UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIA DA COMPUTAÇÃO

Arthur Gabriel Crippa Milanez

Cripparency: Protocolo Baseado em *Certificate Transparency* para Rastreabilidade de Certificados de Uso Único Utilizando *Blockchain*

Florianópolis
2023

Arthur Gabriel Crippa Milanez

Cripparency: Protocolo Baseado em *Certificate Transparency* para Rastreabilidade de Certificados de Uso Único Utilizando *Blockchain*

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.
Orientador: Prof. Ricardo Custódio, Dr.
Coorientador: Lucar Mayr

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Milanez, Arthur Gabriel Crippa
Cripparency: Protocolo Baseado em Certificate Transparency para Rastreabilidade de Certificados de Uso Único Utilizando Blockchain / Arthur Gabriel Crippa Milanez ; orientador, Ricardo Felipe Custódio, coorientador, Lucas Mayr, 2023.
90 p.

2. Transparencia de Certificados. 3. Blockchain. 4. Certificado Digital. 5. Cybersegurança. I. Custódio, Ricardo Felipe . II. Mayr, Lucas. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

Arthur Gabriel Crippa Milanez

Cripparency: Protocolo Baseado em *Certificate Transparency* para Rastreabilidade de Certificados de Uso Único Utilizando *Blockchain*

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Graduação em Ciência da Computação.

Florianópolis, 1 de Agosto de 2023.

Banca Examinadora:

Prof. Ricardo Custódio, Dr.
Orientador
Universidade Federal de Santa Catarina

Prof. Frederico Schardong, Me.
Avaliador
Universidade Federal de Santa Catarina

Prof. Martin Augusto Gagliotti Vigil, Dr.
Avaliador
Universidade Federal de Santa Catarina

Este trabalho é dedicado à minha família.

AGRADECIMENTOS

Gostaria de expressar meus agradecimentos a todos que fizeram parte dessa jornada e contribuíram para a conclusão do meu TCC. Cada um de vocês desempenhou um papel importante na minha trajetória acadêmica, e sou grato por todo o apoio e amizade que recebi ao longo do caminho.

Aos meus pais, Sinara e Edson pelas oportunidades que me proporcionaram a estudar em uma das melhores universidades do pais. Por todo amor, carinho e confiança que me concederam, de sobra, até aqui, para que pudesse me tornar quem sou hoje, amo muito vocês.

A minha namorada Julia, pelo seu amor incondicional e por compreender meus momentos de tristeza e desespero durante a graduação. Por estar sempre do meu lado e, mesmo longe, me apoiar nas decisões, me alertar de perigos e acreditar, mais que ninguém, em mim. Sem você, essa jornada seria impossível. Nessa tempestade chamada faculdade, você foi meu farol de milhas. Te amo.

Ao LabSEC, lugar onde conheci pessoas incríveis, fiz amigos, passei noites estudando, dei risadas, chorei, me desesperei e comemorei aprovações. Local que considerei minha segunda casa durante os anos de graduação.

Ao meu coorientador e colega de trabalho, Lucas Mayr pela ajuda, motivação, broncas, ensinamentos e por confiar no meu potencial para desenvolver o presente trabalho.

Aos amigos que fiz durante a trajetória acadêmica, com vocês a faculdade se tornou um ambiente descontraído, onde riamos da nossa própria tragedia e comemorávamos cada nota e cada aprovação. Espero que tenham muito sucesso na vida pessoal e profissional.

Aos amigos da "call Principal", que levo no coração desde o ensino fundamental, pessoas que, quando estão juntos, a risada e a diversão são garantidas. É satisfatório e tranquilizante saber que posso contar com todos vocês para qualquer momento.

A todos aqueles que, de alguma forma, cruzaram meu caminho durante esses anos, deixaram uma marca positiva e me ajudaram a chegar até aqui, meu mais sincero agradecimento. Suas presenças foram valiosas e contribuíram para minha formação como pessoa e profissional. O apoio de cada um de vocês foi essencial, e serei eternamente grato por isso.

"Se a coisa não sai do jeito que eu quero
Também não me desespero, o negócio é deixar rolar,
Aos trancos e barrancos, lá vou eu,
Sou feliz e agradeço por tudo que Deus me deu."
- Zeca Pagodinho

RESUMO

Com o avanço da cibersegurança, a autenticação de documentos digitais tem sido um desafio. Certificados físicos e digitais estão sujeitos a falhas que comprometem sua autenticidade, resultando em problemas. Para mitigar esses riscos, é necessário reconhecer as imperfeições e buscar soluções adequadas. A detecção de erros na emissão de certificados enfrenta dificuldades dentro das restrições da infraestrutura de chaves públicas. O protocolo *Certificate Transparency* aborda certificados relacionados ao protocolo TLS, mas não o gerenciamento de chaves privadas. Uma solução é o uso de "certificados de uso único", que são exclusivamente usados para assinar um único documento e têm sua chave privada eliminada. No entanto, há risco de vazamento de credenciais do provedor de identidade, permitindo assinaturas fraudulentas. Para contornar isso, propõe-se, neste trabalho, a integração dessas ideias com a tecnologia de blockchain, através do protocolo implementado utilizando *framework HyperLedger Fabric*, Cripparency, que consiste em um registro de certificados de uso único que identifica comprometimentos de credenciais, falhas de emissão e permite o monitoramento de documentos assinados, aumentando a confiabilidade e prevenindo fraudes.

Palavras-chave: Certificado Digital; Transparência de Certificado; Assinatura Digital; Cadeia de Blocos; Cibersegurança.

ABSTRACT

The cybersecurity advancement, became the authentication of digital documents a challenge nowadays. Physical and digital certificates are susceptible to failures that compromise their authenticity, resulting in issues. To mitigate these risks, it is necessary to acknowledge the imperfections and seek appropriate solutions. Detecting errors in certificate issuance faces difficulties within the constraints of the public key infrastructure. The Certificate Transparency protocol addresses certificates related to the TLS protocol but not the management of private keys. One solution is the use of "one-time certificates," which are exclusively used to sign a single document and have their private keys eliminated. However, there is a risk of identity provider credential leakage, enabling fraudulent signatures. To overcome this, this work proposes the integration of these ideas with blockchain technology, through the implemented protocol using the Hyperledger Fabric framework, called Cripparency. It consists of a registry of single-use certificates that identifies credential compromises, issuance failures, and enables monitoring of signed documents, increasing reliability and preventing fraud.

Keywords: Digital Certificate; Certificate Transparency; Digital Signature; Blockchain; Cybersecurity.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AC	Autoridade Certificadora
ICP	Infraestrutura de Chaves Públicas
RFC	Request for Comments
PoW	Proof of Work
CT	Certificate Transparency
SSL	Secure Sockets Layer
TLS	Transport Layer Security
SCT	Signed Certificate Timestamp
API	Application Programming Interface
HLF	HyperLedger Fabric
LGPD	Lei Geral de Proteção dos Dados
SDK	Software Development Kit
RNP	Rede Nacional de Ensino e Pesquisa

SUMÁRIO

1 INTRODUÇÃO

Desde os primórdios, os autores se preocupavam em registrar sua identidade nos escritos através de desenhos e rabiscos. Mais tarde, esse ato foi denominado assinatura e continua sendo utilizado pelas pessoas para validar documentos físicos até os dias atuais. Porém, com os avanços na cibersegurança e sua incorporação no cotidiano da sociedade, ocorreu a necessidade de autenticar a validade dos documentos digitalmente. Nesse contexto, surge o certificado digital, uma ferramenta que, através da criptografia, permite a assinatura digital, garantindo a autenticidade e integridade das informações no ambiente virtual. Entretanto, documentos digitais e físicos são fundamentalmente diferentes. Documentos digitais exigem suporte computacional e criptográfico como (i) par de chaves, (ii) certificado digital, (iii) infraestrutura de chaves públicas, (iv) funções de resumo, etc.

Certificados digitais, assim como físicos, estão sujeitos a falhas, o que levanta preocupações sobre suas consequências, eis que essas ocorrências podem comprometer a autenticidade de documentos, causar prejuízos financeiros e resultar em falhas de autenticação, entre outros. Assim, mostra-se essencial reconhecer essas imperfeições e buscar soluções adequadas para reduzir esses riscos.

A detecção de erros é uma das formas encontradas, a fim de mitigar os danos causados por falhas de emissão. Porém, há certa dificuldade em realizar essa detecção dentro das restrições estabelecidas pela infraestrutura de chaves públicas. Isso porque, havendo uma emissão incorreta, o usuário somente tomará ciência da falha ao se deparar com um documento em que esse certificado falho foi utilizado, podendo, então, dar início ao processo de revogação. Ocorre que não há garantia de que o usuário conseguirá identificar essa situação, tampouco há estimativa do tempo necessário para tanto. Idealmente, a detecção do erro deveria ser instantânea, a fim de evitar danos significativos ao usuário. É por isso que, atualmente, existem sistemas de registro de emissão de certificados e modelos de gerenciamento de chaves privadas que lidam com estes problemas.

Nesse contexto, destaca-se o protocolo *Certificate Transparency*, um *log* público que faz o registro dos certificados digitais e disponibiliza os dados para monitoramento dos clientes. Contudo, devido ao funcionamento da infraestrutura de chave pública (ICP) tradicional, o CT se limita, principalmente, ao registro de certificados, especialmente os relacionados ao protocolo TLS. Isso significa que, caso um atacante consiga ter acesso à chave privada de um usuário, ele ainda conseguirá assinar documentos sem ser facilmente detectado pelo usuário.

Em uma tentativa de contornar a questão do gerenciamento de chaves, destaca-se uma proposta, que foi submetida à publicação, como resultado de um trabalho de mestrado conduzido por um estudante da UFSC, coorientador do presente trabalho. Essa abordagem propõe a adoção de um novo modelo de certificado digital denominado "certificado de uso único", que é utilizado exclusivamente para assinar um único documento, com subsequente eliminação da chave privada e consequente desnecessidade de armazenamento prolongado. Assim, o certificado de uso único não requer revogação, eis que sua validade é restrita ao momento da assinatura. No entanto, caso

ocorra um vazamento de credenciais do provedor de identidade responsável pelo fornecimento das informações à Autoridade Certificadora, os documentos ainda poderão ser assinados em nome de outras pessoas.

Diante desse cenário, este trabalho propõe a integração das ideias mencionadas com a tecnologia de *blockchain*, visando potencializar seus benefícios e promover um ambiente mais seguro. Isso será feito através do Cripparency, um protocolo de registro de certificados de uso único capaz de identificar o comprometimento de credenciais de usuários, falhas de emissão de certificados digitais e permitir o monitoramento de documentos assinados com certificado de uso único.

1.1 OBJETIVOS

Nesta seção, serão descritos os objetivos gerais e específicos do trabalho, que serão a base para o desenvolvimento e implementação do projeto.

1.1.1 Objetivo Geral

Realizar estudos sobre *blockchain* a fim de verificar, dentre as já existentes, qual é a mais adequada para solucionar o problema proposto relacionado ao armazenamento de certificados de uso único. Busca-se utilizar o protocolo de transparência de certificados como base para monitoramento de falhas de emissão desses certificados e identificar a tecnologia de *blockchain* mais eficiente e adequada para atender aos requisitos do projeto.

1.1.2 Objetivos Específicos

- Explorar os conceitos e elementos relacionados à cibersegurança e à certificação digital, visando obter uma compreensão aprofundada dessas áreas.
- Estudar e compreender o uso de diferentes tipos de *blockchain*, com objetivo de selecionar a mais adequada ao armazenamento de certificados digitais, avaliando vantagens e desvantagens em relação às necessidades do trabalho.
- Adaptar o protocolo de transparência de certificado para atender às necessidades dos certificados de uso único, realizando as modificações e ajustes necessários.
- Otimizar o protocolo de transparência de certificado para beneficiar seu uso em uma rede *blockchain*.
- Apresentar e destacar as principais diferenças resultantes das alterações feitas no protocolo, evidenciando o impacto dessas mudanças em seu uso e funcionalidade.
- Descrever, de forma detalhada, o fluxo do novo protocolo em uma rede *blockchain*.

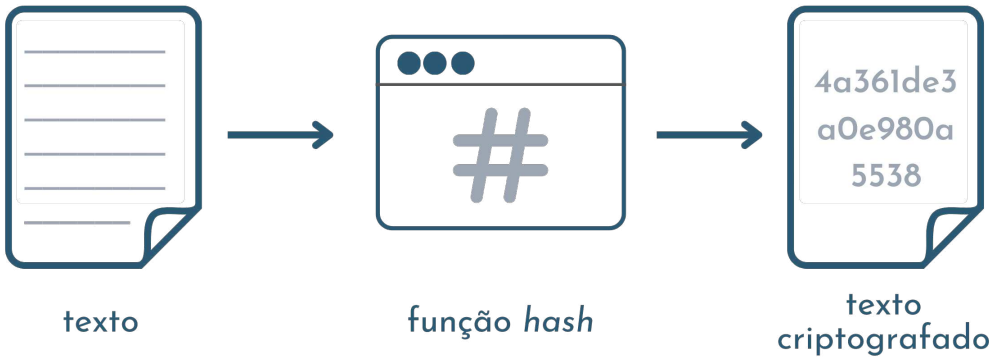
2 PRIMITIVAS CRIPTOGRÁFICAS

A seguir, serão abordados os principais conceitos das primitivas criptográficas, que serão de extrema importância para entendimento do presente trabalho. O capítulo está dividido da seguinte maneira: resumo criptográfico em 2.1, criptografia assimétrica em 2.2, árvore de Merkle e assinatura digital nas seções 2.3 e 2.4 respectivamente.

2.1 RESUMO CRIPTOGRÁFICO

Criptografia é um tipo de operação que transforma um texto original em um texto criptografado utilizando, na maioria das vezes, a técnica de substituição, na qual cada elemento do texto original é mapeado para outro elemento, baseando-se em uma função matemática [1].

Figura 1 – Funcionamento de um resumo criptográfico.



Fonte: Elaborada pelo autor

Nesse contexto, foram criadas as funções de *hash*, que são dadas pela forma $h = H(M)$, onde M é o tamanho do texto original e $H(M)$ é o valor do *hash* de tamanho fixo, que será a saída da função independente do tamanho da entrada. Porém, as funções de *hash* são muito mais complexas do que uma função normal. Alguns critérios para que sejam consideradas funções de *hash* são citados pelo autor Willian Stallings no livro *cryptography and network security principles and practices* [1]:

> Podem ser aplicadas em textos de qualquer tamanho, ou seja, o valor da variável "M" citada acima pode variar para qualquer dimensão. H(M) deve produzir um valor fixo na saída, como já mencionado antes. H(M) deve ser facilmente computável para qualquer que seja o tamanho de M. (tradução livre)

Qualquer função com esses critérios já pode ser considerada função de *hash*, porém, apenas esses critérios não são suficientes para torná-la segura a ponto de ser utilizada no âmbito

da criptografia. Com o atual avanço da computação, uma abordagem de criptoanálise bem executada conseguiria revelar a operação utilizada para gerar o texto criptografado, e, portanto, o texto original. Assim, faz-se necessária a utilização de funções que garantam a segurança e a integridade do documento, de modo que alguns outros critérios são adicionados a fim de transformá-las em *strong one-way-functions*, ou funções de caminho único. Funções de caminho único são computacionalmente fáceis de serem executadas, porém, geralmente, difíceis de serem invertidas [1].

Ainda segundo Stallings (2006), "para qualquer valor de h é computacionalmente inviável encontrar $H(M)$ tal que $H(M) = h$" [1]. Esse critério dá o nome à propriedade *one-way*, que, no contexto da criptografia, se refere à impossibilidade de encontrar a função inicial que gerou a saída h.

Ademais, "é computacionalmente inviável achar um par (x, y) tal que $H(x) = H(y)$". Esse critério se refere à propriedade de resistência de colisões, que será esclarecida adiante, na explicação sobre o conceito de *blockchain*. O resumo criptográfico é, então, o resultado da aplicação de uma função de *hash*, estabelecida previamente, sobre o texto original, ou seja, introduzindo como entrada o texto o qual se deseja criptografar, e recebendo em sua saída um texto de tamanho fixo e ilegível, incompreensível comparado ao inicial. Em outras palavras, um texto criptografado ou resumo criptográfico.

2.2 CRIPTOGRAFIA ASSIMÉTRICA

Contribuindo para o conceito acima exposto, DIFFIE, W. HELLMAN, M., explicam em "*new directions in cryptography*" que criptografia é "o estudo da matemática para resolver dois problemas da segurança: o da privacidade e o da autenticidade" e que privacidade é "garantir que nenhum terceiro tenha acesso ao conteúdo que está sendo comunicado entre dois indivíduos"e a autenticidade é "a garantia que o conteúdo da mensagem original não seja modificado por alguém que não seja o autor" [2].

Dentro desse cenário, existem dois tipos de criptografia: a simétrica, que não terá relevância para o presente trabalho, visto que *Blockchains* em geral, inclusive a que será desenvolvida, não utilizam esse recurso, e criptografia assimétrica, que será diversas vezes retomada durante este desenvolvimento.

A criptografia assimétrica ou criptografia de chaves públicas, apresentada, pela primeira vez, em 1976, por DIFFIE, W. HELLMAN, M., baseia-se no uso de duas chaves criptográficas, a pública e a privada. Ambas são geradas a partir de um mesmo algoritmo matemático. Esse tipo de criptografia foi criado para tornar possível uma troca de mensagens sem a necessidade de um canal seguro. Para melhor entendimento, neste tópico, será designada à chave pública o nome de Ek (Encryption key) e à chave privada, Dk (Decryption key).

Cada uma das chaves realiza a operação inversa da outra, tornando-as chaves que se correlacionam. Não há facilidade computacional de se obter a Dk a partir da Ek, portanto, Dk pode ser divulgada publicamente sem que haja o comprometimento da Dk e da confiabilidade do

sistema. Já a Dk deve ser mantida pelo proprietário das chaves em um local seguro.

