costa et al., 2018). O serviço, chamado de RAP, combina o uso de DLTs, certificação digital e preservação digital para a criação de uma plataforma, escalável e agnóstica, especializada no registro, autenticação e preservação de documentos digitais. Como prova de conceito da plataforma proposta, foi feita a construção de um serviço público para registro e verificação digital da autenticidade de documentos acadêmicos. No protótipo, o registro dos diplomas acadêmicos pode ser feito em duas das DLTs mais populares, Bitcoin e Ethereum, através de uma transação entre a carteira da instituição emissora (uma IES) e a carteira da instituição autenticadora (RNP ou MEC, por exemplo). Neste caso, a publicização inequívoca da propriedade dos endereços de carteiras digitais em pauta poderia garantir a transparência e a segurança das transações de registro.

A comunidade Bitcoin introduziu uma proposta de melhoria do protocolo bitcoin (BIP do inglês *Bitcoin Improvement Proposal*) descrevendo um protocolo de pagamento, que estrutura os dados das transações para evitar erros (Andresen e Hearn, 2013). O **BIP 70** utiliza uma assinatura digital realizada com uma chave associada a um certificado digital X.509 de uma PKI cujo a confiança é baseada em terceiros confiáveis, para identificar o endereço de recebimento no momento da transação. No entanto, como a abordagem foi inicialmente planejada para os comerciantes, percebemos que para a operação do BIP 70 é necessário que cada ator que deseje se identificar na rede Bitcoin faça uma implementação código que forneça um serviço para atender pedidos de software de carteira. Basicamente, o BIP 70 especifica um protocolo de comunicação a ser utilizado por *softwares* de carteira e servidor. Um *software* de carteira estrutura uma transação informando uma URL, ao invés de um endereço, como sendo o destinatário. A implementação *server-side* recebe a intensão de pagamento, assina uma estrutura de dados denominada *PaymentRequest*, contendo o endereço de carteira associado ao website e a devolve juntamente com o certificado X.509. O *software* de carteira por sua vez, valida a assinatura digital e envia a transação à *blockchain* com o endereço contido no *PaymentRequest* como destinatário.

Outra abordagem semelhante foi encontrada na carteira Bitpay (Bitpay, 2019). Essa

abordagem amplia o BIP 70 sugerindo o uso de assinaturas com algoritmo de curva elíptica (ECDSA do inglês *Elliptic Curve Digital Signature Algorithm*)[4], o mesmo algoritmo utilizado pelas principais DLTs, como uma alternativa aos certificados digitais X.509. O protocolo especifica rotas para a distribuição das chaves públicas com descrição dos respectivos donos, como também para a distribuição de chaves PGP utilizadas para as assinaturas das chaves públicas utilizadas no protocolo de pagamento.

[4]Algoritmo de criptografia utilizado por DLTs que utiliza de criptografia de curva elíptica, que gera uma segurança à altura de outros algoritmos com um tamanho de chave menor em quantidade de bits (Stallings, 2015).

Capítulo 4

Trabalho Proposto

4.1 Metodologia

Para atender aos objetivos propostos neste trabalho, foram utilizadas duas metodologias: 1- Revisão da Literatura e 2 - Metodologia de Construção *(Build)*. Em um momento inicial, percebeu-se a falta de clareza e falta de consenso na terminologia envolvida nas tecnologias de livro-razão distribuído. Fez-se necessário uma ampla revisão na literatura, com leitura de diferentes concepções acerca do mesmo tema, para a construção de uma base de conhecimento. Uma vez que o *core* das DLTs é baseado na composição de diversas tecnologias consolidadas, a compreensão do seu funcionamento conduziu a leitura dos conceitos necessários para o desenvolvimento do trabalho. Esta base de conhecimento está compilada na seção 2. A leitura de artigos e trabalhos relacionados corrobora com a importância para a pesquisa realizada neste trabalho. Embora haja uma categoria de DLTs privadas, cujo a identificação dos atores pode ser gerida sem maiores complicações, é evidente a lacuna no quesito de identificação de usuários em DLTs públicas.

A outra metodologia adotada neste trabalho é a metodologia de Construção *(Build)* (Amaral, 2011). Nesta metodologia, almeja-se comprovar uma hipótese através da sua construção. Na etapa de construção, inicialmente foi dedicado um tempo à elaboração de uma solução para atender à questão da identificação, levando-se em consideração: segurança e confiabilidade. Pode-se dizer que a solução proposta inspira-se na ideia de construção das DLTs, isto é, a utilização combinada de uma suíte de tecnologias consolidadas para a resolução de um problema. Em seguida, o esforço foi direcionado para a validação da solução

proposta através de uma prova de conceito. A construção do modelo proposto é especificada neste trabalho, assim como um protótipo funcional foi desenvolvido para a validação do modelo em duas das DLTs públicas mais utilizadas atualmente, **Bitcoin** e **Ethereum**. A construção do protótipo desenvolvido como prova de conceito, denominado *Address Name System* (ANS), foi realizada de acordo com uma abordagem de desenvolvimento incremental e evolutiva (Sommerville, 2011). Por fim, com a primeira versão de um protótipo funcional desenvolvida, foram realizados experimentos em ambiente controlado (em execução local) e em ambientes públicos (instâncias públicas das principais DLTs).

Em suma, podemos enumerar as atividades realizadas neste trabalho da seguinte forma:

1. Revisão da literatura para a elucidação de conceitos acerca de DLTs e formação de uma base de conhecimento em tecnologias relacionadas ao tema em questão (Capítulo 2).

2. Investigação de trabalhos relacionados ao tema da pesquisa, como trabalhos que se relacionam por apontarem a problemática tratada neste trabalho e também trabalhos que demonstram propostas de soluções para o problema (Capítulo 3).

3. Especificação de proposta para a identificação dos atores por trás de endereços de carteiras, com base no conhecimento adquirido durante a revisão bibliográfica (Capítulo 4).

4. Definição de um modelo arquitetural para um sistema de referência que implemente a especificação proposta, de acordo com a metodologia de construção proposta neste trabalho (Seção 4.3.1).

5. Definição de tecnologias (linguagens, *frameworks* e bibliotecas) a serem utilizadas no desenvolvimento do sistema de referência, como também a obtenção de certificado digital real e válido emitido por uma AC da ICP Brasil para os experimentos (Seção 4.3).

6. Prova de conceito com o desenvolvimento de um protótipo funcional de um sistema de referência (Seção 4.3).

7. Realização de experimentos para a validação do sistema e avaliação da viabilidade técnica da arquitetura proposta, inclusive com integração com outras aplicações (Seção 5).

8. Produção e publicação de artigo científico descrevendo a solução desenvolvida e os resultados preliminares (Seção 6.3).

9. Produção da dissertação final e defesa.

4.2 Esboço da Solução

Nesta seção, será apresentada a proposta que visa atender aos objetivos descritos neste trabalho. A solução proposta deve contribuir com a dinâmica das DLTs públicas, provendo uma forma segura de identificar atores associados a endereços de carteiras digitais. Objetiva-se, com a leitura deste capítulo, a compreensão do método proposto, como também a apresentação das implementações de componentes que compõem um sistema de referência desenvolvido para a validação da proposta. O capítulo se inicia com a explanação do esboço da solução, detalhando e referenciando as tecnologias utilizadas, inclusive com detalhamento técnico. Posteriormente, é apresentada uma prova de conceito, com a apresentação de um sistema de referência, que atende aos requisitos descritos no esboço. Por ter o funcionamento semelhante ao do consolidado serviço de tradução de nomes em recursos, *Domain Name System* (DNS), utilizou-se a nomenclatura *Address Name System* (ANS) para fazer referência ao sistema proposto neste trabalho.

Um dos requisitos a serem atendidos para a identificação segura, é a confiabilidade da associação entre os endereços e entidades. Isto é, uma associação confiável, segura, e reconhecida como legítima. Muitos governos consideram a PKI como uma tecnologia para a ligação entre entidades e chaves públicas representadas em certificados digitais, com o objetivo de comunicações digitais garantidas, verificáveis e seguras (Al-Khouri, 2012). A solução proposta utiliza a combinação de assinaturas digitais para prover a associação segura entre endereço de carteira e entidade do mundo real. De um lado, uma assinatura digital realizada por uma chave cujo a associação com uma entidade do mundo real é assegurada por uma autoridade confiável, neste caso, uma autoridade certificadora. De outro lado, uma outra as-

sinatura digital realizada com uma chave privada associada a um endereço de uma DLT. As assinaturas são depositadas em um artefato, análogo a uma credencial verificável, no entanto, em uma sintaxe bem definida para a assinatura digital em documentos XML, o XMLDSig [1]. A confiança na associação entre a entidade e chave pública do certificado digital é provida por uma infraestrutura de chave pública, uma vez que uma autoridade certificadora realizou os procedimentos necessários para a verificação da identidade.

4.2.1 Declaração de Posse de Endereços de Carteiras Digitais

Em uma visão geral, a associação deve ser feita de forma que seja possível comprovar a identidade de uma pessoa/entidade, e comprovar que essa pessoa/entidade é proprietária da chave privada de um determinado endereço. Esta associação é feita através de dois passos: i) **prova de posse** e ii) **prova de identidade**. A **prova de posse** do endereço de carteira indica se a pessoa/entidade possui acesso à chave privada do endereço de carteira para transacionar em uma DLT. A comprovação de posse é feita com a assinatura digital de uma estrutura de dados utilizando a chave privada correspondente à chave pública do endereço reivindicado. A **prova de identidade** tem como objetivo obter a comprovação de que quem está pedindo a associação é quem diz ser e é feita com a assinatura digital realizada com a chave associada a um certificado digital[2]. Neste sentido, um pré-requisito para o registro no ANS é que a pessoa/entidade possua um certificado digital válido e uma premissa para a confiabilidade da identificação é que uma Autoridade Certificadora tenha realizado os procedimentos de verificação de documentos corretamente, obtendo êxito na emissão correta do certificado. Uma premissa fundamental do ANS é que o artefato que contém tais provas seja autocontido e as provas possam ser verificadas de forma autônoma por qualquer interessado em qualquer tempo.

Entretanto, caso utilizadas separadamente, as assinaturas para prova de posse e prova de identidade não fariam qualquer associação entre os proprietários de suas respectivas chaves. É necessária, portanto, a vinculação entre a prova de posse do endereço e a prova de identidade de modo a garantir que o proprietário da chave privada de acesso ao endereço de carteira

[1] https://www.w3.org/TR/xmldsig-core2/

[2] No Brasil, um documento eletrônico assinado por uma chave emitida por uma AC da cadeia ICP-Brasil possui validade jurídica (Brasil, 2001).

é o mesmo proprietário da chave privada que o identifica através do certificado digital. Esta associação é feita com o uso de um documento XML específico, chamado de **ANS Certificate**. A estrutura do **ANS Certificate** segue o padrão XML-Dsig com a nomenclatura da seção a ser assinada inicialmente ($< ToBeSigned >$) baseada no certificado digital X.509. A figura 4.1 ilustra a estrutura principal de um ANS Certificate sinalizando as seções que representam as provas de posse e de identidade. A figura também mostra a relação de dados no ANS Certificate. O endereço reivindicado e presente na seção $< ToBeSigned >$ é derivado da chave pública contida na seção $< DLTSignature >$. O nó $< EntityCertificate >$ da seção $< ToBeSigned >$ contém dados que fazem referência ao certificado X.509 contido na seção $< Signature >$.

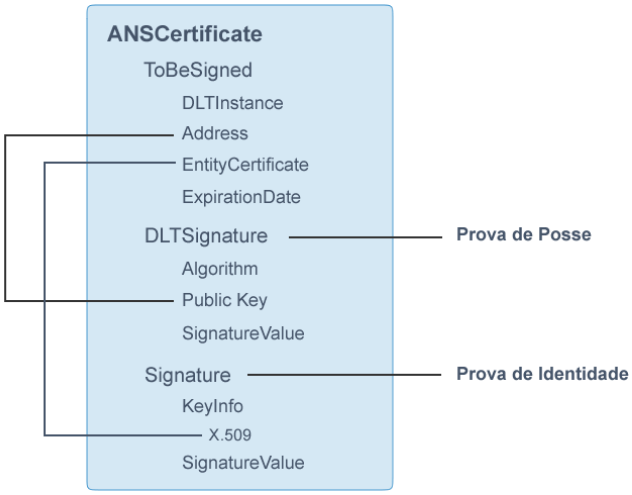

Figura 4.1: Estrutura de um ANS Certificate

Fonte: Próprio autor

Conforme pode ser visto na figura 4.1, a seção $< ToBeSigned >$ foi modelada modelada para fazer referência tanto ao certificado digital do usuário quanto ao endereço da carteira. Para referenciar o certificado digital, o elemento $< EntityCertificate >$ da seção $< ToBeSigned >$ contém os campos *Distinguished Names* da AC emissora e o *serial number* do certificado, que o identifica unicamente dentre os certificados emitidos pela AC, conforme pode ser visto em um ANS Certificate real exibido na figura 4.2.

Além dos dados de referência ao certificado da pessoa/entidade, a seção $< ToBeSigned >$ de um **ANS Certificate** contém ainda os atributos $< DLTInstance >$, $< Address >$, e $< ExpirationDate >$, onde os dois primei-

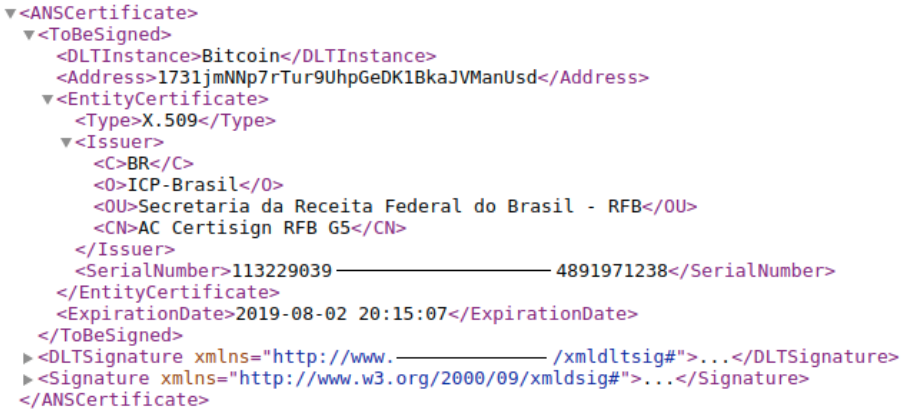

Figura 4.2: Exemplo de um ANS Certificate

Fonte: Próprio autor

ros especificam a instância da DLT e o endereço da carteira, respectivamente, e o último indica a data de validade do **ANS Certificate**. Há ainda o elemento $< DLTSignature >$ usado para armazenar a assinatura digital usada como prova de posse e, finalmente, o elemento $< Signature >$, usado para armazenar a assinatura digital usada em nosso contexto como prova de identidade. As assinaturas são acrescentadas ao final do documento conforme o modelo de assinatura envelopada. Na estratégia de assinatura *Enveloped*, o conteúdo da assinatura é o próprio documento XML e o valor da assinatura é inserido ao final do documento juntamente com o certificado digital do assinante, gerando um artefato final que contém todos os elementos necessários para a verificação de autenticidade da informação criptografada (Bartel et al., 2015). O elemento $< Signature >$ utiliza a estrutura em conformidade com o consolidado padrão internacional W3C, que armazena além da assinatura digital, o certificado associado à chave privada utilizada (Bartel et al., 2015). O elemento $< DLTSignature >$ contém o algoritmo usado para a assinatura, a chave pública referente ao endereço de carteira contido no campo $< Address >$ da seção $< ToBeSigned >$ e a assinatura digital, conforme modelado no *XML Schema Definition* (XSD) e apresentado na Figura 4.3.

Conforme descrito anteriormente, a estratégia adotada para o mapeamento basicamente utiliza a combinação de assinaturas digitais. A pessoa/entidade inicialmente assina digitalmente a seção $< ToBeSigned >$ do **ANS Certificate** com a chave privada do endereço e, em seguida, tanto a seção $< ToBeSigned >$ quanto a seção $< DLTSignature >$ são

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0.0">
 <xs:element name="DLTSignature">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Algorithm" type="xs:string"/>
    <xs:element name="PublicKey" type="xs:string"/>
    <xs:element name="SignatureValue" type="xs:string"/>
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Figura 4.3: *XML Schema Definition* do elemento DLTSignature

Fonte: Próprio autor

assinadas com a chave privada do certificado, gerando um pacote que contém o **ANS Certificate** duplamente assinado, a chave pública associada ao endereço para a verificação da prova de posse e o certificado digital com a respectiva chave pública para a verificação da prova de identidade. Como a assinatura para a prova de identidade é feita com a chave privada do certificado digital que é referenciado no próprio documento, é formado um elo entre a prova de posse do endereço de carteira e a prova de identidade. Este fluxo está ilustrado na Figura 4.4. Percebe-se que o ANS Certificate contém os atributos descritos na seção 2 que caracterizam uma credencial verificável.

