Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise de Soluções de Privacidade em Blockchain

Daniel Silva Almendra

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Orientador
Prof. Dr. Eduardo Alchieri

Brasília
2025

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

AA448a

Almendra, Daniel
Análise de Soluções de Privacidade em Blockchain / Daniel Almendra; orientador Eduardo Alchieri. Brasília, 2025.
126 p.

Dissertação(Mestrado em Informática) Universidade de Brasília, 2025.

1. Blockchain. 2. Privacidade. 3. Segurança. I. Alchieri, Eduardo, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise de Soluções de Privacidade em Blockchain

Daniel Silva Almendra

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

Prof. Dr. Eduardo Alchieri (Orientador)
CIC/UnB

Prof. Dr. Fernando Luis Dotti
PUC-RS

Prof. Dr. Rodrigo Bonifácio
UnB

Prof. Dr. Rodrigo Bonifácio
Coordenador do Programa de Pós-graduação em Informática

Brasília, 27 de março de 2025

Dedicatória

Dedico esta dissertação de mestrado àqueles que me moldaram como pessoa e me apoiaram durante toda a minha vida: meus pais, minha irmã, minha esposa e meu filho.

Aos meus amados pais, Angela e Roberto, que, com sua dedicação e carinho, fizeram de mim quem sou hoje. Suas lições de vida e valores me moldaram profundamente, e tudo o que conquistei é reflexo do amor e da educação que recebi de vocês. Cada sorriso e palavra de incentivo me motivou a seguir em frente. Vocês sempre acreditaram em mim com orgulho genuíno, e isso é minha maior motivação. Vocês foram e sempre serão o meu norte, minha melhor referência.

À minha querida irmã Renata, uma inspiração constante. Professora e doutora, você trilhou seu próprio caminho acadêmico com dedicação e paixão. Seu exemplo reacendeu em mim a vontade de estudar e de aprender cada vez mais. Sua trajetória é uma inspiração diária, e seu amor e orgulho por mim me incentivam a buscar ser sempre melhor.

À minha amada esposa, Lígia, minha companheira fiel em todos os momentos. Seu carinho terno e cuidadoso, seus conselhos sábios e seu apoio são meu maior alicerce. Sua força, inteligência e determinação me inspiram e me motivam a enfrentar e vencer qualquer desafio. Tudo o que conquistei tem, também, um pedaço seu.

E ao meu filho, Gabriel, minha fonte de inspiração e alegria. Você me lembra todos os dias de que vale a pena dar o meu melhor, e meu maior desejo é construir um caminho que lhe traga orgulho no futuro. Sua inocência e entusiasmo pelo mundo me fazem querer ser uma pessoa melhor a cada dia. Busco sempre ser para você um exemplo dos valores mais importantes: generosidade, humildade, integridade, respeito e autenticidade.

A cada um de vocês, minha gratidão eterna por serem a base de tudo o que sou.

Agradecimentos

Gostaria de expressar minha profunda gratidão a todos que contribuíram para a realização deste trabalho de mestrado.

Primeiramente, agradeço ao Banco Central do Brasil pela valiosa oportunidade de realizar este mestrado, concedendo-me uma licença essencial para que eu pudesse acompanhar as disciplinas e me dedicar plenamente à pesquisa. Esse apoio permitiu-me concluir cada etapa com excelência, um valor que o Banco Central representa e inspira fortemente em seus servidores e colaboradores. Em especial, agradeço ao meu orientador institucional, Danilo Dias, por seu grande apoio e orientação cuidadosa, ajudando-me a alinhar esta pesquisa com as expectativas do Bacen e ao projeto do Drex, a futura moeda digital brasileira. Meu agradecimento também se estende a Marcos Euzebio, por sua confiança no meu trabalho e pela inspiração que proporcionou para que eu desse mais este passo em minha trajetória acadêmica.

Sou grato aos professores da Universidade de Brasília (UnB) por todo o conhecimento transmitido, tanto ao longo do curso de mestrado no Departamento de Ciência da Computação quanto durante minha graduação em Engenharia de Redes de Comunicação no início dos anos 2000. Cada disciplina e interação foi uma contribuição valiosa para meu crescimento acadêmico e profissional. Meus sinceros agradecimentos vão especialmente ao meu orientador Eduardo Alchieri, cuja orientação e suporte foram indispensáveis para a realização deste trabalho. Sua experiência e amplo conhecimento me deram a segurança necessária para prosseguir em cada fase da pesquisa.

Por fim, expresso minha gratidão à minha esposa, Lígia, pelo apoio incondicional e encorajamento em todos os momentos do curso. Sua paciência e compreensão nos períodos em que precisei me ausentar ou quando a ansiedade da pesquisa se fez presente foram fundamentais para que eu pudesse seguir adiante e concluir esta jornada.

Resumo

A maioria das plataformas de *blockchain* armazena as transações publicamente no *ledger*. Embora essa abordagem garanta a corretude e a auditabilidade das operações, ela impõe um obstáculo significativo para o desenvolvimento de aplicações que preservem a privacidade dos usuários. Diversas abordagens têm sido propostas para mitigar esse problema, empregando técnicas variadas e oferecendo diferentes níveis de privacidade. No entanto, o campo de pesquisa nessa área demonstra carência de análises abrangentes, comparativas e práticas das soluções de privacidade disponíveis.

Esta dissertação apresenta uma análise conceitual de soluções de privacidade para plataformas de *blockchain*, comparando suas arquiteturas, funcionalidades e limitações. Além disso, é realizada uma avaliação empírica de duas dessas soluções – *Anonymous Zether* e *Zeestar*. Nessa análise, é implementado um estudo de caso no qual essas soluções são utilizadas para aprimorar a privacidade da aplicação *Miles2Coins*, uma plataforma de compra e venda de *tokens* de milhas aéreas desenvolvida para este trabalho. Assim, é possível investigar os impactos sobre o desempenho, os custos transacionais e a complexidade adicional envolvida na implementação de mecanismos de privacidade em aplicações descentralizadas. Os resultados evidenciam os desafios enfrentados pelas abordagens atuais para oferecer uma solução definitiva para a privacidade em *blockchain* sem compromissos consideráveis.

Por fim, este trabalho discute o panorama atual da privacidade em *blockchain*, avaliando os avanços recentes e as dificuldades associadas à adoção de plataformas de *blockchain* com mecanismos de privacidade integrados. O principal obstáculo reside em atender a requisitos rigorosos de privacidade sem comprometer significativamente o desempenho, os custos e a usabilidade das aplicações descentralizadas.

Palavras-chave: Blockchain, Contratos Inteligentes, Privacidade, Segurança

Abstract

Most blockchain platforms store transactions publicly on the ledger. While this approach ensures the correctness and auditability of operations, it poses a significant obstacle to the development of privacy-preserving applications. Various approaches have been proposed to mitigate this issue, employing different techniques and offering varying levels of privacy. However, the research field in this area lacks comprehensive, comparative, and practical analyses of existing privacy solutions.

This dissertation presents a conceptual analysis of privacy solutions for blockchain platforms, comparing their architectures, features, and limitations. Additionally, an empirical evaluation of two of these solutions – Anonymous Zether and Zeestar – is conducted. In this analysis, a case study is implemented in which these solutions are utilized to enhance privacy in Miles2Coins, an airline miles token trading platform developed for this work. This evaluation allows an investigation into the impact of privacy solutions on performance, transaction costs, and the additional complexity involved in their integration into decentralized applications. The results highlight the challenges faced by current approaches in providing a definitive privacy solution for blockchain without significant trade-offs.

Finally, this work discusses the current landscape of privacy in blockchain, assessing recent advances and the difficulties associated with the adoption of blockchain platforms with integrated privacy mechanisms. The main obstacle lies in meeting stringent privacy requirements without significantly compromising the performance, cost, and usability of decentralized applications.

Keywords: Blockchain, Smart Contracts, Privacy, Security

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ABI Application Binary Interface.

ACLs Access Control Lists.

API Application Programming Interface.

CBDC Central Bank Digital Currency.

CBDCs Central Bank Digital Currencies.

CCPA California Consumer Privacy Act.

CIS Center for Internet Security.

CSF Cybersecurity Framework.

DApps Decentralized Applications.

DeFi Decentralized Finance.

DEX Decentralized Exchange.

DLT Distributed Ledger Technology.

DvP Delivery vs Payment.

EMR Electronic Medical Records.

ETH Ether.

eUTXO Extended UTXO.

EVM Ethereum Virtual Machine.

FHE Fully Homomorphic Encryption.

FIPPs Fair Information Practice Principles.

FPC Federal Privacy Council.

GDPR General Data Protection Regulation.

GLBA Gramm-Leach-Bliley Act.

HE Homomorphic Encryption.

HIPAA Health Insurance Portability and Accountability Act.

IoT Internet of Things.

LGPD Lei Geral de Proteção de Dados Pessoais.

LSTM Long Short-Term Memory.

NFT Non-Fungible Token.

NIST National Institute of Standards and Technology.

P2P Peer-to-Peer.

PoA Proof of Authority.

PPSI Programa de Privacidade e Segurança da Informação.

SGX Software Guard Extensions.

SMPC Secure Multi-Party Computation.

TEE Trusted Execution Environment.

TEE-DS Trusted Execution Environment Distributed Storage.

TextCNN Convolutional Neural Network for Text Classification.

TLS Transport Layer Security.

TVL Total Value Locked.

UAW Unique Active Wallets.

UTXO Unspent Transaction Output.

VCDPA Virginia Consumer Data Protection Act.

VM Virtual Machine.

ZKP Zero-Knowledge Proof.

ZKPs Zero-Knowledge Proofs.

zkSNARK Zero-Knowledge Succinct Non-Interactive Argument of Knowledge.

ZSC Zether Smart Contract.

ZTH Zether Token.

Capítulo 1

Introdução

No *white paper* que deu origem ao *Bitcoin* em 2008 [1], foi projetada a *blockchain*, uma nova tecnologia de registro distribuído – *Distributed Ledger Technology (DLT)* – que foi bastante disruptiva ao permitir que duas partes efetuassem transações na nova criptomoeda sem necessidade da intermediação de uma entidade confiável. Desde sua concepção, a *blockchain* ganhou ampla popularidade e, com o advento do *Ethereum* [2, 3], evoluiu para suportar contratos inteligentes *("smart contracts")*. Por serem escritos em uma linguagem *Turing*-completa, esses contratos possibilitam a execução de algoritmos sofisticados, com inúmeras aplicações [4]. Assim, a *blockchain* se popularizou e passou a ser utilizada não apenas para transações com criptomoedas, mas em aplicações variadas, desde leilões, plataformas de *crowdfunding* e votações eletrônicas, até o armazenamento de registros médicos de pacientes e integração de dispositivos *IoT* [5, 4, 6].

No entanto, o próprio conceito da *blockchain* define que as transações devem ser registradas de forma pública no *ledger*, por meio do encadeamento de blocos de transações e seus respectivos *hashes*. Essa estratégia foi idealizada no *Bitcoin* com a finalidade de prevenir o gasto duplo ("*double spending*") e provou-se bastante eficaz ao impossibilitar que um usuário efetuasse transações sucessivas de forma que as saídas de sua carteira tivessem valor maior do que as entradas, já que todas as transações ficam armazenadas de maneira transparente, íntegra e imutável no *ledger*.

Por outro lado, essa publicidade das transações traz à tona uma importante preocupação acerca da privacidade. O artigo original do *Bitcoin* tentou endereçar a questão da privacidade ao relacionar as partes envolvidas apenas por meio de suas chaves públicas, a serem mantidas anônimas e trocadas a cada transação. Porém, ao longo do tempo, surgiram diversas técnicas de desanonimização e vinculação de transações (*"transaction linking"*) que demonstraram a fraqueza dessa abordagem [7, 8, 9]. Ademais, a exposição pública dos dados transacionais – como valores, datas e horários – permite inferências sobre as partes envolvidas, comprometendo ainda mais a privacidade dos usuários.

Dessa forma, observa-se que o *Bitcoin* não oferece a devida privacidade em seu desenho original. O *white paper* [2] e o *yellow paper* [3] do *Ethereum* tampouco abordam a questão da privacidade, o que resulta em um grande obstáculo para a implementação de diversas aplicações em que a privacidade é um requisito absoluto, como por exemplo:

- Leilão de Envelopes Lacrados: Na fase de licitação, os licitantes enviam seus lances em envelopes lacrados a um leiloeiro – nesse caso, um contrato inteligente – sem que nenhum licitante saiba o valor dos outros lances. Em uma fase posterior, os envelopes são abertos e o licitante que enviou o maior lance é declarado vencedor. Se os registros das transações de envio de lances fossem públicos, os licitantes veriam os lances uns dos outros, e poderiam ajustar suas propostas para vencer com uma margem mínima.

- Compartilhamento de Registros Médicos Eletrônicos – *Electronic Medical Records (EMR)*: Os prontuários e outros registros médicos dos pacientes, como exames, imagens, vídeos e documentos, podem ser armazenados de forma descentralizada em uma *blockchain*. Porém, uma vez que se tratam de dados pessoais dos pacientes, eles só devem ser acessados por pessoas autorizadas, o que requer uma *blockchain* com uma camada de privacidade adicional e o devido controle de acesso.