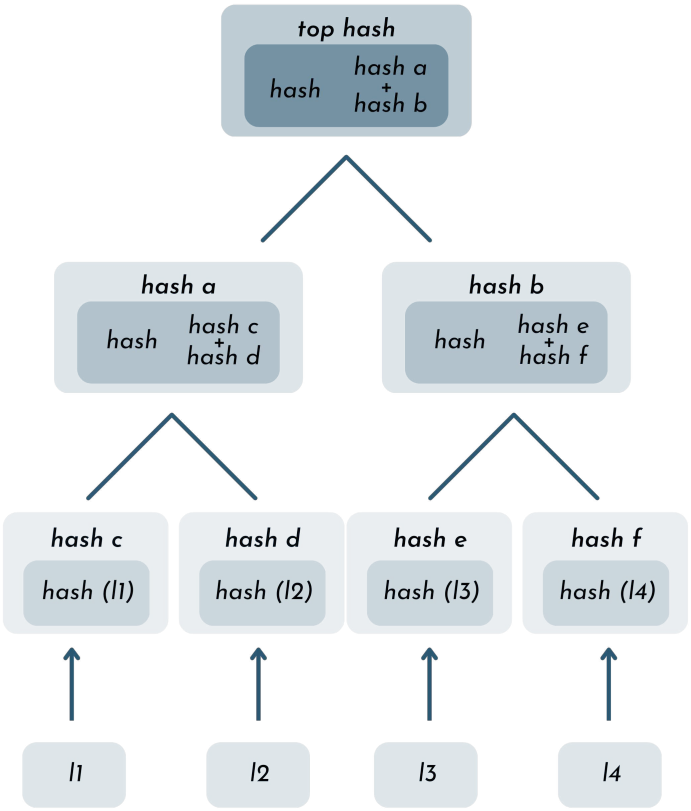
O funcionamento de uma comunicação segura utilizando o método de criptografia assimétrica se dá da seguinte forma: dois indivíduos, A e B, pretendem começar um diálogo que deve ser protegido. Ambos devem criar seu próprio par de chaves contendo uma Ek e uma Dk. Divulgam as suas Ek's, da maneira que lhes for conveniente. Assim, quando A quiser enviar uma mensagem a B, ele utiliza a Ek de B para criptografar e enviar o documento a B. Caso a mensagem seja interceptada no meio do caminho, de nada adianta tê-la pois apenas quem possuir a Dk que forma o par com a Ek utilizada para criptografar a mensagem conseguirá decifra-la

2.3 ÁRVORE DE MERKLE

Árvore de Merkle, mais conhecida por *Merkle Tree* apresentada em [3] por Ralph Merkle, é uma estrutura de dados, do tipo árvore, preenchida com resumos criptográficos. É utilizada para verificação da autenticidade de assinatura digital de uma grande quantidade de dados. A estruturação de uma árvore no contexto da ciência da computação ocorre pela existência de um nodo raiz inicial. A partir do nodo inicial, novos nodos se formam à direita e à esquerda, chamados de nodos folha. Os nodos folha também podem gerar seus próprios nodos folha e assim recursivamente até atingir no tamanho desejado da árvore.

O tamanho da última camada de uma árvore de Merkle é sempre $2n$, visto que, ao contrário das outras árvores, sua leitura se dá das folhas para a raiz. Como todos seus nodos são *hashes*, o nodo pai é o resumo criptográfico da concatenação dos seus filhos. Essa verificação vai sendo feita até chegar na raiz ou até achar um valor de *hash* diferente em um dos nodos do que era na árvore original e com isso, é possível verificar se houve ou não mudança no conteúdo da árvore.

Figura 2 – Exemplo de uma árvore de Merkle.



Fonte: Elaborada pelo autor

2.4 ASSINATURA DIGITAL

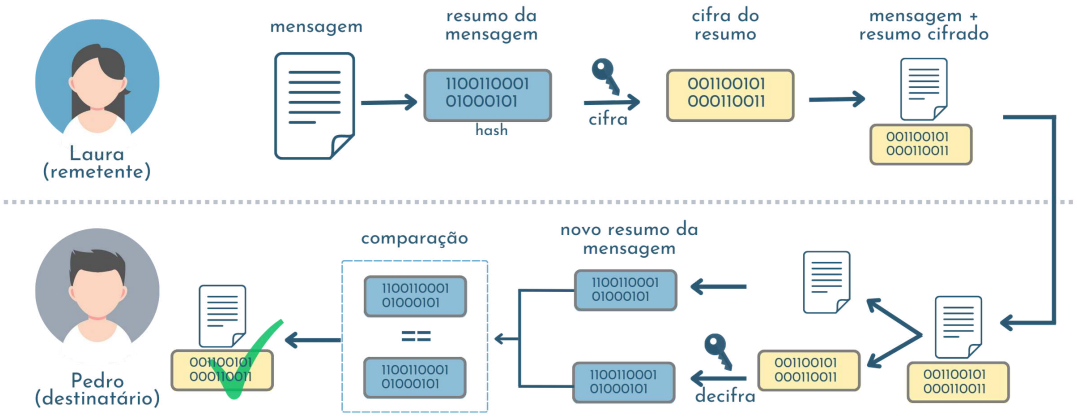
Na atualidade, a assinatura é necessária para qualquer tipo de negócio que utilize contrato e confiança. A falsificação é um entrave no mundo dos negócios, já que as assinaturas à mão são facilmente fraudadas para se obter vantagens ilícitas. Ademais, a utilização da assinatura manuscrita, na maioria das vezes, requer um procedimento extremamente burocrático para verificação de sua autenticidade, o que envolve despesas, deslocamento e tempo.

Nesse sentido, a assinatura digital, também apresentada pela primeira vez por DIFFIE, W. HELLMAN, M. (1976), propõe a criação de um sistema capaz de substituir o uso de assinaturas manuscritas e acabar com grande parte da sua burocracia. A assinatura digital, assim como criptografia assimétrica, utiliza duas chaves, pública e privada. Assim, uma é utilizada para realizar a assinatura e a outra para sua verificação.

A utilização da assinatura digital garante a autenticidade e a integridade do documento. O método funciona de uma forma em que o usuário utiliza a chave privada junto ao o documento para gerar uma saída assinada, que pode ser verificada com o uso da chave pública. A integridade é garantida devido ao fato de que, se o documento for minimamente alterado, a assinatura se torna inválida. A chave privada, que é única para cada pessoa, assegura a autenticidade, já que se

um documento for assinado, é certo que foi assinado pelo dono da chave privada.

Figura 3 – Fluxo de assinatura digital.



Fonte: Elaborada pelo autor

3 INFRAESTRUTURA DE CHAVES PÚBLICAS

A infraestrutura de chaves públicas (ICP) consiste em um conjunto de regras definidas pela *request for comments* (RFC) 2822. RFC é, em suma, um livro de regras que visa garantir a padronização e a segurança de documentos na internet. ICP, por sua vez, é definida por [4] como um "conjunto de hardware, software e pessoas, para manusear, armazenar e distribuir certificados digitais".

A criptografia de chaves públicas consegue, através de uma infraestrutura de mecanismos de segurança, controlar essas informações e assegurar sua confiabilidade. Com a RFC padronizando as regras para a ICP, aplicações dentro da internet podem usar como base essas regras para criar a sua infraestrutura, que é um componente essencial para a estratégia de segurança na internet. [4]

O papel da ICP é emitir, gerenciar, armazenar e revogar certificados digitais, garantindo a confiabilidade e segurança do sistema. Para isso, a ICP utiliza-se de uma cadeia de confiança [5].

3.1 CADEIA DE CONFIANÇA

A fim de verificar a validade do emissor de um determinado certificado, utiliza-se o que se convencionou chamar de “cadeia de confiança”, que consiste, em suma, em uma forma de hierarquia em que um certificado é emitido/assinado por outro certificado que possui um nível de hierarquia superior àquele.

Na prática, a cadeia de confiança é constituída, inicialmente, pela chamada “autoridade certificadora raiz”, ou “autoridade certificadora nível 1”. Essa autoridade certificadora, por originar a cadeia de confiança, funciona como uma âncora de confiança em relação à integridade da cadeia em si. Assim, se a autoridade certificadora raiz existe e é válida, presumem-se válidos os certificados digitais dela decorrentes.

Seguindo o nível de hierarquia da cadeia de confiança, têm-se a "autoridade certificadora intermediária"ou "autoridade certificadora nível 2"a qual consiste, basicamente, em um "isolamento entre a AC e o certificado de autoridade final" [5]. Os certificados intermediários possuem função administrativa, e podem ser utilizados para finalidades específicas (a exemplo de emissão certificados de assinatura de código), e também para conferir a confiança da AC raiz para outras organizações.

Por fim, o certificado de entidade final, último elo da cadeia de confiança, serve para, através da autoridade certificadora intermediária, conferir a confiança da autoridade de certificação raiz a uma entidade - site, empresa, governo - ou pessoa. O certificado de entidade final, ao contrário das autoridades certificadoras raiz e intermediária, não possui capacidade de emitir certificados adicionais; tanto é assim que é conhecido como elo final da cadeia de confiança: dele, não decorrem mais certificados digitais.

Merece menção neste trabalho a existência da “autoridade de registro”. Conhecida no meio da computação simplesmente como "AR", tem como principal papel o de conectar o

usuário e AC, verificando os documentos do titular, conferindo identidade da pessoa física ou pessoa jurídica que solicitou a certificação digital e solicitando à AC a emissão do certificado. Através dessa estrutura, a ICP é capaz de garantir segurança às autoridades certificadoras, ao mesmo tempo em que garante privacidade aos dependentes dos certificados de entidade final. [5]

3.2 CERTIFICADO DIGITAL

Certificado digital consiste em um conjunto de informações sobre a chave pública e sobre a identidade do dono dentro de um único documento. Esse conjunto de elementos deve ser confiado e assinado por um terceiro, geralmente uma autoridade certificadora, que pode ser uma entidade pública ou privada. No Brasil, por exemplo, têm-se a cadeia de autoridades certificadoras da ICP-Brasil, pertencente ao governo federal, realizando o papel desse terceiro e garantindo a confiabilidade do documento.

Hoje, o certificado digital é utilizado como uma espécie de identidade, com validade jurídica equivalente ao CPF e ao CNPJ. A assinatura de documentos tem sido a utilização mais comum do certificado digital. Ao assinar um documento, o certificado com os dados do proprietário são inseridos no documento. Assim, poderá ser realizada a verificação da assinatura e relacioná-la com o respectivo titular.

3.2.1 Emissão de Certificados

Para emitir um certificado, é necessário que a pessoa ou instituição requerente entre em contato com um órgão de registro e repasse suas informações e pretensões, gerando uma requisição, que conterá um nome, a sua chave pública e a sua assinatura, que será usada pela autoridade certificadora para verificação de identidade. Este documento deve ser enviado para uma autoridade certificadora ou algum outro terceiro, que seja publicamente confiável, o qual fará a averiguação das informações que lhe foram repassadas. Caso não aceito, o requerente deve realizar o pedido de uma outra requisição, realizando as alterações que foram anteriormente recusadas pela AC. Caso seja aceita, a AC irá coletar as informações da requisição e a chave pública, a fim de inseri-las em um certificado digital assinado por ela mesma, o qual contém o nome e a chave pública do titular do certificado e o nome e a assinatura da autoridade certificadora [4].

3.2.2 Verificação de Certificado

A verificação de certificados digitais é baseada em confiança e credibilidade da autoridade escolhida para realizar a emissão. Para fazer a averiguação de autenticidade basta realizar a checagem da assinatura. Se a assinatura for de uma autoridade certificadora válida e confiável, então o certificado também é válido e confiável. Para verificar se a autoridade certificadora é confiável, basta aplicar a mesma lógica para toda a cadeia até chegar no seu topo, que chamamos

de *"root"*. Após toda essa verificação, a confiabilidade do certificado inicial pode ser garantida. Trata-se da cadeia de confiança, tema já explorado no tópico 3.1 deste trabalho [6].

Também faz parte da verificação de certificado a análise de sua validade, que consiste em descobrir se o certificado ainda se encontra válido. Para a correta compreensão da verificação da validade do certificado, necessário compreender o conceito de revogação, que consiste, em suma, no "cancelamento do certificado"antes do prazo de vencimento. Há vários motivos que podem gerar a revogação de um certificado: comprometimento de sua chave privada; comprometimento da AC que emitiu o certificado; inserção de informações erradas, etc [4]. A depender do motivo, o próprio usuário faz o pedido de revogação, preenchendo um formulário para provar que ele realmente é o titular do certificado. Em casos de comprometimento da cadeia, a própria AC realiza a revogação. Em ambos os casos, a autoridade certificadora emite uma lista assinada contendo todos o certificados que foram revogados até determinado momento e essa lista deve ser verificada constantemente, para evitar que certificados digitais sejam utilizados de forma incorreta. O local onde essa lista é publicada é inserido dentro do certificado como uma extensão.

3.3 CARIMBO DO TEMPO

Carimbo do tempo consiste em um conjunto de caracteres inseridos dentro de um determinado documento para registrar a data e hora que um determinado evento ocorreu. Sua finalidade é servir como uma prova de integridade, garantindo a existência do documento naquele momento específico [7]. Um carimbo do tempo confiável mantém um rastreio de um documento desde sua criação até o ponto atual, garantindo que ninguém tenha feito nenhuma modificação sem que tenha sido registrado, e dessa forma, as assinaturas digitais são validadas. O carimbo do tempo é emitido por uma terceira parte, que é confiável por ambos os participantes da cerimônia, chamados de Autoridades de Carimbo do Tempo (ACTs). Essas autoridades devem seguir políticas estabelecidas e auditadas por uma fonte confiável. No Brasil, uma das entidades responsáveis por essa regulamentação é o Instituto Nacional de Tecnologia da Informação (ITI).

3.4 CERTIFICADO DE USO ÚNICO

One-Time Certificates (OTC), ou certificado de uso único, expressão adotada neste trabalho, tem o propósito de superar os desafios de gerenciamento de chaves e revogação. Assim, cada certificado digital é utilizado exclusivamente para assinar um único documento, o que impede a invalidação de outras assinaturas caso ocorra a perda da chave privada. Após a assinatura, apenas a chave pública é utilizada, o que permite que a chave privada seja 'destruída' por não ser mais necessária, eliminando-se a necessidade de armazenamento prolongado da chave privada do usuário. Como a assinatura ocorre imediatamente após a emissão do certificado, assume-se que todos os atributos estavam válidos nesse momento [1]. Em decorrência disso, o

[1] A afirmação só pode ser feita pois é responsabilidade da Autoridade Certificadora validar os atributos antes de realizar uma emissão de certificado.

uso de um carimbo do tempo torna-se dispensável, assim como elimina-se a necessidade de validar os dados no momento da assinatura. Ademais, a eliminação da validação de dados e da simplificação do gerenciamento da chave privada tornam a revogação desnecessária. Isso ocorre porque o certificado é restrito a uma única utilização, e os elementos mencionados asseguram a sua validade durante o curto período em que é empregado para realizar a assinatura [8].

3.5 PROVEDOR DE IDENTIDADE

O Provedor de identidade (IdP) armazena e gerencia os dados digitais dos usuários. Ele pode verificar a autenticidade de um usuário através de um conjunto de informações, como usuário e senha. A identidade do usuário está associada, principalmente, a três fatores: (i) Conhecimento, que se refere a um segredo conhecido apenas pelo usuário, como uma senha; (ii) Posse, que implica ser proprietário de algo que pode ser usado como prova, como um celular; e (iii) Características particulares, como impressão digital e retina ocular. O IdP trabalha para estabelecer quais usuários têm acesso a quais tipos de dados, além de ser responsável por armazenar as informações do cliente de forma segura, adotando as medidas necessárias de segurança [9].

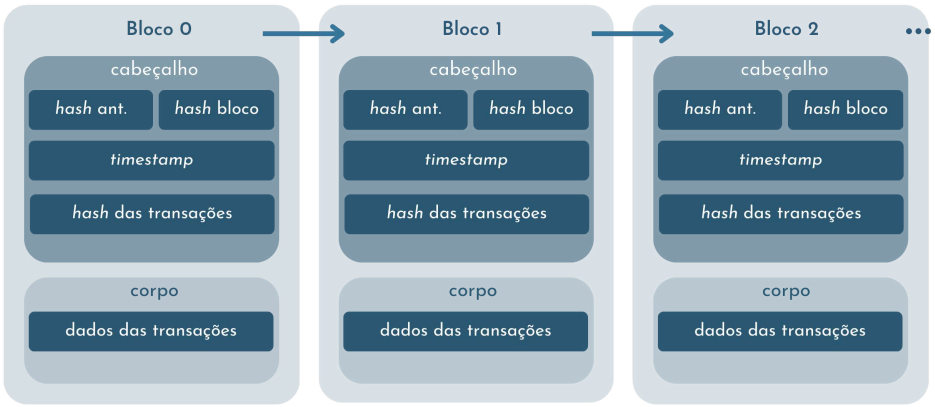
4 *BLOCKCHAIN*

Blockchain, citada pela primeira vez em [10] por Satoshi Nakamoto. No trabalho, foi proposto o conceito de *Bitcoin*, que utiliza a tecnologia *Blockchain* para realizar transações financeiras de maneira confiável, sem uma autoridade centralizadora do poder. A popularização da *Blockchain* na última década aumentou massivamente em função da valorização do *Bitcoin* como moeda monetária. Apesar da tecnologia ter sido desenvolvida com fins financeiros, foi-se percebendo diversas outras aplicações e utilidades em variadas áreas.

Todo o funcionamento da *Blockchain* gira em torno de uma estrutura de dados conhecida por *Merkle Tree*, já explorada no presente trabalho. Esta é utilizada para armazenar dados que são separados em blocos, e cada um desses contém uma estrutura com as informações necessárias para tornar a *blockchain* funcional. Dentro de cada bloco há o cabeçalho e corpo. O cabeçalho é composto pelo *hash* do bloco anterior, o *hash* do bloco atual, um carimbo do tempo (visto em 3.3) e um *hash* das transações ou dados adicionados. O corpo contém os dados que se deseja registrar na *Blockchain*.

Estes blocos são interligados entre si (como uma corrente). Para realizar os encadeamentos entre os blocos, insere-se o resumo criptográfico (*hash*) do bloco antecessor no bloco atual, ou seja, cada bloco contém uma referência ao bloco anterior. Isso implica a inviabilidade de inserção ou alteração de informações no meio da cadeia, pois, para isso, seria necessário que a nova informação contivesse exatamente o mesmo *hash* da informação que se deseja ser substituída, e a dificuldade para realizar colisões de *hash* é demasiadamente complexa para o potencial computacional que se tem hoje em dia.

Figura 4 – Estrutura dos blocos de *blockchain*.



Fonte: Elaborada pelo autor

O uso da *blockchain* tornou-se comum, também pelas suas características de descentralização, pseudo-anonimidade, persistência e auditabilidade. A descentralização se traduz no

fato de que não é necessário um poder centralizado para manter o controle de uma determinada aplicação da *blockchain*. O maior exemplo pode ser o próprio *Bitcoin*, no qual não se faz necessário a existência de uma entidade de controle monetário central para vistoriar, contudo, a moeda é aceita e confiada no mercado financeiro. Apenas a existência da *blockchain* para realizar o registro de transações já é o suficiente, ao contrário de grandes bancos que necessitam de um poder centralizado que realiza a gerência e movimentações financeiras. A anonimidade se beneficia do fato de que cada usuário pode interagir com a *blockchain* utilizando endereços falsos, o que acaba por esconder a sua verdadeira identidade. Porém, algumas pesquisas e operações realizadas por agências de inteligência mostraram que é possível relacionar movimentações dentro da *blockchain* com uma pessoa física, por isso *blockchain* foi deixada de ser considerada totalmente anônima para ser uma implementação pseudo-anônima. Todos as transações precisam ser confirmadas e gravadas em blocos, e para serem adicionados em uma rede, é necessário receber uma confirmação de vários membros dessa rede. Isso implica a dificuldade de realizar falsificações e também a facilidade de verificação, já que as novas transações devem ser checadas antes serem inseridas. Todas as transações na *blockchain* são validadas e registradas junto a um carimbo do tempo. Com isso, o usuário consegue auditar e buscar por registros antigos a partir de qualquer nodo, baseando-se na data e hora que o nodo inicial foi incorporado [11].