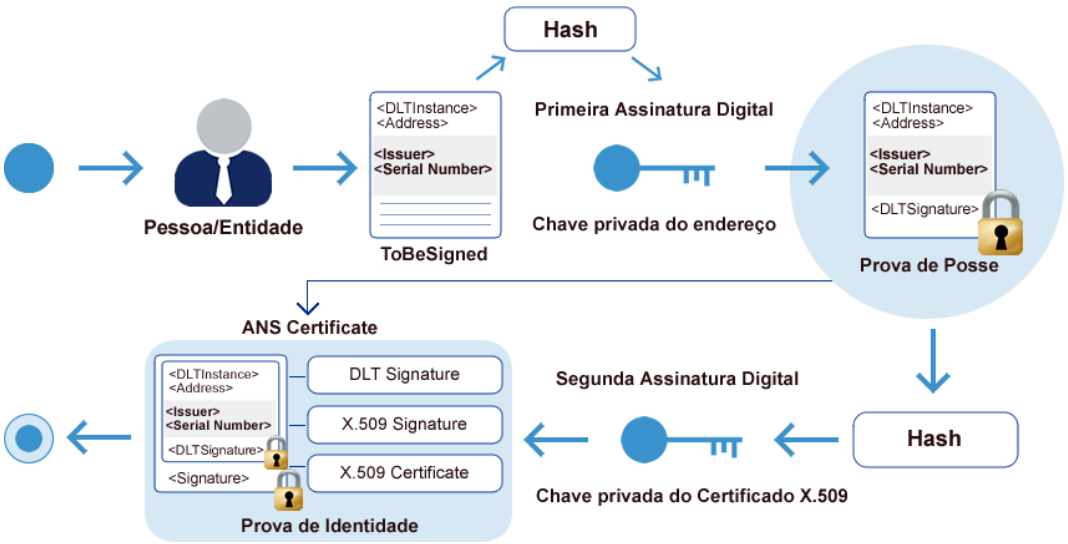

Figura 4.4: Fluxo de assinatura de um ANS Certificate

Fonte: Próprio autor

Conforme demonstrado no fluxo da figura 4.4, as etapas para a produção de um ANS Certificate podem ser elencadas na seguinte sequência:

1. Aplicação de função de *hash* na seção $< ToBeSigned >$.
2. Encriptação com a chave privada do endereço no *hash* produzido.
3. Inserção do resultado da encriptação, juntamente com a chave pública e identificador do algoritmo no elemento $< DLTSignature >$.
4. Aplicação de função de *hash* em todo o elemento raiz ANSCertificate.
5. Encriptação com a chave privada do certificado X.509 no *hash* produzido.
6. Inserção do valor da encriptação e certificado digital no artefato final.

4.2.2 Verificação de Posse de Endereços de Carteiras Digitais

A validação das assinaturas e, consequentemente, da associação entre pessoa/entidade e endereço de carteira, é feita em um processo inverso que usa as respectivas chaves públicas do certificado digital e do endereço de carteira. Tal verificação se baseia em procedimentos bem estabelecidos com algoritmos públicos e padronizados e pode ser feita de forma autônoma a partir do próprio **ANS Certificate**, o qual é um pacote autocontido contendo todos os artefatos necessários para que um interessado possa verificar a autenticidade das informações. Deste modo, utilizando a chave pública contida no certificado digital utilizado, é possível conferir a autenticidade da prova de identidade, e associar o **ANS Certificate** ao proprietário do certificado. Da mesma forma, é possível verificar a assinatura digital da prova de posse utilizando a chave pública do endereço de carteira e associar o **ANS Certificate** ao proprietário do endereço. A Figura 4.5 ilustra o fluxo para a verificação das provas necessárias para a validação da associação endereço-entidade. É possível observar que cada assinatura ao ser decriptada com a chave pública resulta em um resumo *(hash)* do artefato assinado. Ao aplicar a mesma função de *(hash)* no artefato original (não-assinado), é possível comparar o *(hash)* resultante com o *(hash)* da decriptação para a validação da assinatura, conforme explicado no capítulo 2.

Dessa forma, o fluxo para a verificação de um ANS Certificate pode ser dividido nas seguintes etapas:

1. Decriptação do ANS Certificate com a chave pública do certificado X.509.

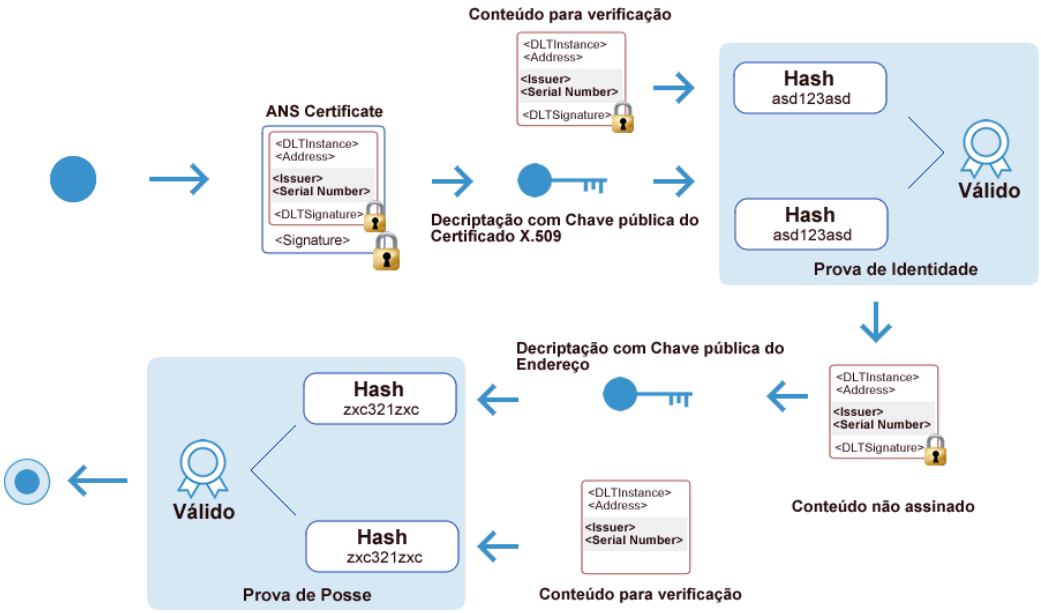

Figura 4.5: Fluxo de verificação de um ANS Certificate

Fonte: Próprio autor

2. Aplicação de função *hash* na subseção do documento a ser verificada.

3. Comparação do *hash* resultante da decriptação da etapa 1 com o *hash* aplicado na etapa anterior.

4. Decriptação da seção $< ToBeSigned >$ com a chave pública do endereço.

5. Aplicação de função *hash* na seção $< ToBeSigned >$.

6. Comparação do *hash* resultante da decriptação da etapa 4 com o *hash* aplicado na etapa anterior.

4.3 Um Protótipo Funcional do *Address Name System*

Se analisarmos na literatura o conceito de sistema, podemos entender como um conjunto de elementos interligados trabalhando para um determinado fim. Tais elementos podem ser: pessoas, *softwares*, processos, etc. O ANS foi desenvolvido para ser pensado como um sistema, com um conjunto de componentes atuando, cada qual com a sua responsabilidade. Em uma visão ampla, o ANS deve prover dois serviços fundamentais: um serviço de registro,

a ser utilizado de maneira voluntária por parte do detentor de um endereço de carteira e de um certificado digital para registrar o mapeamento de endereço para entidade (ou endereço-entidade), e um serviço de consulta, que em uma visão mais ampla, corresponde a um serviço que recebe parâmetros de entrada como um identificador de uma instância de DLT e um endereço de carteira e retorna ao usuário a identidade declarada, se houver uma.

Para isso, foram realizadas implementações em *client-side*, a exemplo da aplicação **ANS Client**, uma aplicação auxiliar desenvolvida para a geração dos ANS Certificates, como também foram realizadas implementações em *server-side*, a exemplo do **ANS Server**, uma implementação que atua como um *web-service* e disponibiliza serviços de registro e consulta. A arquitetura geral do ANS está ilustrada na Figura 4.6. Como mostrado na figura, a geração de ANS Certificates também poderia ser feita em *softwares* de carteira digital, uma vez que tais softwares armazenam as chaves privadas dos endereços de carteiras. Para a prova de conceito, a primeira versão do ANS foi desenvolvida para as instâncias de DLTs públicas da Bitcoin e Ethereum, uma vez que são duas das *blockchains* mais populares, levando-se em consideração o número de nós que compõem a rede. Ainda que a arquitetura dos componentes desenvolvidos foi projetada com padrões de projetos para a adição de suporte a novas DLTs.

4.3.1 Arquitetura

4.3.2 ANS Client

A aplicação chamada de **ANS Client** foi desenvolvida para representar a interface no lado do cliente, contemplando os processos que devem ser realizados por qualquer aplicação cliente que possa ser integrada. Na primeira versão desenvolvida do ANS Client, os seguintes casos de uso foram implementados para atender à prova de conceito: geração/registro de ANS Certificates, consulta de associação endereço-entidade e validação de ANS Certificates. Em uma visão superficial, no caso de uso geração/registro de ANS Certificates, o ANS Client obtém os dados necessários para a produção de um **ANS Certificate**, formata o arquivo XML equivalente, faz as assinaturas digitais conforme descrito na Seção 4.2.1 e envia o **ANS Certificate** gerado para registro em um servidor ANS, chamado **ANS Server**. O caso de uso "consulta de associação endereço-entidade"realiza uma consulta a partir de pa-

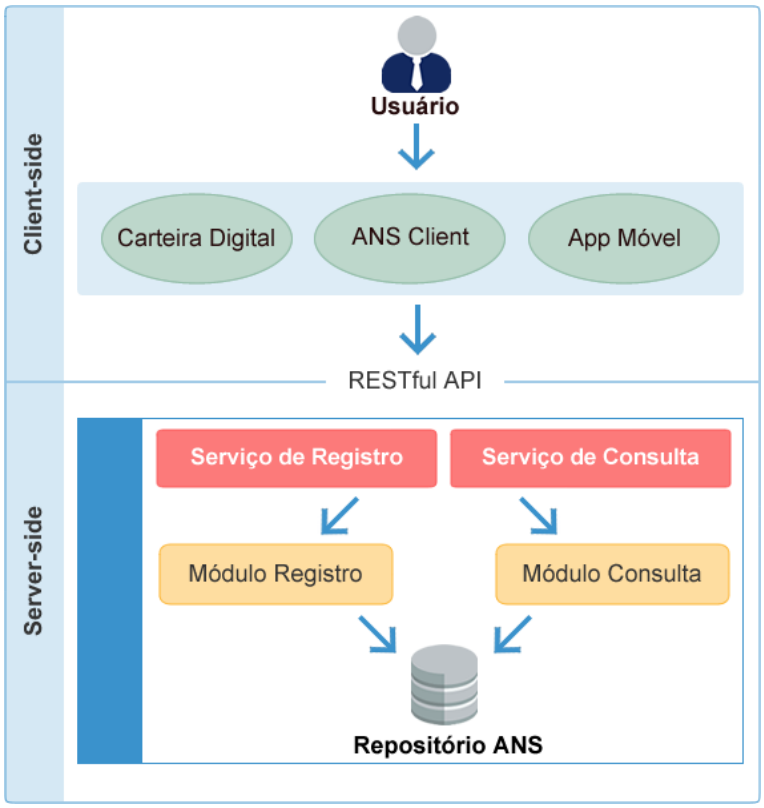

Figura 4.6: Arquitetura do protótipo

Fonte: Próprio autor

râmetros informados, e o caso de uso "validação de ANS Certificates"realiza a validação de um artefato (ANS Certificate) que contém as provas necessárias para a identificação. Esta subseção apresenta o ANS Client mais detalhadamente.

O ANS Client foi desenvolvido utilizando a tecnologia *Java 8* como linguagem de programação combinado com os pacotes gráficos do *JavaFX*. O Java é uma tecnologia consolidada para dar suporte a assinaturas digitais no padrão XMLDSig. Nesta prova de conceito, a assinatura digital de prova de identidade foi realizada utilizando o algoritmo RSA com SHA-256 como função de *hash*. Para lidar com as assinaturas digitais de prova de posse, foram utilizadas as bibliotecas *bitcoinj* e *ethereumj*, ambas implementações em Java de protocolos de assinatura com algoritmo de curva elíptica (ECDSA) para Bitcoin e Ethereum respectivamente.

Para iniciar um registro de posse de um endereço de carteira digital no caso de uso geração/registro de ANS Certificates, a pessoa/entidade deverá informar no **ANS Client** a instância de DLT na qual o endereço foi criado, a chave privada do endereço de carteira, os dados

de acesso à *keystore*[3] onde a chave privada do certificado foi instalada e o seu respectivo certificado digital, conforme pode ser visto na interface mostrada na Figura 4.7. O endereço de carteira é derivado a partir de um *hash* da chave pública[4] correspondente à chave privada informada, sendo calculado pelo próprio **ANS Client** e exibido em tela. Ao clicar no botão *Generate*, as rotinas no *back-end* do ANS Client produzem as assinaturas digitais necessárias, geram o artefato ANS Certificate e realizam uma chamada para uma rota de registro no ANS Server. O **ANS Server** desenvolvido no protótipo é uma implementação de um *web-service* acessível através de uma *API RESTful*, para a condução do registro de associação entidade-endereço, como também para a condução do armazenamento de ANS Certificates em uma camada de repositório. A subseção 4.3.3 detalhará mais profundamente este componente.

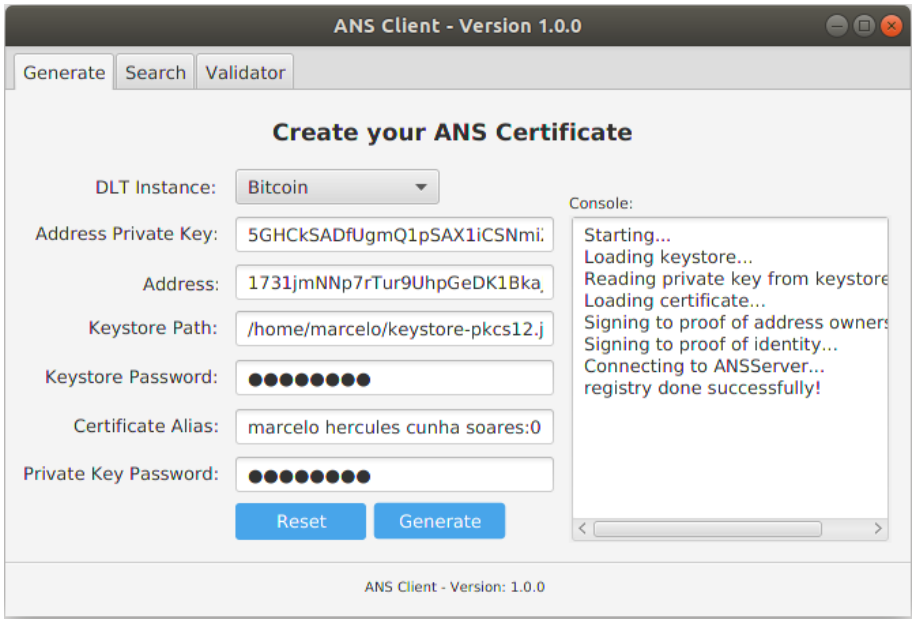

Figura 4.7: ANS Client - Geração e registro de ANS Certificate

Fonte: Próprio autor

Conforme mencionado anteriormente, os componentes do ANS foram desenvolvidos uti-

[3]Nesta primeira versão do protótipo foi utilizado um certificado digital ICP-Brasil do tipo A1, que é um certificado disponibilizado ao usuário como um arquivo digital para que seja instalado no computador.

[4]Na plataforma Ethereum, o endereço de carteira é formado pelos 160 bits mais a direita do resultado de uma função *hash* Keccak-256 da chave pública (Wood, 2017). Na Bitcoin, o endereço possui tamanho de 160 bits e é formado a partir do uso combinado de diferentes funções de hash (RIPEMD-160 e SHA-256) (Bitcoin-Wiki, 2018).

lizando de padrões de projetos para que novas DLTs possam ser suportadas em versões posteriores. Para atender a este requisito, a arquitetura de classes foi implementada com a adoção do padrão de projeto *Template Method*. O *Template Method* é um padrão de projeto comportamental que pode ser aplicado quando há a necessidade de implementar o esqueleto de um algoritmo com partes invariantes uma única vez e implementar comportamentos específicos em subclasses (Gamma et al., 2006). A figura 4.8 apresenta um diagrama de classe com a implementação do referido padrão de projeto no ANS Client no caso de uso de geração/registro de ANS Certificate.