- Finanças Descentralizadas – *Decentralized Finance (DeFi)*: Os contratos inteligentes permitiram o surgimento de uma ampla gama de aplicações descentralizadas – *Decentralized Applications (DApps)* – e, nesse cenário, podem ser elaboradas diversas aplicações de *DeFi*. Um exemplo são as plataformas de empréstimos *peer-to-peer* e as *exchanges* descentralizadas (*DEXs*), onde ativos digitais e criptomoedas são trocados entre os usuários, sem necessidade de uma autoridade intermediária [10]. Nota-se que é imprescindível garantir a confidencialidade das transações em aplicações desse tipo.

- Moedas Digitais de Banco Central (*CBDCs*): Bancos centrais em diversos países estão atualmente estudando, testando ou planejando a emissão de moedas digitais, com muitos considerando plataformas de *blockchain* como a tecnologia subjacente para essas *CBDCs* [11, 12]. Um exemplo notável de *CBDC* é o *Drex*, a futura moeda digital do Brasil, que está sendo desenvolvida pelo Banco Central do Brasil [13]. Ainda em fase piloto, o projeto prevê a utilização de contratos inteligentes em uma *blockchain* permissionada para emitir o *Drex* e permitir transações com ativos tokenizados. A privacidade é um requisito essencial para *CBDCs* como o *Drex*: cidadãos e empresas não devem ser capazes de visualizar as transações uns dos outros, e instituições financeiras não devem ter acesso aos saldos de *CBDCs* ou outros ativos tokenizados de outras instituições.

Após a identificação dos problemas de privacidade do *Bitcoin*, novas criptomoedas foram desenvolvidas com enfoque na confidencialidade do conteúdo das transações, com destaque para o *Zerocash* [14] (hoje denominado *Zcash* [15]) e o *Monero* [16]. No entanto, tais projetos visam trazer privacidade para os pagamentos com essas criptomoedas e não são aplicáveis em contextos mais abrangentes, como as aplicações baseadas em contratos inteligentes. Há também outras plataformas de *DLT*, como o *Hyperledger Fabric* e o *R3 Corda*, que possuem funcionalidades que oferecem maior segurança e privacidade. Entretanto, elas são focadas apenas no modelo de *DLT* permissionado ou privado [17]. Recentemente, surgiram projetos de *blockchains* de camada 2 voltadas para privacidade e escalabilidade, como o *Manta Pacific* [18] e o *Aztec* [19]. Essas soluções permitem preservar a privacidade das transações sem sobrecarregar a camada principal da *blockchain*, reduzindo custos operacionais e melhorando o desempenho. Contudo, essas plataformas ainda enfrentam baixa adoção ou se encontram em estágios iniciais de desenvolvimento. Além disso, apresentam desafios relacionados à interoperabilidade, à complexidade de integração com as *blockchains* de camada 1 e à necessidade de confiança nos operadores da camada secundária [20].

Há, portanto, uma carência por soluções de privacidade que sejam eficientes e, ao mesmo tempo, abrangentes, que permitam sua aplicação em diferentes casos de uso, suportando contratos inteligentes em uma *blockchain* permissionada ou pública. Já foram propostas, não apenas por trabalhos acadêmicos como também pela indústria, uma série de soluções de privacidade para *blockchain*. As abordagens variam bastante quanto às técnicas utilizadas: provas de conhecimento zero – *Zero-Knowledge Proofs (ZKPs)* [21] –, criptografia homomórfica – *Homomorphic Encryption (HE)* [22] –, delegação para ambientes de execução confiáveis – *Trusted Execution Environment (TEE)* [23] –, delegação para terceiros confiáveis [6], ou uma combinação desses paradigmas. Estratégias adicionais, como computação multipartidária segura – *Secure Multi-Party Computation (SMPC)* –, mistura (*mixing*) e assinaturas em anel (*ring signatures*), também podem ser utilizadas para aumentar ainda mais a privacidade [7, 24, 25, 16]. Outra variação diz respeito ao modelo suportado pelas soluções: *Unspent Transaction Output (UTXO)* ou *Account-based* [26, 27]. Existem, ainda, variações no nível de privacidade alcançado: enquanto algumas soluções tornam confidenciais apenas os valores das transações (ou as entradas e saídas da computação realizada), outras garantem também o anonimato das partes envolvidas e, por fim, há aquelas em que toda a computação se torna privada [28].

Este trabalho de mestrado estuda o cenário atual das aplicações baseadas em *blockchain* sob o aspecto da privacidade, buscando identificar os desafios existentes, analisar e comparar as principais soluções de privacidade disponíveis e realizar uma avaliação prática de soluções selecionadas. Para essa avaliação, é desenvolvido um caso de uso simplificado

de *DeFi* por meio da aplicação *Miles2Coins*, cuja principal funcionalidade consiste na compra e venda de *tokens* de milhas aéreas. A partir da identificação dos requisitos de privacidade das aplicações *DeFi* em geral e, mais especificamente, da aplicação criada neste trabalho, são escolhidas duas soluções para serem implementadas e integradas à aplicação, com o objetivo de aprimorar a privacidade. O estudo descreve os obstáculos e dificuldades enfrentados durante o processo de integração e analisa o grau de aprimoramento da privacidade, além dos impactos no desempenho, nos custos transacionais, na complexidade e na usabilidade da aplicação resultante. Ao final, é realizada uma análise do cenário atual das *blockchains* e aplicações descentralizadas existentes, traçando um panorama atualizado da privacidade nesse contexto.

1.1 Objetivos

Esta dissertação de mestrado tem como objetivo geral investigar aspectos de privacidade em *blockchain*, avaliando e comparando soluções de privacidade disponíveis para aplicações baseadas nessa tecnologia. Essa avaliação é complementada pela análise do panorama atual da privacidade nesse contexto. Com este estudo, espera-se contribuir para o campo de pesquisa nesse tema, fornecendo aos desenvolvedores subsídios para escolher as abordagens mais adequadas para atender aos requisitos de privacidade de suas aplicações descentralizadas.

Para atingir este objetivo geral, os seguintes objetivos específicos foram definidos:

- Identificar e analisar os principais desafios de segurança e privacidade inerentes às tecnologias de *blockchain* e contratos inteligentes.
- Levantar os requisitos de privacidade para aplicações *DeFi*, considerando as particularidades desse ecossistema e as demandas regulatórias e técnicas.
- Conduzir uma análise conceitual e uma comparação qualitativa das soluções de privacidade existentes, explorando suas arquiteturas, vantagens e limitações.
- Avaliar a viabilidade prática de soluções selecionadas por meio de sua integração a uma aplicação *DeFi* simplificada, investigando os impactos dessa integração nos aspectos de privacidade, desempenho e custos transacionais.
- Examinar o panorama atual da privacidade em *blockchain*, identificando os avanços realizados e os desafios remanescentes para a pesquisa nessa área.

1.2 Organização do Texto

O restante desta dissertação está organizado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica e a revisão da literatura, abordando as principais áreas de conhecimento relacionadas ao tema em análise. Em seguida, o Capítulo 3 discute os requisitos de privacidade para aplicações *blockchain* e apresenta a análise conceitual e a comparação das soluções de privacidade. O Capítulo 4 investiga os desafios práticos da privacidade em *blockchain* por meio da implementação da aplicação *Miles2Coins* e sua integração com duas soluções de privacidade, encerrando com uma avaliação do panorama atual da privacidade em aplicações descentralizadas. Por fim, o Capítulo 5 apresenta as conclusões do trabalho.

Capítulo 2

Fundamentação Teórica e Revisão da Literatura

Neste capítulo, são detalhados os fundamentos teóricos sobre os temas relacionados ao presente estudo. São abordados os sistemas distribuídos, os requisitos de segurança cibernética e o conceito de privacidade de forma ampla. Em seguida, as tecnologias de *blockchain* e contratos inteligentes são descritas, assim como os aspectos de segurança e privacidade inerentes às aplicações implementadas sobre essas tecnologias. Por fim, é conduzida uma revisão da literatura sobre o tema, onde se evidencia a contribuição deste trabalho para o campo da privacidade em *blockchain* e contratos inteligentes.

O restante deste capítulo está estruturado da seguinte forma. A Seção 2.1 apresenta uma introdução aos sistemas distribuídos, destacando suas principais características. Em seguida, a Seção 2.2 aborda os fundamentos da segurança cibernética, juntamente com o conceito de privacidade. A Seção 2.3 explora os conceitos essenciais de *blockchain* e contratos inteligentes. Na sequência, a Seção 2.4 discute os desafios específicos de segurança e privacidade associados às tecnologias analisadas. A Seção 2.5 analisa as principais abordagens e técnicas utilizadas para aprimorar a privacidade de aplicações baseadas nesse ecossistema. A Seção 2.6 reúne a revisão da literatura existente sobre o tema e, por fim, a Seção 2.7 apresenta as considerações finais do capítulo.

2.1 Sistemas Distribuídos

Há múltiplas definições para o conceito de sistemas distribuídos. Enquanto [29] define sistema distribuído como "uma coleção de computadores independentes que, para os usuários, aparentam ser um único sistema", [30] afirma que trata-se de um sistema "em que os componentes de *hardware* ou *software*, localizados em computadores interligados em rede, comunicam-se e coordenam suas ações apenas por meio de mensagens". De todo

modo, os autores convergem ao abordar as características e funcionalidades dos sistemas distribuídos, que normalmente não são encontradas em sistemas centralizados [29, 30]:

- Transparência: Embora os componentes de um sistema distribuído estejam fisicamente separados, em redes distintas ou mesmo espalhados em regiões ou até países diferentes, essa distribuição geralmente é transparente aos usuários e desenvolvedores de aplicação, que percebem o sistema como sendo único e coeso.
- Escalabilidade: Indica que o sistema permanecerá operando de forma eficaz mesmo quando houver um aumento significativo no número de recursos e de usuários. Idealmente, os componentes de *software* do sistema não devem precisar ser alterados quando o sistema cresce em escala.
- Elasticidade: Refere-se à habilidade de oferecer capacidade computacional redimensionável de acordo com a demanda. Um sistema distribuído pode oferecer elasticidade do tipo horizontal – em que o número de componentes pode ser aumentado ou diminuído conforme a necessidade – ou vertical – onde um mesmo recurso ou nó do sistema é redimensionado com a mudança na carga de trabalho – ou ambos, simultaneamente.
- Concorrência: Os serviços e aplicações oferecidos por um sistema distribuído geralmente permitem que múltiplas requisições de clientes sejam processadas simultaneamente. Para tanto, seus processos devem funcionar de forma concorrente, utilizando os recursos compartilhados sem interferência entre eles.
- Tolerância a falhas: A implementação de componentes de *hardware* e *software* redundantes permite que um sistema distribuído seja capaz de continuar funcionando mesmo que um ou mais de seus componentes deixem de funcionar. A configuração física e lógica das redes que interconectam sistemas distribuídos também costuma prever rotas alternativas entre os componentes, tornando-os resilientes a problemas em roteadores, cabos ou outros elementos de rede.

De maneira mais abrangente que o conceito de tolerância a falhas, existe a concepção de *dependabilidade* (tradução literal a partir do termo em inglês "*dependability*"), que consiste na capacidade do sistema de fornecer serviços que possam ser justificadamente confiáveis [31]. Tal ideia engloba atributos como disponibilidade, confiabilidade, integridade e manutenibilidade e está intimamente ligada ao conceito de segurança, que adiciona um último atributo ao grupo: confidencialidade. Assim, idealmente, um sistema distribuído deve possuir a característica de *dependabilidade*, além de atender aos requisitos de segurança, abordados na Seção 2.2 a seguir, juntamente com os aspectos de privacidade.

2.2 Segurança e Privacidade

O conceito de segurança está relacionado à proteção das informações contra acesso não autorizado, uso ou divulgação indevida, interrupção, modificação ou destruição [32]. Isso envolve a preservação de três requisitos mínimos [33]:

- Confidencialidade: A informação não deve ser disponibilizada ou revelada a pessoas, sistemas, órgãos ou entidades não autorizados.
- Integridade: A informação não deve ser modificada ou destruída de maneira não autorizada ou acidental.
- Disponibilidade: A informação deve estar acessível e utilizável, sob demanda, por uma pessoa ou determinado sistema, órgão ou entidade devidamente autorizado.

Outros requisitos associados à segurança da informação são a autenticidade e o não-repúdio. Para atendê-los, é necessário assegurar, de forma irrefutável, que a informação foi produzida, modificada ou destruída por determinada pessoa, sistema, órgão ou entidade.

A segurança cibernética envolve a proteção de sistemas, redes e outros ativos de tecnologia da informação contra ameaças cibernéticas, garantindo a confidencialidade, integridade e disponibilidade das informações armazenadas, trafegadas ou processadas por esses ativos [34, 33].

As principais ameaças de segurança cibernética se tornam cada vez mais sofisticadas à medida que a tecnologia evolui. Enquanto nos anos 90 a maioria dos *malwares* eram simples e apenas destruíam informações, na última década as ameaças mudaram o enfoque para os dispositivos móveis e a computação em nuvem, com os *hackers* inclusive utilizando inteligência artificial para perpetrar ataques de maior efetividade e impacto [35, 36, 37].