4.1 TAXONOMIA DA *BLOCKCHAIN*

Existem, hoje, tipos diferentes de *blockchains*, como: públicas, privadas e *consortium*. Todas possuem as mesmas propriedades, porém se diferenciam nas características de cada uma dessas propriedades, que são: determinação de consenso; centralização; eficiência; imutabilidade e permissão de leitura [12].

- Determinação do consenso: a determinação do consenso na *blockchain* deve ser determinada pelos seus participantes. Em uma *blockchain* pública, todos os nodos participantes podem se juntar ao processo; já na comissionada, somente participantes selecionados irão realizar a confirmação do bloco. Por fim, em uma *blockchain* privada, uma autoridade central realiza a referida confirmação.

- Permissão de leitura: a *blockchain* pública geralmente tem permissão de escrita pública, ou seja, qualquer usuário pode realizar a inserção de um bloco. Em contra partida, nas *blockchains* privada e comissionada, apenas usuários selecionados têm esse direito. Dessa forma, os organizadores podem definir quais informações serão inseridas, e dentre essas, quais serão públicas.

- Imutabilidade: como já mencionado, as transações devem ser validadas por todas as partes, e ainda são conectadas por resumos criptográficos. Com isso, a alteração de informações já armazenadas torna-se quase impossível nas *blockchains* públicas; já nas privadas e *consortium*, depende da vontade de seu administrador.

- Eficiência: na *blockchain* pública, há vários validadores e cada inserção de uma nova transação deve ser validada por, se não todos, uma grande quantidade de participantes. Somando isso ao fato de que tem-se um apelo em manter a segurança dos dados, sua eficiência deixa a desejar, diferente dos outros tipos mais restritos, que com menos validadores podem ser mais eficazes.

- Centralização: essa característica se diferencia para cada um dos tipos citados acima (pública, privada e *consortium*): pública contém uma descentralização total do poder; a *consortium* é parcialmente centralizada; já a privada é controlada por uma autoridade central.

4.2 *SMARTCONTRACTS*

A utilização da *blockchain* para uso financeiro, um recurso sensível e que requer segurança computacional dobrada, fez necessária a criação de um mecanismo inteligente para realizar operações e manter sua confiabilidade. Os *smartcontracts* surgiram para suprir a necessidade descrita. São códigos de programação armazenados na *blockchain* e são executados quando pré condições, definidas no contrato, forem alcançadas.

Sua principal função é automatizar acordo entre as partes e caso executado, implica a concordância de todos os participantes com seus termos. A agilidade obtida por se usar um contrato inteligente é expressiva, em função de que suas regras e operações são postas em prática e armazenadas na *blockchain* no instante que a condição é aceita, evitando o dispêndio de tempo e diminuindo a burocracia. Ademais, pelo fato de ele ser inserido na *blockchain*, torna-se praticamente impossível que uma das partes altere as condições do contrato para seu auto-benefício, em razão de este ser parte do conteúdo de um bloco. A transparência também é um dos benefícios, pois dispensa o envolvimento de terceiros para garantir a confiabilidade.

4.3 CONSENSO

O mecanismo de consenso é um método que busca validar entradas em bases de dados, de forma a mantê-las seguras. No âmbito da *blockchain*, essa entrada equivale à inserção de um novo bloco. Existem vários consensos para determinar qual bloco receberá o direito de ser introduzido na *blockchain*. Os mais conhecidos são *proof of work* e *proof of stake*.

O primeiro, utilizado na implementação da *Bitcoin*, baseia-se na quantidade de poder computacional que pode, analogamente, ser visto como trabalho, ou seja, quanto mais poder computacional é utilizado (ou quanto mais trabalho é feito), mais altas são as chances de receber o direito de ter o seu bloco introduzido na *blockchain*. O trabalho de encontrar o bloco a ser encadeado é realizado pelos "mineradores", em razão de o processo da procura ser chamado de "mineração". A cada bloco encadeado, uma recompensa é dada ao dono do bloco como uma forma de retribuição pelo esforço realizado para encontrar o bloco correto.

Proof of stake foi criado como uma alternativa ao *proof of work*, pois reduz consideravelmente a quantidade de poder computacional para verificar blocos, uma vez que são utilizados recursos - muitas vezes, criptomoedas - para sortear quem se tornará um validador. Os proprietários dos recursos os oferecem como uma forma de aposta, a fim de receber o direito de validar e inserir um novo bloco. Quanto maior a oferta, maior a chance de "vencer". Esse consenso busca promover a sustentabilidade ambiental, diferente do *PoW*, que necessita de uma quantidade massiva de energia para manter os robustos equipamentos de mineração online [11].

4.4 APLICAÇÕES

A versatilidade da *blockchain* em ser adaptada para inúmeros usos em diversas áreas também contribuiu para que ganhasse destaque como uma tecnologia inovadora nos últimos anos. Os parágrafos a seguir se debruçarão sobre breves resumos de aplicações da *blockchain* nos dias atuais [12].

4.4.1 Segurança

Pode-se utilizar essa nova tecnologia para aumentar a segurança de mecanismos já existentes, como por exemplo os anti-vírus. O funcionamento de um anti-vírus hoje em dia baseia-se no reconhecimento de padrões de arquivos maliciosos. Para isso, existe um servidor central que armazena todo os modelos de vírus para poder realizar uma filtragem no computador do usuário. Contudo, esse servidor pode sofrer ataques maliciosos e ter seus arquivos alterados, fazendo com que parem de detectar certos padrões, o que dificilmente aconteceria caso esses dados estivessem armazenados em uma *blockchain*, devido à sua característica de imutabilidade, anteriormente mencionada [12].

4.4.2 Financeiro

A aplicação mais comum para sua utilização mostrou e ainda mostra que tem grande potencial para expandir dentro do mercado financeiro. Desde o surgimento do *Bitcoin* novas moedas são criadas com alta frequência, como a *Etherium*, hoje tão conhecida quanto *Bitcoin*. Nos últimos anos, se vivenciou a explosão dos *non-fungible tokens* (NFTs), que são basicamente certificados de autenticidade de posse. Em [13], discute-se o potencial que criptomoedas possuem de se equipararem a bancos financeiros "físicos".

4.4.3 Social

Novamente fundamentando-se na característica da imutabilidade, a tecnologia *blockchain* tem sido aplicada com sucesso em diversos setores sociais. No âmbito do registro de terras, informações como propriedade, descrição do terreno e alterações realizadas por órgãos

governamentais já começaram a ser inseridas na *blockchain*, proporcionando maior transparência e segurança. Também há uma tentativa de encorajamento à adoção de fontes de energia sustentáveis através de recompensas em criptomoeda. Funcionaria da seguinte forma: o usuário investe na geração de energia solar e vende o excedente ao governo, realizando a transação por meio do pagamento com a criptomoeda "solarcoin", como mencionado em [14].

5 TRABALHOS RELACIONADOS

Neste capitulo, é apresentada uma revisão sistemática bibliográfica em 5.1, com intuito de identificar ideias que possam auxiliar no desenvolvimento do presente trabalho. Em seguida, na seção 5.2, é apresentada a seleção baseada nos resultados da revisão, e a analise dos artigos na seção 5.3. Logo, é realizada uma descrição detalhada do trabalho que mais se assemelhou à proposta, conforme abordado em 5.5. Além disso, é apresentado o protocolo que fundamentou e embasou a pesquisa realizada neste documento.

5.1 REVISÃO SISTEMÁTICA DA BIBLIOGRAFIA

Nesta seção, apresenta-se o processo de revisão sistemática da bibliografia para o projeto de criação de uma *blockchain* com finalidade de armazenar certificados de uso único. Para realizar o processo de levantamento do estado da arte, foram utilizadas as recomendações de kitchenham [15]. Este documento está organizado como segue. Na seção 5.1.1, apresenta-se o conjunto de palavras-chave que definem o objeto de pesquisa. Na Seção 5.1.2 apresenta-se as consultas e as base de dados de trabalhos científicos escolhidas. Por fim, a Seção 5.2 apresenta o processo de seleção e um resumo sobre os trabalhos mais relevantes para a presente pesquisa.

A continuidade do trabalho será realizada a partir dos trabalhos escolhidos ao final da seleção.

5.1.1 Palavras-chave e Sinônimos

- **Blockchain** - Distributed ledger, cryptographic ledger, digital ledger
- **Certificate Transparency** -
- **Digital Certificate** - Identity certificates, public key certificates
- **Timestamp** - Timecode

5.1.2 Consultas

A Tabela 1 apresenta as consultas utilizadas em cada uma das base de dados selecionadas.

Tabela 1 – Consultas.

Base de dados	Consulta
Google Scholar	"blockchain or distributed ledger or cryptographic ledger or digital ledger"and "certificate transparency"and "digital certificate or identity certificates or public key certificates"and "timestamp or timecode"
ACM	"blockchain or distributed ledger or cryptographic ledger or digital ledger"and "certificate transparency"and "digital certificate or identity certificates or public key certificates"and "timestamp or timecode"
IEEE	"blockchain or distributed ledger or cryptographic ledger or digital ledger"and "certificate transparency"and "digital certificate or identity certificates or public key certificates"and "timestamp or timecode"
Springer	"blockchain or distributed ledger or cryptographic ledger or digital ledger"and "certificate transparency"and "digital certificate or identity certificates or public key certificates"and "timestamp or timecode"

5.1.3 Resultados

Tabela 2 – Reultados da Revisão Sistemática.

Base de dados	Pesquisa inicial	Selecão 1	Selecão 2
Google Scholar	88	3	2
ACM	3	0	0
IEEE	6	2	2
Springer	4	0	0
Total	**101**	**5**	**4**

5.2 SELEÇÃO

Nesta seção, descreve-se o processo de seleção dos trabalhos relacionados, cujos resultados são apresentados na tabela 2. Inicialmente, foram realizadas consultas nas bases de dados de revistas e conferências, utilizando palavras-chave definidas na seção 5.1.1. A Seleção 1 foi composta por trabalhos cujos títulos estavam relacionados à temática. Na Seleção 2, foram escolhidos os trabalhos com resumos relacionados. Por fim, na seção 5.3, é apresentado um breve resumo sobre os trabalhos selecionados após a leitura.

5.2.1 Seleção 1

1. Google Scholar

    - **CertLedger: A new PKI model with Certificate Transparency based on blockchain [16]**.

    - **PBCert: Privacy-Preserving Blockchain-Based Certificate Status Validation Toward Mass Storage Management [17]**.

    - **Certificate validation using blockchain [18]**.

2. IEEE

    - **Certificate Transparency Using Blockchain [19]**.

    - **TrustCA: Achieving Certificate Transparency Through Smart Contract in Blockchain Platforms [20]**.

5.2.2 Seleção 2

1. Google Scholar

    - **CertLedger: A new PKI model with Certificate Transparency based on blockchain [16]**.

    - **PBCert: Privacy-Preserving Blockchain-Based Certificate Status Validation Toward Mass Storage Management [17]**.

2. IEEE

    - **Certificate Transparency Using Blockchain [19]**.

    - **TrustCA: Achieving Certificate Transparency Through Smart Contract in Blockchain Platforms [20]**.

5.3 ANÁLISE DOS TRABALHOS SELECIONADOS

Nesta seção, são apresentados os resumos dos trabalhos selecionados. Nas seções 5.5 e 5.4, realiza-se um aprofundamento nos trabalhos que apresentaram maior alinhamento com as propostas desta pesquisa.

5.3.1 *Certificate Transparency Using Blockchain*

O projeto de Transparência de Certificados do Google fez com que vários trabalhos de pesquisa analisassem a adição de transparência para melhor monitoramento da autoridade certificadora, efetivamente por meio de *logs* públicos de todos os certificados emitidos pelas autoridades de certificação. Neste artigo, aproveitando o progresso recente na tecnologia *blockchain*, é proposto um novo sistema, chamado CTB, que torna impossível para uma CA emitir um certificado para um domínio sem obter o consentimento do proprietário. Além disso, a CTB foi equipada com o mecanismo de revogação de certificado.

5.3.2 *TrustCA: Achieving Certificate Transparency Through Smart Contract in Blockchain Platforms*

Neste artigo, são utilizadas as características naturais de imutabilidade e transparência em plataformas *blockchain* para criar uma entidade denominada AC *proxy*, a qual gerencia o ciclo de vida dos certificados digitais. É proposta uma nova arquitetura de sistema, que permite uma fácil integração da AC *proxy* com as AC atuais por meio da aplicação do serviço *blockchain*. Nesta arquitetura, a AC *proxy* e as ACs podem realizar o gerenciamento dos certificados. Aproveitando a característica de transparência da *Blockchain* para gerenciar o ciclo de vida dos certificados digitais, é possível resolver as deficiências do modelo de confiança das Autoridades Certificadoras (ACs) tradicionais, além de aprimorar a segurança.

5.3.3 *PBCert: Privacy-Preserving Blockchain-Based Certificate Status Validation Toward Mass Storage Management*

A sincronização de dados entre os servidores de *log* da *Certificate Transparency* pode resultar em latência na rede. O trabalho propõe esquemas de infraestrutura de chave pública baseados em *blockchain*, nos quais todas as operações de certificados são registradas na *blockchain* para fins de auditoria pública. Além disso, o trabalho aborda o armazenamento de certificados revogados, especialmente em cenários com grande volume de dados.

5.4 TRANSPARÊNCIA DE CERTIFICADOS

Certificate Transparency (ou CT) é um protocolo de segurança na internet que tem como objetivo corrigir várias vulnerabilidades presentes no padrão *SSL/TLS*. Suas especificações estão definidas na *RFC* 6962, que foi onde o CT foi introduzido pela primeira vez, e na *RFC* 9162, versão 2.0 que aborda falhas estruturais da versão anterior. Em ambas as versões, o CT é definido como uma estrutura de dados na qual é permitido apenas anexar informações, sem a possibilidade de removê-las. Essa estrutura funciona de maneira semelhante a um *log*,

registrando os certificados emitidos pelas Autoridades Certificadoras e tornando esse registro publicamente acessível para verificação por qualquer pessoa [21].

Um dos principais problemas da infraestruturas de chaves públicas da web (ICP) é a necessidade de confiar cegamente nas autoridades certificadoras, o que aumenta o risco de emissão de certificados falsos caso uma autoridade seja comprometida. Um exemplo notável ocorreu em 2011, quando a DigiNotar, uma autoridade certificadora holandesa, foi hackeada, resultando na emissão de certificados com domínio do Google (*.google.com) e comprometimento dos dados de usuários iranianos. Esse ataque foi detectado rapidamente pela Google, utilizando uma técnica muito arriscada chamada *keyPinning*, a qual só costuma ser executada por empresas com grande experiência. Se a técnica não fosse utilizada, essa detecção poderia levar meses ou anos. O CT busca mitigar o problema de detecção de erros de emissão, sem recorrer a técnicas arriscadas, adicionando todos os certificados emitidos a um *log* público. O *log* em si não impede que erros ocorram, mas reduz o tempo necessário para identificá-los. Para que o *log* seja confiável por determinados *browsers*, ele deve seguir algumas regras como: ter o registro dos documentos e dados, utilizando tipos e estruturas definidas na RFC, manter-se ativo 99% do tempo, e realizar a inserção de novos certificados no *log* dentro de um prazo conhecido como *Maximum Merge Delay* (MMD) [21].

A submissão de novos certificados ao *log* pode ser realizada por qualquer entidade. É possível submeter um certificado ou um pré certificado, que consiste em um arquivo CMS contendo todas as informações do certificado. Quando uma AC envia um pré certificado ao *log*, ela confirma sua intenção de assinar esse certificado. Considera-se um erro de emissão validar um pré certificado de um certificado que não foi assinado. Cada submissão no *log* deve vir acompanhada por todos os certificados adicionais necessários para verificar a cadeia até alcançar uma autoridade certificadora raiz confiável. Se todos os certificados adicionais enviados forem válidos, fica a critério de quem executa o *log* aceitar ou não a submissão. Caso seja aceita, o *log* retorna um *Signed Certificate Timestamp* (SCT) e um arquivo contendo a prova de inclusão. O SCT é uma estrutura que contem um carimbo do tempo, o "*LogID*", que é um valor único referente ao *log* específico e uma assinatura. Caso um pré-certificado tenha sido submetido, essa estrutura deve ser incluída como parte do certificado final quando for emitido. Se a submissão for feita a partir de um certificado já emitido, o proprietário do certificado deve se encarregar de incluir o SCT junto com o certificado, de alguma forma [21].

A adoção do protocolo de Transparência de Certificados ganhou destaque em 2015, quando o navegador *Google Chrome*, passou a exigir o registro de certificados emitidos para novos sites em um CT confiável pelo navegador. Em 2018, essa exigência se tornou obrigatória por todos os sites. O registro em um *log* público aumenta a confiabilidade do certificado, pois além da confiança na AC, também existe a confiança de quem fiscaliza a CT [21].

A implementação do CT trouxe novas funções para a estrutura da ICP, como operadores do *log*, auditores e monitores. O operador do *log* é responsável por receber e adicionar novos certificados à lista de certificados do CT. Essa adição deve ocorrer dentro de um período de tempo conhecido como *Maximum Merge Delay*(MMD), que é o tempo máximo definido na

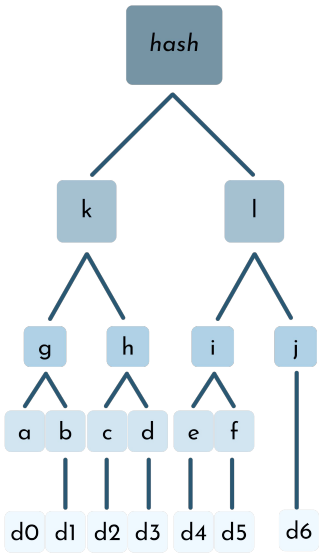
RFC 9162 para que o novo certificado seja adicionado ao registro de transparência. Atualmente, esse tempo é de 24 horas. Esta etapa é uma das várias regras que um CT deve seguir para ser considerado funcional para o mercado de ICP.

A função do monitor é fiscalizar cada passo de um CT *log*. Sempre que um novo certificado é adicionado no *log*, o monitor comunica ao domínio daquele site que um novo certificado foi emitido em seu nome, permitindo a identificação rápida de falhas na emissão e a solicitação de revogação do certificado.

O auditor verifica se as regras estabelecidas pela RFC e pelo *browser* estão sendo seguidas pelo CT, como manter-se ativa em 99% do tempo, incluir novos certificados dentro do tempo definido pelo *Maximum Merge Delay*(MMD) que, por exemplo, no *browser Chrome* é de 24 horas, entre outras regras. O processo de auditoria pode ser realizado por monitores do CT, ou por qualquer outra pessoa interessada. Grandes corporações têm interesse em monitorar esses registros, pois são alvos potenciais de ataques, o que leva empresas como *Facebook* e *Cloudflare* a implementar em seus próprios sistemas de monitoramento [21].