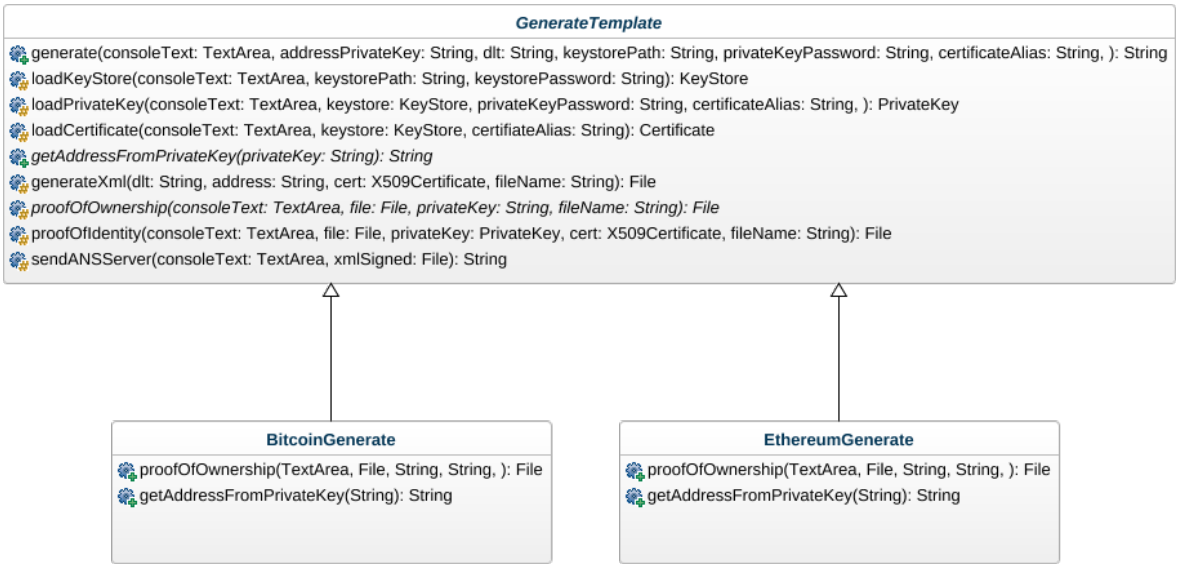

Figura 4.8: Padrão *Template Method* na implementação do ANS Client

Fonte: Próprio autor

O método *generate* da classe *GenerateTemplate* é responsável por encapsular o algoritmo, isto é, a sequência de outras operações necessárias para realizar a geração e registro de um ANS Certificate. A figura 4.8 mostra a sequência de métodos na ordem em que são chamados pelo primeiro método, o *generate*. O método *proofOfIdentity* que contém a lógica para a prova de identidade está implementado na mesma superclasse *GenerateTemplate* uma vez que é uma operação comum independente da instância de DLT a ser utilizada. Os métodos *proofOfOwnership* e *getAddressFromPrivateKey*, responsáveis pela prova de identidade e por obter o endereço a partir de uma chave pública, respectivamente, são abstratos na superclasse, com cada especificidade implementada em métodos concretos nas subclasses

de acordo com a instância de DLT, neste caso, nas subclasses *BitcoinGenerate* e *EthereumGenerate*. O ANS Client, a partir da DLT selecionada pelo usuário, instancia a respectiva subclasse que por sua vez herda da superclasse que contém o template do algoritmo.

Para consultar uma associação entre entidade e endereço, é necessário informar a instância da DLT e o endereço o qual deseja ser consultado. O ANS Client realiza uma chamada a uma rota de consulta do ANS Server, que por sua vez devolve os dados validados contidos no ANS Certificate, caso haja um registro para esta DLT e endereço. A interface para o respectivo caso de uso pode ser vita na figura 4.9.

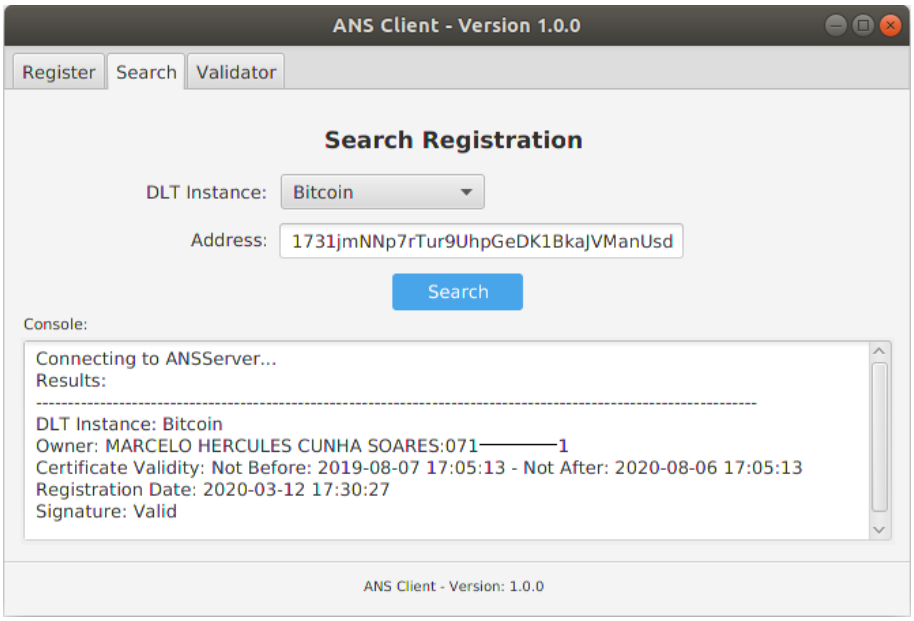

Figura 4.9: ANS Client - Consulta de associação endereço-entidade

Fonte: Próprio autor

O ANS Client também disponibiliza um caso de uso para a validação do artefato que corresponde ao ANS Certificate, que possa ter sido disponibilizado ao usuário por qualquer outra via. Para realizar a validação, o usuário precisa submeter o arquivo XML. O ANS Client realizará as validações necessárias e descritas na subseção 4.2.2 e exibirá os dados da entidade validada na prova de identidade para o usuário. A interface de validação de ANS Certificates pode ser vista na figura 4.10.

O procedimento para a validação de uma associação, isto é, a validação de um ANS Certificate, também é implementada de acordo com o padrão *Template Method*, uma vez que a validação de prova de posse pode conter especificidades de acordo

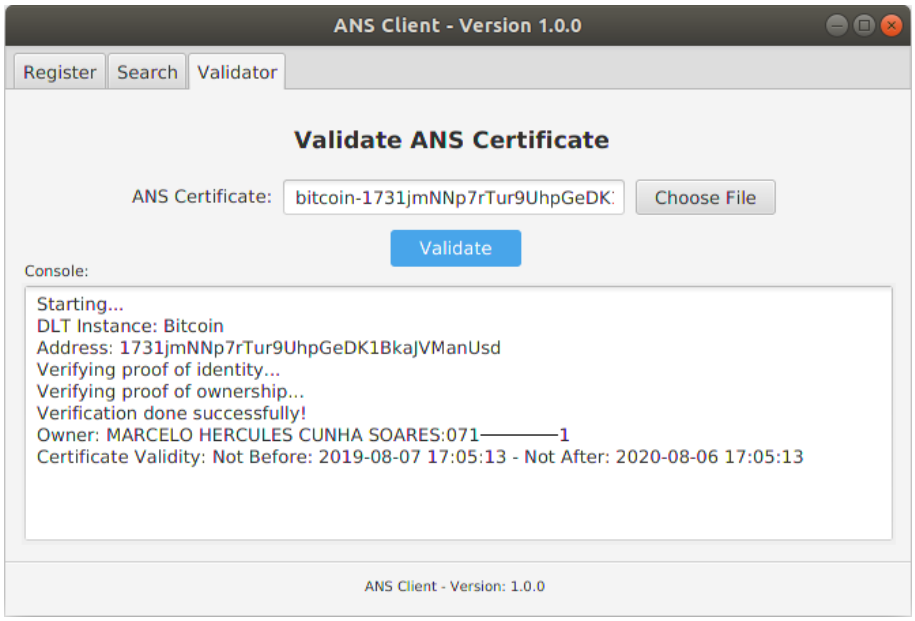

Figura 4.10: ANS Client - Validação de ANS Certificates

Fonte: Próprio autor

com a DLT do endereço em questão. Uma superclasse *ValidatorTemplate* contém o algoritmo, representado em uma sequência de métodos, para a validação de um ANS Certificate. A lógica para a validação da prova de identidade é implementada na superclasse no método *validateProofOfIdentity*, enquanto que a validação da prova de posse está implementada nas respectivas subclasses, neste caso, *BitcoinValidator* e *EthereumValidator*, de acordo com a DLT utilizada, no método *validateProofOfOwnership*. A aplicação instancia a respectiva subclasse de acordo com o valor do nó $< DLTInstance >$ do ANS Certificate. A arquitetura das classes envolvidas no caso de uso de validação de ANS Certificates está representada no diagrama de classes na figura 4.11.

Uma vez gerado, o ANS Certificate corresponde ao artefato que contém as provas criptográficas necessárias para associar uma entidade a um endereço de carteira digital, e pode ser distribuído entre partes interessadas por qualquer via desejável pelo seu detentor.

4.3.3 ANS Server

O **ANS Server** é o componente do sistema que representa uma aplicação *server-side* responsável por conduzir o processo de registro e consulta de associação entidade-endereço, através de uma API *RESTful* disponibilizada para o ANS Client. O ANS Server foi desenvolvido

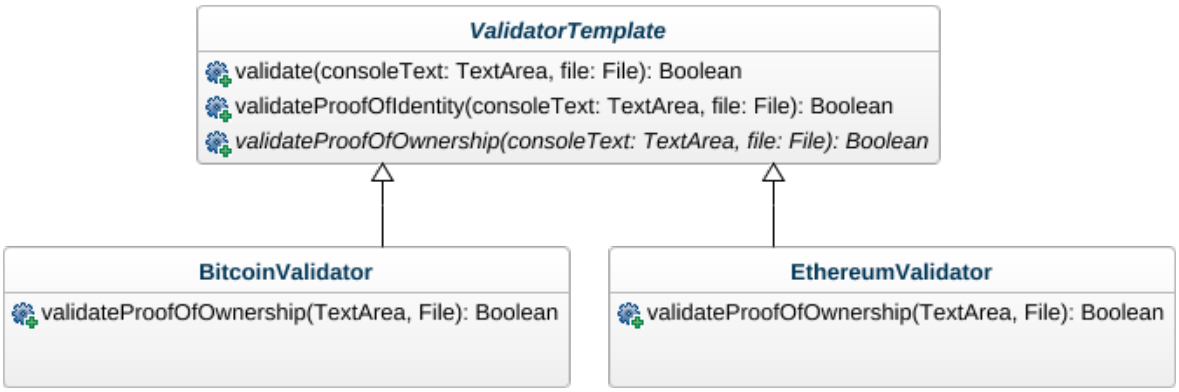

Figura 4.11: Padrão *Template Method* na validação de ANS Certificates

Fonte: Próprio autor

com tecnologia *Java 8* como linguagem de programação em combinação com o *framework Spring Boot*[5]. A arquitetura de código segue o padrão de desenvolvimento *MVC* (acrônimo de *Model-View-Controller*), padrão este que propõe o agrupamento de código em camadas de acordo com a sua responsabilidade. A camada *view* representa a interface do usuário (UI do inglês *User Interface*), a camada *model* contém as classes de domínio da aplicação, como representação de entidades, regras de negócio, acesso a dados, etc. A camada *controller* serve como uma ponte entre as *views* e os *models*. O código agrupado nas classes de controladores é responsável por receber requisições, ou mesmo acionar eventos específicos para componentes visuais, dependendo da tecnologia utilizada, e manipular *models* para realizar operações, podendo ainda devolver dados para a visão (Krasner e Pope, 1998). A figura 4.12 representa a arquitetura MVC do ANS Server.

É possível perceber que o ANS Server fornece na camada de visão, uma interface web para que um interessado possa consultar um registro de associação de forma interativa, sem a necessidade da implementação de uma aplicação cliente. Esta interface realiza a sua requisição para a classe *IndexController* da camada *controller*. Uma outra classe na mesma camada, a *ANSServiceController* é responsável por expor uma *API RESTful* para atender a requisições oriundas de aplicações clientes, como o ANS Client. As respostas dessas requisições são retornadas em formato *JSON* (do inglês *JavaScript Object Notation*)[6]. As classes responsáveis

[5]https://spring.io/projects/spring-boot

[6]https://www.json.org/

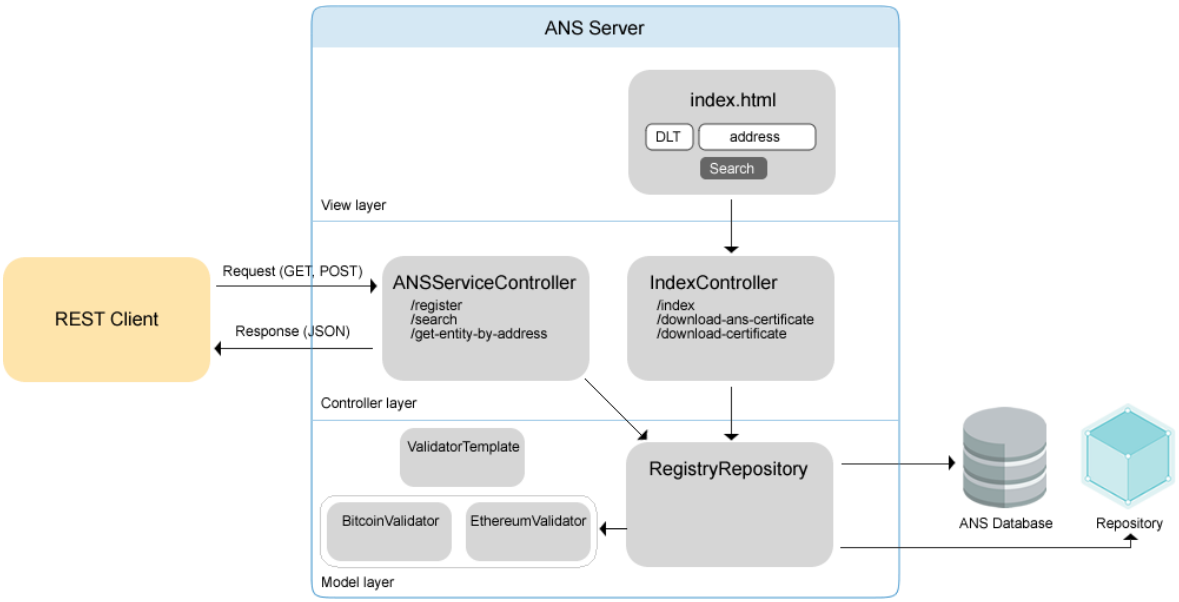

Figura 4.12: Arquitetura MVC do ANS Server

Fonte: Próprio autor

pela validação do ANS Certificate, apresentadas na subseção 4.3.2, foram compactadas em um pacote *jar* (*Java Archive*) para serem reaproveitadas no ANS Server.

O ANS Server conduz o processo de registro de um novo ANS Certificate aplicando as verificações necessárias para a garantia da confiabilidade da associação endereço-entidade. Basicamente, são realizadas as seguintes tarefas: validação das assinaturas com as respectivas chaves públicas, verificação do elo entre as duas provas, inserção de um registro em sua tabela de mapeamento e armazenamento do ANS Certificate em um repositório distribuído denominado **ANS Repository**. O registro na tabela de mapeamento do ANS Server corresponde a uma inserção de uma tupla em uma tabela da sua base de dados, armazenando os dados da DLT, o endereço em questão, e o caminho para a recuperação do artefato que contém as provas criptográficas, isto é, o ANS Certificate, no ANS Repository. A tecnologia utilizada para armazenar dados de um registro de mapeamento foi o banco de dados *PostgreSQL*[7].