Ataques de vazamento de dados têm se tornado cada vez mais comuns e causam grande preocupação às empresas, que sofrem prejuízo médio de quase 4 milhões de dólares nesse tipo de incidente [37]. Os dados pessoais de clientes e funcionários, se vazados, podem gerar grandes prejuízos financeiros, devido a processos e multas, além de danos à reputação das empresas. Ademais, as pessoas cujos dados foram vazados ficam sujeitas a golpes, estelionatos e outros crimes. Nesse sentido, é imprescindível resguardar a privacidade dos dados produzidos, armazenados e processados pelos sistemas de informação.

É importante notar que os conceitos de confidencialidade e privacidade estão intimamente relacionados, mas não são sinônimos. O primeiro está focado na informação em si, já o último está relacionado às pessoas e suas informações pessoais. Assim, enquanto confidencialidade se refere à não divulgação ou revelação de informações a entes não autorizados, o conceito de privacidade pode ser definido como o direito de um indivíduo de manter indevassados os dados e informações que lhe digam respeito, ou seja, trata-se do

controle que se assegura ao indivíduo sobre a divulgação ou exposição de manifestações próprias de sua vida privada [38].

A publicação de normas como a *General Data Protection Regulation (GDPR)* [39] na União Europeia e a Lei Geral de Proteção de Dados Pessoais (LGPD) [40] no Brasil mostram a crescente preocupação dos governos com a privacidade. Tais normativos têm como foco a proteção dos direitos individuais em relação aos dados pessoais, estabelecendo diretrizes para a coleta, armazenamento, processamento e compartilhamento dessas informações, ressaltando também a responsabilidade das organizações no uso adequado e seguro das informações pessoais sob seu controle.

Há diversos mecanismos e práticas disponíveis para melhorar a segurança cibernética das instituições e suas aplicações, além de aprimorar a privacidade dos usuários. Nesse sentido, existem *frameworks* que fornecem princípios, boas práticas e controles que ajudam empresas e instituições a implementar medidas eficazes para prevenir incidentes. Nesse contexto, destacam-se os *frameworks* descritos abaixo:

- *NIST Cybersecurity Framework (CSF)* [41]: Atualmente na versão 2.0, o *CSF* apresenta um conjunto de objetivos de segurança que orientam empresas, instituições e outras organizações na gestão de riscos de segurança cibernética. Diferentemente de um *checklist* prescritivo, o *CSF* oferece uma estrutura adaptável às necessidades específicas de cada organização. Para auxiliar na identificação e implementação de controles, o *NIST* publicou o documento detalhado "*Security and Privacy Controls for Information Systems and Organizations*" [42], que fornece um catálogo abrangente de controles alinhados ao *CSF* e a outros normativos americanos.
- *NIST Privacy Framework* [43]: Lançado em 2020, este *framework* é organizado de maneira semelhante ao *CSF*, mas com foco específico nos aspectos relacionados à privacidade. Ele visa ajudar organizações a identificar e gerenciar riscos à privacidade enquanto atendem às regulamentações e expectativas de proteção de dados.
- *CIS Controls* [44]: Ao contrário dos *frameworks* do *NIST*, que têm uma abordagem não-prescritiva, o *CIS Controls* tem um enfoque mais prático, prevendo controles e ações detalhadas para prevenir e mitigar riscos de segurança. Atualmente, o documento está na versão 8.1, lançada em junho de 2024.
- *CIS Controls Privacy Companion Guide* [45]: Lançado em 2022, este guia complementa os *CIS Controls* ao interpretar os aspectos de privacidade relacionados a cada controle. O documento promove o alinhamento entre os controles do *CIS*, os princípios de privacidade e regulamentações como a GDPR, permitindo a análise das implicações de privacidade relacionadas a cada controle.

- *Framework* de Privacidade e Segurança da Informação [46]: Desenvolvido pelo governo brasileiro no âmbito do Programa de Privacidade e Segurança da Informação (PPSI), este *framework* oferece diretrizes para instituições públicas, visando identificar, acompanhar e corrigir lacunas relacionadas à privacidade e à segurança da informação. Baseado principalmente nos *CIS Controls*, o *framework* também se inspira no *NIST Privacy Framework* e está alinhado à legislação brasileira, especialmente à LGPD.

Os controles e atividades previstos em tais plataformas incluem não apenas a implantação de sistemas físicos e lógicos de segurança, mas também de processos de gerenciamento de ativos e de identidades, práticas de conscientização em segurança, gerenciamento de riscos e elaboração de planos de recuperação de incidentes.

Considerando a crescente sofisticação dos ataques e a necessidade de se garantir a privacidade dos usuários, é primordial que os sistemas que lidam com dados pessoais sejam construídos de maneira segura e considerando a privacidade como requisito fundamental desde a sua concepção – princípio conhecido como "*privacy by design*" [47].

2.3 *Blockchain* e Contratos Inteligentes

Idealizada originalmente no *white paper* do *Bitcoin* [1], a *blockchain* pode ser definida como uma estrutura de dados baseada em uma cadeia de blocos interligados, em que cada bloco armazena um conjunto de transações ou outras informações. Cada novo bloco a ser incluído na cadeia inclui o *hash* do bloco anterior, criando assim um encadeamento (vide Figura 2.1), de forma que qualquer alteração em um bloco anterior invalidaria toda a cadeia. Assim, a *blockchain* permite o registro das transações de forma cronológica e íntegra.

Uma plataforma de *blockchain* consiste em uma rede descentralizada, onde os nós são independentes, porém, trabalham cooperativamente seguindo um protocolo comum. Cada nó da rede mantém uma cópia integral da cadeia de blocos. A inclusão de um novo bloco na cadeia é realizada através de um consenso entre os nós da rede e todos os nós atualizam sua cópia local da cadeia para refletir essa inclusão. Uma vez que os nós mantêm, de forma distribuída, uma mesma cadeia ou registro, pode-se definir uma plataforma de *blockchain* como um tipo de *Distributed Ledger Technology (DLT)* e, de maneira mais abrangente, como um sistema distribuído.

Atualmente, existem diversas plataformas de *blockchain*, que variam quanto às aplicações, modelos de consenso suportados, escalabilidade, desempenho, dentre outros fatores. Uma classificação comum é quanto à abertura da *blockchain* para a participação dos nós na rede [48, 49]:

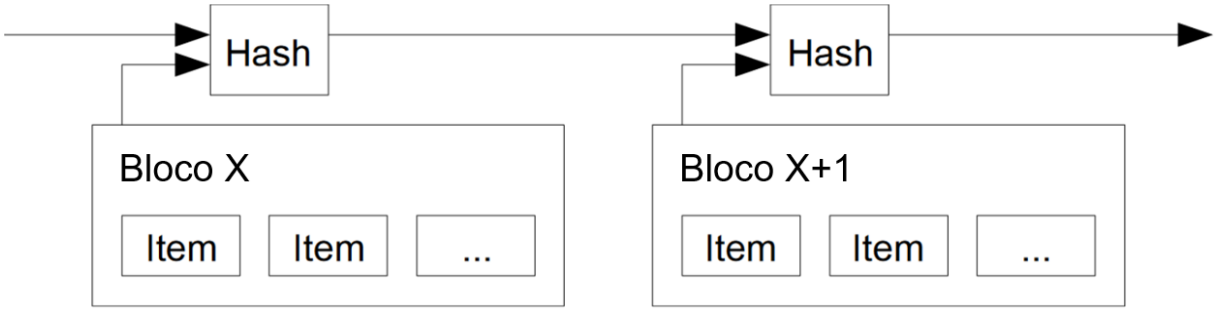

Figura 2.1: Encadeamento de blocos (Adaptado de [1]).

- *Blockchains* públicas: Também chamadas de não permissionadas, consistem em *blockchains* totalmente descentralizadas, onde os nós podem entrar ou sair da rede de forma irrestrita. Nelas, os dados ficam públicos na cadeia e disponíveis para leitura e modificação por qualquer nó participante. Os principais exemplos de *blockchains* públicas são o *Bitcoin* [1] e o *Ethereum* [3].
- *Blockchains* permissionadas: Nelas, a participação é restrita a determinados nós previamente autorizados, e os dados da cadeia são visíveis somente aos participantes. Dentro dessa categoria, pode-se enquadrar as *blockchains* de consórcio – onde um conjunto de organizações define as regras de participação e realiza a governança do sistema – ou privadas – pertencentes a um indivíduo ou organização. Um exemplo de *blockchain* permissionada é o *Hyperledger Fabric* [50].
- *Blockchains* híbridas: São plataformas mais flexíveis, que combinam funcionalidades e características tanto de *blockchains* públicas como permissionadas. Atualmente, existem diversos projetos de soluções híbridas disponíveis. Um exemplo desse tipo de solução é o *Dragonchain* [51].

Embora originalmente, no artigo do *Bitcoin* [1], os blocos que compunham a *blockchain* tenham sido concebidos para armazenar apenas transações realizadas com aquela criptomoeda, isso mudou com o surgimento do *Ethereum*, uma plataforma de *blockchain* com suporte a contratos inteligentes. Na definição original que consta no *white paper* do *Ethereum* [2], os contratos inteligentes seriam "sistemas que movem automaticamente ativos digitais de acordo com regras arbitrárias pré-especificadas". Porém, uma vez que o *Ethereum* introduziu uma nova linguagem *Turing*-completa – posteriormente denominada *Solidity* [52] – para desenvolvimento dos contratos, eles podem ser programas bastante complexos, suportando qualquer tipo de computação. Assim, em uma definição mais atual, um contrato inteligente é um programa descentralizado implantado em uma *blockchain* que garante a execução de protocolos e acordos sem a necessidade de envolver terceiros ou estabelecer confiança mútua entre as partes [4].

Com o advento dos contratos inteligentes, portanto, tornou-se viável a implementação de uma ampla gama de aplicações descentralizadas (*DApps*) utilizando *blockchain*, abrangendo desde plataformas de jogos e redes sociais até soluções de identidade digital e outros casos de uso inovadores [53, 54]. Um campo de grande importância nesse cenário são as aplicações de finanças descentralizadas (*DeFi*), como as plataformas de empréstimos *peer-to-peer*, as *exchanges* descentralizadas (*DEX*) e as aplicações de operações com derivativos e de gerenciamento automatizado de ativos. O mercado de *DeFi* é bastante significativo – no final de 2024, um total de US$ 121 bilhões circulava em aplicações desse tipo [55].

Após o *Bitcoin* e o *Ethereum*, diversas plataformas de *blockchain* foram desenvolvidas, como *Solana* [56], *BNB Smart Chain* [57], *Base* [58], entre outras. Ademais, além do *Solidity*, outras linguagens de programação passaram a ser utilizadas para desenvolver contratos inteligentes, como *Vyper*, *Go*, *Rust*, e *Move* [59].

2.3.1 Conceitos Relacionados

Para melhor entendimento do restante deste trabalho, é importante o detalhamento de determinados conceitos específicos relacionados às tecnologias de *blockchain* e contratos inteligentes.

Criptomoedas, *Stablecoins* e *CBDCs*

Criptomoedas são formas de dinheiro digital que utilizam criptografia para armazenar as transações de maneira segura em um registro distribuído *(DLT)* sem requerer agentes centralizados para sua operação [60]. O *Bitcoin* [61] é a primeira e mais conhecida criptomoeda, mas hoje já existem milhares de criptomoedas em operação, como *ETH (Ethereum)* [62], *SOL (Solana)* [56], *ADA (Cardano)* [63] e *BNB (Binance)* [64]. Geralmente, as criptomoedas são descentralizadas e operam sobre plataformas *peer-to-peer* baseadas em *blockchain*.

As criptomoedas podem ser divididas em duas categorias principais: criptomoedas não lastreadas e *stablecoins*. O primeiro grupo inclui moedas que não possuem respaldo em ativos financeiros como ouro ou dólar, portanto seu valor pode apresentar volatilidade elevada. Já as *stablecoins* compreendem criptomoedas projetadas para serem respaldadas por uma ou mais moedas fiduciárias ou por ativos como ouro e outras *commodities*, buscando, assim, apresentar maior estabilidade de valor em comparação com criptomoedas tradicionais [65]. Alguns exemplos de *stablecoins* são o *Tether (USDT)* [66] e *USD Coin (USDC)* [67], atreladas ao dólar americano, e o *Paxos Gold (PAXG)* [68], ancorado ao ouro.

Por fim, as *Central Bank Digital Currencies (CBDCs)* representam formas digitais de moedas emitidas por um banco central. Ao contrário das criptomoedas, as *CBDCs* são regulamentadas por autoridades governamentais, podem possuir paridade com a moeda soberana de determinado país e serem instituídas como meio legal de pagamento. As *CBDCs* podem ser divididas em *CBDCs* de varejo, que atuam como uma versão digital do dinheiro em espécie, universalmente acessível aos cidadãos, e *CBDCs* de atacado, que, de forma similar a contas de liquidação em bancos centrais, seriam acessíveis apenas a instituições financeiras. Espera-se que as *CBDCs* utilizem *DLT* como tecnologia subjacente, embora isso não seja obrigatório [60].