A prova de inclusão, enviada ao usuário como um dos parâmetros de retorno após a inclusão de um novo certificado, consiste em uma lista de *hashes* necessários para calcular o *hash* da raiz da árvore de inclusão de certificados. A imagem abaixo ilustra o que foi explicado:

Figura 5 – Estrutura de uma árvore de Merkle no contexto de *Certificate Transparency*.



Fonte: Elaborado pelo autor

Observando a imagem, verifica-se que, se a nova informação adicionada fosse o elemento ’*d*0’, a lista retornada seria o *hash* dos nodos b,h,l. Dessa forma, a partir de ’*d*0’, já é possível obter o valor de ’*a*’. Ao realizar o cálculo da soma de ’*a*’ e ’*b*’, considerando que ’*b*’ foi retirado da lista de retorno, pode-se alcançar o *hash* ’*g*’, e assim por diante, até chegar na raiz, provando a existência da inclusão desta folha [21].

5.5 TRANSPARÊNCIA DE CERTIFICADOS USANDO *BLOCKCHAIN*

O trabalho intitulado *certificate transparency using blockchain* [19] propõe a utilização do protocolo de *Certificate Transparency* utilizando uma *blockchain*, abreviada como CTB. De acordo com os autores, essa abordagem adiciona uma nova camada de confiança ao protocolo de comunicação SSL/TLS. O fluxo inicial é semelhante ao da CT, no qual um domínio solicita a emissão de um certificado a uma autoridade certificadora. No entanto, é necessário que essa AC faça parte de uma rede da CTB e tenha capacidade de inserir o certificado na blockchain. Neste contexto, para verificar a autenticidade de um certificado, o cliente pode realizar a busca na *ledger* da rede CTB e confirmar sua inclusão.

Outra proposição feita pelo trabalho é a melhora do gerenciamento de certificados revogados. A existência de um certificado é necessária, porém, não é suficiente para determinar a sua validade, e para isso, é necessária a emissão de uma lista de certificado revogados, que apesar de funcional, é vulnerável. Uma autoridade certificadora emite uma CRL, de tempo em tempo e com isso, gera uma brecha de tempo onde um certificado já revogado ainda não se encontra na CRL, pois o tempo decorrido não foi suficiente para emitir uma nova lista. O documento, então, propõe o uso de uma funcionalidade disponibilizada pelo *framework* de *blockchain* escolhido, chamada *world state*, que consiste no armazenamento da ultima alteração feita em um determinado dado da blockchain em um banco de dados. A proposta explica que com a possibilidade de alterar a validade do certificado, bastaria pesquisar a última alteração realizada no dado referente ao certificado de interesse para ter ciência da sua condição.

6 ESCOLHA DE ARQUITETURA

O presente trabalho utiliza um protocolo - denominado Cripparency - desenvolvido para pôr à prova a proposta que será apresentada. Durante a implementação do protótipo, foram realizadas diversas escolhas relacionadas à arquitetura de software. A escolha de maior relevância consistiu na eleição do *framework* de *blockchain* que será utilizado no presente trabalho, decisão da qual decorreram todas as demais escolhas. Assim, elegeu-se o *framework HyperLedger Fabric*, sob a justificativa de que se apresenta como uma ferramenta sólida e satisfatória em relação às funcionalidades disponíveis. Para plena compreensão da motivação da escolha, passa-se a detalhar a estrutura que compõe uma rede *HyperLedger Fabric* [22].

6.1 HYPERLEDGER FABRIC

HypherLedger Fabric é um projeto de código aberto que possui funcionalidades singulares, diferenciando-se das *blockchains* tradicionais como *Etherium* e *Bitcoin*. Criado pela *HyperLedger Foundation*, o projeto *Fabric* possui, atualmente, mais de 200 desenvolvedores de mais de 25 empresas, tais como *Linux Foundation* e IBM, os quais dão suporte necessário para manter o projeto sólido e atualizado. O *Fabric* representou verdadeira inovação no âmbito das *blockchains*, pois permitiu que os contratos inteligentes fossem escritos em linguagens de programação usuais - Java, Go e Node.js. [23] - tornando mais acessível sua utilização pelos programadores, o que contribuiu para o seu sucesso [22].

A configuração do ambiente e arquitetura é dotada de ampla liberdade de alteração pelo desenvolvedor. A *Hyperledger* permite que o desenvolvedor atue com autonomia ao ajustar organizações, usuários, permissões, entre outros componentes da arquitetura da rede [22].

Por não utilizar o consenso de *Proof of Work (PoW)*, não há necessidade do uso de uma *criptomoeda* de incentivo financeiro a fim de atrair mineradores para gerar novos blocos. Por outro lado, o mecanismo de consenso comumente utilizado no projeto *HyperLedger Fabric* é customizável, gerando uma gama de possibilidades que tonará possível a adequação do protocolo à necessidade de qualquer caso concreto. Esse conjunto de características implica melhoria de desempenho no processamento de transações, e também qualifica a rede a aumentar a privacidade das transações [24].

Diante do exposto e considerando que a flexibilidade proporcionada pela *HyperLedger Fabric* em relação às decisões de arquitetura adequou-se às finalidades da pesquisa, optou-se pela utilização deste *framework*.

6.1.1 Permissionada

Uma *blockchain* permissionada, como a *HyperLedger Fabric*, parte da premissa de que os integrantes da rede são conhecidos. Não significa, contudo, que os participantes confiam uns nos outros, mas apenas que se conhecem. Essa propriedade é o que permite que o mecanismo de

consenso utilizado seja dotado de tamanha liberdade, possibilitando que cada rede o modifique, a fim de adequá-lo às suas requisições. Tradicionalmente, utilizam-se os mecanismos de consensos de *crash fault tolerant* (CFT) [1] ou *byzantine fault tolerant* (BFT) [2], os quais dispensam mineração [25].

Além da liberdade proporcionada pelo mecanismo de consenso, há aumento da confiança na rede, diante da maior dificuldade de realização de alterações maliciosas. Por conhecer os usuários da rede, cada alteração, invocação de *chaincodes* e utilização da aplicação são registradas na blockchain seguindo as regras estabelecidas pela rede. Assim, nas *blockchains* permissionadas, diferentemente do que ocorre nas *blockchains* não permissionadas, caso seja comprovado que a alteração foi realizada de forma a prejudicar a integridade do sistema, é possível identificar o autor e adotar as providências cabíveis [25].

6.2 CHAINCODE

Chaincode é um programa que pode ser escrito em Java, Go e Node.js. Este programa age como uma *interface* entre os participantes e a rede, que descreve a lógica das operações eventualmente invocadas, como busca e inserção. *Chaincode* representa, no âmbito da *Hyperledger* o mesmo que os contratos inteligentes representam para outras *blockchains*. Um canal pode ter instâncias de várias *Chaincodes*, possibilitando a definição de formas diversas de realização da mesma operação, mas seu uso não pode ser restrito a uma organização específica, devendo ser passível de utilização por todos os integrantes do canal [26] [27]. Por exemplo, havendo, no canal C1, as organizações A e B, e estando presentes neste canal as *Chaincodes* S1 e S2, tanto A quanto B podem realizar as operações descritas em ambas as *Chaincodes*.

Qualquer alteração em uma *Chaincode* deve ser aprovada pelos usuários do canal, por quórum previsto pelas regras da rede. O quórum de aprovação da alteração de uma *Chaincode* consiste, normalmente, na maioria das organizações que compõem o canal. Havendo a aprovação, a *Chaincode* é atualizada para incorporar a alteração realizada. Além disso, uma mesma *Chaincode* pode ser instanciada em um número indeterminado de canais dentro da rede. Porém, a partir do momento em que ela é introduzida em um canal, eventual alteração realizada por uma organização só produzirá efeitos dentro daquele canal, não influenciando outras instâncias [26] [28].

---

[1] É um conceito importante em sistemas distribuídos que visa alcançar o consenso entre as entidades participantes, mesmo em falhas benignas. Ao contrário do *Byzantine Fault Tolerance* (BFT), que lida com falhas maliciosas ou arbitrárias, o CFT assume que as falhas ocorrem de forma não arbitrária.

[2] É um conceito fundamental em sistemas distribuídos que visa garantir a segurança e o funcionamento correto mesmo quando algumas entidades do sistema se comportam de maneira maliciosa ou apresentam falhas arbitrárias. Para garantir essa tolerância, o protocolo emprega estratégia de comunicação e consenso robustas, geralmente envolvendo uma comunicação intensiva entre os nós e múltiplas etapas de verificação e validação.

6.2.1 Linguagem da Chaincode

- Java: Java é uma linguagem de programação de alto nível, orientada a objetos e de propósito geral. Ela foi desenvolvida pela Sun Microsystems (agora de propriedade da Oracle Corporation) e lançada em 1995. Desde então, tornou-se uma das linguagens de programação mais populares, sendo amplamente utilizada em todo o mundo [29].

- Node.js: Node.js é uma plataforma de desenvolvimento de software que permite a execução de código JavaScript. Diferentemente do JavaScript tradicional, que é executado no navegador, o Node.js estende o uso da linguagem JavaScript para ambientes de servidor, como servidores web [30].

- Go: Go, também conhecida como Golang, é uma linguagem de programação de código aberto criada pela Google em 2007. Ela foi projetada para ser eficiente, concorrente, segura e de fácil utilização. Go combina características de linguagens como C e C++ com recursos modernos para facilitar o desenvolvimento de software escalável e de alto desempenho [31].

6.3 TRANSAÇÕES

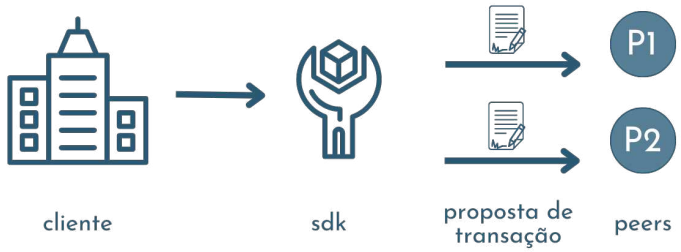
Para o êxito da execução da operação, ela deve conter os devidos parâmetros, os quais são concedidos por quem está executando a ação e postos em uma estrutura organizada, a qual é inserida na *blockchain*. Nesse contexto, a transação representa o conjunto de informações que foram utilizadas para realizar determinada operação na rede [32].

Nesse sentido, a transação é a forma utilizada para manter o registro dos dados relacionados ao histórico de operações ocorridas na rede dentro dos blocos da *blockchain* [32]. No *Bitcoin*, por exemplo, para saber como uma carteira obteve determinada quantidade de dinheiro, basta verificar todas as transações em que esta carteira esteve envolvida, seja de adição ou retirada da moeda. Ao final, a soma do valor de todos os dados das transações deve corresponder ao valor exato da carteira atualmente [10].

O registro de uma transação deve obedecer às regras da rede. No contexto da *Hyper-Ledger Fabric*, a realização de uma ação por uma organização se dá através de uma *Application Programming Interface* (API), que gerencia os dados da operação, convertendo-os no formato requerido pelo *Fabric* e cria uma proposta de transação, que nada mais é que um pedido de invocação de uma *chaincode*, assinando-a com as credenciais do requerente, a qual somente será executada se aceita pelas organizações alvos [32].

O nodo ordenador do alvo, por sua vez, aceita a proposta se verificar sua conformidade com as regras, notadamente em relação à necessidade de utilização de um formato adequado, validade da assinatura e aptidão do remetente à realização da ação descrita na proposta de transação. Além disso, deverá apurar se a transação sob análise não se encontra submetida, a fim

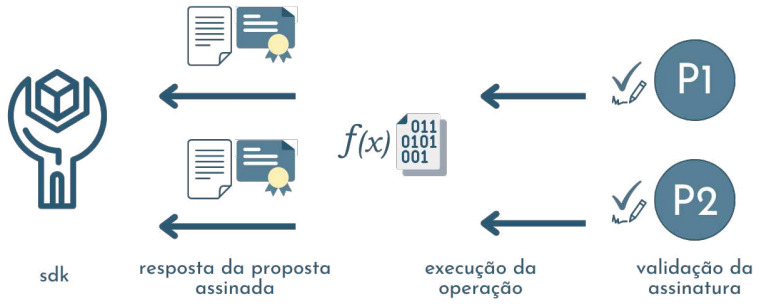
Figura 6 – Organização envia proposta de transação para *peers* das organizações alvo.



Fonte: Elaborada pelo autor

de evitar a duplicidade de dados e ocorrência de erros [3]. Aceita a proposta, o nodo ordenador realiza a chamada da função descrita na transação da *Chaincode*, produzindo seu resultado e assinando-o. Este, então, é inserido na proposta de transação, que é enviada à API que verifica as assinaturas e se a resposta condiz com a proposta inicial [32].

Figura 7 – *Peers* validam a assinatura, invocam a operação descrita na proposta de transação e devolvem a resposta à API.



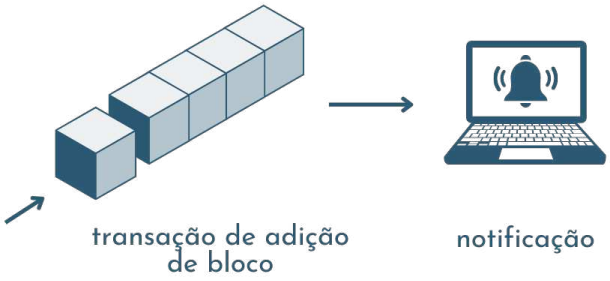
Fonte: Elaborada pelo autor

Verificada a validade das assinaturas e da resposta, inicia-se a geração do bloco que acomodará a transação proposta e será inserido na cadeia. A partir disto, os nodos recebem a nova atualização para replicação das mudanças em seus próprios *ledgers*, mantendo, assim, a sincronia dos dados. Por fim, notificam o cliente inicial acerca do êxito da incorporação da nova transação [32].

No âmbito deste trabalho, as transações são realizadas por diferentes entidades envolvidas na infraestrutura de chaves públicas (ICP), como a Autoridade Certificadora (AC), Provedor de Identidade e o Assinador. O assinador é responsável pela assinatura de documentos de forma digital. Geralmente, o usuário, ao utilizar esse recurso, já possui o certificado e o documento

[3] O HLF denomina esse tipo de proteção de *replay-attack protection*

Figura 8 – Atualização da cadeia e notificação da incorporação.



Fonte: Elaborada pelo autor

que deseja assinar. No entanto, neste trabalho, o uso de certificados de uso único requer que o próprio assinador entre em contato com a autoridade certificadora e o provedor de identidade a cada solicitação, para emitir um novo certificado. Autoridade Certificadora e Provedor de Id foram detalhadas na seções 3.1 e 3.5, respectivamente.

Como mencionado na seção 5.4, a detecção de falhas de emissão é baseada no monitoramento das atualizações no *log*. O sistema procura pela publicação de um certificado de seu interesse e verifica se houve pedido prévio para aquela emissão [21]. No presente trabalho, as transações serão monitoradas e utilizadas para identificar possíveis falhas, seguindo uma abordagem semelhante ao CT.

6.4 ESTRUTURA HLF

A *blockchain* do projeto *Fabric* diferencia-se das *blockchains* mais populares por ter, entre suas características, uma divisão peculiar de canais, organizações, *peers* e políticas organizacionais com vistas a auxiliar o controle de separação de tarefas. A seguir, passa-se ao detalhamento destas divisões.

6.4.1 Canais

O canal é uma funcionalidade fornecida pelo *Fabric* com objetivo de atuar como meio de comunicação entre as organizações da rede. Não há restrição quanto ao número de canais simultâneos, assim como pode ser inserido um número ilimitado de *peers* ou organizações [33]. Para uma melhor compreensão desse conceito, pode-se fazer uma analogia com um grupo de amigos. As organizações podem ser comparadas a um grupo de amigos, em que várias pessoas fazem parte e compartilham interesses e informações em comum. Esses interesses são o que distinguem um grupo de amigos (ou organizações, nesse caso) dos demais.

Para integrar uma *blockchain Fabric*, o participante deverá ingressar em um canal da rede já existente ou então, criado por ele, caso seja possível. Neste canal, há a criação de uma *Ledger* específica, a qual só pode ser lida e escrita por seus participantes. Assim, a troca de informações dentro de determinado canal é feita de forma anônima em relação aos participantes da rede que não integram o canal [33]. Exemplifica-se o uso desta funcionalidade com o caso em que empresas concorrentes de um mesmo ramo façam parte de uma mesma rede *blockchain*, preservando informações sigilosas de suas transações. Nesse caso, faculta-se que as empresas integrem canais distintos, junto a organizações aliadas de seus interesses. Não se exclui a possibilidade, ainda, de aliados integrarem, simultaneamente, os canais das duas empresas, comunicando-se com ambas.

6.4.2 Políticas

As políticas configuram uma rede *HyperLedger*. Através delas, são definidas as regras de aceitação da rede, a fim de mitigar falhas ao gerenciar os dados que nela trafegam. Para isso, existem dois tipos de políticas: *SignaturePolicy* e *ImplicitMetaPolicy* [28]. A página oficial da *HyperLedger Fabric*, assim as define:

> *SignaturePolicy*: esse tipo de política é o mais poderoso e especifica uma combinação de regras de avaliação para os membros. Ele suporta combinações arbitrárias de AND, OR e NOutOf, permitindo a construção de regras extremamente poderosas.
>
> *ImplicitMetaPolicy*: É uma politica menos flexível e é válida apenas no contexto da configuração. Ela agrega o resultado da avaliação de políticas mais profundas na hierarquia de configuração, que são definidas em última análise por *SignaturePolicies*.

As políticas do canal, cuja obediência deve ser observada por todos os seus integrantes, consistem em um conjunto de regras definidas pelo próprio canal, tendo em vista suas necessidades e peculiaridades. As políticas - de assinatura, de controle de acesso, etc - têm por finalidade regular e disciplinar o modo de interação entre as organizações e o canal [28].

Conforme mencionado no 6.3, a proposta de transação enviada aos nodos da rede vem acompanhada da assinatura do requerente, a qual é utilizada pra verificar o cumprimento das políticas, no que se refere às permissões daquele *peer*. A definição das políticas se dá através do arquivo de configuração utilizado para criação da rede, conforme se observa no exemplo a seguir:

Verifica-se, da análise do código da figura 9, que há definição expressa de quais participantes são leitores (*readers*) e/ou escritores (*writers*). Assim, caso o nodo *'Org1MSP.peer'* tente realizar uma operação de escrita na rede, será verificado que ele não possui permissão para efetuá-la.