O serviço para consulta de associação entidade-endereço recebe como entrada os parâmetros: *dlt-instance* e *address*, correspondentes à instância da DLT e o endereço que deseja

[7]https://www.postgresql.org/

ser pesquisado respectivamente. Ao receber uma solicitação de consulta, o ANS Server pesquisa na sua tabela de mapeamento a partir dos parâmetros informados. Caso haja um registro, é realizada a obtenção do respectivo ANS Certificate no ANS Repository, a partir do *path* armazenado na base do ANS Server, com a validação das assinaturas novamente antes de devolver o ANS Certificate para o solicitante. A Figura 4.13 mostra um exemplo de retorno do serviço de consulta do ANS Server.


```
{
    "id": 5,
    "dlt": {
        "id": 2,
        "name": "Ethereum"
    },
    "address": "0xcd2a3d9f938e13cd947ec05abc7fe734df8dd826",
    "registrationDate": "2020-03-11T23:19:32.047+0000",
    "ansCertificateCID": "QmbDWHHDA49rRfvAH2p6Y6xzsWLWCFsi32qwgRvQprF8Je",
    "owner": "MARCELO HERCULES CUNHA SOARES:071————1",
    "notBefore": "2018-08-02T23:15:07.000+0000",
    "notAfter": "2019-08-02T23:15:07.000+0000",
    "signatureValid": true
}
```


Figura 4.13: Retorno do serviço de consulta do ANS Server

Fonte: Próprio autor

Conforme visto na figura 4.12, o ANS Server também disponibiliza uma interface com um formulário web para a consulta de registros de associações entidade-endereço. A interface fornece campos para que o usuário informe a instância da DLT e o endereço da carteira e, caso haja o registro, o mesmo procedimento de recuperação e validação do ANS Certificate é realizado e os dados do registro e do ANS Certificate são exibidos em tela. Para uma maior confiabilidade, a interface web também disponibiliza a opção de fazer *download* do ANS Certificate para que o usuário possa fazer o processo de verificação de forma autônoma, caso deseje. Também é possível exportar e fazer o *download* do certificado digital X.509 contido no ANS Certificate para a verificação da identidade da pessoa/entidade associada ao endereço. Tal interface é mostrada na Figura 4.14.

4.3.4 ANS Repository

Para completar o mecanismo proposto, é necessário disponibilizar uma forma de que os **ANS Certificates** autoproduzidos pelos detentores dos endereços possam ser facilmente recuperados por qualquer parte interessada. Tecnologias de livro-razão distribuído, quando

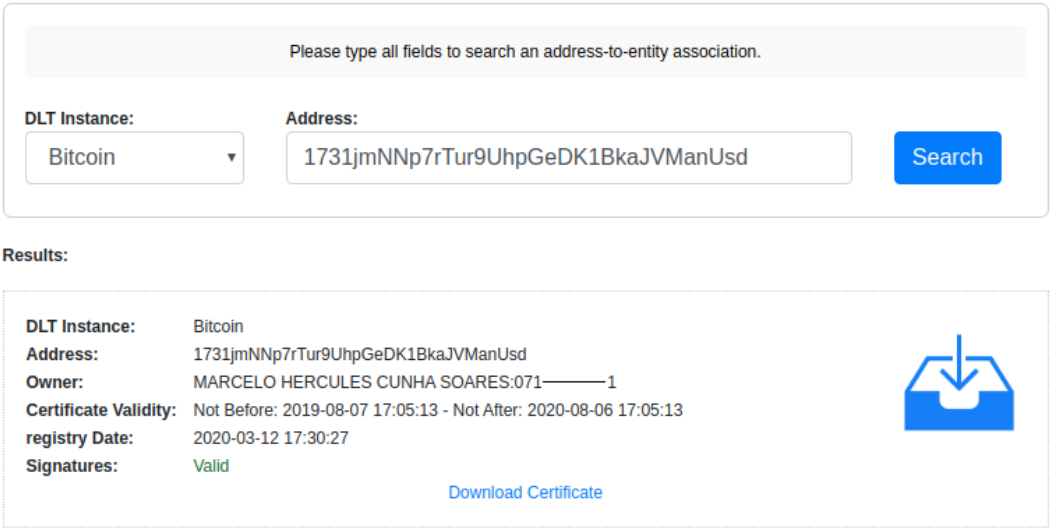

Figura 4.14: Protótipo: Interface interativa do ANS Server

Fonte: Próprio autor

utilizadas para o armazenamento de dados arbitrários, armazenam valores de *hash* de arquivos no livro-razão ao invés dos *bytes* dos arquivos, uma vez que pode sair extremamente caro armazenar arquivos diretamente em DLTs. Durante o estudo das DLTs públicas, percebeu-se que diversos projetos desenvolvidos para a plataforma Ethereum utilizam uma tecnologia de armazenamento distribuído denominada *InterPlanetary File System* (IPFS).

O IPFS é um sistema de arquivos distribuídos *peer-to-peer*, cujo o arquivo é endereçável de acordo com o seu conteúdo, isto é, um *hash* do conteúdo do arquivo é gerado para servir como um índice para a recuperação deste arquivo, garantindo assim, que cada arquivo é único, já que qualquer alteração em um arquivo, acarretaria em um novo *hash* (Benet, 2014). O *hash* para endereçamento de arquivos no IPFS é chamado de *Content Identifier* (CID). O IPFS é baseado na ideia de participação, onde uma comunidade possui e compartilha arquivos de outros para torná-los disponíveis. Cada nó armazena apenas conteúdo de seu interesse, e pode manter um *cache* de determinados arquivos. A figura 4.15 mostra o monitoramento de nós conectados na rede IPFS.

Uma tecnologia como o IPFS para o armazenamento de ANS Certificates, proporciona uma maior efetividade do ANS já que a disponibilidade dos artefatos é relativamente maior em comparação a um repositório centralizado. Quando um nó IPFS é executado como um *daemon*, uma API é disponibilizada para a interação com o nó. Através da API, é possível

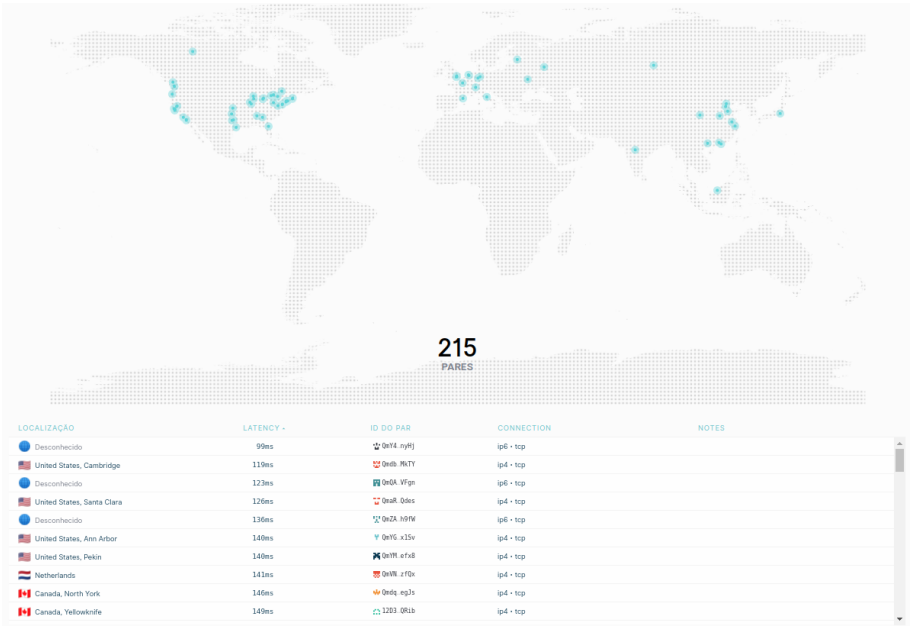

Figura 4.15: Rede distribuída do IPFS

Fonte: *Print screen* da aplicação web console do IPFS

utilizar o IPFS diretamente de dentro de uma aplicação própria, como no caso do ANS Server.

Conforme mencionado, o IPFS não recupera um artefato armazenado com base em um caminho. Em vez disso, o IPFS endereça um arquivo de acordo com o seu conteúdo, através de um identificador composto por um *hash* e o identificador do algoritmo usado. A instalação padrão do software IPFS contém uma aplicação web que se comunica com a API e serve como um *gateway* para a obtenção de arquivos sem a necessidade de implementar uma aplicação cliente. Um exemplo de recuperação de conteúdo através do *gateway* pode ser visto na seguinte URL: `http://ipfs.io/ipfs/QmbDWHHDA49rRfvAH2p6Y6xzsWLWCFsi32qwgRvQprF8Je`. A string após o parâmetro /ipfs/ representa o CID de um ANS Certificate armazenado na rede.

Para a comunicação do ANS Server com a API do IPFS, foi utilizada uma biblioteca Java desenvolvida para tal finalidade. A biblioteca *java-ipfs-http-client*[8] contém classes que encapsulam comandos do próprio IPFS que podem ser invocados através de métodos. Ao instanciar a classe IPFS da biblioteca *java-ipfs-http-client*, é necessário passar um *endpoint* de uma API disponível em um nó IPFS. Uma vez instanciada a classe IPFS, é possível

[8]https://github.com/ipfs/java-ipfs-http-client

adicionar conteúdo à rede através do comando IPFS *add*, invocado pelo método de mesmo nome, ou mesmo realizar a leitura dos *bytes* de um arquivo através do comando *cat*, também implementado como um método da classe IPFS. Tais métodos são invocados no momento de registro de uma associação entidade-endereço (método *add*) como também na consulta (método *cat*), conforme descrito na subseção 4.3.3. A figura 4.16 ilustra a comunicação entre os sistemas ANS Server, a base de dados e o IPFS para a obtenção de um ANS Certificate.

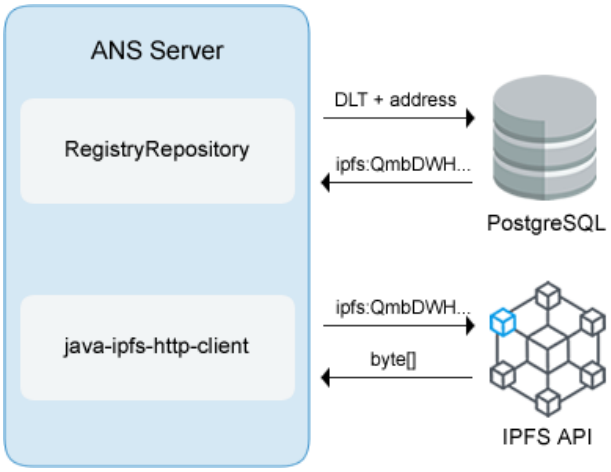

Figura 4.16: Recuperação de ANS Certificate

Fonte: Próprio autor

4.3.5 ANS Oracle

Até este ponto, o protótipo do ANS Server funciona sobre uma rede, seja ela privada, ou mesmo uma rede pública como a internet. No entanto, há uma categoria de aplicações escritas em linguagens específicas e que executam dentro das próprias DLTs em contratos inteligentes, cujo a problemática exposta neste trabalho também é comum a esta categoria. Em alguns casos, há a necessidade de que tais aplicações necessitem da identificação do proprietário por trás de um endereço, como no caso do BNDESToken (Júnior et al., 2018). Portanto, faz-se necessário um meio de que esta classe de aplicações, denominadas *dApp* (do inglês *Decentralized Application*), consigam ser clientes no ANS.

Como a camada do ANS responsável por prover um serviço de consulta de associação está em uma rede fora das DLTs, isto é, os dados estão *off-chain*, é necessário a interação das dApps com dados do "mundo"externo às DLTs. Este requisito pode ser atendido através

de um **oráculo**, isto é, um contrato inteligente cujo a finalidade é obter dados externos e servir a outros contratos. O componente **ANS Oracle** foi escrito com a linguagem *Solidity*[9] e compilado na versão 0.5.0.

Diferentemente da Bitcoin, onde dados arbitrários são inseridos através da exploração de campos designados para outros fins, na plataforma Ethereum, é possível armazenar dados permanentes em uma área associada a um contrato inteligente, denominada *storage*. Para armazenar os dados de uma associação entidade-endereço foi utilizada a estrutura de dados *mapping(KeyType => ValueType)* que pode ser vista como uma *hashTable* com chaves que mapeiam para valores. No momento em que o ANS Oracle é populado, as chaves do *mapping* são preenchidas com o endereço em questão e para o valor foi utilizado um tipo de variável personalizado, chamado *struct*, que corresponde a uma estrutura de dados a ser definida, conforme pode ser visto nas linhas 8-12 do código-fonte exibido na figura 4.17. A *struct* foi criada para armazenar os seguintes valores: *index*, *entity* e *ansCertificateCID* que representam um atributo único da entidade contido no certificado digital, uma descrição da entidade e o CID do ANS Certificate no IPFS, respectivamente.

Para popular o contrato do ANS Oracle, é necessário uma camada externa para intermediar a consulta à uma API e a criação de uma nova transação para alterar o estado do contrato e alimentar o oráculo. Este componente denominado *ANS Listener* é uma aplicação desenvolvida em *javascript* executada como um serviço *nodejs* conectada a um nó da rede Ethereum através de uma conexão por *web-socket*. A biblioteca javascript *web3*[10] foi utilizada na escrita dos métodos que interagem com o ANS Oracle. O fluxo para a utilização do ANS por uma dAPP se inicia com um contrato inteligente solicitando ao ANS Oracle por uma associação para um determinado endereço. O ANS Oracle busca em sua *hashTable* e caso não encontre, emite um evento que está sendo escutado pelo ANS Listener, que por sua vez realiza uma requisição para o serviço de consulta da API *RESTful* do ANS Server repassando os parâmetros necessários para a consulta. Caso o ANS Server encontre um registro de associação e retorne os respectivos dados, o ANS Listener cria uma transação e chama a função *register* do ANS Oracle que por sua vez registra em sua *hashTable*. Este

[9] Linguagem de programação de alto nível, orientada a objetos e tipada, para a implementação de contratos inteligentes. Site: `https://solidity.readthedocs.io/en/v0.6.4/`

[10] https://web3js.readthedocs.io/

```
pragma solidity ^0.5.0;

contract ANSOracle {

    /**
     * struct to represent an Entity
     */
    struct Entity{
        uint64 index;
        string entity;
        string ansCertificateCID;
    }

    /**
     * mapping to store associations. The key is an index (eg: cpf or cnpj), the value is an Entity struct
     */
    mapping(address => Entity) public addressToEntity;

    /**
     * Array to store all address registered. TODO Check if it is usable
     */
    address[] public entityAccts;

    event EntityRequest(address _address);
    event AddressRegistered(string _entityDescription);

    /**
     * ANS Listerner is listening this function
     */
    function searchEntityByAddress(address _address) public returns (uint, string memory, string memory) {
      if(addressToEntity[_address].index != 0){
        return (addressToEntity[_address].index, addressToEntity[_address].entity, addressToEntity[_address].ansCertificateCID);
      }else{
        emit EntityRequest(_address);
        return (0 , "", "");
      }
    }

    /**
     * Return an entity by an address
     */
    function getEntityByAddress(address _address) view public returns (uint, string memory, string memory){
        return (addressToEntity[_address].index, addressToEntity[_address].entity, addressToEntity[_address].ansCertificateCID);
    }

    /**
     * Function to register an association
     */
    function register(address _address, uint64 _index, string memory _entityDescription, string memory _ansCertificateCID)
                public returns (string memory){
        Entity storage entity = addressToEntity[_address];

        entity.index = _index;
        entity.entity = _entityDescription;
        entity.ansCertificateCID = _ansCertificateCID;

        entityAccts.push(_address) -1;

        emit AddressRegistered(_entityDescription);
        return _entityDescription;
    }

}
```

Figura 4.17: Código-fonte do ANS Oracle em Solidity

Fonte: Próprio autor

fluxo representado no diagrama de sequência na figura 4.18.

Ainda que o ANS Oracle armazene em seu *mapping* o CID do ANS Certificate no repositório distribuído, as linguagens de programação de contratos em Ethereum não fornecem suporte para a codificação de rotinas de validação de assinaturas digitais em documentos XML (XAdES). Neste caso, a confiança da associação segura é baseada, em parte, na validação das assinaturas digitais realizada no ANS Server, antes da resposta para o ANS Listener. Conforme mencionado, o ANS Oracle mantém uma referência para uma evidência externa que permita a auditabilidade completa da associação, uma vez que qualquer usuário de dApps pode obter um ANS Certificate no repositório distribuído e realizar a sua validação. Após o

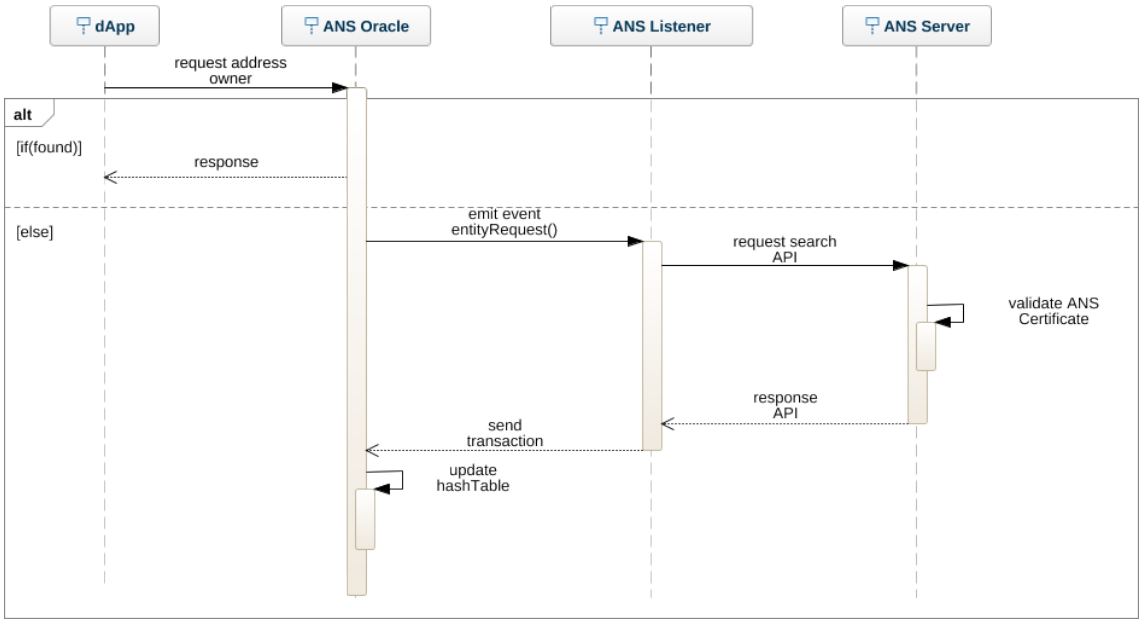

Figura 4.18: Fluxo de alimentação de um ANS Oracle

Fonte: Próprio autor

fluxo completo, o ANS Oracle cumpre o seu papel de fornecer dados de uma fonte externa à DLT, podendo servir a outras dApps clientes, permitindo também a utilização do ANS com aplicações descentralizadas escritas em contratos inteligentes. Embora a primeira versão do protótipo do ANS tenha considerado a Ethereum para a prova de conceito, outras tecnologias com suporte a contratos inteligentes poderiam ser integradas ao ANS.