Enquanto países como Nigéria e Bahamas já lançaram suas próprias *CBDCs* [69, 70], mais de cem outros países estão avançando nos estudos sobre a implementação dessas moedas digitais [71]. Alguns projetos de *CBDC* estão na fase piloto, como ocorre no Brasil com o Real Digital *(Drex)*, conforme mencionado no Capítulo 1.

Tokens **fungíveis e não-fungíveis**

No contexto de *blockchain*, *tokens* são ativos digitais que representam valores ou bens e são construídos sobre plataformas de contratos inteligentes, como o *Ethereum* [72]. Eles são implementados por meio de contratos inteligentes específicos, que estabelecem funções e eventos associados, permitindo operações como criação (*"mint"*), destruição (*"burn"*) e transferência (*"transfer"*) entre diferentes contas. Além disso, os *tokens* possuem a função padrão *"approve"*, que permite a um usuário conceder permissão a um contrato inteligente para transferir, em seu nome, uma quantidade específica de seus *tokens* [73]. Os *tokens* podem ser classificados como fungíveis ou não-fungíveis, de acordo com suas características:

- *Tokens* fungíveis: Aqueles cujas unidades individuais são intercambiáveis, onde cada unidade é indistinguível e tem o mesmo valor das demais [74]. Exemplos desse tipo de *token* são as criptomoedas e *stablecoins*. No *Ethereum*, o *ERC-20* [75] é o padrão mais amplamente adotado para *tokens* fungíveis, estabelecendo um conjunto de regras e funções padronizadas para sua criação e gestão.
- *Tokens* não-fungíveis (*NFT*): São únicos, indivisíveis em unidades menores e geralmente estão vinculados a ativos físicos ou digitais, como colecionáveis, obras de arte e outros bens [76, 74]. O padrão *ERC-721* [77] define esse tipo de *token* no *Ethereum*.

Existem também outros padrões de *tokens*, onde se destaca o padrão *multi-token ERC-1155* [78]. Com ele, é possível especificar múltiplos *tokens*, fungíveis e/ou não-fungíveis,

em um único contrato inteligente. Mais eficiente, esse padrão simplifica a implementação de *tokens* e reduz custos ao habilitar operações em lote, como transferências e consulta de saldo de diversos *tokens* em uma única transação.

Delivery vs Payment (DvP)

Como já mencionado, as aplicações de *DeFi* atuam de maneira descentralizada, sem a necessidade de intermediários, como bancos, cartórios ou outras instituições. Nesse contexto, o termo *Delivery vs Payment (DvP)* é um conceito frequentemente associado a transações de ativos digitais, onde a entrega dos ativos ocorre simultaneamente com o pagamento. Essa abordagem é utilizada para assegurar que ambas as partes em uma transação cumpram suas obrigações, garantindo a integridade e a segurança das operações na ausência de intermediários centralizados [79].

Modelos *UTXO* e *Account-Based*

A *blockchain* consiste em uma estrutura de dados que registra todos os eventos e transações ao longo do tempo. Diferentes plataformas de *blockchain* empregam modelos distintos para atualizar esses registros no *ledger*. Dois modelos amplamente reconhecidos são o *Unspent Transaction Output (UTXO)*, adotado pelo *Bitcoin*, e o *Account-Based*, utilizado no *Ethereum*.

No *Bitcoin*, as transações incluem informações sobre remetentes, destinatários, entradas (*"inputs"* – valores sendo gastos) e saídas (*"outputs"* – valores sendo recebidos). Assim, no modelo *UTXO*, os saldos são representados pelas saídas (*"outputs"*) de transações não gastas – daí advém o nome *"Unspent Transaction Output"*. Cada transação consome *UTXOs* como *"inputs"* e gera novos *UTXOs* como *"outputs"*. Isso permite o rastreamento da propriedade de determinado ativo de forma simples e direta, uma vez que cada *UTXO* é vinculado a uma ou mais transações anteriores. Embora o modelo *UTXO* seja eficiente para transações simples, como transferências de moeda, utilizar esse modelo pode ser desafiador na implementação de contratos inteligentes mais complexos [26, 27].

O modelo *Account-Based* difere do modelo *UTXO* ao associar o saldo diretamente a cada conta. As transações neste modelo geram alterações diretas nos saldos das contas do remetente e do destinatário. Os saldos são armazenados em um estado global e são atualizados a cada transação. Essa abordagem é mais flexível e eficiente para implementações complexas de contratos inteligentes, pois não requer o rastreamento de múltiplos *UTXOs*. No entanto, esse modelo pode ser ineficiente em casos de uso mais simples [26, 27]. Existem também outros modelos, como o *Extended UTXO (eUTXO)* [80], que estendeu o modelo *UTXO* com *scripts* para suportar funcionalidades mais avançadas.

EVM e *Gas*

Uma vez que o *Ethereum* é, atualmente, a *blockchain* com suporte a contratos inteligentes mais relevante, é essencial definir dois conceitos fundamentais dessa plataforma: a *Ethereum Virtual Machine (EVM)* e o *Gas*.

A *EVM* é o ambiente de execução responsável pelo processamento de contratos inteligentes no *Ethereum*. Trata-se de uma máquina virtual descentralizada e determinística, replicada em todos os nós da rede para garantir que as mesmas instruções sejam processadas e resultem no mesmo estado final. Os contratos são escritos em linguagens de alto nível, como *Solidity* [52], e compilados para *bytecode*. Esse código é interpretado e executado pela *EVM* de maneira uniforme em toda a rede, assegurando a segurança da execução [2].

O *gas*, por sua vez, é a unidade de medida do custo computacional das operações na rede *Ethereum*. Cada instrução executada pela *EVM* possui um custo fixo em *gas*, conforme definido na especificação da rede. O preço do *gas* é expresso em *gwei* (onde 1 *gwei* equivale a 10^{-9} *ETH*) e varia dinamicamente conforme a demanda da rede. O custo total de uma transação é calculado multiplicando a quantidade de *gas* consumida pelo preço do *gas* no momento da execução. Esse modelo serve para evitar sobrecarga na rede e incentiva a otimização dos contratos inteligentes para reduzir custos operacionais [2].

Transações *On-Chain* e *Off-Chain*

As transações em aplicações descentralizadas podem ser processadas de diferentes formas, dependendo de onde ocorrem e de como são registradas [81, 82]:

- *On-Chain*: Refere-se às transações realizadas diretamente na *blockchain* principal. Essas transações são validadas por nós da rede e registradas de forma imutável no *ledger*, garantindo transparência e integridade. Exemplos incluem transferências de criptomoedas e execução de contratos inteligentes na rede principal, como *Bitcoin* e *Ethereum*.
- *Off-Chain*: São transações que ocorrem fora da *blockchain* principal. Essas operações podem ser realizadas em sistemas secundários, sem a necessidade de registrar cada detalhe na *blockchain*. A realização de transações *off-chain* permite aumentar a escalabilidade e reduzir custos, pois evita a sobrecarga da rede principal. Um exemplo são as transações em *blockchains* de camada 2, discutidas a seguir.

Blockchains de Camada 1 e Camada 2

A evolução da tecnologia de *blockchain* levou ao desenvolvimento de arquiteturas que se diferenciam entre *blockchains* de camada 1 e camada 2 [82]:

- *Blockchain* de Camada 1 *("Layer 1")*: Representa a infraestrutura principal onde todas as transações e contratos inteligentes são processados e validados. Ela é responsável pela segurança, consenso e operação básica da rede. Exemplos incluem *Bitcoin* [61], *Ethereum* [62] e *Solana* [56]. A camada 1 é a fundação sobre a qual outras soluções podem ser construídas. Seu desempenho, contudo, é limitado, pois cada transação deve ser validada por todos os nós da rede, resultando em uma taxa relativamente baixa de transações por segundo.
- *Blockchain* de Camada 2 *("Layer 2")*: Refere-se a soluções construídas sobre uma *blockchain* de camada 1 para melhorar sua escalabilidade, eficiência [83, 58] e, em determinados casos, privacidade [18]. Essas soluções processam transações fora da rede principal *(off-chain)* e posteriormente as registram na camada 1, reduzindo a carga e os custos. Exemplos de *blockchains* de camada 2 incluem *Optimism* [83] e *Base* [58].

2.4 Desafios de Segurança e Privacidade em *Blockchain*

A arquitetura inovadora da tecnologia *blockchain*, combinada com a expressividade dos contratos inteligentes e a diversidade de plataformas de *blockchain* disponíveis, traz novos obstáculos para a segurança das aplicações e a privacidade dos usuários. Esses desafios, que diferem dos encontrados em sistemas convencionais, surgem principalmente devido à natureza descentralizada e transparente das *blockchains*.

2.4.1 Desafios de Segurança

Os principais entraves relacionados à segurança das aplicações sobre *blockchain* são descritos abaixo.

1. Dificuldade de correção de vulnerabilidades: Toda transação ou informação inserida na *blockchain*, incluindo os próprios contratos inteligentes, torna-se imutável após ser publicada. Enquanto isso assegura a integridade e disponibilidade do contrato, proporcionando previsibilidade em seu funcionamento, também apresenta um desafio significativo: a incapacidade de corrigir falhas ou vulnerabilidades após a publicação do contrato. Embora existam estratégias para mitigar esse problema, como contratos atualizáveis, todas apresentam contrapartidas que limitam sua eficácia [84, 85].

2. Análise complexa de segurança: Contratos inteligentes são "*stateful*", de forma similar a máquinas de estado, pois podem alcançar estados específicos após uma sequência determinada de transações. Esse comportamento cria um obstáculo para as soluções de análise de segurança, uma vez que elas podem não conseguir cobrir todo o código, todas as funcionalidades e todos os estados possíveis de determinado contrato [86].

3. Interação com usuários e outros contratos: Determinado contrato inteligente pode enviar e receber transações, comunicando-se não somente com usuários, mas também com outros contratos, que podem ser maliciosos, explorando eventuais vulnerabilidades do contrato. Além disso, um contrato pode interagir com fontes externas ("*external oracles*"), o que, por sua vez, pode resultar no vazamento de informações sensíveis ou em ataques provenientes dessas fontes [87].

4. Bloqueio de valores: Os contratos inteligentes podem ser capazes de receber e armazenar valores, como *Ether (ETH)* ou outras criptomoedas e *tokens*. Esses valores ficam bloqueados nos contratos e podem ser comprometidos caso haja a exploração de uma vulnerabilidade que permita o desvio desses ativos para uma carteira maliciosa ou para um contrato controlado por um agente mal-intencionado. Ademais, os valores podem ser "congelados" (do inglês "*freeze*") caso um invasor consiga enviá-los para uma conta inexistente de forma intencional [88].

5. Publicidade dos contratos inteligentes: No *Ethereum* e em outras *blockchains* sem enfoque na privacidade, as informações dos contratos ficam públicas na *blockchain*. Isso permite que agentes maliciosos visualizem o valor atual armazenado em determinado contrato inteligente, além de inferir suas funcionalidades e analisar seu "*bytecode*". Como resultado, contratos que gerenciam altas quantidades de determinada criptomoeda ou ativo digital tornam-se alvos fáceis para análises minuciosas por criminosos em busca de vulnerabilidades que possam ser exploradas [89].

2.4.2 Desafios de Privacidade

Em cenários como *DeFi* e outras aplicações que lidam com dados pessoais, financeiros ou informações sensíveis, a privacidade é uma preocupação primordial. Porém, no ambiente de *blockchain* há obstáculos significativos para resguardar a privacidade, onde se destacam:

1. Transparência das transações: Os detalhes das transações realizadas em *blockchain* são acessíveis a todos os participantes da rede, permitindo que qualquer nó visualize as chamadas de funções de contratos, seus inputs e outputs. Essa transparência, embora aumente a confiança e a auditabilidade da rede, gera riscos significativos

à privacidade, pois os históricos de transações dos usuários podem ser rastreados, potencialmente revelando informações sensíveis sobre suas atividades financeiras, identidades e relacionamentos, podendo resultar também no roubo de propriedade intelectual e outras violações de privacidade [7].

2. Vinculação de transações (*"transaction linking"*): A natureza pseudo-anônima das principais plataformas *blockchain*, como o *Bitcoin* e o *Ethereum*, não é suficiente para garantir a privacidade [24]. Diversas técnicas permitem vincular as transações aos usuários envolvidos, analisando e correlacionando registros de transações para associar endereços de chaves públicas aos seus respectivos usuários, o que pode expor suas identidades e atividades financeiras [7, 9, 90]. A análise de grafos de transações, uma técnica comumente empregada nesse contexto, envolve a construção de um grafo das transações na *blockchain*, onde os nós representam endereços ou carteiras, e as arestas representam transações entre esses nós [91]. Ao examinar esses grafos, é possível descobrir padrões e, potencialmente, identificar as entidades reais por trás desses endereços [90]. Algoritmos de agrupamento ajudam a agrupar endereços que provavelmente pertencem ao mesmo usuário e heurísticas adicionais podem refinar esses grupos. Por exemplo, se múltiplos endereços forem utilizados como entradas em uma única transação, é provável que pertençam ao mesmo usuário [91].