No mesmo sentido, a política de *ImplicitMeta* também se presta à definição das políticas do canal. Assim, em um canal, primeiro faz-se a verificação das políticas de assinatura para cada

Figura 9 – Exemplo de arquivo de definição de políticas da rede.

```
1    - &Org1
2  ...
3  Policies:
4      Readers:
5          Type: Signature
6          Rule: "OR('Org1MSP.admin', 'Org1MSP.peer', 'Org1MSP.client')"
7      Writers:
8          Type: Signature
9          Rule: "OR('Org1MSP.admin', 'Org1MSP.client')"
10     Admins:
11        Type: Signature
12        Rule: "OR('Org1MSP.admin')"
13     Endorsement:
14        Type: Signature
15        Rule: "OR('Org1MSP.peer')"
```

Fonte: Documentação disponibilizada pelo*HyperLedger Fabric* [34]

organização, através da aprovação ou não da política. Em seguida, verifica-se se o quórum de aprovação satisfaz a *ImplicitMeta* [28]. Segue, abaixo, exemplo a fim de esclarecer o tema:

O código mostrado na figura 10 determina o tipo e a regra que será utilizada para cada gênero de participante (escritor, leitor, administrador, etc) para prática de determinada ação, como por exemplo, para realizar uma leitura em um *ledger* onde a politica é *ImplicitMeta* e a regra dada por *'ANY Readers'*, significa que para realizar uma leitura basta que qualquer *peer* com permissão para leitura aprove a operação. No caso da regra *'MAJORITY Admins'* por exemplo, requer que a maioria dos *peers* com permissão de *admin* façam a aprovação.

Figura 10 – Exemplo de definição da política *ImplicitMeta*.

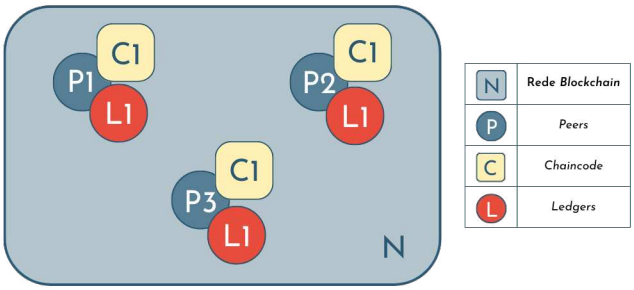
```
Policies:
    Readers:
        Type: ImplicitMeta
        Rule: "ANY Readers"
    Writers:
        Type: ImplicitMeta
        Rule: "ANY Writers"
    Admins:
        Type: ImplicitMeta
        Rule: "MAJORITY Admins"
    LifecycleEndorsement:
        Type: ImplicitMeta
        Rule: "MAJORITY Endorsement"
    Endorsement:
        Type: ImplicitMeta
        Rule: "MAJORITY Endorsement"
```

Fonte: Documentação disponibilizada pelo*HyperLedger Fabric* [34]

6.4.3 *Peers*

Peers são elementos cruciais para rede *Fabric*, eis que armazenam e manipulam uma cópia da *Ledger* e dos *Smartcontracts* (*Chaincode*). Representam, em suma, o ponto de comunicação entre uma organização e os elementos da rede. Os *peers* podem ser criados, editados e até removidos de um canal [27].

Figura 11 – Estrutura básica de *peers* participantes de uma rede.

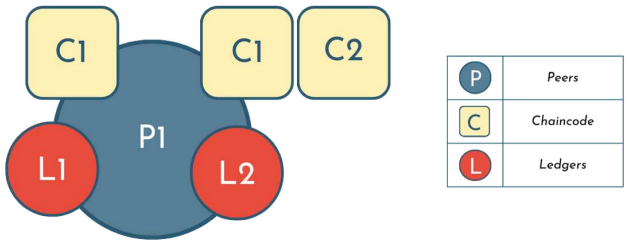


Fonte: Elaborada pelo autor

Peers podem comunicar-se entre si através de canais. Para isso, ambos os *peers* devem participar de um mesmo canal e concordar com as regras por ele definidas. Cada *peer* mantém uma instância de uma *ledger* e uma instância de uma *chaincode*, gerando um vínculo entre todos

os *peers* de um canal, possibilitando a troca de informações entre eles [27]. Um *peer* pode manter instâncias de várias *ledgers* e *chaincodes*. É comum que exista pelo menos uma *chaincode* com permissão de acesso para cada uma das instâncias de *ledger* [35]: ou seja, para uma *ledger* L1 existe uma *chaincode* S1 que pode realizar operações em L1, e para L2 existe uma S2 que possa acessar L2. Contudo, é possível existência de várias *chaincodes* que modificam uma mesma *ledger*, como se observa no exemplo abaixo:

Figura 12 – Exemplo onde um único *peer* possui instâncias múltiplas de *chaincodes* e *ledgers*



Fonte: Elaborada pelo autor

Os *peers* são o ponto central por onde ocorrem as comunicações da rede, dentre elas vale destacar as seguintes [27]:

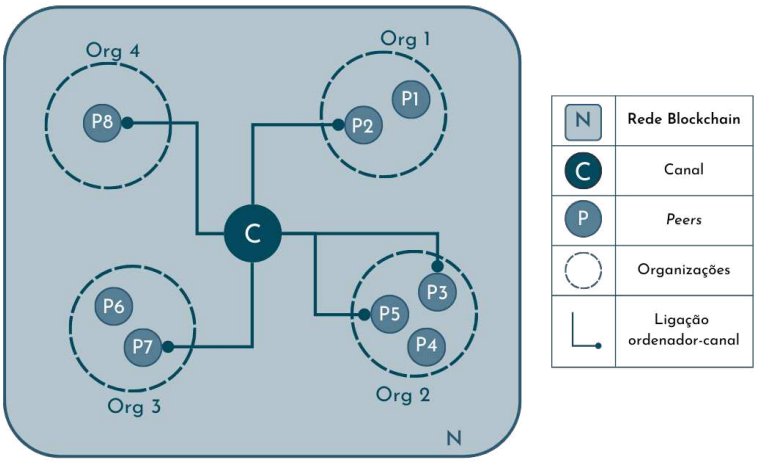
- Comunicação *Peer*-Canal: *Peers* podem comunicar-se entre si através de canais de forma privada. Para isso, ambos os *peers* devem participar de um mesmo canal e concordar com as regras por ele definidas.
- Comunicação *Peer*-Organização: Uma rede blockchain é gerenciada por organizações e os *peers* são o ponto de comunicação entre elas e a rede.
- Comunicação *Peer*-Ordenadores: As atualizações aprovadas por ordenadores, são enviadas a cada um dos *peers* para a atualização da *Ledger*.

6.4.4 Organizações

A administração da blockchain se dá pelas organizações da rede, as quais têm, como ponto central, um *peer*. Não há restrição quanto ao número de organizações que um canal pode conter. As organizações podem ser compreendidas, de forma geral como instituições/empresas, sendo certo que as elas precisam se comunicar com suas parceiras, portanto, o fazem através de um canal [36].

A rede só irá existir se as organizações disponibilizarem recursos, ou *peers*, nesse caso. Sem este artificio, a existência da rede é despropositada, pois não haveria pontos de comunicação e ela permaneceria estagnada. Entretanto, a rede não depende de uma determinada organização para existir, ela continuará existindo até que a última organização, seja ela qual for, se retire [36].

Figura 13 – Exemplo de estrutura de uma rede *HyperLedger Fabric* envolvendo seus principais elementos.



Fonte: Elaborada pelo autor

6.5 MINIFABRIC

Minifabric consiste em um projeto de código aberto baseado em *Hyperledger Fabric,* destacando-se pela redução da complexidade das operações. Apesar de pequeno, o projeto *Minifabric* abrange todas as funcionalidades disponibilizadas pelo *Fabric*. Além disso, o *Minifabric* disponibiliza interessantes aplicações que possibilitam o monitoramento das transações através de uma interface gráfica. É considerado não só uma ferramenta de desenvolvimento, mas também de aprendizado, eis que permite ao desenvolvedor experimentar o *HyperLedger Fabric* como um desenvolvedor, administrador ou usuário da rede [37] [38]. Assim, o projeto *Minifabric* foi utilizado na implementação do presente trabalho.

7 PROPOSTA

A detecção de falhas de assinatura digital ainda é um desafio nos dias de hoje. Protocolos, como o *Certificate Transparency* (CT), têm sido criados para facilitar o rastreio de falhas de assinatura digital. Com a a invenção de novos tipos de assinatura, surge a necessidade de criação de novos protocolos, a fim de se adaptar às necessidades da inovação tecnológica. Assim, busca-se, com o presente trabalho, utilizar a base do protocolo de *certificate transparency* para realizar o rastreamento da emissão dos certificados de uso único e assinatura dos documentos, a fim de aumentar o nível de segurança, provar a validade daquela emissão e, principalmente, detectar erros de assinaturas de maneira ágil. O princípio básico protocolo foi idealizado para utilização em uma rede de *blockchain*, que deve ser criada especificamente para registro e monitoramento de certificados de uso único.

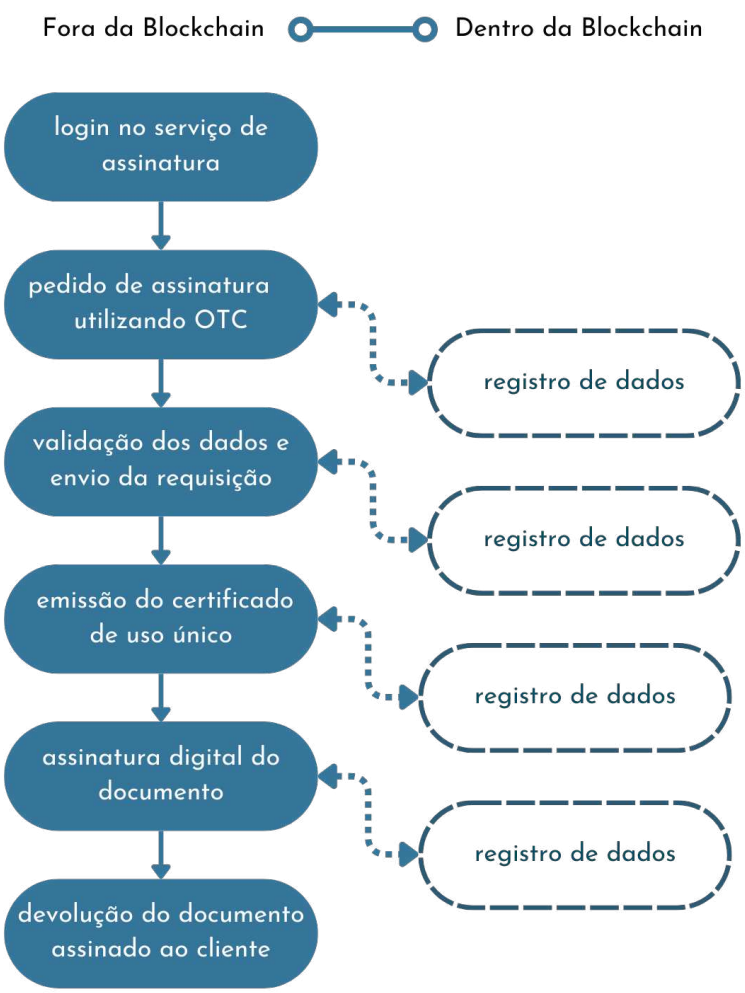
Contrastando com a *Certificate Transparency*, em que a inserção de novos certificados poderia ser realizada por qualquer usuário com interesse, este trabalho propõe que apenas participantes autorizados tenham permissão para inserir documentos na rede. Para isso, o atributo de permissão, detalhado na seção 6.1.1, que compõe o projeto *HyperLedger Fabric*, é explorado de forma a moldar as permissões dos dados para se adequar à necessidade do presente trabalho.

Para a detecção dos erros, faz-se necessária a manutenção da rastreabilidade dos dados gerados entre as etapas de criação do certificado de uso único, desde a requisição de assinatura, criação do certificado e assinatura do documento. Os passos previstos para utilização protocolo proposto seguiriam, como apresenta a figura 14, da seguinte forma: (i) para realizar a assinatura de um documento, o usuário deve estar logado em um serviço de Assinatura, para isso ele deve criar uma conta; (ii) após o *login*, ele realiza o pedido de assinatura de um documento com certificado de uso único e insere o documento de interesse; (iii) o assinador realiza o pedido de validação de dados à um provedor de identidade; (iv) os dados de interesse são inseridos na blockchain; (v) caso aceito, o provedor de identidade, então, envia uma requisição à uma autoridade certificadora, que emite o certificado de uso único (vi) o assinador recebe o certificado de uso único com as informações requisitadas pelo usuário; (vii) a AC insere o hash do certificado na blockchain; (viii) o assinador assina o documento e disponibiliza ao cliente; (ix) por fim, o assinador insere o hash do documento assinado na blockchain.

Propõe-se, também, que as operações na *blockchain* sejam rastreadas por um *token*, que é gerado durante o processo e reúne todas as suas etapas, permitindo o monitoramento completo do ciclo de geração de certificado de uso único até a devolução do documento, já assinado, ao usuário, além de comprovação do registro das etapas na *blockchain*.

A detecção de erros segue os princípios básicos do protocolo CT, no qual se realiza um monitoramento da rede para verificar inserção de novos dados. Se um dado for adicionado ao nome de alguém que não tenha solicitado essa operação, os dados estarão visíveis para monitoramento e detecção de erro. Encontrado o erro por um monitor, há a possibilidade de uma auditoria para verificação dos motivos e circunstâncias do erro, com base nos dados registrados na operação. Havendo dúvidas acerca da validade de um documento, mostra-se possível a

Figura 14 – Representação do fluxo de assinatura do documento e registro de dados.



Fonte: Elaborada pelo autor

realização de uma busca na rede da *blockchain*, a fim de verificar se houve o registro da operação de emissão do documento. Assim, imprescindível que todos os documentos emitidos sejam registrados em uma rede *blockchain*, assim como ocorre no protocolo CT.

Para detecção de falhas de emissão do documento assinado com o certificado de uso único, nem todas as especificações do *Certificate Transparency* são necessárias. Além disso, a utilização da *blockchain* faz com que seu uso se torne desvantajoso, ante à existência de restrições desnecessárias. Verificou-se, por exemplo, no item 5.4, que, havendo a adição de um certificado, o *log* deve retornar um *Signed Certificate Timestamp*, que funciona como uma assinatura do *log*, a qual será adicionada ao certificado. No protocolo proposto neste trabalho, busca-se contornar a necessidade de verificação de várias assinaturas a fim de otimizar o aproveitamento do espaço e tempo de verificação. Além disso, eventual utilização do SCT implicaria, necessariamente, o armazenamento da chave pública utilizada para gerar o SCT,

de forma segura, em determinado local, para que fosse possível sua verificação. No CT, por exemplo, a chave pública fica armazenada no *browser*. Em contrapartida, no protocolo proposto, não há necessidade de armazenamento da chave, o que aumenta a confiabilidade do protocolo.

Ao adotar a *blockchain* como uma das bases do protocolo, reforçam-se as características encontradas no CT, tais como: credibilidade temporal, imutabilidade do *log*, capacidade de rastrear e validar os documentos, descentralização de poder e, por fim, a restrição do uso.

O reforço à credibilidade temporal é alcançado através do registro do tempo de criação de cada bloco na *blockchain*. Embora o certificado de uso único não se beneficie de um carimbo de tempo, como discutido na seção 3.4, a existência desse registro temporal confirma a validade e a existência dos dados contidos no bloco.

A imutabilidade do *log*, principio básico da *blockchain*, garante a não alteração dos dados. A *blockchain* registra todas as transações em blocos encadeados, formando uma cadeia de blocos imutável. Todos os registros das transações, incluindo a emissão e uso de certificados de uso único, são permanentes e não podem ser alterados. Essa característica preserva a integridade dos certificados.

A capacidade de rastrear e validar os documentos se deve ao registro de diversas informações relacionadas à operação de assinatura. A *blockchain* oferece total transparência das transações de certificados de uso único, registrando todas as transações em tempo real e tornando-as visíveis para todos os participantes da rede *blockchain*. Isso facilita a realização de auditorias e verificações independentes, permitindo que qualquer pessoa acompanhe o histórico completo de cada certificado de uso único, desde a emissão até o uso. Essa transparência fortalece a confiabilidade e a verificabilidade dos documentos.

A descentralização do poder, por sua vez, permite diversas entidades possam validar e inserir dados na rede. Essa abordagem descentralizada elimina a dependência de uma única autoridade, reduzindo o risco de fraudes e manipulações.

Por fim, a restrição do uso, viabilizada pelo *framework HyperLedger Fabric,* garante que apenas entidades cadastradas na rede possam utilizar o sistema, impedindo a inserção de dados por usuários que não foram previamente aceitos e cadastrados na rede. Isso reduz significativamente as vulnerabilidades à inserção de informações maliciosas sem a detecção do verdadeiro infrator. Além disso, a adição de novos membros requer acordo entre os participantes existentes, o que contribui para um ambiente mais seguro, onde apenas membros confiáveis são aceitos

Essas características fortalecem a segurança, a confiabilidade e a integridade dos certificados, contribuindo para um ambiente mais confiável e eficiente na emissão e verificação desses documentos. A adoção da *blockchain* representa um avanço significativo na detecção e prevenção de falhas, garantindo a qualidade e a autenticidade dos certificados de uso único.

8 IMPLEMENTAÇÃO

Para simular o uso do protocolo de verificação de falhas de emissão de certificado de uso único proposto, faz-se necessária a criação de uma rede de *blockchain*, o que será realizado utilizando uma organização existente. Ressalta-se que a referida seleção tem finalidade meramente exemplificativa. Além disso, durante a implementação foram realizadas escolhas arquiteturais, as quais serão justificadas no decorrer do texto. Dentre elas, estão: (i) modelagem da rede de *Blockchain*; (ii) decisões estruturais dos blocos; (iii) linguagem da *Chaincode*; (iv) modelagem da *Chaincode*.

8.1 EXEMPLO DE MODELO DA REDE NACIONAL DE ENSINO E PESQUISA (RNP)

A Rede Nacional de Ensino e Pesquisa (RNP) é uma plataforma digital de comunicação e colaboração que trabalha para promover e implementar aplicações de tecnologia da informação. Ela conecta mais de 4 milhões de estudantes, professores e pesquisadores em universidades e instituições educacionais em todo o Brasil. A opção pela RNP como modelo para criação da rede de *blockchain* ocorreu por dois principais motivos. Primeiramente, destaca-se que a RNP estabelece uma sólida ligação entre a cibersegurança e as instituições educacionais através da plataforma ICPEdu, que possibilita a emissão de certificados digitais pessoais para membros da comunidade acadêmica. Ademais, a RNP mantém estreita relação com o Laboratório de Segurança em Computação (LabSEC) da UFSC, que se destaca como um centro de excelência em pesquisa e desenvolvimento de soluções de segurança.

Considerando um cenário hipotético, supõe-se que a RNP tenha o interesse de desenvolver um projeto para criar uma rede de comunicação entre todas as universidades e institutos federais do Brasil. O objetivo desse projeto seria possibilitar a troca de informações sobre projetos assinados por cada órgão e promover a tomada de decisões conjuntas em relação a diversos assuntos. No projeto em questão, a RNP tem interesse em utilizar os certificados de uso único para realizar as assinaturas digitais. Supondo o cenário hipotético acima, podemos modelar a rede como descreve as seções seguintes.