Capítulo 5

Experimentos

A realização dos experimentos foi feita em dois cenários. Primeiramente foram feitos experimentos em um ambiente local, com utilização do ANS Client para a geração do ANS Certificate, uma instância de um ANS Server executada em um servidor de aplicações local e uma DLT Ethereum inicializada localmente sem conexão com outros nós. Em um segundo caso, foram feitos experimentos em instâncias públicas reais das DLTs Bitcoin e Ethereum. Os experimentos contemplam a integração do ANS com uma gama de aplicações, como: *Block Explorers*, Aplicações Web e Aplicações Descentralizadas (dApps). Em ambos os cenários, foi utilizado um certificado digital real emitido por uma Autoridade Certificadora da ICP-Brasil com o valor de *Subject Name*: MARCELO HERCULES CUNHA SOARES:071XXXXXXX1 [1]. Outros detalhes de ambiente são descritos nas subseções seguintes referentes a cada experimento.

5.1 Integração em um ambiente local

5.1.1 Preparação do ambiente

Para a execução dos experimentos em um ambiente com uma DLT em execução localmente, foi utilizada a implementação de *blockchain* Go Ethereum, que corresponde a uma das 3 implementações do protocolo Ethereum, escrita em linguagem Go. Para utilizar o Go Ethereum é necessário instalar o seu aplicativo cliente autônomo *geth*. O *geth* é uma ferramenta em

[1] Os caracteres "X"ocultam um número de CPF real.

linha de comando que executa um nó Ethereum completo na máquina, fornecendo funções em linha de comando para a execução de transações e disponibilizando também um servidor para chamadas de procedimento remoto (RPC).

5.1.2 Registro no ANS Server

Inicialmente foi necessário criar endereços de carteiras para serem usados nas transações. Foram criados dois endereços através da plataforma MEW[2]. O endereço `0xcd2a3d9f938e13cd947ec05abc7fe734df8dd826` foi importado na instância local da *blockchain* Ethereum através do console do *geth*. O ANS Client foi executado para realizar o procedimento de geração e registro do ANS Certificate descrito na seção 4.3.2, associando o endereço importado ao certificado digital de *Subject Name*: MARCELO HERCULES CUNHA SOARES:071XXXXXX41.

Neste caso, foi feita uma transação entre o endereço de coinbase[3] criado pelo próprio geth como o endereço de origem o endereço `0xcd2a3d9f938e13cd947ec05abc7fe734df8dd826` como o endereço de destino da transação. A transação foi adicionada a um bloco que por sua vez foi minerado e adicionado à cadeia.

5.2 Integração com *Block Explorer*

Para facilitar a visualização da identidade dos usuários envolvidos no contexto de uma determinada aplicação que utiliza de uma DLT, foi feita a integração do ANS com uma ferramenta de *Block Explorer*. Um *Block Explorer* é uma ferramenta que fornece informações sobre blocos, transações e endereços de DLTs, desde uma visão mais macro (últimos blocos minerados), até uma visão mais detalhada (transações de um determinado endereço).

[2]https://www.myetherwallet.com

[3]Primeira conta criada por padrão durante a instalação do geth.

5.2.1 *Block Explorer* para Ethereum

Para a integração com o ANS em uma plataforma Ethereum, foi utilizado o *Block Explorer* de código aberto *Ethnamed Block Explorer*[4], conectado a *blockchain* local da Ethereum através das interfaces de conexão RPC da biblioteca javascript *web3*. O código da ferramenta foi customizado para fazer chamadas ao serviço de consulta do ANS com o objetivo de obter informações da entidade proprietária do endereço envolvido na transação. Na tela inicial do *Block Explorer* é possível visualizar os últimos blocos minerados, isto é, adicionados à cadeia. Dentro de cada bloco é possível visualizar os dados das transações contidas. A maneira tradicional de visualização de dados de transações em tais ferramentas é a exibição do endereço de carteira/contrato de origem, o valor do ativo transacionado, e o endereço de carteira/contrato de destino. Para cada transação contida dentro de um bloco, o *Block Explorer* faz uma chamada à API *RESTful* do ANS Server, passando os parâmetros necessários para o serviço de consulta, conforme descrito na seção 4.3.3. Caso haja uma ocorrência de registro do endereço consultado no ANS Server, o mesmo retorna uma resposta em formato *json* para o *Block Explorer*, com os dados do detentor do endereço, incluindo o CID para acessar o respectivo ANS Certificate.

O *Block Explorer* por sua vez exibe ao lado do endereço, caso haja um retorno do ANS Server, o atributo *Subject Name* do certificado digital contido no ANS Certificate e utilizado para a prova de identidade. Como é possível visualizar na Figura 5.1, o usuário tem a possibilidade de fazer o *download* do ANS Certificate, caso deseje escrever o seu próprio mecanismo de validação, ou mesmo fazer o *download* do Certificado Digital X.509 contido no ANS Certificate para verificar as informações do detentor do endereço, como também informações sobre a Autoridade Certificadora responsável por assinar o certificado.

Uma observação importante é que, neste caso, apenas uma das contas possuía uma declaração de posse registrada, isto é, enquanto um participante da transação permite a sua identificação, o outro participante permaneceu de forma anônima, conforme pode ser visto na figura 5.2.

[4]https://github.com/web3space/eth-explorer

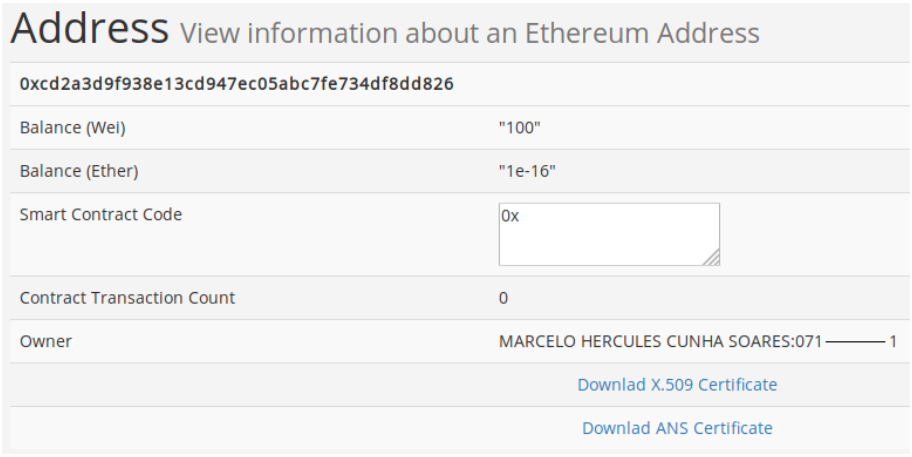

Figura 5.1: Visualização de dados do endereço de carteira no *Block Explorer*

Fonte: Próprio autor

Figura 5.2: Visualização de transação no *Block Explorer*

Fonte: Próprio autor

5.2.2 *Block Explorer* para Bitcoin

Para o experimento de integração do ANS com um Block Explorer para Bitcoin, foi realizada a integração com a ferramenta de código aberto *BTC RPC Explorer*[5]. Para este experimento, o endereço de carteira `mxfAw8LFPYnVWK4WfGFoGzW3Um4tDA7Yjj` através do software de carteira BitPay, na rede pública de testes da bitcoin, denomiada *testnet*. Os procedimentos de geração e registro do ANS Certificate foram realizados através dos componentes do ANS descritos na seção 4.3. Para a visualização de informações associadas ao endereço, é necessário que pelo menos uma transação tenha sido escrita no

[5]https://github.com/horizontalsystems/bitcoin-block-explorer

ledger. Para este fim, uma transação[6] foi realizada através de uma plataforma de de distribuição de frações de bitcoins da rede testnet. Sites de distribuição de bitcoins em redes de testes são popularmente chamados de *faucets*. Para este experimento, foi utilizado o `https://bitcoinfaucet.uo1.net/`. Redes de testes são amplamente utilizadas para fins de desenvolvimento e experimentos pois evitam os custos de transações das redes principais.

Diferentemente do experimento na rede Ethereum, cujo uma *blockchain* foi inicializada localmente, para o experimento com a Bitcoin, foi necessária a inicialização de um nó sincronizado com os dados da rede de testes. Para esta sincronização, foi realizada a instalação do *Bitcoin Core* e seus utilitários: **bitcoin-cli** *(Bitcoin command line interface)* e o *bitcoind*, que executa o protocolo Bitcoin como um *daemon*. As configurações foram setadas em um arquivo de configuração **bitcoin.conf** para indicar a rede de testes e o nó foi inicializado como um *daemon* com o comando: **bitcoind -testnet -daemon**. A sincronização completa do *ledger* com cerca de 24GB levou aproximadamente 5 horas. Uma vez com o nó bitcoin em execução, o arquivo de configuração do *BTC RPC Explorer* foi alterado com a inserção dos dados para o estabelecimento da conexão RPC com o nó Bitcoin. O código-fonte da aplicação foi alterado para que as requisições de consulta ao ANS Server fossem realizadas em determinados pontos do *Block Explorer*. O procedimento de geração e registro do ANS Certificate descrito na seção 4.3 foi executado, com a prova de posse do endereço `mxfAw8LFPYnVWK4WfGFoGzW3Um4tDA7Yjj` associada à prova de identidade do sujeito identificado no certificado digital real emitido por uma AC ICP-Brasil, conforme descrito no início deste capítulo. A figura 5.3 mostra a tela de visualização de dados de endereços no *BTC RPC Explorer*. O retorno da consulta ao ANS Server é exibido ao lado da label **Owner** com os links para *download* do ANS Certificate ou do certificado X.509, como foi feito no experimento do *block explorer* para Ethereum.

Ao visualizar a tela de detalhes de uma transação, é possível perceber que foi realizada a partir de um endereço para 2 endereços destinatários (seção 1Input, 2Outputs), conforme mostrado na figura 5.4. Na plataforma Bitcoin, isso é comumente feito quando o valor a ser

[6]Hash: `ad94a0506fee0c94b087768a6634c450344b79dde2e35ec36dbb6ab2e6eb8d96`. A transação pode ser visualizada em qualquer *Block Explorer* conectado a uma rede de testes, como por exemplo o `https://live.blockcypher.com/`.

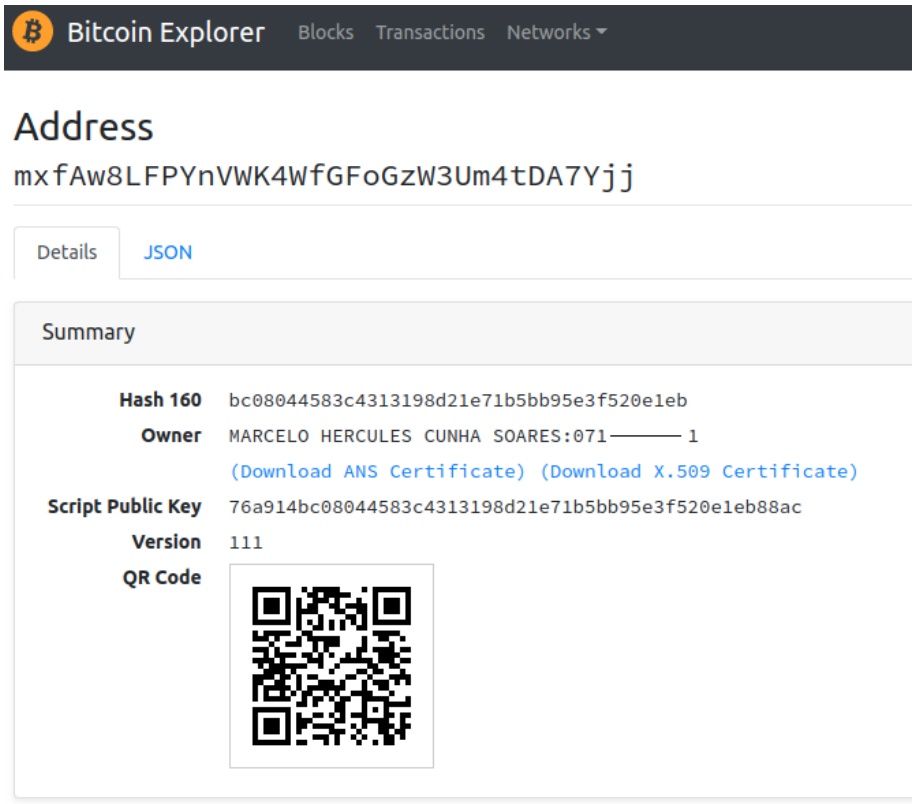

Figura 5.3: Visualização de dados de endereço no *BTC RPC Explorer*

Fonte: Próprio autor

transacionado é menor que o saldo referenciado na transação de entrada, sendo necessário que uma segunda saída seja especificada para o troco, que geralmente é um endereço de propriedade do próprio emissor da transação. É possível perceber que a transação envolve dois atores. Um emissor, que nesse contexto permanece anônimo, e um receptor que possui um registro de reivindicação de posse do endereço em um servidor ANS, com a respectiva prova armazenada em um repositório distribuído. Esse cenário demonstra mais uma vez que a integração com o ANS não implica na quebra generalizada do anonimato no escopo da rede. Isto é, os atores que desejam o anonimato não são afetados pela integração com o ANS.