3. Rastreamento via rede: Como as transações em *blockchain* ocorrem em uma rede *peer-to-peer*, geralmente na Internet, o tráfego de rede pode ser analisado para rastrear usuários através de seus endereços *IP* [8]. Embora esse risco possa ser mitigado com o uso de métodos de aprimoramento de privacidade, como o *TOR* [92] para ofuscar o endereço *IP* de origem das transações, essas técnicas não são completamente infalíveis, agregam complexidade adicional e podem ser vulneráveis a outras formas de ataques [93].

4. Gestão de chaves privadas: A forma como os usuários gerenciam as chaves privadas relacionadas às suas contas na *blockchain* é outra questão crítica que impacta na privacidade. As chaves privadas são essenciais para assinar transações, e seu comprometimento pode resultar em roubo de identidade e perda de fundos. As carteiras *offline* oferecem proteção ao reduzir a superfície de ataque, mas as chaves podem ser perdidas ou furtadas se não forem armazenadas e copiadas de forma cuidadosa. Por outro lado, terceirizar a gestão de chaves privadas para serviços de terceiros exige total confiança nesses provedores, que se tornam alvos preferenciais para ataques cibernéticos [7].

5. Impossibilidade de exclusão de dados: Uma vez que os dados são escritos na *blockchain*, eles não podem ser removidos ou modificados, dificultando a conformidade

com regulamentações de privacidade como a LGPD que, no seu artigo 18, resguarda ao titular o direito à eliminação dos seus dados [40]. Soluções como armazenamento *off-chain* ou mecanismos para permitir a exclusão de dados privados na *blockchain* podem mitigar esse problema, mas exigem uma implementação cuidadosa para garantir tanto a privacidade quanto a conformidade [94, 95].

6. Privacidade *vs.* usabilidade: Desenvolvedores frequentemente enfrentam dificuldades com a complexidade de implementação e gestão dos mecanismos de preservação de privacidade. Interfaces amigáveis e camadas de abstração são necessárias para tornar essas tecnologias acessíveis e práticas. Além disso, com o surgimento de centenas de *blockchains*, cada uma com sua própria arquitetura e características específicas, garantir a interoperabilidade entre diferentes cenários habilitados para *blockchain*, enquanto se mantém a privacidade, é fundamental, mas bastante desafiador [7].

2.5 Técnicas para Aprimorar a Privacidade

Diante do cenário desafiador descrito anteriormente, diversas abordagens têm sido propostas para aprimorar a privacidade dos usuários e garantir a confidencialidade das transações em aplicações baseadas em *blockchain* e contratos inteligentes. De modo geral, essas abordagens fundamentam-se no uso de criptografia, na delegação da computação de elementos privados para entidades confiáveis ou em outras técnicas. Estas estratégias, que também podem ser utilizadas em conjunto dependendo da necessidade, serão apresentadas a seguir.

2.5.1 Técnicas de Criptografia

Provas de Conhecimento Zero

As Provas de Conhecimento Zero – *Zero-Knowledge Proofs (ZKPs)* – são protocolos que permitem que uma das partes envolvidas – o provador ("*prover*") – convença outra parte – o verificador ("*verifier*") – de que uma afirmação sobre determinados dados é verdadeira, sem revelar ou vazar qualquer informação além da veracidade dessa afirmação. Em outras palavras, as *ZKPs* possibilitam a comprovação da validade de afirmações lógicas relacionadas a dados privados sem a necessidade de revelar os próprios dados. Por exemplo, utilizando *ZKP*, pode-se provar que uma transação está transferindo determinada moeda, sem revelar a origem, o destino e o valor transferido, mas ainda garantindo a validade da transação [21].

Criptografia Homomórfica

A criptografia homomórfica é um esquema criptográfico que permite a execução de operações matemáticas sobre dados criptografados sem a necessidade de decifrá-los anteriormente. Existem diferentes níveis de criptografia homomórfica: aditiva, multiplicativa e totalmente homomórfica. Na criptografia homomórfica aditiva, é possível realizar operações de adição ou subtração nos dados criptografados. Na criptografia homomórfica multiplicativa, podem ser realizadas operações de multiplicação e divisão nos dados cifrados. Por fim, a criptografia totalmente homomórfica – *Fully Homomorphic Encryption (FHE)* – vai além, permitindo uma gama completa de operações aritméticas sobre os dados cifrados, preservando a privacidade ao longo do processo de computação [28].

2.5.2 Delegação

Outra abordagem adotada por determinadas soluções de privacidade consiste na delegação da computação dos dados privados para uma entidade externa, que pode ser um ambiente de execução confiável em *hardware* ou terceiros confiáveis, comumente denominados *"managers"*.

Delegação para Ambiente de Execução Confiável

Existem tecnologias que permitem a criação de um ambiente de execução confiável – *Trusted Execution Environment (TEE)* – que consiste em uma área isolada, chamada de "enclave seguro", dentro dos processadores. Esse enclave proporciona um espaço dedicado na memória para que as aplicações executem operações de forma segura e isolada, sem interferência de outros processos ou sistemas operacionais em execução no mesmo *hardware*. Essa abordagem permite proteger dados sensíveis contra ameaças como *softwares* maliciosos ou administradores de sistema não confiáveis, proporcionando uma camada adicional de segurança em nível de *hardware* para garantir a confidencialidade e integridade dos dados. Adicionalmente, um enclave seguro pode oferecer uma atestação ("*attestation*"), que é uma declaração baseada em *hardware* que comprova, de maneira irrefutável, as operações executadas dentro do enclave, permitindo a verificação por terceiros [96]. A tecnologia de enclave seguro em *hardware* mais utilizada é o *Intel Software Guard Extensions (SGX)* [97], mas outros fabricantes, como a *AMD* [98] e a *ARM* [99], também oferecem tecnologias semelhantes, com diferentes funcionalidades e limitações. Existem também propostas de enclaves seguros baseados em *software* [23].

Como será visto mais adiante neste trabalho, há propostas de *blockchains* cuja privacidade é baseada na delegação para ambientes de execução confiáveis, repassando para enclaves seguros a computação de elementos privados das transações. Um ponto de dis-

cussão importante é que já foram descobertas várias vulnerabilidades relacionadas ao *Intel SGX* [100] e outras tecnologias de enclave seguro [101], cuja exploração resultaria na potencial exposição das chaves privadas associadas ao enclave e, portanto, comprometeria as transações protegidas pelo enclave vulnerável.

Delegação para Terceiros Confiáveis

Outra abordagem para aprimorar a privacidade é delegar operações que envolvem dados privados a terceiros confiáveis, geralmente chamados de "*managers*". Eles são responsáveis por processar, de maneira segura e *off-chain*, os dados sensíveis, garantindo que apenas os resultados necessários sejam publicados na *blockchain*, sem expor informações confidenciais. Nesse modelo, a integridade dos "*managers*" é um fator crítico, pois confia-se que eles cumprirão seu papel de preservar o sigilo dos dados [28].

2.5.3 Outras Técnicas

Além das técnicas de criptografia e de delegação, a privacidade em sistemas de *blockchain* pode ser aprimorada por meio de métodos complementares, incluindo:

Computação Multipartidária Segura *(SMPC)*

Um protocolo *Secure Multi-Party Computation (SMPC)* permite que um grupo de participantes, sem confiança entre si, execute funções sobre suas entradas privadas, revelando apenas o resultado da computação. Esse processo é projetado para garantir a privacidade, assegurando que nenhum participante possa acessar informações sobre as entradas individuais dos demais, além do que pode ser inferido pelo resultado. No contexto de *blockchains* que oferecem recursos de privacidade, os protocolos *SMPC* podem ser utilizados para executar computações *off-chain* sobre as entradas de múltiplos usuários, substituindo entidades confiáveis centralizadas, entre outras possibilidades [28].

Mistura (*Mixing*)

Esta técnica envolve a ofuscação da relação entre endereços de envio e recebimento de transações. Isso pode ser alcançado combinando múltiplas transações de diferentes usuários em uma única transação ou utilizando serviços de mistura, que atuam como intermediários para embaralhar as moedas dos usuários e, em seguida, enviar novas transações com valores equivalentes aos destinatários [24, 25].

Assinaturas em Anel (*Ring Signatures*)

Este método permite que um usuário assine uma transação em nome de um grupo, sem revelar qual membro realmente produziu a assinatura. Ele oferece anonimato ao assinante dentro do grupo, tornando impossível determinar quem assinou a transação [24, 16].

2.6 Revisão da Literatura

Para este trabalho, foi conduzida uma extensa revisão da literatura, incluindo a análise de pesquisas relacionadas à segurança e privacidade em *blockchain* e contratos inteligentes e o estudo de diferentes soluções focadas na segurança e na privacidade dessas tecnologias. A seguir, são destacados os principais trabalhos encontrados e suas contribuições, divididos em três categorias: pesquisas sobre segurança e privacidade, soluções de segurança e soluções de privacidade.

2.6.1 Pesquisas sobre Segurança e Privacidade

Foram estudadas diversas pesquisas abordando os desafios e soluções relacionados à segurança de *blockchain* e contratos inteligentes e à privacidade das aplicações baseadas nessas tecnologias. No recente trabalho de Chaliasos et al. [102], os autores conduziram uma importante pesquisa envolvendo a avaliação da eficácia de cinco ferramentas automatizadas de análise de vulnerabilidades na detecção dos problemas de segurança em contratos inteligentes relacionados a 127 ataques conhecidos, cujo impacto financeiro total foi de US$ 2,3 bilhões. Os resultados revelaram uma descoberta alarmante: as ferramentas analisadas só possuíam oráculos – padrões e regras para identificar vulnerabilidades – para detectar as falhas de segurança relacionadas a 32 ataques. Além disso, as vulnerabilidades exploradas em somente 11 ataques – 8% do total – teriam sido de fato detectadas pelas ferramentas, o que evitaria a perda de US$ 271 milhões – apenas 12% do total perdido nos ataques.

Em [4], além de desenvolver uma taxonomia de classificação para as ferramentas de segurança em contratos inteligentes, os autores criam um mapa de cobertura de vulnerabilidades e analisam a evolução das ferramentas. O estudo foi bastante amplo, incluindo a análise de 133 ferramentas cujas técnicas utilizadas variam bastante, abrangendo desde análise estática e execução simbólica até interceptação de transações, passando por estratégias como *fuzzing* e aprendizado de máquina. O artigo identifica os desafios futuros e tendências nessa área, onde se destaca a necessidade de as ferramentas reforçarem o enfoque na interceptação dinâmica de transações, considerada uma técnica eficiente para detecção de vulnerabilidades, porém negligenciada pela maioria das ferramentas.

No artigo [103], é conduzido um estudo de mapeamento sistemático para identificar os avanços na área de segurança e privacidade de contratos inteligentes e *blockchain*, avaliando também as principais lacunas nos campos de pesquisa sobre esses temas. O artigo elenca seis categorias para as ferramentas de segurança, definindo uma categoria específica para aquelas focadas em prover privacidade a contratos inteligentes. Como ideia de pesquisa futura na área de privacidade, os autores sugerem que uma abordagem interessante seria a combinação das técnicas de criptografia – como as já citadas *ZKP* e criptografia homomórfica – com enclaves seguros em *hardware*.

Bernabe et al. [7] apresentam uma revisão sistemática sobre abordagens para preservação da privacidade em *blockchain*. O artigo detalha os principais desafios existentes e analisa, de forma abrangente, as técnicas para mitigá-los. O estudo avalia mais de trinta artigos propondo estratégias para endereçar a privacidade em *blockchain*, englobando domínios como governo eletrônico, *IoT* e compartilhamento de registros médicos. Os autores destacam que as plataformas de *blockchain* atuais não atendem aos requisitos mínimos de privacidade, dificultando a conformidade com regulamentações de proteção de dados, como a GDPR [39]. O artigo enfatiza a necessidade de pesquisas futuras para equilibrar privacidade e conformidade regulatória, abordando simultaneamente questões de usabilidade, escalabilidade e interoperabilidade.

O estudo conduzido por Almashaqbeh e Solomon [28] fornece uma análise abrangente das soluções de privacidade para *blockchain* e contratos inteligentes, buscando organizar o conhecimento existente e sistematizar as abordagens adotadas. As autoras discutem os desafios existentes nesse contexto e descrevem as principais técnicas utilizadas para aprimoramento da privacidade em aplicações descentralizadas. A partir dessa base, propõem uma taxonomia que classifica as soluções de privacidade em três grupos, de acordo com o escopo da proteção oferecida. A primeira categoria, *"Private Payments"*, abrange soluções voltadas para a privacidade de transações de pagamento, focadas na ocultação dos valores transferidos e no anonimato das partes envolvidas. A segunda, *"Private Computation"*, inclui técnicas que possibilitam a execução de contratos inteligentes preservando a privacidade das entradas e saídas das operações, permitindo casos de uso mais complexos. Já a terceira categoria, *"Function Privacy"* refere-se a abordagens que protegem não apenas os dados manipulados, mas também a lógica computacional executada. A pesquisa analisa dez soluções propostas entre 2014 e 2020 e observa uma evolução progressiva das abordagens, com um movimento crescente na direção das categorias mais avançadas, que buscam oferecer privacidade em um espectro mais amplo de funcionalidades.