8.1.1 Arquivos necessários

Verificou-se, durante este trabalho, que a criação da rede contou com o auxílio do software de código aberto *Minifabric*, baseado em *HyperLedger Fabric*. Para definir a estrutura da rede, é necessário utilizar um arquivo de configuração que descreve todos os componentes necessários para seu funcionamento. O arquivo [1] é gerado no diretório onde o comando de criação da rede será executado. A partir desse arquivo de configuração, são gerados outros

[1] O nome do arquivo deve, obrigatoriamente, ser *"spec.yaml"*, pois é seguindo esse padrão que o *Minifabric* identifica se há um arquivo para se basear, ou se deve iniciar utilizando o padrão, que é pré definido em sua instalação.

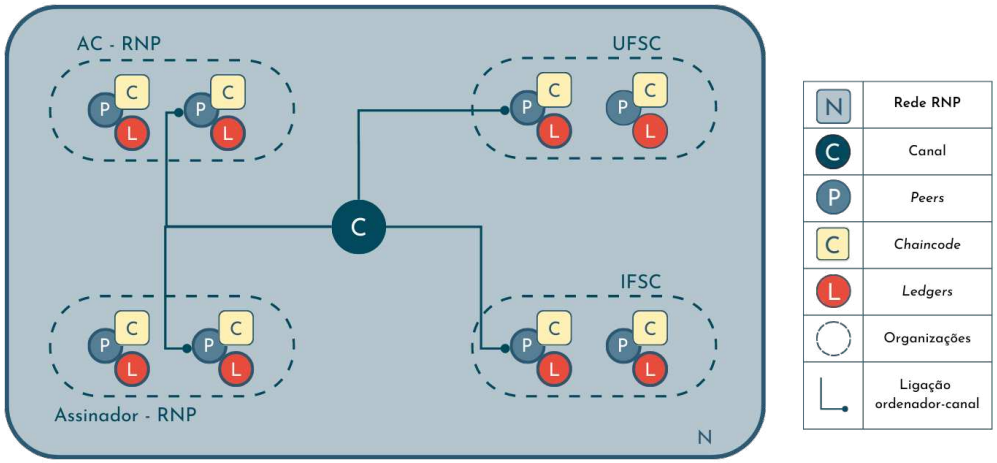
arquivos essenciais para o correto funcionamento da rede. Esses arquivos incluem as identidades digitais de cada nodo, como pares de chaves e certificados, o arquivo de configuração das políticas da rede, o bloco gênesis da cadeia, entre outros.

8.1.2 Rede

Para a implementação, utilizou-se como referência o caso da RNP, em que todos os *peers* têm igual relevância e capacidade de comunicação entre si. Diante disso, não se mostra necessária a criação de mais de um canal para a rede, eis que todas as organizações compartilham da mesma necessidade.

Em contrapartida, será necessária a criação de mais de uma organização: uma para cada ponto de acesso à rede. Assim, deverá ser criada uma organização para casa assinador, uma para cada provedor de identidade e uma para cada autoridade certificadora. Dentro das organizações, optou-se por conferir, a cada organização, o mesmo poder decisório. Desse modo, toda e qualquer organização será composta por um *peers* ordenador e um *peer* comum. A rede, então, pode ser representada da seguinte forma:

Figura 15 – Exemplo de rede no caso da RNP.



Fonte: Elaborada pelo autor

Neste exemplo a RNP faz o papel de Autoridade Certificadora e Assinador, enquanto as universidades UFSC e IFSC, representam, no exemplo, os provedores de identidade. A comunicação ocorre dessa forma pois as universidades providenciam os dados de alunos e projetos à ICPEdu, plataforma de emissão de certificados da RNP.

Figura 16 – Parte do arquivo de configuração onde são definidos os *peers*, organizações e seus ordenadores.

```
1 fabric:
2 [...]
3   peers:
4   - "peer0.ac.example.com"
5   - "peer1.ac.example.com"
6   - "peer0.id.example.com"
7   - "peer1.id.example.com"
8   - "peer0.sig.example.com"
9   - "peer1.sig.example.com"
10  orderers:
11  - "orderer1.example.com"
12  - "orderer2.example.com"
13  - "orderer3.example.com"
14 [...]
```

Fonte: Elaborado pelo autor

8.1.2.1 Criando a rede

A rede representada acima foi criada utilizando como base, o arquivo de configuração disponibilizado pelo *Minifabric* [39]. A partir dele, um comando de criação da rede é executado, e o *Minifabric* realiza os procedimentos e gera todos os arquivos fundamentais para tornar capaz o funcionamento de uma rede blockchain utilizando HLF.

A seguir, serão expostos trechos extraídos do arquivo mencionado, com o fim de exemplificar e esclarecer a maneira em que a rede foi configurada. O seguinte trecho descreve as organizações e seus respectivos *peers* e seus papeis:

Analisando o código, percebe-se que existem três organizações - "ac", "id"e "sig-, cada uma contendo dois *peers* e um nodo ordenador

O comando `"minifab netup"` cria a rede seguindo os passos: "download images, generate certificates, start network"; e após isso, a rede está inicializada e apta a ser utilizada.

8.1.3 Dados dos blocos

Os dados que deverão ser armazenados nas transações são: (i) nome do usuário do Provedor de Identidade; (ii) resumo Criptográfico do documento sem assinatura; (iii) resumo Criptográfico do certificado digital; (iv) resumo Criptográfico do documento assinado; (v) atributo *Common Name* do certificado; (vi) data e hora de acesso ao banco de dados de Identidade Eletrônica.

A escolha dos dados foi planejada de forma a manter o registro de partes importantes do processo que possam facilitar a identificação de falhas de e ajudar na comprovação da existência

de uma falha, sem expor dados sensíveis do usuário que poderiam violar a LGPD.

O item (i) foi pensado para manter o registro da conta que foi utilizada para realizar o pedido. Os itens (ii), (iii) e (iv), podem ser utilizados para detectar a alteração indesejada no documento e em qual etapa ocorreu. O item (v) pode ser utilizado para controle de emissões feitas por Monitores da rede, como explicado em 5.4 e 7. Por fim, o item (vi), pode ser utilizado como forma de provar a (in)capacidade de um usuário realizar uma determinada emissão, caso demonstrada sua inaptidão na data e hora registrada.

8.1.3.1 LGPD

A Lei Geral de Proteção de Dados (LGPD) foi baseada no Regulamento Geral de Proteção de Dados (GDPR) da União Europeia e entrou em vigor no Brasil em 2020. A LGPD estabelece regras claras sobre como as empresas e organizações devem coletar, armazenar, processar e compartilhar dados pessoais dos indivíduos. Além de promover a transparência no tratamento de dados pessoais, visa proteger a privacidade das pessoas. A lei também determina que a Autoridade Nacional de Proteção de Dados (ANPD) fica responsável por realizar a fiscalização das empresas e pode impor sanções àquelas que não cumprirem com as obrigações [40] [41].

8.1.4 Desenvolvimento *Chaincode*

Após a configuração e inicialização da rede de forma desejada, baseando-se nas etapas da seção 8.1.2, deve-se dar início ao desenvolvimento da *chaincode.* O primeiro passo para a produção de uma *Chaincode* é a escolha, dentre as opções elencadas em 6.2.1, da linguagem que será utilizada. Neste trabalho, optou-se pela linguagem Go. Considerando que a documentação disponibilizada pelo HLF não menciona nem recomenda nenhuma linguagem específica, a escolha se baseou na facilidade do uso e na existência de uma documentação muito bem detalhada acerca da linguagem.

A *chaincode* desenvolvida tem como papel principal inserir e buscar os dados da *blockchain* a partir de um identificador único. Ambas operações serão detalhadas nas seções seguintes. O desenvolvimento da *chaincode* utilizada nesse trabalho foi baseado em um exemplo disponibilizado pelo próprio *HyperLedger Fabric*. A partir desse exemplo pode-se entender seu funcionamento e com isso iniciar o processo de criação de uma *chaincode* que se adequasse ao protocolo proposto.

8.1.4.1 Inserção

O protocolo será composto por diversas etapas, cada qual responsável pela coleta e registro de parte das informações na *blockchain*. Os dados são referenciados por um identificador

único[2], que será gerado aleatoriamente toda vez que um novo conjunto de dados for inserido na rede. Inicialmente, é criada uma *composite key*[3] vazia, que será preenchida no decorrer do processo à medida em que os dados forem coletados por cada participante. Dessa forma o registro apresentará a contribuição de cada um no procedimento.

Os dados que serão registrados, que foram detalhados na seção 8.1.3, e qual dos participantes proverá cada um deles se dá pela seguinte forma: após o login e pedido de assinatura, o Assinador, agindo como uma organização da rede, dá inicio à criação de uma nova transação e uma nova *Primary Key* e registra o nome do usuário e o *hash* do documento sem estar assinado. A operação continua com o Provedor de Identidade, que realiza a verificação da identidade do usuário no momento que o usuário aceita a permissão de acesso e faz o registro da data e hora que o banco de dados de Identidade Eletrônica foi acessado. Logo em seguida, a Autoridade Certificadora faz a emissão do certificado e cadastra o *hash* na *blockchain*. Finalmente, o certificado retorna ao Assinador, que utiliza para assinar o documento e conclui inserindo o *hash* do documento já assinado na *blockchain*. Vale ressaltar que durante essa comunicação entre as entidades, o identificador único é encaminhado junto as outras informações.

8.1.4.2 Busca

A busca de dados pode ser feita por qualquer pessoa interessada em verificar a validade da assinatura. Para tanto, é necessário um *token* de identificação único referente ao documento que queira verificar. Em caso de monitores, a busca pode ser feita utilizando o *Common Name* para tentar identificar se houve a publicação de um documento sem permissão. Salienta-se, que para essa segunda estilo de busca, não há garantia de encontrar apenas um único documento, e nem mesmo de encontrar um documento sequer.

A busca deve ser feita no *World State*, um componente pertencente a *legder* do *Hyper-Ledger Fabric*. Também conhecido por *current state* (estado atual), ele armazena apenas o estado mais atualizado de uma determinada chave, evitando a necessidade de percorrer toda a cadeia de blocos. O estado atual será atualizado toda vez que o valor de uma chave for alterado. Realizada a busca pelos dados requeridos e obtida uma resposta, resta provada a validade da assinatura. Assim, significa dizer que aquela assinatura foi emitida por um órgão válido, contudo, não se pode atestar a inexistência de falha de emissão da assinatura.

8.2 BIBLIOTECA UTILIZADA

A biblioteca *shim*, desenvolvida pela *HyperLedger*, fornece uma API para ser utilizada no desenvolvimento das *chaincodes*. Ela fornece o suporte necessário para que o programador

[2] O identificador único é bastante utilizado dentro da esfera de Banco de Dados, é mais conhecido por *Primary Key*. Essa chave primaria, e única, faz referencia a um ou mais dados especificados pelo programador do banco [42].

[3] *Composite Key* é uma forma de referenciar varias dados, dentro de um banco de dados, utilizando uma única *Primary Key* como referência [43].

acesse os estados das variáveis, contextos das transações e até chamada de outras *chaincodes*, caso haja dependências de operações.

9 CONCLUSÃO

O contínuo avanço da tecnologia e a transição para documentos digitais trouxeram consigo a necessidade de garantir a autenticidade e integridade dos documentos no ambiente virtual. Nesse contexto, surgiram os certificados digitais. Dentre eles, o certificado de uso único, tecnologia inovadora no âmbito da cibersegurança, teve especial relevância neste trabalho. O presente trabalho se propôs à realização de estudos sobre *blockchain*, a fim de verificar qual a mais adequada para solucionar o problema proposto, relacionado ao armazenamento de certificados de uso único, levando em conta critérios de eficiência e adequação aos requisitos do trabalho. Ademais, buscou-se a utilização do protocolo de transparência de certificados como base para criação de um protocolo inédito para monitoramento de falhas de emissão dos certificados de uso único.

Ao longo deste trabalho, foram aprofundados conceitos fundamentais de criptografia e segurança computacional, permitindo uma investigação otimizada do monitoramento de documentos e/ou certificados, com o objetivo de mitigar falhas presentes no atual modelo de infraestrutura de chaves públicas. A busca por uma *blockchain* compatível com a proposta resultou em um amplo conhecimento do *HyperLedger Fabric*, um dos principais *frameworks* do mercado atual. Os resultados alcançados neste estudo atingiram o objetivo de encontrar respostas e soluções para os desafios enfrentados durante o desenvolvimento e implementação, mesmo considerando a natureza inovadora dos certificados de uso único.

Ao enfrentar o desafio de trabalhar com uma tecnologia pioneira como os certificados de uso único, este estudo contribuiu com a criação de um protocolo de segurança baseado em *Certificate Transparency*, que possibilita o monitoramento das emissões de documentos assinados utilizando certificado de uso único. Essa abordagem permitiu identificar emissões maliciosas e falhas no fluxo de execução, elevando significativamente o nível de segurança da cadeia. Esse avanço é especialmente relevante no contexto dos certificados de uso único, proporcionando um aumento de confiança ao utilizar uma tecnologia ainda pouco explorada.

O protocolo de *Certificate Transparency*, com o qual o presente trabalho estabeleceu estreita relação durante todo seu desenvolvimento, serviu como principal fonte de inspiração e base para o protocolo proposto. O CT, como já antes visto, tem como propósito elevar o padrão de segurança proposto pela infraestrutura de chaves públicas para certificados SSL/TLS. Tendo isso em vista, o protocolo Cripparency introduziu mudanças como: remoção da assinatura do *log* no certificado, visando agilizar a autenticação dos dados; descentralização do nível de permissões e do registro de informações, permitindo que mais de uma organização autorize ou registre novos dados; armazenamento de informações além do certificado, possibilitando maior efetividade na busca pelo responsável por falhas de emissões; restrição do uso do protocolo apenas a instituições autorizadas, dentre outras citadas durante o trabalho. Essas modificações, visando aprimorar o desempenho do protocolo em um ambiente que atenda aos requisitos do trabalho, foram possíveis graças às vantagens do uso da *blockchain*, em particular o *HyperLedger Fabric.*

Após cuidadosas análises, pesquisas e conclusões detalhadas, foi constatado que o protocolo de *Certificate Transparency* contém normas e regras interessantes para fortalecer a segurança digital, podendo ser amplamente explorado e utilizado como base para diversos outros mecanismos de confiança a serem criados. Bem como o certificado de uso único, que apresenta uma nova forma de utilizar o certificado digital, mitigando alguns dos principais desafios, como a revogação e o gerenciamento de chaves. Além disso, foi realizado um estudo detalhado dos principais atributos e funcionalidades oferecidos pelo *HyperLedger Fabric*, que foram os principais motivadores para a escolha desse *framework* como núcleo da implementação. Neste trabalho, cada uma das aplicações do HLF recebeu sua devida atribuição e responsabilidade, desempenhando um papel fundamental na compreensão e modelagem da rede na qual o protocolo foi aplicado. Essa abordagem possibilitou uma melhor adaptação do protocolo às necessidades singulares da infraestrutura de certificados de uso único, contribuindo significativamente para o êxito do trabalho.

Por fim, percebeu-se que, embora a infraestrutura de chaves públicas seja amplamente utilizada para garantir a segurança e autenticidade dos documentos digitais em todo o mundo, a constante evolução das tecnologias exige aprimoramento contínuo, inclusive em protocolos robustos. A busca incansável por esse aprimoramento é o lembrete constante de que a pesquisa acadêmica desempenha um papel de extrema importância para a sociedade, proporcionando ideias, inovações e soluções que impulsionam, não apenas a comunidade de computação, mas todas as áreas, rumo a um futuro mais seguro e confiável.

9.1 TRABALHOS FUTUROS

Ao longo da implementação do trabalho, surgiram ideias para aprimorar a proposta. No entanto, considerando a necessidade de manter o projeto em um nível viável, essas ideias foram documentadas e detalhadas abaixo para serem exploradas em trabalhos futuros.

9.1.1 OID com token

No presente trabalho, o *token* de identificação para busca do certificado na *blockchain* retorna ao final da operação, e o usuário pode armazená-lo e compartilhá-lo a seu critério. Entretanto, se não manuseado da forma correta e ocorrendo sua perda, não é passível de recuperação. Nesse contexto, em um trabalho futuro, seria possível inserir o *token* como um *Object Identifier*, de modo que fosse parte integrante do certificado, e consequentemente, do documento assinado. Assim, dispensa-se a necessidade de armazenamento e envio do *token* pelo usuário toda vez que utiliza o documento referente àquele identificador. Haveria, desta forma, o aprimoramento da busca pelo documento, eis que o *token* estaria sempre à disposição e sem erros, não sendo mais necessária a busca por outros parâmetros.

9.1.2 Parâmetros Privados

Como mencionado anteriormente na seção 8.1.3.1, os dados coletados por um determinado site devem ser considerados como dados privados, a menos que o usuário conceda permissão ao site para acessá-los. No contexto do nosso trabalho, é importante ter uma atenção redobrada, pois além de manipular esses dados, eles também se tornam públicos. Para futuras aplicações práticas da proposta, seria interessante que o usuário disponibilizasse alguns dados mais sensíveis, que permitissem comprovar de forma mais clara a validade da assinatura, caso haja a necessidade de verificação em questões judiciais.

No entanto, é importante ressaltar que dados críticos, como o endereço IP do usuário, não devem ser expostos publicamente. Para garantir a segurança e privacidade dos usuários, parâmetros privados podem ser processados apenas por nós específicos, conforme definido na política estabelecida. Isso significa que somente a empresa responsável pela inserção dos dados teria a capacidade de manipulá-los, o que permitiria reter informações suficientes para garantir a segurança do usuário sem expor dados críticos a qualquer pessoa. A decisão final seria deixada ao usuário, permitindo que ele escolha disponibilizar dados críticos que possam, potencialmente, inocentá-lo no futuro ou optar por não compartilhar esses dados, confiando na cadeia de certificados e na autoridade certificadora de sua escolha. Essa abordagem busca encontrar um equilíbrio entre a segurança do usuário e a necessidade de provar a validade da assinatura em cenários jurídicos.

9.1.3 Monitoramento automático

Com o objetivo de agilizar a detecção de falhas de emissão e auxiliar os monitores, uma abordagem interessante seria por modelar uma rede em que o registro do e-mail do emissor fosse obrigatório na *blockchain*, em conjunto com à utilização de contratos inteligentes (*smart contracts*) para automatização de tarefas. Desse modo, toda vez que um certificado fosse emitido usando uma determinada conta, um e-mail seria encaminhado alertando sobre essa emissão, trazendo praticidade para os usuários da rede. Essa estratégia permitiria uma maior eficiência na detecção e notificação de emissões de certificados, contribuindo para a rápida identificação de possíveis falhas e facilitando a supervisão do sistema como um todo.

REFERÊNCIAS

1 STALLINGS, W. **Cryptography and network security, 4/E**. [S.l.]: Pearson Education India, 2006.

2 DIFFIE, W.; HELLMAN, M. New directions in cryptography. **IEEE transactions on Information Theory**, IEEE, v. 22, n. 6, p. 644–654, 1976.