5.3 Integração com RAP/SIGAA

A aplicação de registro, autenticação e preservação de documentos digitais, denominada **RAP**, desenvolvida no Laboratório de Aplicações de Vídeo Digital (LAVID) da Universi-

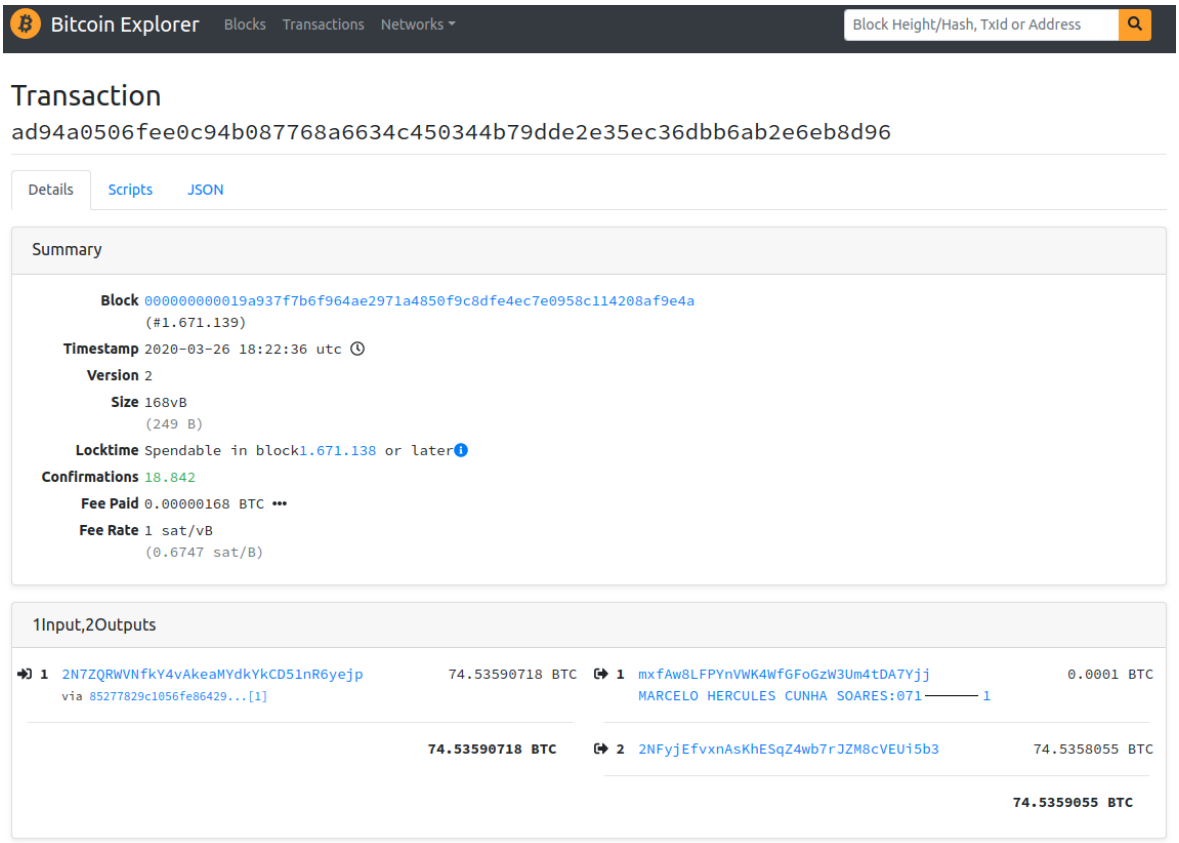

Figura 5.4: Visualização de dados de transação no *BTC RPC Explorer*

Fonte: Próprio autor

dade Federal da Paraíba foi utilizada pela Superintendência de Tecnologia da Informação (STI) da UFPB em um projeto piloto para a emissão de diplomas digitais para alunos concluintes dos cursos de Ciências da Computação e Engenharia da Computação[7]. O serviço RAP, ao receber uma solicitação de registro de documento, além de registrar o documento em sua respectiva tecnologia de preservação, também realiza o registro de uma representação resumida em uma tecnologia de livro-razão distribuído, neste caso, a DLT pública da plataforma Ethereum. Com o registro na *blockchain* da Ethereum, é possível garantir que o documento existia naquele determinado momento em que foi registrado, como também, a

[7] http://g1.globo.com/pb/paraiba/bom-dia-pb/videos/t/edicoes/v/estudantes-da-ufpb-substituem-o-diploma-de-papel-por-diploma-digital/7558389/

sua integridade pode ser verificada por qualquer parte que possua o documento original[8].

O RAP é um exemplo de aplicação que utiliza de uma DLT pública e pode ser aplicada em contextos com a finalidade de prover maior segurança para documentos eletrônicos e, consequentemente, evitar fraudes, como no caso dos diplomas digitais. Isto implica que as informações apresentadas pelo RAP aos usuários sejam confiáveis. A identificação do ator remetente da transação de registro de diplomas se faz necessária para que as informações fornecidas pelo serviço sejam comprovadamente verdadeiras e verificáveis. No ponto de vista de arquitetura de sistemas, no projeto piloto mencionado, o cliente do serviço RAP é o Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA). O SIGAA é o sistema utilizado pela UFPB para a gestão acadêmica, o que inclui a gestão dos diplomas emitidos pela instituição.

No portal externo do SIGAA há um link para a autenticação de diplomas digitais, com um formulário para que o usuário realize o *upload* do documento a ser verificado. No momento da verificação, o SIGAA faz uma chamada ao serviço RAP através de uma API RESTful, que por sua vez retorna os dados relativos ao registro do diploma no serviço, como também os dados relativos ao registro na DLT, como o *hash* da transação realizada na *blockchain*. Após o retorno dos dados de registro pelo RAP, o SIGAA realiza uma chamada ao ANS Server, passando como parâmetros o identificador da DLT Ethereum e o endereço de carteira utilizado na transação. O ANS Server por sua vez, retorna os dados registrados ao SIGAA, uma vez que uma associação foi encontrada, e por sua vez o SIGAA os exibe em tela para o usuário, conforme pode ser visto na figura 5.5.

Para este experimento, foi gerado um ANS Certificate assinado com chave privada associada ao endereço `0x6033341f9A8b019D82aae3e8F265a5e8CeB61c28` para a prova de posse. Como não foi possível o acesso à chave e certificado digital de pessoa jurídica da UFPB, a assinatura para a prova de identidade foi realizada com a chave privada associada ao certificado real ICP Brasil descrito no início deste capítulo. Ainda que o link para *download* do ANS Certificate tenha sido disponibilizado, foi feita uma validação da assinatura perante à infraestrutura de chaves públicas que emitiu as chaves para a prova de identidade, neste caso, a ICP Brasil. O Instituto Nacional de Tecnologia da Informação (ITI),

[8]Um exemplo de transação com um registro de um diploma pode ser visto em `https://etherscan.io/tx/0xf9d9a7ba103061ee978c0a74980d6fd9bf21f572317838c69c3fa3d692f0a315`

Figura 5.5: Interface para validação de diplomas digitais no SIGAA

Fonte: *Printscreen* do SIGAA

órgão responsável por manter e executar as políticas da ICP Brasil, disponibiliza um verificador de conformidade de assinatura digital[9] para que documentos assinados em padrões CAdES, XAdES e PAdES possam ser validados. O ANS Certificate é então recuperado do ANS Repository e enviado em uma requisição ao validador do ITI. Até o momento em que este experimento foi realizado, o ITI não disponibilizada um serviço para consultas ao validador através de padrões bem definidos de consumos de *web-services*, como o *REST* ou *SOAP*. Isto é, ao enviar um documento para validação, um relatório da validação é exibido em uma página HTML como resposta. A validação do ANS Certificate através do formulário disponibilizado pelo ITI pode ser vista na fitura 5.6.

A solução encontrada foi enviar uma requisição HTTP para a rota de validação do ve-

[9]O validador de conformidade pode ser acessado em `https://verificador.iti.gov.br/verifier-2.5.1/`

Figura 5.6: Interface para validação de assinaturas digitais ICP Brasil

Fonte: *Printscreen* do Verificador de Conformidade do ITI

rificador do ITI, com tratamento da estrutura HTML da resposta para a obtenção do status do documento. O resultado da verificação é exibido ao lado da label **Validade ICP Brasil** conforme pode ser visto na figura 5.5. Este experimento foi realizado no ambiente de testes da Superintendência de Tecnologia da Informação da UFPB, para que possíveis ocorrências durante o experimento não interferissem no funcionamento do sistema no ambiente de produção.

5.4 Integração *on chain*

Conforme mostrado na seção 4.3, o ANS foi pensado para que possa ser integrado com aplicações descentralizadas (dApps). Para este requisito, é necessária a utilização do componente ANS Oracle, a fim de popular os dados *on chain* com dados oriundos do escopo externo à rede, providos pelo ANS Server. Este experimento foi dividido em duas etapas. Na primeira etapa foi realizada a integração com um contrato inteligente simples, cujo o papel é representar uma dApp cliente do ANS Oracle. Uma *blockchain* da Ethereum foi inicializada localmente para este experimento e interações com o ANS Oracle foram feitas a partir da

IDE *remix* [10]. O *remix* permite estabelecer uma conexão com uma rede Ethereum, seja uma rede local ou remota, como também, fornece uma interface amigável provendo uma interface com os *inputs* necessários para interagir com contratos inteligentes. A figura 5.7 mostra o retorno de uma consulta à função *getEntityByAddress* ao ANS Oracle realizada na interface do Remix com o seu respectivo retorno.

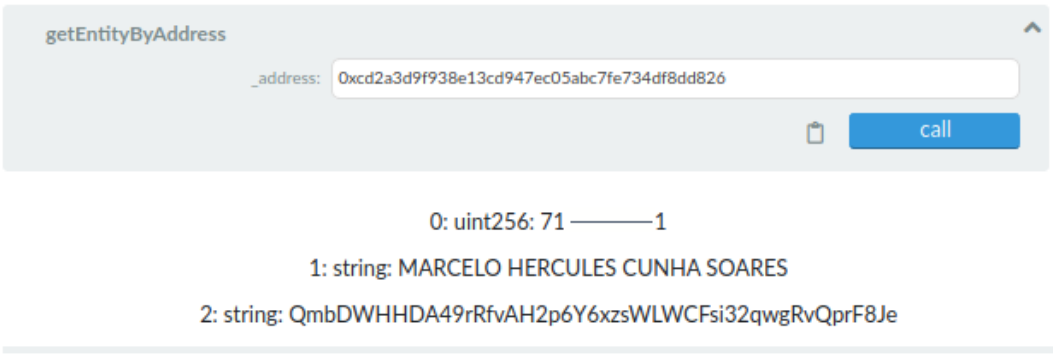

Figura 5.7: Retorno de consulta no ANS Oracle na IDE Remix

Fonte: Próprio autor

Para iniciar o experimento, foi necessário realizar o deploy do ANS Oracle em uma rede Ethereum. Um código utilitário foi escrito em javascript com *nodejs* para o deploy do contrato utilizando a biblioteca *solc.js*, que fornece uma API para as operações do compilador solidity. A estratégia adotada para a alimentação do ANS Oracle é através da emissão de um evento na dApp que é escutado pelo ANS Listener, através de uma conexão *websocket* entre a aplicação e o nó que executa o *geth*, neste caso, a própria máquina local. O ANS Listener entra em ação quando o evento *EntityRequest* é disparado. O *listener* por sua vez consulta o ANS Server passando como parâmetro o identificador da rede Ethereum e o endereço enviado no evento. Após a resposta do ANS Server, caso uma ocorrência seja encontrada, o *listener* recupera e faz a validação do ANS Certificate no verificador de conformidade do ITI, seguindo a mesma estratégia descrita na subseção anterior. Em seguida, executa uma transação para o ANS Oracle passando como parâmetros necessários os dados retornados do ANS Server. A utilização do ANS Oracle por dApps clientes se dá através da importação do ANS Oracle nos contratos inteligentes. A figura 5.8 mostra o código de uma dApp que recebe o endereço do ANS Oracle e cria uma referência em uma variável do contrato. Desse modo, é

[10] https://remix.ethereum.org/

possível realizar chamadas a funções do ANS Oracle fazendo invocações pelos respectivos nomes na variável que contém a referência.

```
pragma solidity ^0.5.0;

import './ANSOracle.sol';

contract ANSOracleClient {

    ANSOracle oracle;

    constructor(ANSOracle _oracle) public{

        oracle = ANSOracle(_oracle);

    }

    /**
     * Returns an entity CNPJ by an address
     */
    function getEntityIndex(address _address) view public returns (uint64){
        uint64 cnpj = oracle.getEntityIndex(_address);
        return cnpj;
    }

}

```

Figura 5.8: Exemplo de uma dApp simples consumindo o ANS Oracle

Fonte: Próprio autor

5.4.1 Integração com BNDESToken

O BNDES disponibiliza em seu repositório no GitHub[11], o código-fonte dos contratos inteligentes pertencentes ao projeto BNDESToken. Para avaliar mais profundamente a viabilidade do ANS Oracle, foi realizado também um experimento de integração com o BNDESToken. Pela falta de documentação no repositório público do projeto, o entendimento do código-fonte foi baseado na interpretação de nomes de variáveis e funções, na leitura de comentários e documentação a nível de código, como também da análise da relação entre os contratos existentes. Os seguintes contratos foram obtidos e implantados: *BNDESRegistry* e *BNDESToken*. Com uma análise do código, percebeu-se que o contrato BNDESRegistry atua como um contrato auxiliar, provendo funcionalidades e dados ao contrato BNDESToken. Associações entre CNPJs e endereços são armazenados em uma variável do tipo *mapping*, que chaveia de endereços (tipo *address*) para uma *struct* definida e chamada *LegalEntityInfo*. A alimentação dessas associações é feita na função *registryLegalEntity*. O contrato BNDESRe-

[11] https://github.com/bndes/bndestoken

gistry foi alterado para receber uma referência do ANS Oracle em seu construtor, semelhante ao que foi mostrado na figura 5.8, como também o método *registryLegalEntity* foi alterado para realizar uma consulta a uma instância do ANS Oracle, como pode ser visto na figura 5.9, mais precisamente nas linhas 198 à 204.

```
190     function registryLegalEntity(uint64 cnpj, uint64 idFinancialSupportAgreement, uint32 salic,
191         address addr, string memory idProofHash) onlyTokenAddress public {
192
193         // Endereço não pode ter sido cadastrado anteriormente
194         require (isAvailableAccount(addr), "Endereço não pode ter sido cadastrado anteriormente");
195
196         require (isValidHash(idProofHash), "O hash da declaração é inválido");
197
198         uint64 cnpjRetorno = oracle.getEntityIndex(addr);
199
200         bool isICPBrasil = (cnpjRetorno != 0) ? true : false;
201
202         require(isICPBrasil, "Registro não encontrado no ANS Oracle");
203
204         require(cnpj == cnpjRetorno, "CNPJ não confere com o do e-CNPJ");
205
206         legalEntitiesInfo[addr] = LegalEntityInfo(cnpj, idFinancialSupportAgreement, salic,
207                                                   idProofHash, BlockchainAccountState.WAITING_VALIDATION);
208
209         // Não pode haver outro endereço cadastrado para esse mesmo subcrédito
210         if (idFinancialSupportAgreement > 0) {
211             address account = getBlockchainAccount(cnpj,idFinancialSupportAgreement);
212             require (isAvailableAccount(account), "Cliente já está associado a outro endereço. Use a função Troca.");
213         }
214         else {
215             address account = getBlockchainAccount(cnpj,0);
216             require (isAvailableAccount(account), "Fornecedor já está associado a outro endereço. Use a função Troca.");
217         }
218
219         cnpjFSAddr[cnpj][idFinancialSupportAgreement] = addr;
220
221         emit AccountRegistration(addr, cnpj, idFinancialSupportAgreement, salic, idProofHash);
222     }
```

Figura 5.9: Integração entre BNDESToken e ANS Oracle

Fonte: Adaptação de código fonte encontrado em

`https://github.com/bndes/bndestoken`

Uma vez que o ANS Oracle está populado com associações cujo a identidade está representada em um certificado digital validado pela ICP Brasil, o BNDESRegistry só conseguirá popular a sua tabela de associações endereços retornados pelo ANS Oracle, e o BNDESToken por consultar esta associação, só será transferido entre contas cujo a identidade dos proprietários são representadas em certificados do tipo e-CNPJ, conforme requisito apontado no projeto. Este experimento buscou mostrar a utilização do ANS em um contexto *on-chain* através da integração do ANS Oracle com aplicações descentralizadas. Ainda que a integração com o BNDESToken tenha sido realizada com base na percepção do código, o experimento obteve êxito na validação da interação entre os componentes e mostrou que o ANS Oracle é funcional.

5.5 Integração com DNSLink

É comum encontrar páginas web que exibam endereços de carteira. O ANS nestes casos pode ser utilizado para relacionar o endereço de carteira à mesma entidade proprietária do website. Um experimento foi realizado com a utilização do *DNSLink*, um protocolo que aproveita a infraestrutura distribuída do DNS para vincular conteúdos e serviços[12]. De acordo com a documentação do DNSLink, sugere-se a adição de uma entrada do tipo TXT para o sudbomínio **_dnslink** com o seguinte conteúdo: `dnslink=/ipfs/:contentID`. Quando um cliente IPFS tenta resolver um endereço, esta entrada é buscada para a leitura do arquivo referenciado pelo :contentID. Para este experimento, a seguinte entrada do tipo TXT foi adicionada para o subdomínio `ans.meudominio.com.br` com o CID do ANS Certificate: `dnslink=/ipfs/QmbDWHHDA49rRfvAH2p6Y6xzsWLWCFsi32qwgRvQprF8Je`. Uma entrada do tipo CNAME também foi inserida resolvendo o subdomínio `ans.meudominio.com.br` para o gateway IPFS de recuperação de arquivos (`http://ipfs.io/`).

Ao acessar a URL `ans.meudominio.com.br`, o *gateway* IPFS responde devido à entrada CNAME, e busca pelo valor da entrada TXT conforme descrito anteriormente, para então devolver o ANS Certificate ao usuário. O experimento com o *DNSLink* foi realizado até este ponto. Entretanto, *softwares* de carteira poderiam ser adaptados para realizarem transações tendo um padrão de subdomínio preenchido como campo de destinatário. Internamente, a obtenção do ANS Certificate seria realizada conforme descrito, e o endereço poderia ser obtido do próprio ANS Certificate. Abordagem esta, semelhante à utilizada no BIP 70, porém com a vantagem do aproveitamento de uma tecnologia existente como o DNS para o encaminhamento até um ANS Certificate, sem a necessidade de que os usuários precisassem implementar um código *server-side* para atender requisições, como ocorre com o BIP 70. A figura 5.10 ilustra em um diagrama de sequência a proposta de utilização do ANS com *DNSLink* por uma aplicação cliente.