Por fim, Qi et al. [104] apresentam uma sistematização do conhecimento sobre soluções de privacidade para contratos inteligentes, dividindo-as em duas categorias: esquemas baseados em criptografia, que utilizam técnicas como *ZKPs*, *SMPC* e criptografia homo-

mórfica; e métodos baseados em *TEE*. Os autores destacam desafios como as limitações de expressividade e eficiência das abordagens baseadas em *ZKP* e a forte dependência das soluções baseadas em *TEE* no *Intel SGX* [97]. Além disso, o artigo explora direções futuras de pesquisa, incluindo o aprimoramento da eficiência dos esquemas baseados em *ZKP* por meio da delegação da geração de provas para servidores *TEE* e a redução da dependência de um único provedor de *TEE*, integrando implementações heterogêneas de *TEE* aos sistemas *blockchain*.

Este trabalho amplia as pesquisas existentes na área ao fornecer não apenas uma análise conceitual abrangente e atualizada das soluções de privacidade, mas também uma avaliação prática, integrando duas dessas soluções a uma aplicação *DeFi*. Essa abordagem permite investigar, de forma aplicada, os desafios, benefícios e limitações da adoção dessas técnicas no contexto de finanças descentralizadas e de *blockchain* em geral.

2.6.2 Soluções de Segurança

Esta seção discute diversos trabalhos que abordam as principais vulnerabilidades de *blockchain* e contratos inteligentes e os ataques relacionados, bem como as abordagens propostas para detecção e prevenção de vulnerabilidades nesse contexto. Embora o artigo de Luu et al. [105] não seja muito recente (a publicação é de 2016), a ferramenta *Oyente* apresentada no trabalho é citada em muitas pesquisas por ter sido uma das primeiras ferramentas de análise de vulnerabilidades propostas pela academia. A *Oyente* é baseada em execução simbólica, que consiste em uma técnica para explorar sistematicamente caminhos de execução potencialmente problemáticos em um programa, sem a necessidade de executar o código com valores concretos. A ferramenta analisa o *bytecode* de contratos inteligentes para detectar quatro tipos de vulnerabilidades, dentre elas, a vulnerabilidade de "reentrância" ("*Reentrancy*"). Bastante explorada até os dias atuais, essa vulnerabilidade ocorre quando um contrato permite que um invocador externo execute novamente uma função antes que a execução anterior seja concluída ou validada. Isso pode levar a comportamentos inesperados, onde o estado do contrato é alterado antes que as operações pendentes sejam finalizadas, permitindo que o invocador execute operações adicionais, potencialmente causando danos ou manipulando o estado do contrato de maneira não prevista pelo seu desenvolvedor. Um dos ataques mais conhecidos que exploraram a vulnerabilidade de reentrância foi o ataque ao contrato inteligente denominado *"The DAO" ("Decentralized Autonomous Organization")* no *Ethereum* em 2016. Nesse incidente, um *hacker* explorou a falha de reentrância no código do *"DAO"* para desviar mais de US$ 50 milhões em *Ether* para uma conta controlada por ele [85].

Outra abordagem bastante utilizada para detecção de vulnerabilidades em contratos inteligentes é a técnica de *fuzzing*. Utilizada por diversas ferramentas, o *fuzzing* envolve

a geração automatizada e aleatória de entradas para um contrato inteligente, visando encontrar falhas de segurança ao fornecer entradas inválidas. Em [86], é apresentada a ferramenta *SMARTIAN*, que implementa uma etapa prévia de análise estática do *bytecode* dos contratos no intuito de identificar sequências de transações críticas para gerar as sementes ("*seeds*") da fase de *fuzzing* propriamente dita. A ferramenta conta, ainda, com um mecanismo de *feedback* que monitora dinamicamente os fluxos de dados entre variáveis de estado durante o *fuzzing*, permitindo a atualização mais efetiva das sementes do *fuzzer* em tempo de execução. Essa abordagem de mesclar diferentes técnicas para melhorar a eficácia e o desempenho da ferramenta é um diferencial importante do *SMARTIAN*, que o destaca de outros *fuzzers*.

Existem também ferramentas de análise de vulnerabilidades de contratos inteligentes que utilizam aprendizado de máquina, como o *SVScanner*, proposto em [106]. Partindo do código-fonte de um contrato inteligente na linguagem *Solidity*, a ferramenta combina o modelo "*Bidirectional LSTM*" com um mecanismo de atenção e a arquitetura *"TextCNN"* para obter informações semânticas do código e identificar vulnerabilidades.

Entre as abordagens de análise dinâmica, destaca-se a ferramenta *TxSpector*, detalhada no artigo [107]. Essa ferramenta analisa o histórico de transações mantido na *blockchain*, reproduzindo-as para gerar rastros ("*traces*") em nível de *bytecode* da *Ethereum Virtual Machine (EVM)* a fim de identificar possíveis ataques e detectar as vulnerabilidades exploradas nos contratos inteligentes associados.

Apesar desta pesquisa não ser focada diretamente em ferramentas de análise de vulnerabilidades de contratos inteligentes e *blockchain*, o estudo destes trabalhos auxiliou na definição da direção desta dissertação, além de ampliar os conhecimentos sobre segurança e os desafios adicionais introduzidos por essas tecnologias.

2.6.3 Soluções de Privacidade

Uma vez que, neste estudo, se deseja analisar e comparar as soluções de privacidade no contexto de *blockchain* e contratos inteligentes, a pesquisa nessa área foi bastante abrangente, visando catalogar os artigos com propostas desse tipo de solução a partir de 2016. A seguir, são descritos alguns desses trabalhos.

Uma das primeiras soluções de privacidade de contratos inteligentes foi o *Hawk*, proposto em 2016 [6], que consiste em uma plataforma de contratos inteligentes que inclui uma nova linguagem, compilador e protocolos próprios para prover privacidade. No *Hawk*, os contratos incluem funções públicas e privadas, sendo que a computação das funções privadas é delegada a um terceiro confiável denominado "*manager*", que utiliza *ZKPs* para comprovar a corretude das operações realizadas. Um ponto de atenção dessa abordagem é a confiança de que o *"manager"* não irá divulgar as informações privadas das operações

executadas por ele. Para mitigar esse risco, foi proposta a solução *zkHawk* [108], que utiliza *SMPC* no lugar do "*manager*", ou seja, múltiplos participantes são encarregados de realizar as computações com dados privados de maneira segura, sem que eles precisem conhecer os dados.

No trabalho proposto por Steffen et al. [109], por sua vez, é apresentado o *zkay*, que constitui uma linguagem própria para contratos inteligentes prevendo anotações especiais para denotar dados privados e uma ferramenta de transformação desses contratos para a linguagem *Solidity*, usando *ZKPs* de forma a não divulgar os dados privados na *blockchain*.

No artigo que apresentou a solução *Zether* [110], os autores propõem uma abordagem diferente, ao prever um contrato especial, persistido na *blockchain Ethereum*, para executar as operações com os dados, utilizando criptografia homomórfica e *ZKP* para manter a privacidade das transações. No artigo original do *Zether*, prover anonimato para as partes envolvidas nas transações é colocado como um aprimoramento futuro. Essa funcionalidade foi incluída posteriormente, na solução *Anonymous Zether*, descrita em [111].

No artigo de Chen et al. [112], é proposta a solução *Ekiden*, que combina a tecnologia de *blockchain* com a utilização de ambientes de execução confiáveis – *TEEs*. Assim, a computação com os dados privados ocorre *off-chain*, por nós denominados "*compute nodes*" utilizando enclaves seguros como o *Intel SGX*, proporcionando um desempenho muito superior às abordagens em que a computação é feita *on-chain*. Porém, como já visto, essa abordagem de delegação está sujeita a riscos caso os enclaves seguros apresentem alguma vulnerabilidade explorável por um agente malicioso.

Observa-se, portanto, que existem múltiplas abordagens para prover privacidade a aplicações baseadas em *blockchain*, cada uma com diferentes níveis de complexidade, funcionalidade e eficácia. Diante dessa diversidade, torna-se fundamental compreender as vantagens e limitações dessas soluções antes de definir a abordagem mais adequada para cenários reais.

2.7 Considerações Finais do Capítulo

Este capítulo apresentou os fundamentos teóricos necessários para o desenvolvimento deste trabalho, abordando conceitos essenciais sobre sistemas distribuídos, segurança cibernética e privacidade. Além disso, foram detalhadas as tecnologias de *blockchain* e contratos inteligentes, com ênfase nos aspectos de segurança e privacidade inerentes a esse ecossistema.

A revisão da literatura evidenciou a evolução das pesquisas na área, destacando as principais técnicas e soluções desenvolvidas para mitigar os problemas de privacidade em *blockchain*. Apesar dos avanços, a análise revelou que a privacidade ainda representa

um desafio crítico para a adoção ampla dessas tecnologias, reforçando a relevância deste estudo. Nesse contexto, a avaliação e comparação de diversas soluções de privacidade realizadas neste trabalho visam auxiliar desenvolvedores a compreender as diferentes estratégias disponíveis, possibilitando uma escolha mais informada e alinhada às necessidades das aplicações descentralizadas.

No próximo capítulo, será conduzida uma análise conceitual das soluções de privacidade existentes, explorando sua arquitetura, técnicas utilizadas, impacto nos custos transacionais e desempenho.

Capítulo 3

Análise Conceitual de Soluções de Privacidade

É fundamental estabelecer uma base teórica robusta para compreender os desafios de privacidade em *blockchain*, a fim de avaliar as abordagens para superá-los. A parte teórica deste trabalho é apresentada neste capítulo. Após descrever a metodologia adotada nesta etapa da pesquisa, são elencados os principais requisitos de privacidade no contexto de aplicações descentralizadas. Em seguida, é realizada uma pesquisa sobre as soluções de privacidade disponíveis, analisando-as sob diferentes aspectos e comparando-as qualitativamente. Esta etapa fornece o suporte necessário para a fase prática do trabalho, que envolve a implementação e avaliação de duas soluções específicas.

Este capítulo está organizado da seguinte forma. A Seção 3.1 descreve a metodologia adotada nesta etapa do estudo. Na sequência, a Seção 3.2 discute os requisitos de privacidade para aplicações baseadas em *blockchain*, com ênfase em *DeFi*, considerando normativos sobre proteção de dados pessoais e *frameworks* de segurança e privacidade. As seções 3.3 e 3.4 abordam, respectivamente, a análise conceitual das soluções de privacidade para *blockchain* e a comparação dessas soluções sob diversos aspectos. Por último, a Seção 3.5 sintetiza as considerações finais do capítulo.

3.1 Metodologia - Etapa Teórica

Nesta seção, é descrita a metodologia adotada para a etapa teórica do trabalho, que abrange o levantamento de requisitos de privacidade, o estudo conceitual das soluções existentes e a comparação qualitativa dessas soluções.

3.1.1 Análise de Requisitos de Privacidade

O primeiro passo consiste na identificação dos requisitos de privacidade aplicáveis a *blockchain* e contratos inteligentes, com especial ênfase nas aplicações *DeFi*. Para tanto, são conduzidas as seguintes atividades:

- Revisão de normativos e *frameworks* sobre privacidade: Normas como a Lei Geral de Proteção de Dados Pessoais (LGPD) [40] e a *General Data Protection Regulation (GDPR)* [39] são analisadas para compreender como os princípios de proteção de dados pessoais podem ser atendidos no contexto de aplicações baseadas em *blockchain* e contratos inteligentes. Também são avaliados os aspectos de privacidade previstos em *frameworks* de instituições renomadas como o *NIST* [41, 43] e o *CIS* [44, 45].
- Identificação de requisitos essenciais de aplicações *DeFi*: Com base na revisão realizada, são elencados os requisitos essenciais para a privacidade de aplicações *DeFi*, como confidencialidade das transações, anonimato das partes e controle sobre os dados transacionados.

3.1.2 Análise Conceitual das Soluções de Privacidade

Esta etapa inclui a realização de uma pesquisa exploratória sobre as soluções de privacidade disponíveis para *blockchain* e contratos inteligentes, seguida da análise conceitual de cada solução. Para isso, segue-se o método abaixo, visando assegurar que as soluções analisadas sejam representativas no contexto estudado:

- Pesquisa em fontes acadêmicas e técnicas: São consultadas publicações científicas em bases como *IEEE Xplore* [113], *ACM Digital Library* [114] e *Google Scholar* [115], utilizando filtros e palavras-chave para encontrar artigos relacionados a ferramentas, soluções e abordagens para aprimorar a privacidade no contexto de *blockchain* e contratos inteligentes.
- Seleção de soluções relevantes: São selecionadas as soluções de privacidade com suporte a casos de uso mais complexos utilizando contratos inteligentes. Soluções com foco apenas na privacidade de transações com criptomoedas ficam fora do escopo da análise. Artigos que apresentam abordagens muito específicas para prover privacidade apenas a casos de uso limitados, sem elaborar uma arquitetura para a solução proposta, também não são selecionados.
- Análise conceitual das soluções: Com base nos artigos e documentações analisados, as soluções são categorizadas de acordo com as principais técnicas empregadas. A análise considera diversos aspectos de cada abordagem, incluindo sua arquitetura, os

mecanismos utilizados, o nível de privacidade alcançado, além do impacto previsto no desempenho e nos custos transacionais.