3 MERKLE, R. C. A certified digital signature. In: SPRINGER. **Conference on the Theory and Application of Cryptology**. [S.l.], 1989. p. 218–238.

4 WEISE, J. Public key infrastructure overview. **Sun BluePrints OnLine, August**, p. 1–27, 2001.

5 SSL. **Question types**. 2022. https://www.ssl.com/pt/faqs/what-is-a-certificate-authority/.

6 IBM. **Digital certificates and certificate authorities**. 2021. Https://www.ibm.com/. Acesso em: 17 de Maio de 2023. Disponível em: https://www.ibm.com/docs/en/db2/11.1?topic=SSEPGG_11.1.0/com.ibm.db2.luw.admin.sec.doc/doc/c00535

7 HABER, S.; STORNETTA, W. S. How to time-stamp a digital document. In: SPRINGER. **Conference on the Theory and Application of Cryptography**. [S.l.], 1990. p. 437–455.

8 MAYR, e. a. One-time certificates for reliable, inclusive and secure document signing. Artigo submetido para publicação. 2023.

9 CLOUDFLARE. **What is an identity provider (IdP)?** 2023. Https://www.cloudflare.com/. Acesso em: 17 de Maio de 2023. Disponível em: https://www.cloudflare.com/learning/access-management/what-is-an-identity-provider/.

10 NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. **Decentralized Business Review**, p. 21260, 2008.

11 MONRAT, A. A.; SCHELÉN, O.; ANDERSSON, K. A survey of blockchain from the perspectives of applications, challenges, and opportunities. **IEEE Access**, IEEE, v. 7, p. 117134–117151, 2019.

12 ZHENG, Z. et al. Blockchain challenges and opportunities: A survey. **International Journal of Web and Grid Services**, Inderscience Publishers (IEL), v. 14, n. 4, p. 352–375, 2018.

13 PETERS, G. W.; PANAYI, E. Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money. In: **Banking beyond banks and money**. [S.l.]: Springer, 2016. p. 239–278.

14 GOGERTY, N.; ZITOLI, J. Deko–currency proposal using a portfolio of electricity linked assets. **Available at SSRN 1802166**, 2011.

15 KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, n. 2004, p. 1–26, 2004.

16 KUBILAY, M. Y.; KIRAZ, M. S.; MANTAR, H. A. Certledger: A new pki model with certificate transparency based on blockchain. **Computers & Security**, v. 85, p. 333–352, 2019. ISSN 0167-4048. Disponível em: https://www.sciencedirect.com/science/article/pii/S0167404818313014.

17 YAO, S. et al. Pbcert: Privacy-preserving blockchain-based certificate status validation toward mass storage management. **IEEE Access**, IEEE, v. 7, p. 6117–6128, 2018.

18 GAYATHIRI, A.; JAYACHITRA, J.; MATILDA, S. Certificate validation using blockchain. In: **2020 7th International Conference on Smart Structures and Systems (ICSSS)**. [S.l.: s.n.], 2020. p. 1–4.

19 MADALA, D. V.; JHANWAR, M.; CHATTOPADHYAY, A. Certificate transparency using blockchain. In: **2018 IEEE International Conference on Data Mining Workshops (ICDMW)**. Los Alamitos, CA, USA: IEEE Computer Society, 2018. p. 71–80. Disponível em: https://doi.ieeecomputersociety.org/10.1109/ICDMW.2018.00018.

20 ZHAO, J. et al. Trustca: Achieving certificate transparency through smart contract in blockchain platforms. In: **2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)**. Los Alamitos, CA, USA: IEEE Computer Society, 2020. p. 1–6. Disponível em: https://doi.ieeecomputersociety.org/10.1109/HPBDIS49115.2020.9130581.

21 LAURIE, B. et al. **Certificate Transparency Version 2.0**. RFC Editor, 2021. RFC 9162. (Request for Comments, 9162). Disponível em: https://www.rfc-editor.org/info/rfc9162.

22 FOUNDATION, H. **Introduction**. 2023. Https://hyperledger-fabric.readthedocs.io/en/. Acesso em: 10 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/whatis.html.

23 FOUNDATION, H. **Writing Your First Chaincode**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/chaincode4ade.html#what-is-chaincode.

24 FOUNDATION, H. **Consensus**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 2 de Abril de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/fabric_model.html#consensus.

25 FOUNDATION, H. **Permissioned vs Permissionless Blockchains**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/whatis.html#permissioned-vs-permissionless-blockchains.

26 FOUNDATION, H. **Smart Contracts**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html#smart-contracts.

27 FOUNDATION, H. **Peers**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/peers/peers.html.

28 FOUNDATION, H. **Policies**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 12 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/policies/policies.html.

29 ORACLE. **Java**. 2023. Https://www.oracle.com. Acesso em: 17 de Maio de 2023. Disponível em: https://www.oracle.com/java/.

30 FOUNDATION, O. **About Node.js**. 2023. Https://nodejs.org/. Acesso em: 17 de Maio de 2023. Disponível em: https://nodejs.org/en/about.

31 GOOGLE. **The Go Project**. 2023. Https://go.dev/. Acesso em: 17 de Maio de 2023. Disponível em: https://go.dev/project.

32 FOUNDATION, H. **Transaction Flow**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/txflow.html#transaction-flow.

33 FOUNDATION, H. **Creating a channel**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 11 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/create_channel/create_channel_overview.html.

34 FOUNDATION, H. **How are policies implemented**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 12 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/policies/policies.html#how-are-policies-implemented.

35 FOUNDATION, H. **Ledgers and Chaincode**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/peers/peers.html#ledgers-and-chaincode.

36 FOUNDATION, H. **Organizations**. 2023. Https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 15 de Fevereiro de 2023. Disponível em: https://hyperledger-fabric.readthedocs.io/en/release-2.5/create_channel/create_channel_config.html#organizations.

37 FOUNDATION, H. 2021. Https://labs.hyperledger.org/. Acesso em: 8 de Março de 2023. Disponível em: https://labs.hyperledger.org/labs/minifabric.html.

38 GROUP, M. 2021. Https://github.com/hyperledger-labs/minifabric/blob/main/. Acesso em: 8 de Março de 2023. Disponível em: https://github.com/hyperledger-labs/minifabric/blob/main/docs/README.md.

39 GROUP, M. **spec.yaml**. 2021. Https://github.com/hyperledger-labs/minifabric/blob/main/. Acesso em: 8 de Março de 2023. Disponível em: https://github.com/hyperledger-labs/minifabric/blob/main/spec.yaml.

40 BRASIL. **LEI Nº 13.709, DE 14 DE AGOSTO DE 2018**. 2018. Https://www.planalto.gov.br. Acesso em: 20 de Maio de 2023. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm.

41 BRASIL, M. d. E. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. 2018. Https://www.gov.br. Acesso em: 20 de Maio de 2023. Disponível em: https://www.gov.br/esporte/pt-br/acesso-a-informacao/lgpd.

42 KROENKE, D. M. et al. **Database concepts**. [S.l.]: Prentice Hall Upper Saddle River, NJ, 2010.

43 IBM. **Defining Composite Primary and Foreign Keys**. 2023. Https://www.ibm.com/docs/en/informix-servers/14.10. Acesso em: 5 de Junho de 2023. Disponível em: https://www.ibm.com/docs/en/informix-servers/14.10?topic=format-defining-composite-primary-foreign-keys.

APÊNDICE A – ARTIGO SBC

Cripparency: Protocolo Baseado em *Certificate Transparency* para Rastreabilidade de Certificados de Uso Único Utilizando *Blockchain*

Arthur G. C. Milanez[1]

[1]Departamento de Informática e Estatística (INE)
Universidade Federal de Santa Catarina (UFSC)
CEP: 88040-370 – Florianópolis – SC – Brasil

arthur.crippa@grad.ufsc.br

Abstract. *With the advancement of cybersecurity, authentication of digital documents has been a challenge. Physical and digital certificates are subject to flaws that compromise their authenticity. The detection of errors in issuing certificates faces difficulties and the Certificate Transparency protocol addresses certificates related to the TLS protocol. One solution is the use of "single-use certificates", which are exclusively used to sign a single document. To overcome this, this work proposes the integration of these ideas with blockchain technology, through the protocol implemented using framework HyperLedger Fabric, Cripparency, which consists of a record of single-use certificates that identify failures of issuance.*

Resumo. *Com o avanço da cibersegurança, a autenticação de documentos digitais tem sido um desafio. Certificados físicos e digitais estão sujeitos a falhas que comprometem sua autenticidade. A detecção de erros na emissão de certificados enfrenta dificuldades e o protocolo Certificate Transparency aborda certificados relacionados ao protocolo TLS. Uma solução, é o uso de "certificados de uso único", que são exclusivamente usados para assinar um único documento. Para contornar isso, propõe-se, neste trabalho, a integração dessas ideias com a tecnologia de blockchain, através do protocolo implementado utilizando framework HyperLedger Fabric, Cripparency, que consiste em um registro de certificados de uso único que identifica falhas de emissão.*

1. Introdução

Com os avanços na cibersegurança e sua incorporação no cotidiano da sociedade, ocorreu a necessidade de autenticar a validade dos documentos digitalmente. Nesse contexto, surge o certificado digital, uma ferramenta que, através da criptografia, permite a assinatura digital, garantindo a autenticidade e integridade das informações no ambiente virtual. Documentos digitais exigem suporte computacional e criptográfico como (i) par de chaves, (ii) certificado digital, (iii) infraestrutura de chaves públicas, (iv) funções de resumo, etc.

Certificados digitais, assim como físicos, estão sujeitos a falhas, o que levanta preocupações sobre suas consequências, que podem comprometer a autenticidade de documentos. Há certa dificuldade em realizar a detecção dentro das restrições estabelecidas

pela infraestrutura de chaves públicas. Idealmente, a detecção do erro deveria ser instantânea, a fim de evitar danos significativos ao usuário. É por isso que, atualmente, existem sistemas de registro de emissão de certificados que lidam com estes problemas.

Nesse contexto, destaca-se o protocolo *Certificate Transparency*, um *log* público que faz o registro dos certificados digitais e disponibiliza os dados para monitoramento dos clientes. Destaca-se também, uma proposta que foi submetida à publicação, como resultado de um trabalho de mestrado conduzido por um estudante da UFSC. Essa abordagem propõe a adoção de um novo modelo de certificado digital denominado certificado de uso único, que é utilizado exclusivamente para assinar um único documento.

Diante desse cenário, este trabalho propõe a integração das ideias mencionadas com a tecnologia de *blockchain*, visando potencializar seus benefícios e promover um ambiente mais seguro. Isso será feito através do Cripparency, um protocolo de registro de certificados de uso único capaz de identificar o comprometimento de credenciais de usuários, falhas de emissão de certificados digitais e permitir o monitoramento de documentos assinados com certificado de uso único.

2. Fundamentação Teórica

A seguir, serão abordados os principais conceitos, que serão de extrema importância para entendimento do presente trabalho.

2.1. Infraestrutura de Chaves Públicas

A infraestrutura de chaves públicas (ICP) consiste em um conjunto de regras definidas pela *request for comments* (RFC) 2822. RFC é, em suma, um livro de regras que visa garantir a padronização e a segurança de documentos na internet. ICP, por sua vez, é definida por [Weise 2001] como um ”conjunto de hardware, software e pessoas, para manusear, armazenar e distribuir certificados digitais”.

A criptografia de chaves públicas consegue, através de uma infraestrutura de mecanismos de segurança, controlar essas informações e assegurar sua confiabilidade. Com a RFC padronizando as regras para a ICP, aplicações dentro da internet podem usar como base essas regras para criar a sua infraestrutura, que é um componente essencial para a estratégia de segurança na internet. [Weise 2001]

O papel da ICP é emitir, gerenciar, armazenar e revogar certificados digitais, garantindo a confiabilidade e segurança do sistema. Para isso, a ICP utiliza-se de uma cadeia de confiança [ssl 2022].

2.2. Assinatura Digital

A assinatura digital, também apresentada pela primeira vez por DIFFIE, W. HELLMAN, M. (1976), propõe a criação de um sistema capaz de substituir o uso de assinaturas manuscritas e acabar com grande parte da sua burocracia. A assinatura digital, assim como criptografia assimétrica, utiliza duas chaves, pública e privada. Assim, uma é utilizada para realizar a assinatura e a outra para sua verificação. A utilização da assinatura digital garante a autenticidade e a integridade do documento. O método funciona de uma forma em que o usuário utiliza a chave privada junto ao o documento para gerar uma saída assinada, que pode ser verificada com o uso da chave pública.

2.3. Certificado Digital

Certificado digital consiste em um conjunto de informações sobre a chave pública e sobre a identidade do dono dentro de um único documento. Esse conjunto de elementos deve ser confiado e assinado por um terceiro, geralmente uma autoridade certificadora (AC), que pode ser uma entidade pública ou privada. No Brasil, por exemplo, têm-se a cadeia de autoridades certificadoras da ICP-Brasil, pertencente ao governo federal, realizando o papel desse terceiro e garantindo a confiabilidade do documento.

Hoje, o certificado digital é utilizado como uma espécie de identidade, com validade jurídica equivalente ao CPF e ao CNPJ. A assinatura de documentos tem sido a utilização mais comum do certificado digital. Ao assinar um documento, o certificado com os dados do proprietário são inseridos no documento. Assim, poderá ser realizada a verificação da assinatura e relacioná-la com o respectivo titular.

2.4. Certificado de uso único

One-Time Certificates (OTC), ou certificado de uso único, expressão adotada neste trabalho, tem o propósito de superar os desafios de gerenciamento de chaves e revogação. Assim, cada certificado digital é utilizado exclusivamente para assinar um único documento, o que impede a invalidação de outras assinaturas caso ocorra a perda da chave privada. Após a assinatura, apenas a chave pública é utilizada, o que permite que a chave privada seja 'destruída' por não ser mais necessária, eliminando-se a necessidade de armazenamento prolongado da chave privada do usuário. Como a assinatura ocorre imediatamente após a emissão do certificado, assume-se que todos os atributos estavam válidos nesse momento [1]. Em decorrência disso, o uso de um carimbo do tempo torna-se dispensável, assim como elimina-se a necessidade de validar os dados no momento da assinatura. Ademais, a eliminação da validação de dados e da simplificação do gerenciamento da chave privada tornam a revogação desnecessária. Isso ocorre porque o certificado é restrito a uma única utilização, e os elementos mencionados asseguram a sua validade durante o curto período em que é empregado para realizar a assinatura [Mayr 2023].

3. Transparência de Certificado

O CT busca mitigar o problema de detecção de erros de emissão, sem recorrer a técnicas arriscadas, adicionando todos os certificados emitidos a um *log* público. O *log* em si não impede que erros ocorram, mas reduz o tempo necessário para identificá-los. Para que o *log* seja confiável por determinados *browsers*, ele deve seguir algumas regras como: ter o registro dos documentos e dados, utilizando tipos e estruturas definidas na RFC, manter-se ativo 99% do tempo, e realizar a inserção de novos certificados no *log* dentro de um prazo conhecido como *Maximum Merge Delay* (MMD) [Laurie et al. 2021].

A submissão de novos certificados ao *log* pode ser realizada por qualquer entidade. É possível submeter um certificado ou um pré certificado, que consiste em um arquivo CMS contendo todas as informações do certificado. Quando uma AC envia um pré certificado ao *log*, ela confirma sua intenção de assinar esse certificado. Considera-se um erro de emissão validar um pré certificado de um certificado que não foi assinado. Cada submissão no *log* deve vir acompanhada por todos os certificados adicionais necessários

[1] A afirmação só pode ser feita pois é responsabilidade da Autoridade Certificadora validar os atributos antes de realizar uma emissão de certificado.

para verificar a cadeia até alcançar uma autoridade certificadora raiz confiável. Se todos os certificados adicionais enviados forem válidos, fica a critério de quem executa o *log* aceitar ou não a submissão. Caso seja aceita, o *log* retorna um *Signed Certificate Timestamp* (SCT) e um arquivo contendo a prova de inclusão. O SCT é uma estrutura que contem um carimbo do tempo, o "*LogID*", que é um valor único referente ao *log* específico e uma assinatura. Caso um pré-certificado tenha sido submetido, essa estrutura deve ser incluída como parte do certificado final quando for emitido. Se a submissão for feita a partir de um certificado já emitido, o proprietário do certificado deve se encarregar de incluir o SCT junto com o certificado, de alguma forma [Laurie et al. 2021].

A adoção do protocolo de Transparência de Certificados ganhou destaque em 2015, quando o navegador *Google Chrome*, passou a exigir o registro de certificados emitidos para novos sites em um CT confiável pelo navegador. Em 2018, essa exigência se tornou obrigatória por todos os sites. O registro em um *log* público aumenta a confiabilidade do certificado, pois além da confiança na AC, também existe a confiança de quem fiscaliza a CT [Laurie et al. 2021].

A implementação do CT trouxe novas funções para a estrutura da ICP, como operadores do *log*, auditores e monitores. O operador do *log* é responsável por receber e adicionar novos certificados à lista de certificados do CT. Essa adição deve ocorrer dentro de um período de tempo conhecido como *Maximum Merge Delay*(MMD), que é o tempo máximo definido na RFC 9162 para que o novo certificado seja adicionado ao registro de transparência. Atualmente, esse tempo é de 24 horas. Esta etapa é uma das várias regras que um CT deve seguir para ser considerado funcional para o mercado de ICP.

A função do monitor é fiscalizar cada passo de um CT *log*. Sempre que um novo certificado é adicionado no *log*, o monitor comunica ao domínio daquele site que um novo certificado foi emitido em seu nome, permitindo a identificação rápida de falhas na emissão e a solicitação de revogação do certificado.

O auditor verifica se as regras estabelecidas pela RFC e pelo *browser* estão sendo seguidas pelo CT, como manter-se ativa em 99% do tempo, incluir novos certificados dentro do tempo definido pelo *Maximum Merge Delay*(MMD) que, por exemplo, no *browser Chrome* é de 24 horas, entre outras regras. O processo de auditoria pode ser realizado por monitores do CT, ou por qualquer outra pessoa interessada. Grandes corporações têm interesse em monitorar esses registros, pois são alvos potenciais de ataques, o que leva empresas como *Facebook* e *Cloudflare* a implementar em seus próprios sistemas de monitoramento [Laurie et al. 2021].

4. Blockchain e HyperLedger Fabric

O presente trabalho utiliza um protótipo - denominado Cripparency - desenvolvido para pôr à prova a proposta que será apresentada. Durante a implementação do protótipo, foram realizadas diversas escolhas relacionadas à arquitetura de software. A escolha de maior relevância consistiu na eleição do *framework* de *blockchain* que será utilizado no presente trabalho, decisão da qual decorreram todas as demais escolhas. Assim, elegeu-se o *framework HyperLedger Fabric*, sob a justificativa de que se apresenta como uma ferramenta sólida e satisfatória em relação às funcionalidades disponíveis. Para plena compreensão da motivação da escolha, passa-se a detalhar a estrutura que compõe uma rede *HyperLedger Fabric* [Foundation 2023b].