[12] https://dnslink.io/

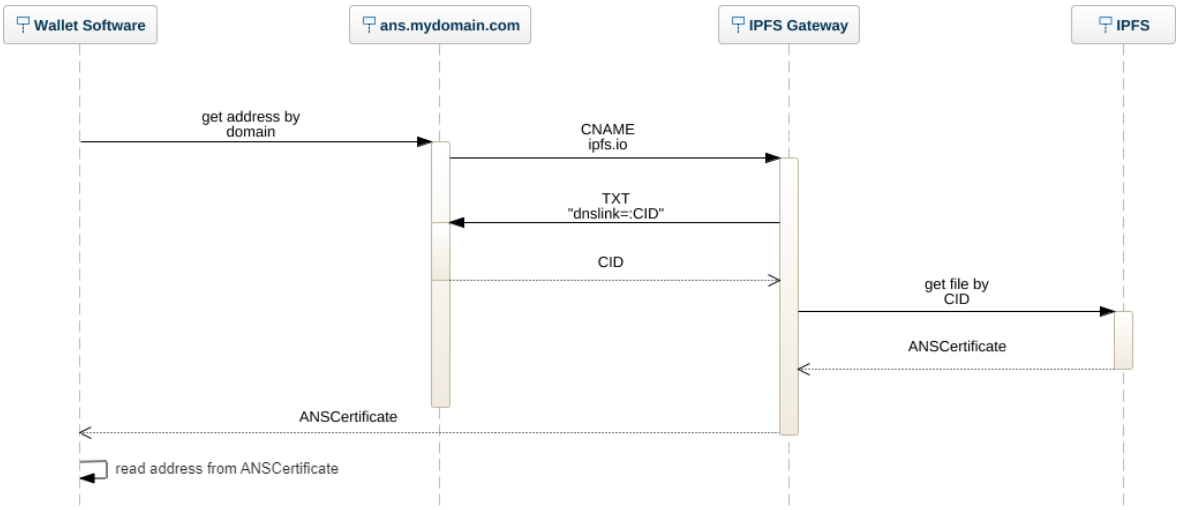

Figura 5.10: Utilização do ANS com o *DNSLink*

Fonte: Próprio autor

5.6 Considerações Finais

Esta seção apresentou a validação do ANS através de sua utilização em cenários de integração com aplicações reais. A integração ocorreu de forma pacífica em todos os cenários, uma vez que é realizada com um método consolidado de comunicação entre sistemas. O consumo a uma API RESTful é uma operação comum no âmbito de sistemas distribuídos. Portanto, tal operação pode ser implementada sem maiores dificuldades por aplicações escritas em diferentes linguagens de programação, assim como ocorreu com os experimentos realizados.

Em comparação com as abordagens relacionadas existentes, percebe-se algumas vantagens em relação à utilização do ANS. Basicamente, as abordagens citadas no capítulo 3 fornecem apenas um protocolo semelhante ao que faz o HTTPS para a internet, sendo diferente da abordagem da ANS que extrapola o contexto da sessão de operação. Por exemplo, o ANS permite a integração com ferramentas de exploração de bloco, permitindo a visualização de todos os envolvidos em transações de um determinado aplicativo baseado em DLT que exige transparência.

As abordagens elencadas no capítulo 3 permitem a identificação de destinatários de transações, mas não identificam endereços de remetentes, como a ANS. Além disso, o ANS é menos invasivo e mais dissociado, não exigindo a implementação de um servidor para cada

usuário. O uso combinado de chave de certificado digital e assinaturas de chaves ECDSA não apenas proporciona maior segurança, mas também uma associação confiável com entidades do mundo real. É importante mencionar que o caráter voluntário e auto-verificável da ANS permite a coexistência pacífica de publicidade e privacidade das carteiras no mesmo DLT.

Outro quesito importante é a flexibilização da arquitetura do ANS para a adição de suporte à novas implementações de DLTs. As abordagens dos trabalhos relacionados são implementações para a Bitcoin. O ANS, embora tenha sido desenvolvido e validado inicialmente para Bitcoin e Ethereum, foi estruturado para que o suporte a novas DLTs possa ser acoplado apenas pela extensão dos métodos de assinatura/validação.

Capítulo 6

Conclusão e Trabalhos futuros

Este capítulo conclui o trabalho e apresenta as considerações finais sobre o projeto de pesquisa, como também traz para a discussão limitações da proposta que não foram consideradas no escopo deste trabalho. Durante a leitura, foi possível perceber que considerações finais são feitas em alguns capítulos do trabalho, especificamente sobre o conteúdo do capítulo. Neste capítulo, as considerações elaboradas se referem ao contexto geral do trabalho. Este trabalho foi estruturado de modo que a leitura seja melhor conduzida a fim de proporcionar um compreendimento não só acerca do tema, mas também do ponto de vista do autor sobre como a problemática é apresentada. Entende-se que este trabalho possui relação com várias áreas de pesquisa, como: tecnologias de livro-razão distribuído, identidade digital e segurança da informação, e que a perspectiva a partir de cada uma dessas áreas poderia gerar outras narrativas para a problemática. No trabalho realizado, a pesquisa originou-se a partir da investigação de DLTs.

6.1 Considerações finais

Durante o desenvolvimento deste trabalho foi possível observar a dinâmica na evolução das tecnologias de livro-razão distribuído e entender que a sua capacidade de adaptação em diferentes contextos traz também novos desafios. Percebeu-se que o anonimato e a privacidade em determinadas situações podem ser dispensados, abrindo uma grande lacuna para a investigação de soluções que possam ajudar a identificar, legítima e inequivocamente, as entidades por trás de endereços de carteiras digitais.

Neste sentido, a abordagem apresentada propôs a utilização de tecnologias e conceitos consolidados nas próprias DLTs para a resolução de um problema específico. A solução proposta sugere a utilização de infraestruturas de chave pública existentes para o provimento da confiança no quesito da identificação dos atores por trás dos endereços. Percebeu-se que, com a ampla utilização de PKIs como soluções para identidade com reconhecimento legal, esta tecnologia poderia ser aproveitada para atuar em parte do problema. O modelo proposto para atingir o objetivo de prover uma associação confiável é baseado na utilização de assinatura digital para a geração de um artefato análogo a uma credencial verificável. O mecanismo proposto pode ser utilizado por aplicações onde é necessária a identificação dos atores envolvidos em transações de forma não exclusiva e com suporte a múltiplas instâncias de DLTs. O protótipo desenvolvido como prova de conceito ajudou a demonstrar a viabilidade da declaração espontânea de uma relação endereço-entidade como também a sua recuperação e verificação de forma independente.

Com o intuito de avaliar a proposta, foram realizados experimentos de integração do protótipo funcional com aplicações reais, demonstrando ser uma solução viável, uma vez que é de fácil integração e atinge a finalidade para o qual foi proposto. A figura 6.1 ilustra uma análise comparativa entre o ANS e algumas soluções encontradas na investigação dos trabalhos relacionados, considerando características importantes do ANS. O ANS pode ser considerado em parte descentralizado, uma vez que partes da solução podem ser consideradas centralizadas, como a utilização de PKI. Um outro quesito importante a ser considerado é que o ANS é multi-usuário, isto é, uma única instância do ANS pode atender a vários usuários sem a necessidade da instanciação de um serviço para cada ator que deseje reivindicar a posse de um endereço. Conforme descrito na especificação da proposta, a arquitetura de código do ANS foi iniciada adotando padrões que permitam a adição de código para suporte a múltiplas DLTs, ainda que em a versão inicial do protótipo já tenha validado a estratégia com a implementação para as duas DLTs públicas mais populares. O ANS foi concebido para que também possa ser utilizado por aplicações decentralizadas (on-chain), recurso não encontrado em outras soluções.

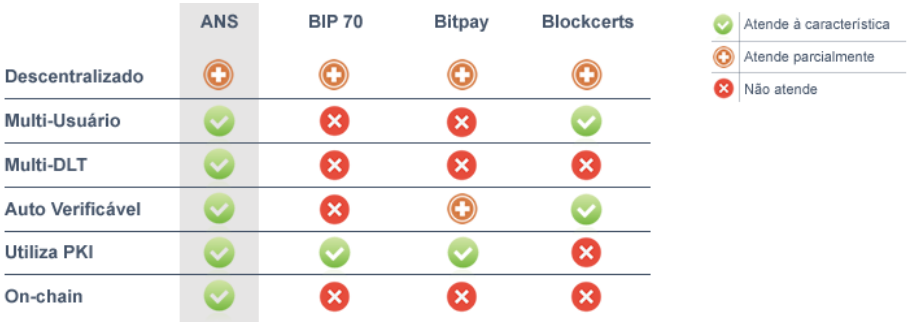

Figura 6.1: Análise comparativa entre soluções

Fonte: Próprio autor

6.2 Trabalhos futuros

Ainda que os objetivos desta pesquisa tenham sido atingidos e uma versão inicial de um sistema de referência tenha sido validado em um escopo delimitado para este trabalho, questões importantes foram percebidas e podem ser tratadas posteriormente em uma segunda etapa da pesquisa. Tais questões são apresentadas e discutidas nesta seção.

Revogação de ANS Certificates Uma questão importante que deve ser tratada na continuação da pesquisa está relacionada à possibilidade de perda da chave privada associada ao endereço de carteira. Este cenário já é considerado no âmbito da prova de identidade, com a revocação de certificados digitais através da publicação de listas de certificados revogados pelas autoridades certificadoras. Em suma, deve ser pensado em um mecanismo de revogação de ANS Certificates para o caso do comprometimento do acesso a alguma das chaves privadas.

Registro em base distribuída Conforme descrito na especificação do sistema de referência, para a primeira versão do ANS, a inserção de dados de mapeamento necessários para a recuperação de ANS Certificates foi feita em uma base de dados centralizada, ainda que os ANS Certificates sejam armazenados em um sistema de arquivos distribuído. Uma tecnologia de banco de dados chave-valor distribuído poderia ser utilizada em alternativa ao PostgreSQL, aumentando a tolerância à falhas em pontos centralizados do sistema. Bancos de Dados com suporte a replicação também poderiam ser considerados. Uma outra possibilidade a ser investigada é a utilização da infraestrutura do DNS neste sentido. O DNS por

ser amplamente distribuído e replicado, é uma tecnologia tolerante à falhas que poderia ser utilizada para o armazenamento de dados de associações endereço-entidade.

Gateway para validação de PKI Durante a realização dos experimentos, percebeu-se que a confiança da identificação é baseada na confiança em uma autoridade certificadora e consequentemente na infraestrutura utilizada. Em alguns experimentos, foi feita a validação da assinatura de prova de identidade do ANS Certificate em uma URL do órgão que mantém a infraestrutura utilizada, nesse caso, o ITI. Neste sentido, um novo componente do ANS poderia fazer o papel de um *gateway* para intermediar requisições para serviços de validação de assinaturas providos por infraestruturas de chave pública.

Generalização de recuperação de ANS Certificate Para a primeira validação do ANS, o repositório usado para o armazenamento de ANS Certificates foi o IPFS. O serviço de registro armazena o CID do artefato para posterior recuperação. O ANS Server, internamente carrega as bibliotecas necessárias para estabelecer uma conexão com um nó IPFS e recuperar os artefatos. Para uma próxima versão do ANS, sugere-se que a arquitetura de código seja flexibilizada para que diferentes tecnologias possam ser utilizadas em alternativa ao IPFS. Metadados sobre o tipo de tecnologia poderiam ser armazenados e padrões de projetos poderiam ser aplicados para permitir a implementação de escrita e recuperação de ANS Certificates para cada tecnologia.

Análise da LGPD Deve ser levada em consideração a realização de uma análise minuciosa da Lei Geral de Proteção de Dados Pessoais (LGPD) no caso do ANS ser utilizado sob juridição da justiça brasileira. Ainda que a autodeclaração e registro de posse de um endereço seja voluntária, o ANS Server poderia ser customizado para que permitisse a adição de restrições para que a identidade pudesse ser divulgada apenas para determinados usuários. Assinaturas digitais poderiam ser utilizadas para autenticação e consequentemente, acesso a ANS Certificates.

Amadurecimento do ANS Oracle Embora o ANS Oracle seja capaz de servir outras dApps fornecendo dados alimentados pelo ANS Server, ainda é necessário o amadurecimento dos meios de integração e utilização, como a definição do momento em que o Oracle

seria alimentado, levando em consideração custos transacionais e usabilidade.

Aplicação modelos de GIDs existentes Conforme mencionado no início deste capítulo, a temática deste trabalho se relaciona com diferentes áreas e uma delas é a gestão de identidades digitais. Este trabalho descreve um problema e sua solução através de métodos para autenticação, softwares auxiliares e serviços. O ANS Certificate poderia ser experimentado e explorado em modelos de gestão de identidades digitais e acessos existentes.

Integração com DIDs Conforme mencionado na seção 2, endereços de carteiras de DLTs públicas são usados para a composição de identificadores descentralizados (DIDs). DIDs são utilizados em identidades descentralizadas, a exemplo da identidade auto-soberana, em seus respectivos artefatos utilizados por tais tecnologias, como as credenciais verificáveis. Na literatura que envolve o tema, é comum encontrar exemplos de casos de tais identidades citando como exemplo a emissão de informações de uma autoridade confiável, como uma instituição pública, para um usuário solicitante, através de credenciais verificáveis. Embora que em sua essência, as identidades descentralizadas sejam baseadas em redes de confiança dos próprios usuários, um modelo híbrido poderia ser proposto para que apenas atores com *status* de emissor de credenciais pudessem ter a sua identidade legitimada pela "ancoragem"em uma PKI. Nesse contexto, uma vez que as instituições seriam associadas a DIDs, uma problemática seria garantir aos usuários a confiabilidade de que o DID realmente pertence à instituição. O ANS poderia ser utilizado para atender a este quesito. Em sua especificação, um DID é resolvido por um componente *resolver* para uma estrutura denominada *DID Document* montada com dados armazenados em DLTs, que fazem o papel de *DID Registries*. Um *DID Document* é um documento associado a um DID, que além de prover chaves-públicas para a verificação de provas criptográficas, expõe serviços para a interação com o sujeito. Sugere-se que o serviço do ANS Server poderia ser adicionado como uma entrada de serviço no *DID Document*.

6.3 Contribuição

Durante o desenvolvimento deste projeto de pesquisa, foi publicado um artigo nos anais do Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, onde

o mesmo foi apresentado no Workshop de Gestão de Identidades Digitais, conforme referência a seguir. O artigo foca na propositura do ANS com a especificação da proposta e o desenvolvimento de uma prova de conceito.

SOARES, Marcelo; COSTA, Rostand. Auto Identificação Voluntária e Verificável de Participantes em Aplicações Baseadas em Livros-Razão Distribuídos. In: WORKSHOP DE GESTÃO DE IDENTIDADES DIGITAIS - SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSEG), 18. , 2018, 1. Anais Estendidos do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. Porto Alegre: Sociedade Brasileira de Computação, oct. 2018 . p. 99 - 112.

Em meio ao desenvolvimento deste trabalho, o Banco Nacional de Desenvolvimento Econômico e Social (BNDES) publicou uma consulta pública 05/2018[1], para ouvir a comunidade acerca de questões em aberto no desenvolvimento do BNDESToken (Júnior et al., 2018). Foi submetida a inscrição do ANS para atender ao assunto (iii) associação de contas a CNPJs e/ou CPFs. Em 20 de dezembro de 2018 ocorreu uma reunião técnica entre a equipe técnica envolvida no desenvolvimento do BNDESToken e os pesquisadores da UFPB: Marcelo Soares e Rostand Costa. Durante a reunião, foi apresentada a proposta do ANS, e foram tiradas várias dúvidas sobre o seu funcionamento. Em seguida, o BNDES publicou o Relatório de Análise das Plataformas, no âmbito da Consulta 05/2018 avaliando que o ANS atende integralmente a todos os requisitos técnicos do BNDESToken, conforme pode ser visto na figura 6.2. Posteriormente, outras 3 reuniões ocorreram entre ambas as equipes, para discutir questões relacionadas à integração de BNDESToken e ANS.

A pesquisa em questão também resultou no registro do *software* "ANS - Address Name System - Sistema de associação de endereços de carteiras digitais e entidades do mundo real"no Instituto Nacional da Propriedade Industrial (INPI), conforme pode ser visto na figura 6.3.

[1] www.bndes.gov.br/consultablockchain

PROPONENTES: ROSTAND COSTA E MARCELO SOARES

Rostand Costa e Marcelo Soares são servidores da UFPB. Rostand trabalha no Laboratório de Aplicações de Vídeo Digital (LAVID), e Marcelo trabalha na Superintendência de Tecnologia da Informação (STI). Durante 2018, eles desenvolveram juntos o projeto ANS - Address Name System.