- Análise da documentação técnica de projetos relevantes: Nos casos de soluções que são implementadas como projetos disponíveis para uso pela comunidade, sua documentação é avaliada para complementar a análise e entender suas funcionalidades e particularidades de forma mais técnica e aprofundada.

3.1.3 Comparação Qualitativa das Soluções

Com base na análise conduzida na etapa anterior, realiza-se uma comparação qualitativa das soluções, considerando critérios técnicos pré-definidos, como o nível de privacidade alcançado, a necessidade de terceiros confiáveis, o impacto no desempenho e outros fatores relevantes para o contexto de aplicações descentralizadas.

3.2 Análise de Requisitos de Privacidade

Para avaliar de maneira criteriosa os requisitos de privacidade, é interessante iniciar a análise partindo dos normativos sobre proteção de dados, passando pelos *frameworks* de privacidade existentes e, por fim, contextualizando a análise para o cenário de aplicações descentralizadas, com foco especial nas aplicações *DeFi*.

3.2.1 Normativos sobre Privacidade

No Brasil, a principal norma sobre privacidade é a Lei nº 13.709, publicada em agosto de 2018, mais conhecida como Lei Geral de Proteção de Dados Pessoais (LGPD) [40]. No art. 3º da LGPD, consta:

—

Esta lei aplica-se a qualquer operação de tratamento realizada por pessoa natural ou por pessoa jurídica de direito público ou privado, independentemente do meio, do país de sua sede ou do país onde estejam localizados os dados, desde que:

I – a operação de tratamento seja realizada no território nacional;

II – a atividade de tratamento tenha por objetivo a oferta ou o fornecimento de bens ou serviços ou o tratamento de dados de indivíduos localizados no território nacional; ou

III – os dados pessoais objeto do tratamento tenham sido coletados no território nacional.

(...)

—

Já no art. 5º da LGPD, constam as definições de "dado pessoal" e "tratamento". O primeiro é definido como *"informação relacionada a pessoa natural identificada ou identificável"*, enquanto o segundo corresponde a *"toda operação realizada com dados pessoais"*, englobando ações como acesso, utilização, processamento, modificação, transferência e eliminação.

Avançando na avaliação da LGPD, o artigo 17 versa que *"Toda pessoa natural tem assegurada a titularidade de seus dados pessoais e garantidos os direitos fundamentais de liberdade, de intimidade e de privacidade, nos termos desta Lei"*. O artigo 18 também contém diretrizes importantes referentes ao direito do titular dos dados de requerer a anonimização, bloqueio ou eliminação dos seus dados, bem como revogar o consentimento para tratamento dado anteriormente.

Por fim, destaca-se o artigo 46, onde se lê:

Os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito.

(...)

§ 2º As medidas de que trata o caput deste artigo deverão ser observadas desde a fase de concepção do produto ou do serviço até a sua execução.

A LGPD também define requisitos importantes para os agentes de tratamento de dados, como: obtenção do consentimento explícito do titular dos dados pessoais, transparência no tratamento dos dados, divulgação de incidentes, entre outros.

Outros países também avançaram na definição de normativos com foco na privacidade. Na Europa, a *General Data Protection Regulation (GDPR)* [39] é a principal regulamentação sobre proteção de dados no continente. Publicada em 2016, a GDPR serviu de base para a elaboração da LGPD no Brasil, razão pela qual ambas compartilham princípios e diretrizes bastante similares. Nos Estados Unidos, a abordagem regulatória é fragmentada, com leis federais de privacidade geralmente voltadas para setores específicos. Exemplos incluem o *Privacy Act* de 1974 [116], que regula como as agências federais podem coletar e usar dados sobre cidadãos; o *Health Insurance Portability and Accountability Act (HIPAA)* [117], que protege informações de saúde; e o *Gramm-Leach-Bliley Act (GLBA)* [118], que estabelece regras de privacidade a serem seguidas pelas instituições financeiras americanas. Embora os EUA ainda não possuam uma lei federal abrangente sobre privacidade de dados, há iniciativas importantes nesse sentido, como a criação, em 2016, da agência *Federal Privacy Council (FPC)* [119], estabelecida para melhorar as

práticas de privacidade no governo americano. Um documento importante da *FPC* é o *Fair Information Practice Principles (FIPPs)* [120], que define os princípios de privacidade a serem seguidos pelas agências americanas. Ademais, diversos estados americanos têm promulgado suas próprias regulamentações sobre proteção de dados, destacando-se o *California Consumer Privacy Act (CCPA)* [121] e o *Virginia Consumer Data Protection Act (VCDPA)* [122] [123, 124].

3.2.2 *Frameworks* sobre Segurança e Privacidade

Conforme abordado na Seção 2.2, o *NIST* e o *CIS*, instituições norte-americanas de renome no cenário da segurança cibernética, elaboraram *frameworks* que visam orientar as instituições na melhoria da segurança de seus sistemas e no aprimoramento da privacidade dos usuários cujos dados são tratados por suas aplicações [41, 43, 42, 44, 45]. O governo brasileiro, por sua vez, desenvolveu seu próprio *framework* [46], fortemente baseado nos *CIS Controls* [44]. Nesta seção, são destacados os principais aspectos de privacidade previstos nesses *frameworks*.

No *NIST Privacy Framework* [43], uma função relevante é a *"Controle-P"*, que prevê o desenvolvimento de atividades para permitir que organizações ou indivíduos gerenciem dados com granularidade suficiente para gerenciar riscos de privacidade. Nesta função, existe uma categoria denominada "Processamento Desassociado", que contém subcategorias onde se destacam alguns pontos primordiais a respeito da preocupação com a privacidade no tratamento de dados:

- *CT.DP-P1*: Os dados são processados para limitar a observabilidade e a *"linkabilidade"* (ex: ações de dados ocorrem em dispositivos locais, criptografia que preserva a privacidade).
- *CT.DP-P2*: Os dados são processados para limitar a identificação de indivíduos (ex: técnicas de privacidade de desidentificação, tokenização).

No *framework CIS Controls* [44], complementado pelo *CIS Controls Privacy Companion Guide* [45], constam princípios de privacidade baseados nos já citados *Fair Information Practice Principles (FIPPs)* [120]. Abaixo, são destacados dois desses princípios:

- Limitação da Coleta: Dados pessoais devem ser coletados de forma limitada, legal e justa, com o conhecimento ou consentimento do titular, sempre que apropriado.
- Medidas de Segurança: Os dados pessoais devem ser protegidos contra riscos, como perda, acesso não autorizado, destruição, modificação ou divulgação, por meio de medidas de segurança adequadas.

O *framework* do *CIS* prevê diversos controles, onde se destacam os abaixo:

- 3.10: Preconiza que os dados sensíveis devem ser criptografados quando estiverem em trânsito, por exemplo, usando *TLS*.
- 3.11: Prevê que os dados sensíveis armazenados em servidores, aplicações e bancos de dados devem ser criptografados.
- 16.1: Sugere que seja estabelecido um processo de desenvolvimento seguro, abrangendo padrões de design e codificação seguros, testes de vulnerabilidade, segurança de códigos de terceiros e treinamento dos desenvolvedores para lidar com dados pessoais e integrar considerações de privacidade no design das aplicações.

3.2.3 Requisitos de Privacidade de Aplicações *DeFi*

Conforme abordado anteriormente, a *blockchain*, em sua concepção original, registra as informações das transações de forma transparente no *ledger*, o que traz à tona questões sobre a confidencialidade dos dados e a privacidade dos usuários das aplicações que utilizam essa tecnologia. Já existem inúmeras aplicações implementadas usando *blockchain* e contratos inteligentes, com uma variedade de casos de uso que vão desde os mais simples, como um jogo de cara ou coroa [125], até aplicações complexas, como uma plataforma de empréstimos descentralizados [126]. Portanto, é evidente que cada aplicação pode ter requisitos específicos de segurança e privacidade. Enquanto em determinadas aplicações as transações podem ser públicas, em outras é imprescindível garantir um alto nível de privacidade.

Um cenário onde se torna evidente a necessidade de preocupação com a privacidade é o das aplicações de finanças descentralizadas *(DeFi)*. Um exemplo marcante são as *DEXs*, que permitem a negociação direta de ativos digitais, como *tokens* e criptomoedas, entre usuários, utilizando contratos inteligentes e eliminando intermediários. A transparência inerente à *blockchain* expõe dados sensíveis, como endereços das contas envolvidas, valores transacionados e históricos de negociações, comprometendo a privacidade e revelando padrões financeiros. Além do impacto na privacidade dos usuários, essa exposição também aumenta a vulnerabilidade a ataques, como o *front-running* – estratégia em que um agente mal-intencionado monitora transações pendentes e antecipa suas próprias ordens para obter vantagens financeiras [127]. Sem a implementação de mecanismos de segurança robustos, ameaças como essa podem comprometer a integridade e a confiabilidade das aplicações *DeFi*.

Diante dos desafios descritos, e considerando os normativos e *frameworks* sobre privacidade discutidos nas seções 3.2.1 e 3.2.2, é possível mapear os principais requisitos de privacidade para aplicações *DeFi*.

Aplicabilidade da LGPD

Considerando os artigos 3º e 5º da LGPD [40], já mencionados, a interpretação de que aplicações *DeFi* realizam o tratamento de dados pessoais é bastante direta. Essas aplicações lidam com informações pessoais e financeiras dos usuários, além de processarem transações envolvendo *tokens* que, em última instância, são de propriedade de uma pessoa natural ou jurídica, conforme previsto nos artigos citados.

Trazendo os artigos 17 e 46 da LGPD ao cenário das aplicações *DeFi*, também é natural concluir que tais aplicações devem zelar pela privacidade dos dados pessoais processados por elas, adotando as medidas necessárias para a proteção dos dados desde sua concepção até a sua execução.

Por fim, com a consolidação de normativos e regulamentações de proteção de dados, como a LGPD e a GDPR, em diversos países, e considerando que as aplicações *DeFi* podem processar dados de pessoas de diferentes nacionalidades, a ausência de medidas eficazes para assegurar a privacidade dos usuários pode expor a aplicação a sérias implicações legais e reputacionais.

Requisitos de Privacidade Mapeados para *DeFi*

Os princípios e controles previstos nos normativos sobre proteção de dados e nos *frameworks* de segurança e privacidade devem ser considerados por quaisquer aplicações que lidem com dados pessoais. Nesta seção, os requisitos de privacidade para o contexto específico das aplicações *DeFi* são identificados e relacionados com os normativos e *frameworks* estudados.

- **R.DeFi-1** – *Privacy by Design*: Incorporar requisitos de privacidade desde a fase inicial do projeto, garantindo que a proteção de dados seja um atributo fundamental da arquitetura do sistema.
  - Referências: LGPD – art. 46 §2; GDPR – art. 25; *NIST Privacy Framework – CT.DM-P10*; *CIS Controls Privacy Companion Guide* – controles 16.1 e 16.10.
- **R.DeFi-2** – Consentimento: Garantir que os usuários forneçam consentimento explícito para o tratamento e o compartilhamento de seus dados. Isso inclui interfaces amigáveis, configurações de privacidade ajustáveis e informações claras sobre o uso dos dados.
  - Referências: LGPD – arts. 7-II e 8; GDPR – art. 6-1-a; *CIS Controls Privacy Companion Guide – FIPPs-1.*

- **R.DeFi-3** – Minimização da Coleta de Dados: Dados coletados devem ser limitados ao necessário para os fins específicos.

  – Referências: LGPD – arts. 6-III; GDPR – art. 5-1-*b,c,e* e art. 25; *CIS Controls Privacy Companion Guide – FIPPs*-1,2,3,4.

- **R.DeFi-4** – Proteção dos Dados Pessoais e Confidencialidade das Transações: Adotar medidas como criptografia, pseudonimização ou anonimização para proteção de dados pessoais. Implementar técnicas que garantam que os usuários não possam ser identificados diretamente a partir de suas transações. As informações a respeito de cada transação devem ser mantidas confidenciais e acessíveis apenas para as partes diretamente envolvidas na transação ou explicitamente autorizadas.

  – Referências: LGPD – arts. 6-VII,VIII e 46; GDPR – arts. 5-1-f, 24 e 32; *NIST Privacy Framework – CT.DP-P1,P2,P3* e *PR.DS-P1,P2,P5*; *CIS Controls Privacy Companion Guide – FIPPs-5* e controles 3.3, 3.10 e 3.11.

- **R.DeFi-5** – Controle por Parte do Usuário: Os usuários devem ser informados de maneira clara sobre quais dados estão sendo processados e com que finalidade. Além disso, devem ter ferramentas que lhes permitam exercer seus direitos de acesso, retificação ou exclusão dos dados.

  – Referências: LGPD – arts. 6-VI, 9 e 18; GDPR – art. 12 e arts. 15 a 17; *NIST Privacy Framework – CT.PO-P3*, *CT.DM-P1,P2,P3,P4*; *CIS Controls Privacy Companion Guide – FIPPs-7*.