HypherLedger Fabric é um projeto de código aberto que possui funcionalidades singulares, diferenciando-se das *blockchains* tradicionais como *Etherium* e *Bitcoin*. Criado pela *HyperLedger Foundation*, o projeto *Fabric* possui, atualmente, mais de 200 desenvolvedores de mais de 25 empresas, tais como *Linux Foundation* e IBM, os quais dão suporte necessário para manter o projeto sólido e atualizado. O *Fabric* representou verdadeira inovação no âmbito das *blockchains*, pois permitiu que os contratos inteligentes fossem escritos em linguagens de programação usuais - Java, Go e Node.js. [Foundation 2023e] - tornando mais acessível sua utilização pelos programadores, o que contribuiu para o seu sucesso [Foundation 2023b].

A configuração do ambiente e arquitetura é dotada de ampla liberdade de alteração pelo desenvolvedor. A *Hyperledger* permite que o desenvolvedor atue com autonomia ao ajustar organizações, usuários, permissões, entre outros componentes da arquitetura da rede [Foundation 2023b].

Por não utilizar o consenso de *Proof of Work (PoW)*, não há necessidade do uso de uma *criptomoeda* de incentivo financeiro a fim de atrair mineradores para gerar novos blocos. Por outro lado, o mecanismo de consenso comumente utilizado no projeto *HyperLedger Fabric* é customizável, gerando uma gama de possibilidades que tonará possível a adequação do protocolo à necessidade de qualquer caso concreto. Esse conjunto de características implica melhoria de desempenho no processamento de transações, e também qualifica a rede a aumentar a privacidade das transações [Foundation 2023a].

4.1. Estrutura HLF

A *blockchain* do projeto *Fabric* diferencia-se das *blockchains* mais populares por ter, entre suas características, uma divisão peculiar de canais, organizações, *peers* e políticas organizacionais com vistas a auxiliar o controle de separação de tarefas. A seguir, passa-se ao detalhamento destas divisões.

O código mostrado na figura **??** determina o tipo e a regra que será utilizada para cada gênero de participante (escritor, leitor, administrador, etc) para prática de determinada ação, como por exemplo, para realizar uma leitura em um *ledger* onde a politica é *ImplicitMeta* e a regra dada por *'ANY Readers'*, significa que para realizar uma leitura basta que qualquer *peer* com permissão para leitura aprove a operação. No caso da regra *'MAJORITY Admins'* por exemplo, requer que a maioria dos *peers* com permissão de *admin* façam a aprovação.

4.2. *Peers*

Peers são elementos cruciais para rede *Fabric*, eis que armazenam e manipulam uma cópia da *Ledger* e dos *Smartcontracts* (*Chaincode*). Representam, em suma, o ponto de comunicação entre uma organização e os elementos da rede. Os *peers* podem ser criados, editados e até removidos de um canal [Foundation 2023d].

Peers podem comunicar-se entre si através de canais. Para isso, ambos os *peers* devem participar de um mesmo canal e concordar com as regras por ele definidas. Cada *peer* mantém uma instância de uma *ledger* e uma instância de uma *chaincode*, gerando um vínculo entre todos os *peers* de um canal, possibilitando a troca de informações entre eles [Foundation 2023d]. Um *peer* pode manter instâncias de várias *ledgers* e *chaincodes*. É comum que exista pelo menos uma *chaincode* com permissão de acesso para cada uma das instâncias de *ledger* [Foundation 2023c]: ou seja, para uma *ledger* L1 existe uma *chaincode* S1 que pode realizar operações em L1, e para L2 existe uma S2 que possa acessar L2. Contudo, é possível existência de várias *chaincodes* que modificam uma mesma *ledger*, como se observa no exemplo abaixo:

Os *peers* são o ponto central por onde ocorrem as comunicações da rede, dentre elas vale destacar as seguintes [Foundation 2023d]:

- Comunicação *Peer*-Canal: *Peers* podem comunicar-se entre si através de canais de forma privada. Para isso, ambos os *peers* devem participar de um mesmo canal e concordar com as regras por ele definidas.
- Comunicação *Peer*-Organização: Uma rede blockchain é gerenciada por organizações e os *peers* são o ponto de comunicação entre elas e a rede.
- Comunicação *Peer*-Ordenadores: As atualizações aprovadas por ordenadores, são enviadas a cada um dos *peers* para a atualização da *Ledger*.

5. Proposta

A detecção de falhas de assinatura digital ainda é um desafio nos dias de hoje. Protocolos, como o *Certificate Transparency* (CT), têm sido criados para facilitar o rastreio de falhas de assinatura digital. Com a a invenção de novos tipos de assinatura, surge a necessidade de criação de novos protocolos, a fim de se adaptar às necessidades da inovação tecnológica. Assim, busca-se, com o presente trabalho, utilizar a base do protocolo de *certificate transparency* para realizar o rastreamento da emissão dos certificados de uso único e assinatura dos documentos, a fim de aumentar o nível de segurança, provar a validade daquela emissão e, principalmente, detectar erros de assinaturas de maneira ágil. O princípio básico protocolo foi idealizado para utilização em uma rede de *blockchain*, que deve ser criada especificamente para registro e monitoramento de certificados de uso único.

Contrastando com a *Certificate Transparency*, em que a inserção de novos certificados poderia ser realizada por qualquer usuário com interesse, este trabalho propõe que apenas participantes autorizados tenham permissão para inserir documentos na rede. Para isso, o atributo de permissão, detalhado na seção **??**, que compõe o projeto *Hyper-Ledger Fabric*, é explorado de forma a moldar as permissões dos dados para se adequar à necessidade do presente trabalho.

Para a detecção dos erros, faz-se necessária a manutenção da rastreabilidade dos dados gerados entre as etapas de criação do certificado de uso único, desde a requisição

de assinatura, criação do certificado e assinatura do documento. Os passos previstos para utilização protocolo proposto seguiriam, como apresenta a figura 1, da seguinte forma: (i) para realizar a assinatura de um documento, o usuário deve estar logado em um serviço de Assinatura, para isso ele deve criar uma conta; (ii) após o *login*, ele realiza o pedido de assinatura de um documento com certificado de uso único e insere o documento de interesse; (iii) o assinador realiza o pedido de validação de dados à um provedor de identidade; (iv) caso aceito, o provedor de identidade, então, envia uma requisição à uma autoridade certificadora, que emite o certificado de uso único (v) o assinador recebe o certificado de uso único com as informações requisitadas pelo usuário; (vi) o assinador assina o documento e disponibiliza ao cliente.

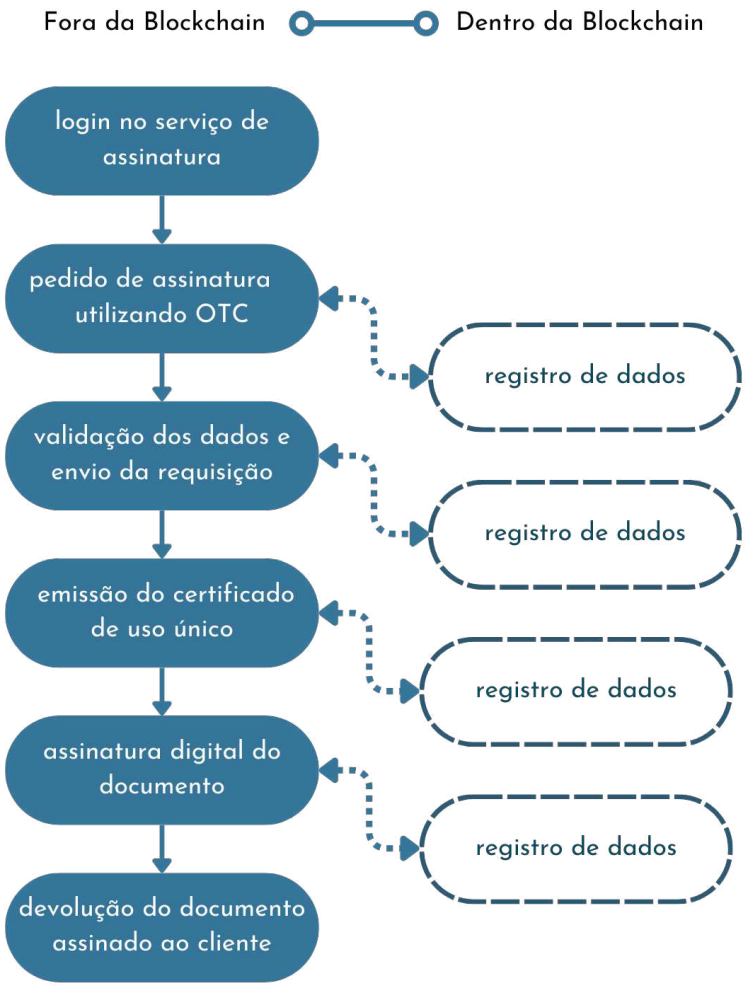


Figure 1. Fluxo de assinatura do documento e registro de dados.

Propõe-se, também, que as operações na *blockchain* sejam rastreadas por um *token*, que é gerado durante o processo e reúne todas as suas etapas, permitindo o monitoramento completo do ciclo de geração de certificado de uso único até a devolução do documento, já assinado, ao usuário, além de comprovação do registro das etapas na

blockchain.

A detecção de erros segue os princípios básicos do protocolo CT, no qual se realiza um monitoramento da rede para verificar inserção de novos dados. Se um dado for adicionado ao nome de alguém que não tenha solicitado essa operação, os dados estarão visíveis para monitoramento e detecção de erro. Encontrado o erro por um monitor, o documento pode ser revogado, havendo a possibilidade de uma auditoria para verificação dos motivos e circunstâncias do erro, com base nos dados registrados na operação. Havendo dúvidas acerca da validade de um documento, mostra-se possível a realização de uma busca na rede da *blockchain*, a fim de verificar se houve o registro da operação de emissão do documento. Assim, imprescindível que todos os documentos emitidos sejam registrados em uma rede *blockchain*, assim como ocorre no protocolo CT.

Para detecção de falhas de emissão do documento assinado com o certificado de uso único, nem todas as especificações do *Certificate Transparency* são necessárias. Além disso, a utilização da *blockchain* faz com que seu uso se torne desvantajoso, ante à existência de restrições desnecessárias. Verificou-se, por exemplo, no item **??**, que, havendo a adição de um certificado, o *log* deve retornar um *Signed Certificate Timestamp*, que funciona como uma assinatura do *log*, a qual será adicionada ao certificado. No protocolo proposto neste trabalho, busca-se contornar a necessidade de verificação de várias assinaturas a fim de otimizar o aproveitamento do espaço e tempo de verificação. Além disso, eventual utilização do SCT implicaria, necessariamente, o armazenamento da chave pública utilizada para gerar o SCT, de forma segura, em determinado local, para que fosse possível sua verificação. No CT, por exemplo, a chave pública fica armazenada no *browser*. Em contrapartida, no protocolo proposto, não há necessidade de armazenamento da chave, justamente por não haver adição da assinatura ao documento, o que aumenta a confiabilidade do protocolo.

Ao adotar a *blockchain* como uma das bases do protocolo, reforçam-se as características encontradas no CT, tais como: credibilidade temporal, imutabilidade do *log*, capacidade de rastrear e validar os documentos, descentralização de poder e, por fim, a restrição do uso.

O reforço à credibilidade temporal é alcançado através do registro do tempo de criação de cada bloco na *blockchain*. Embora o certificado de uso único não se beneficie de um carimbo de tempo, como discutido na seção 2.4, a existência desse registro temporal confirma a validade e a existência dos dados contidos no bloco.

A imutabilidade do *log*, principio básico da *blockchain*, garante a não alteração dos dados. A *blockchain* registra todas as transações em blocos encadeados, formando uma cadeia de blocos imutável. Todos os registros das transações, incluindo a emissão e uso de certificados de uso único,são permanentes e não podem ser alterados. Essa característica preserva a integridade dos certificados.

A capacidade de rastrear e validar os documentos se deve ao registro de diversas informações relacionadas à operação de assinatura. A *blockchain* oferece total transparência das transações de certificados de uso único, registrando todas as transações em tempo real e tornando-as visíveis para todos os participantes da rede *blockchain*. Isso facilita a realização de auditorias e verificações independentes, permitindo que qualquer pessoa acompanhe o histórico completo de cada certificado de uso único, desde a emissão

até o uso. Essa transparência fortalece a confiabilidade e a verificabilidade dos documentos.

A descentralização do poder, por sua vez, permite diversas entidades possam validar e inserir dados na rede. Essa abordagem descentralizada elimina a dependência de uma única autoridade, reduzindo o risco de fraudes e manipulações.

Por fim, a restrição do uso, viabilizada pelo *framework HyperLedger Fabric,* garante que apenas entidades cadastradas na rede possam utilizar o sistema, impedindo a inserção de dados por usuários que não foram previamente aceitos e cadastrados na rede. Isso reduz significativamente as vulnerabilidades à inserção de informações maliciosas sem a detecção do verdadeiro infrator. Além disso, a adição de novos membros requer acordo entre os participantes existentes, o que contribui para um ambiente mais seguro, onde apenas membros confiáveis são aceitos

Essas características fortalecem a segurança, a confiabilidade e a integridade dos certificados, contribuindo para um ambiente mais confiável e eficiente na emissão e verificação desses documentos. A adoção da *blockchain* representa um avanço significativo na detecção e prevenção de falhas, garantindo a qualidade e a autenticidade dos certificados de uso único.

6. Implementação

7. Conclusão

O contínuo avanço da tecnologia e a transição para documentos digitais trouxeram consigo a necessidade de garantir a autenticidade e integridade dos documentos no ambiente virtual. Nesse contexto, surgiram os certificados digitais. Dentre eles, o certificado de uso único, tecnologia inovadora no âmbito da cibersegurança, teve especial relevância neste trabalho. O presente trabalho se propôs à realização de estudos sobre *blockchain*, a fim de verificar qual a mais adequada para solucionar o problema proposto, relacionado ao armazenamento de certificados de uso único, levando em conta critérios de eficiência e adequação aos requisitos do trabalho. Ademais, buscou-se a utilização do protocolo de transparência de certificados como base para criação de um protocolo inédito para monitoramento de falhas de emissão dos certificados de uso único.

Ao longo deste trabalho, foram aprofundados conceitos fundamentais de criptografia e segurança computacional, permitindo uma investigação otimizada do monitoramento de documentos e/ou certificados, com o objetivo de mitigar falhas presentes no atual modelo de infraestrutura de chaves públicas. A busca por uma *blockchain* compatível com a proposta resultou em um amplo conhecimento do *HyperLedger Fabric*, um dos principais *frameworks* do mercado atual. Os resultados alcançados neste estudo atingiram o objetivo de encontrar respostas e soluções para os desafios enfrentados durante o desenvolvimento e implementação, mesmo considerando a natureza inovadora dos certificados de uso único.

Ao enfrentar o desafio de trabalhar com uma tecnologia pioneira como os certificados de uso único, este estudo contribuiu com a criação de um protocolo de segurança baseado em *Certificate Transparency*, que possibilita o monitoramento das emissões de documentos assinados utilizando certificado de uso único. Essa abordagem permitiu identificar emissões maliciosas e falhas no fluxo de execução, elevando significativamente o

nível de segurança da cadeia. Esse avanço é especialmente relevante no contexto dos certificados de uso único, proporcionando um aumento de confiança ao utilizar uma tecnologia ainda pouco explorada.

O protocolo de *Certificate Transparency*, com o qual o presente trabalho estabeleceu estreita relação durante todo seu desenvolvimento, serviu como principal fonte de inspiração e base para o protocolo proposto. O CT, como já antes visto, tem como propósito elevar o padrão de segurança proposto pela infraestrutura de chaves públicas para certificados SSL/TLS. Tendo isso em vista, o protocolo Cripparency introduziu mudanças como: remoção da assinatura do *log* no certificado, visando agilizar a autenticação dos dados; descentralização do nível de permissões e do registro de informações, permitindo que mais de uma organização autorize ou registre novos dados; armazenamento de informações além do certificado, possibilitando maior efetividade na busca pelo responsável por falhas de emissões; restrição do uso do protocolo apenas a instituições autorizadas, dentre outras citadas durante o trabalho. Essas modificações, visando aprimorar o desempenho do protocolo em um ambiente que atenda aos requisitos do trabalho, foram possíveis graças às vantagens do uso da *blockchain*, em particular o *HyperLedger Fabric.*

Após cuidadosas análises, pesquisas e conclusões detalhadas, foi constatado que o protocolo de *Certificate Transparency* contém normas e regras interessantes para fortalecer a segurança digital, podendo ser amplamente explorado e utilizado como base para diversos outros mecanismos de confiança a serem criados. Bem como o certificado de uso único, que apresenta uma nova forma de utilizar o certificado digital, mitigando alguns dos principais desafios, como a revogação e o gerenciamento de chaves. Além disso, foi realizado um estudo detalhado dos principais atributos e funcionalidades oferecidos pelo *HyperLedger Fabric*, que foram os principais motivadores para a escolha desse *framework* como núcleo da implementação. Neste trabalho, cada uma das aplicações do HLF recebeu sua devida atribuição e responsabilidade, desempenhando um papel fundamental na compreensão e modelagem da rede na qual o protocolo foi aplicado. Essa abordagem possibilitou uma melhor adaptação do protocolo às necessidades singulares da infraestrutura de certificados de uso único, contribuindo significativamente para o êxito do trabalho.

Por fim, percebeu-se que, embora a infraestrutura de chaves públicas seja amplamente utilizada para garantir a segurança e autenticidade dos documentos digitais em todo o mundo, a constante evolução das tecnologias exige aprimoramento contínuo, inclusive em protocolos robustos. A busca incansável por esse aprimoramento é o lembrete constante de que a pesquisa acadêmica desempenha um papel de extrema importância para a sociedade, proporcionando ideias, inovações e soluções que impulsionam, não apenas a comunidade de computação, mas todas as áreas, rumo a um futuro mais seguro e confiável.

References

Foundation, H. (2023a). Consensus. https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 2 de Abril de 2023.

Foundation, H. (2023b). Introduction. https://hyperledger-fabric.readthedocs.io/en/. Acesso em: 10 de Fevereiro de 2023.

Foundation, H. (2023c). Ledgers and chaincode. https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023.

Foundation, H. (2023d). Peers. https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023.

Foundation, H. (2023e). Writing your first chaincode. https://hyperledger-fabric.readthedocs.io/en/release-2.5/. Acesso em: 10 de Fevereiro de 2023.

Laurie, B., Langley, A., Kasper, E., Messeri, E., and Stradling, R. (2021). Certificate Transparency Version 2.0. RFC 9162.

Mayr, e. a. (2023). One-time certificates for reliable, inclusive and secure document signing. Artigo submetido para publicação.

ssl (2022). Question types. `https://www.ssl.com/pt/faqs/what-is-a-certificate-authority/`.

Weise, J. (2001). Public key infrastructure overview. *Sun BluePrints OnLine, August*, pages 1–27.