DECLARAÇÃO DE REQUISITOS ATENDIDOS

Agrupamento	Item da Consulta	Atende Integralmente	Atende Parcialmente	Não atende
Gestão de Contas	2.2.a.a			
Gestão de Contas	2.2.a.b			
Gestão de Contas	2.2.a.c			
Gestão de Contas	2.2.a.d			
Gestão de Contas	2.2.a.e			
Gestão de Contas	2.2.a.f			
Acompanhamento de transações	2.2.b.a			
Acompanhamento de transações	2.2.b.b			
Acompanhamento de transações	2.2.b.c			
Acompanhamento de transações	2.2.b.d			
Acompanhamento de transações	2.2.b.e			
Associação de contas a CNPJs e/ou CPFs	2.2.c.a	X		
Associação de contas a CNPJs e/ou CPFs	2.2.c.b	X		
Associação de contas a CNPJs e/ou CPFs	2.2.c.c	X		
Associação de contas a CNPJs e/ou CPFs	2.2.c.d	X		
Associação de contas a CNPJs e/ou CPFs	2.2.c.e	X		
Requisitos técnicos	2.3.a	X		
Requisitos técnicos	2.3.b	X		
Requisitos técnicos	2.3.c	X		

Figura 6.2: Reunião entre autores do ANS e equipe do BNDESToken

Fonte: Retirado de `https://www.bndes.gov.br/wps/wcm/connect/site/fd4f0c33-57e5-4f2f-8ff5-2b936c2328f6/Relatorio+das+Demonstra%C3%A7%C3%B5es+-+ERRATA.pdf?MOD=AJPERES&CVID=mA0YlCC.`

INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL

02/07/2019 870190061691 12:48

29409191905469102

Pedido de Registro de Programa de Computador - RPC

Número do Processo: 512019001385-9

Dados do Titular

Titular 1 de 1

Nome ou Razão Social: UNIVERSIDADE FEDERAL DA PARAIBA
Tipo de Pessoa: Pessoa Jurídica
CPF/CNPJ: 24098477000110
Nacionalidade: Brasileira
Qualificação Jurídica: Instituição de Ensino e Pesquisa
Endereço: Cidade Universitária
Cidade: João Pessoa
Estado: PB
CEP: 58059-900
País: Brasil
Telefone: (83) 32167558
Fax:
Email: inova@reitoria.ufpb.br

Dados do Programa

Data de Publicação: 25/10/2018

- § 2º do art. 2º da Lei 9.609/98: "Fica assegurada a tutela dos direitos relativos a programa de computador pelo prazo de cinquenta anos contados a partir de 1º de janeiro do ano subsequente ao da sua publicação ou, na ausência desta, da sua criação"

Título: ANS - Address Name System - Sistema de associação de endereços de carteiras digitais e entidades do mundo real.

Algorítimo hash: SHA-512 - Secure Hash Algorithm

Resumo digital hash: 84e099af7a631cc3c9cc864e4e07440effa0e3f189a4c145b54e528022404d2e82c208b0d874ddab81944ccb25ff8d8388ad536ef55fc131b356a2163c5a8808

§1º e Incisos VI e VII do §2º do Art. 2º da Instrução Normativa: O titular é o responsável único pela

Figura 6.3: Registro de Software no INPI

Fonte: Próprio autor

Referências Bibliográficas

Abood, O. e Guirguis, S. (2018). A survey on cryptography algorithms. *International Journal of Scientific and Research Publications*, 8:495–516.

Al-Khouri, A. M. (2012). Pki in government digital identity management systems.

Albarqi, A., Alzaid, E., Ghamdi, F., Asiri, S., e Kar, J. (2015). Public key infrastructure: A survey. *Journal of Information Security*, 06:31–37. https://www.researchgate.net/publication/272955036_Public_Key_Infrastructure_A_Survey. [Online; accessed 15-December-2019].

Amaral, J. N. (2011). About computing science research methodology.

Andresen, G. e Hearn, M. (2013). Bitcoin improvement proposal 70.

Aste, T., Tasca, P., e Matteo, T. D. (2017). Blockchain technologies: The foreseeable impact on society and industry. *IEEE Computer Society*, 50:18 – 28.

Averkamp, A. (2017). Accounting basics - part 5 | explanation.

Baird, L. C. (2016). The swirlds hashgraph consensus algorithm: fair, fast, byzantine fault tolerance. Technical Report SWIRLDS-TR-2016-01, Swirlds, Inc.

Bakker, E.-J. (2018). Brazil's beginning blockchain business.

Bartel, M., Boyer, J., Fox, B., LaMacchia, B., e Simon, E. (2015). Xml signature syntax and processing. https://www.w3.org/TR/xmldsig-core2/. [Online; accessed 05-August-2018].

Bech, M. L. e Garratt, R. (2017). Central bank cryptocurrencies.

Benet, J. (2014). IPFS - content addressed, versioned, P2P file system. *CoRR*, abs/1407.3561.

Benkler, Y. (2006). *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, New Haven, CT, USA.

Bitcoin (2018a). Some things you need to know. https://bitcoin.org/en/you-need-to-know. [Online; accessed 04-August-2018].

Bitcoin (2018b). Transactions guide - bitcoin. https://bitcoin.org/en/transactions-guide. [Online; accessed 04-August-2018].

Bitcoin-Wiki (2017). Base58check encoding. https://en.bitcoin.it/wiki/Base58Check_encoding [Online; accessed 05-August-2018].

Bitcoin-Wiki (2018). Technical background of version 1 bitcoin addresses. https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses. [Online; accessed 05-August-2018].

Bitpay (2019). Json payment protocol specification v2.

Blockcerts (2016). Introduction - blockcerts : The open standard for blockchain credentials. [Online; accessed 20-August-2019].

BNDES (2018). Public call notice aarh 05/2018 – bndes. https://www.bndes.gov.br/wps/wcm/connect/site/711aecd2-20da-42f8-ad56-bec520832d75/Relat%C3%B3rio+Final+-+An%C3%A1lise+das+Plataformas+-+Vers%C3%A3o+em+ingl%C3%AAs.pdf?MOD=AJPERESCVID=mCRkiQn. [Online; accessed 04-August-2018].

Boeyen, S., Santesson, S., Polk, T., Housley, R., Farrell, S., e Cooper, D. (2008). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280.

Bragg, S. (2017). Ledger - accountingtools.

Brasil (2001). Medida provisória no 2.200-2, de 24 de agosto de 2001.

Brennan, C. e Lunn, W. (2016). The trust disrupter. Technical report, Credit Suisse.

Buterin, V. (2014). Ethereum: A next-generation smart contract and decentralized application platform. `https://github.com/ethereum/wiki/wiki/White-Paper`. Accessed: 2016-08-22.

Cameron, K. (2005). The laws of identity. `https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf`.

Chaum, D. (1992). Achieving electronic privacy. 267:96–101.

Choudhury, S., Bhatnagar, K., e Haque, W. (2002). *Public Key Infrastructure Implementation and Design*. John Wiley Sons, Inc., USA, 1st edition.

Christopher Allen, Arthur Brock, V. B. J. C. D. D. C. L. P. K. J. N. D. R. M. S. G. S. N. T. H. T. W. (2015). Decentralized public key infrastructure. https://github.com/WebOfTrustInfo/rwot1-sf/blob/master/final-documents/dpki.pdf. [Online; accessed 10-January-2020].

Clauundefined, S. e Köhntopp, M. (2001). Identity management and its support of multilateral security. *Comput. Netw.*, 37(2):205–219.

Costa, R., Faustino, D., Lemos, G., Queiroga, A., Djohnnatha, C., Alves, F., Lira, J., e Pires, M. (2018). Uso não financeiro de blockchain: Um estudo de caso sobre o registro, autenticação e preservação de documentos digitais acadêmicos. *Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain-SBRC)*, 1(1/2018).

Davidson, S., De Filippi, P., e Potts, J. (2016). Disrupting governance: The new institutional economics of distributed ledger technology.

Deshpande, A., Stewart, K., Lepetit, L., e Gunashekar, S. (2017). Distributed ledger technologies/blockchain: Challenges, opportunities and the prospects for standards. Technical report, British Standards Institution (BSI).

Dhar, S. e Bose, I. (2016). Smarter banking: Blockchain technology in the indian banking system. *Asian Management Insights*, 3:46–53. https://ink.library.smu.edu.sg/ami/3.

Diffie, W. (1988). The first ten years of public-key cryptography. *Proceedings of the IEEE*, 76(5):560–577.

Durand, A., Gremaud, P., e Pasquier, J. (2017). Decentralized web of trust and authentication for the internet of things. Em *Proceedings of the Seventh International Conference on the Internet of Things*, IoT '17, New York, NY, USA. Association for Computing Machinery.

Fischer, M. J., Lynch, N. A., e Paterson, M. S. (1985). Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382.

Fromknecht, C., Velicanu, D., e Yakoubov, S. (2014). A decentralized public key infrastructure with identity retention. *IACR Cryptology ePrint Archive*, 2014:803. https://eprint.iacr.org/2014/803.pdf. [Online; accessed 15-December-2019].

Gamma, E., Johnson, R., Helm, R., e Vlissides, J. (2006). *Padrões de Projetos: Soluções Reutilizáveis*. Bookman.

Helland, P. (2015). Immutability changes everything. *Queue*, 13(9):40:101–40:125.

IDC (2018). New idc spending guide sees worldwide blockchain spending growing to $9.7 billion in 2021. https://www.idc.com/getdoc.jsp?containerId=prUS43526618. [Online; accessed 04-August-2018].

ITI (2018). Glossário - instituto nacional de tecnologia da informação - iti. http://www.iti.gov.br/glossario. [Online; accessed 04-August-2018].

Joyce, R. e Gupta, G. (1990). Identity authentication based on keystroke latencies. *Commun. ACM*, 33(2):168–176.

Júnior, G. M. A., Jr., J. N. D., Onodera, M. T., de Borba Maranhão Moreno, S. M., e da Rocha Santos Almeida, V. (2018). Bndestoken: Uma proposta para rastrear o caminho de recursos do bndes. *Workshop em Blockchain: Teoria, Tecnologias e Aplicações (WBlockchain_SBRC)*, 1(1/2018).

Krasner, G. e Pope, S. (1998). A cookbook for using the model - view controller user interface paradigm in smalltalk - 80. *Journal of Object-oriented Programming - JOOP*, 1.

Kuhn, D. R., Hu, V., Polk, W. T., e Chang, S.-j. H. (2001). Sp 800-32. introduction to public key technology and the federal pki infrastructure. Technical report, Gaithersburg, MD, USA. https://dl.acm.org/doi/pdf/10.5555/2206241?download=true. [Online; accessed 09-December-2019].

Lackey, E. D. (2012). Red hat certificate system common criteria certification 8.1.

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565.

Lemieux, V., Flores, D., e Lacombe, C. (2018). Real estate transaction recording in the blockchain in brazil (rcplac-01) - case study 1.

Longley, D., Burnett, D., Zundel, B., Sporny, M., e Noble, G. (2019). Verifiable credentials data model 1.0. W3C recommendation, W3C. https://www.w3.org/TR/2019/REC-vc-data-model-20191119/.

Lyons, T., Courcelas, L., e Timsit, K. (2019). Blockchain and digital identity. Technical report, European Union Blockchain Observatory Forum. https://www.eublockchainforum.eu/reports. [Online; accessed 12-February-2019].

Maltese, M. E. G. (2015). Singapore prime minister said national banks can use blockchain. https://cointelegraph.com/news/singapore-prime-minister-said-national-banks-can-use-blockchain. [Online; accessed 04-August-2018].

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., e Savage, S. (2013). A fistful of bitcoins: Characterizing payments among men with no names. Em *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pgs. 127–140, New York, NY, USA. ACM.

Merkle, R. C. (1982). Method of providing digital signatures. https://patentimages.storage.googleapis.com/69/ab/d9/2ff9f94fada6ea/US4309569.pdf. [Online; accessed 04-February-2020].

Möser, M. e Böhme, R. (2017). The price of anonymity: empirical evidence from a market for bitcoin anonymization. *Journal of Cybersecurity*, 3(2):127–135.

Nakamoto, S. (2008a). Bitcoin: A peer-to-peer electronic cash system.

Nakamoto, S. (2008b). Bitcoin p2p e-cash paper. The Cryptography Mailing List.

Narayanan, A. e Clark, J. (2017). Bitcoin's academic pedigree. *Commun. ACM*, 60(12):36–45.

Natarajan, H., Krause, S., e Gradstein, H. (2017). Distributed ledger technology (dlt) and blockchain. FinTech note; no. 1. Washington, D.C. : World Bank Group. http://documents.worldbank.org/curated/en/177911513714062215/Distributed-Ledger-Technology-DLT-and-blockchain. [Online; accessed 06-August-2018].

Olson, E. T. (2019). Personal identity. Em Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2019 edition.

Orman, H. (2018). Blockchain: the emperors new pki? *IEEE Internet Computing*, 22(2):23–28.

Pagliery, J. (2013). Fbi shuts down online drug market silk road. https://money.cnn.com/2013/10/02/technology/silk-road-shut-down/index.html. [Online; accessed 04-August-2018].

Parameswaran, M., Susarla, A., e Whinston, A. (2001). P2p networking: An information-sharing alternative. *Computer*, 34(7):31–38.

Parlamento Europeu (2014). Regulamento (ue) n.o 909/2014 do parlamento europeu e do conselho de 23 de julho de 2014. *Jornal Oficial da União Europeia*, 257:73–114. https://eur-lex.europa.eu/legal-content/PT/TXT/PDF/?uri=OJ:L:2014:257:FULLfrom=PT. [Online; accessed 12-September-2019].

Perekalin, A. (2017). How to protect vs. wannacrypt. https://www.kaspersky.com/blog/wannacry-ransomware/16518/. [Online; accessed 04-August-2018].

Popov, S. (2018). The tangle. https://iota.org/IOTA_Whitepaper.pdf. [Online; accessed 02-August-2018].

Prodanov, C. e Freitas, E. C. (2013). *Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico - 2ª Edição*. Editora Feevale.

Rashid, F. Y. (2014). A closer look at how criminals steal your bitcoins, and how to stop them. https://www.itproportal.com/2014/03/08/a-closer-look-at-how-criminals-steal-your-bitcoins-and-how-to-stop-them/. [Online; accessed 04-August-2018].

Reid, F. e Harrigan, M. (2011). An Analysis of Anonymity in the Bitcoin System. *ArXiv e-prints*.

Sommerville, I. (2011). *Software Engineering*. Ninth Edition, 9 edition.

Sporny, M., Sabadello, M., Allen, C., Longley, D., e Reed, D. (2019). Decentralized identifiers (DIDs) v1.0. W3C working draft, W3C. https://www.w3.org/TR/2019/WD-did-core-20191209/.

Stallings, W. (2013). *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press, USA, 6th edition.

Stallings, W. (2015). *Criptografia E Segurança De Redes*. PEARSON BRASIL.

Swanson, T. (2015). Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. `http://www.ofnumbers.com/wp-content/uploads/2015/04/Permissioned-distributed-ledgers.pdf`. Accessed: 2018-05-01.

Sward, A., Vecna, I., e Stonedahl, F. (2018). Data insertion in bitcoin's blockchain. *Ledger*, 3(0). https://ledgerjournal.org/ojs/index.php/ledger/article/download/101/93. [Online; accessed 10-September-2018].

Tanenbaum, A., Wetherall, D., e Translations, O. (2011). *Redes de computadores*. PRENTICE HALL BRASIL.

Tanenbaum, A. S. e Steen, M. v. (2006). *Distributed Systems: Principles and Paradigms (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Taylor, S., Brown, R. G., Lehdonvirta, V., Ali, R., Sasse, A., Godsiff, P., Mulligan, C., e Curry, P. (2016). Distributed ledger technology: beyond block chain. Technical report, Government Office for Science. [Online]. https://medium.com/aid-tech/distributed-ledger-technology-beyond-block-chain-a-report-by-the-uk-government-chief-scientific-2eb8e0bda140, Accessed: 2018-05-01.

Tennant, L. (2017). Improving the anonymity of the iota cryptocurrency. http://iotafeed.com/wp-content/uploads/2017/08/anonymity-iota.pdf. [Online; accessed 21-September-2018].

Thomsen, S. e Knudsen, L. (2009). *Cryptographic Hash Functions*. PhD thesis.

Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., e Kishigami, J. (2016). Blockchain contract: Securing a blockchain applied to smart contracts. Em *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pgs. 467–468.

Wood, G. (2017). Ethereum: A secure decentralised generalised transaction ledger eip-150 revision (759dccd - 2017-08-07). https://ethereum.github.io/yellowpaper/paper.pdf. [Online; accessed 04-August-2018].