- **R.DeFi-6** – Auditoria e Responsabilização: Estabelecer trilhas de auditoria e mecanismos de monitoramento para detectar usos indevidos de dados pessoais, além de planos de resposta a incidentes de privacidade. Caso ocorra uma falha, deve haver mecanismos para notificar rapidamente os usuários afetados.

  – Referências: LGPD – arts. 6-X, 42 e 48; GDPR – arts. 5-2, 24, 33, 34; CIS *Controls Privacy Companion Guide – FIPPs-8* e controles 8 e 17.

Atender simultaneamente a todos os requisitos mapeados é um desafio significativo. A natureza pública da *blockchain* torna os dados e transações visíveis por padrão, exigindo a adoção de medidas para aprimorar a privacidade como um primeiro passo. No entanto, uma aplicação que privilegie a privacidade pode limitar a possibilidade de auditoria. Soma-se a isso o desafio imposto pela característica imutável e descentralizada da *blockchain*, que dificulta a implementação de mecanismos eficazes para controle dos dados

pelos usuários, como a remoção ou alteração de informações, de forma compatível com os normativos de proteção de dados.

É importante ressaltar que, embora os requisitos elencados sejam focados na privacidade, a aplicação deve considerar também aspectos gerais de segurança. Por exemplo, é preciso garantir que as transações sejam realizadas de forma segura e não permitam fraudes ou manipulações indevidas. No cenário de *blockchain* e contratos inteligentes, isso envolve a proteção contra ataques como reentrância, gasto duplo, entre outros [128].

Para aplicações *DeFi*, a utilização de uma *blockchain* pública sem mecanismos adequados de privacidade é temerária, pois expõe os usuários a riscos significativos de violação de privacidade. Mesmo em *blockchains* permissionadas – por exemplo, que envolvam apenas instituições como cartórios ou bancos – a privacidade dos usuários deve ser garantida, evitando que dados pessoais de clientes de uma instituição sejam acessados por outras.

A transparência inerente à *blockchain* exige que as aplicações adotem soluções suplementares para garantir a privacidade ou utilizem plataformas de *blockchain* que já ofereçam recursos de privacidade. É nesse contexto que entram em cena as soluções de privacidade examinadas na etapa a seguir.

3.3 Análise Conceitual das Soluções de Privacidade

Nesta seção, são analisadas 20 soluções de privacidade para *blockchain* e contratos inteligentes, publicadas entre 2016 e 2024, com base em uma pesquisa abrangente de artigos científicos, conforme descrito na Seção 3.1.2. É relevante ressaltar que, neste estudo, o termo "soluções" é empregado de forma ampla, abarcando diversas abordagens para assegurar a privacidade no contexto em análise. Essas abordagens podem envolver conjuntos de bibliotecas, novas linguagens e compiladores ou até mesmo novas plataformas de *blockchain* e os protocolos associados.

3.3.1 Desafios e Limitações da Análise

Nesta etapa, a análise é realizada em um nível conceitual, sem a implementação ou a execução de testes práticos das soluções avaliadas. O estudo baseia-se inteiramente nos artigos que apresentam as soluções de privacidade e em pesquisas sobre o tema, como as mencionadas na Seção 2.6.1. Embora essa abordagem forneça uma visão geral e uma comparação teórica das soluções, ela apresenta certos desafios e limitações, conforme detalhado a seguir.

1. Complexidade das técnicas criptográficas: Muitas soluções dependem de técnicas criptográficas avançadas, como Provas de Conhecimento Zero *(ZKPs)* e criptografia

homomórfica. Compreender os princípios e implicações dessas técnicas exige certo conhecimento sobre esses conceitos. A diversidade nas abordagens criptográficas também dificulta a comparação direta das soluções de maneira uniforme.

2. Documentação incompleta: Diversos artigos carecem de documentação detalhada sobre as técnicas e os protocolos utilizados, dificultando a compreensão completa das metodologias propostas e de suas implicações práticas. Em determinados casos, a ausência de uma implementação impede a avaliação da aplicabilidade e do desempenho das soluções em cenários reais.

3. Diversidade nas implementações e métricas de avaliação: As métricas de avaliação utilizadas nos artigos estudados variam bastante, dificultando a realização de uma comparação mais precisa. Alguns artigos focam no desempenho teórico, enquanto outros priorizam a implementação prática. Quando exemplos de implementação são fornecidos, as especificações de hardware, software e ambiente também são distintas. Essa variação exige um certo grau de normalização e interpretação dos resultados para gerar comparações significativas.

4. Integração e suporte: A integração de uma solução de privacidade em aplicações existentes depende fortemente do suporte da comunidade e do ecossistema. Para avaliar a aplicabilidade das soluções, é necessário considerar fatores como a disponibilidade de um repositório público, ferramentas para desenvolvedores, engajamento da comunidade e suporte de plataformas *blockchain* relevantes. Diversas soluções não possuem um repositório de código mantido ativamente, e algumas nem foram implementadas em ambiente de teste. Essa diversidade no nível de maturidade das soluções analisadas também é um fator que torna a avaliação significativamente desafiadora.

3.3.2 Escopo das Soluções Estudadas

Desde que as preocupações acerca da privacidade no *Bitcoin* ganharam destaque e atenção da comunidade acadêmica, foram propostas novas soluções focadas em transações de pagamento com privacidade. Tais soluções incluem não apenas criptomoedas como o *Zerocash* [14] (agora denominado *Zcash* [15]) e *Monero* [16], já mencionadas no Capítulo 1, mas também plataformas como o *zkLedger* [129], um sistema de *DLT* para transações de pagamento com privacidade e suporte a auditoria.

No entanto, este estudo se concentra especificamente na análise das soluções de privacidade com suporte a contratos inteligentes, viabilizando a implementação de aplicações descentralizadas mais avançadas que vão além das transações de pagamento. Assim, so-

luções de privacidade que se restrinjam à funcionalidade de transações de pagamento não fazem parte do escopo analisado.

Considerando a diversidade das aplicações baseadas em *blockchain*, que abrangem múltiplos casos de uso com requisitos distintos de segurança e privacidade, a análise apresentada neste capítulo não busca avaliar o cumprimento dos requisitos de privacidade para aplicações específicas, tampouco verificar a conformidade das soluções com os normativos de proteção de dados pessoais. O foco recai sobre as próprias soluções de privacidade, explorando suas características gerais, pontos fortes e limitações. O enfoque no atendimento aos requisitos de privacidade específicos para o contexto de *DeFi* é dado no Capítulo 4, que detalha a implementação da aplicação *Miles2Coins* e sua integração prática com as soluções *Anonymous Zether* [111] e *Zeestar* [130], inicialmente estudadas de forma teórica nesta etapa.

Considerando que o *Ethereum*, a primeira *blockchain* a suportar contratos inteligentes, foi lançado em julho de 2015, esta pesquisa abrange o período de 2015 a agosto de 2024. Conforme a metodologia descrita na Seção 3.1, foram revisados mais de 100 artigos, incluindo mais de 50 trabalhos propondo soluções de privacidade, além de pesquisas sobre o tema. As soluções que atenderam aos seguintes critérios foram selecionadas para análise detalhada:

1. O artigo apresenta uma explicação detalhada da arquitetura e do protocolo da solução, em vez de apenas conceitos ou ideias de alto nível.
2. A solução é abrangente e aplicável a diversos cenários, não se limitando a transações de pagamento ou a casos de uso muito específicos, como compartilhamento de registros médicos eletrônicos, *IoT*, aprendizado federado *("federated learning")* ou votação eletrônica.
3. A solução se mostra compatível com a implementação em *blockchains* públicas.
4. O foco principal da solução é o aprimoramento da privacidade, e não apenas a melhoria de escalabilidade, desempenho ou outros recursos, com a privacidade em segundo plano.

Com base nesses aspectos, 20 soluções foram selecionadas para análise, cada uma empregando diferentes técnicas e estratégias para prover privacidade.

3.3.3 Categorização das Soluções de Privacidade

As soluções de privacidade geralmente empregam as estratégias descritas na Seção 2.5, individualmente ou em combinação. Neste trabalho, as soluções analisadas são categorizadas com base nas principais técnicas adotadas. As categorias são:

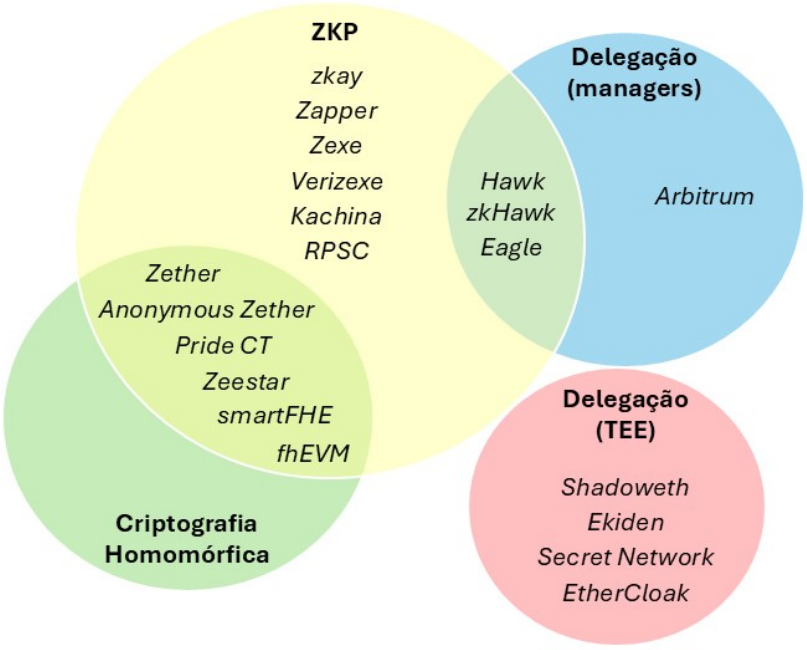

Figura 3.1: Diagrama de Soluções e Técnicas Empregadas

1. Delegação: Soluções que se baseiam majoritariamente na delegação da computação dos dados privados para terceiros confiáveis ("*managers*") ou ambientes de execução confiáveis *(TEEs)*.

2. *ZKP*: Soluções que utilizam, como elemento principal, esquemas de provas de conhecimento zero *(ZKPs)* como base para prover privacidade às transações.

3. Criptografia homomórfica e *ZKP*: Soluções que fazem uso das técnicas de criptografia homomórfica e *ZKP* em conjunto para oferecer privacidade.

O diagrama na Figura 3.1 ilustra a adoção dessas técnicas pelas soluções avaliadas. Observa-se que os esquemas de *ZKP* são amplamente utilizados, tanto de forma independente quanto em combinação com outras estratégias. Nota-se também que a delegação para *TEE* não é combinada com outras técnicas nos trabalhos estudados.

Embora as técnicas *SMPC*, "*mixing*" e "*ring signatures*" descritas na Seção 2.5.3 sejam utilizadas por algumas soluções para aprimorar a privacidade, elas não são comumente empregadas de forma isolada ou como abordagem principal. Portanto, este trabalho não define uma categoria específica para elas.

A seguir, as 20 soluções analisadas são agrupadas de acordo com as categorias mencionadas acima, e suas características, funcionalidades, pontos fortes e deficiências são descritos.

3.3.4 Soluções Baseadas em Delegação

As soluções que adotam a técnica de delegação, seja para terceiros confiáveis ("*managers*") ou para *TEE*, são detalhadas nesta seção.

Hawk. Proposto em 2016, o *Hawk* [6] é uma das primeiras soluções de privacidade para *blockchain* e contratos inteligentes, projetado como uma extensão do *Zerocash* [14] para oferecer programabilidade com privacidade. No *Hawk*, os desenvolvedores podem escrever contratos inteligentes sem a necessidade direta de implementar criptografia, já que o compilador *Hawk* gera um protocolo de interação entre as partes envolvidas e a *blockchain* utilizando técnicas criptográficas, como *ZKPs*. Esse compilador divide o contrato em uma parte pública e outra privada, onde a parte pública não contém dados sensíveis. No protocolo especificado, os usuários criptografam suas entradas utilizando a chave pública de uma entidade confiável, denominada "*manager*", e submetem esses criptogramas ao contrato. O "*manager*" é quem realiza a computação intensiva *off-chain*, o que inclui a geração das *ZKPs* em nome dos usuários, preservando a privacidade de suas entradas.

Para geração das *ZKPs*, o *Hawk* implementa a técnica *Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zkSNARK)*, que requer um "*trusted setup*" – processo de geração dos parâmetros matemáticos para a criação e verificação das *ZKPs*. Nesse processo, que precisa ser realizado para cada aplicação, são armazenados alguns parâmetros *on-chain*, incluindo a chave de verificação das *ZKPs*. Quanto ao desempenho da implementação apresentada no artigo, enquanto a computação feita *on-chain* é bastante leve, as operações criptográficas realizadas *off-chain* pelo "*manager*" são pesadas, levando-se quase 3 minutos para elaborar as *ZKPs* em uma aplicação de exemplo com 100 participantes, utilizando uma máquina com 4 *cores*.

A abordagem do *Hawk* consta na categoria de delegação devido à confiança depositada no "*manager*", uma vez que ele não deve divulgar os dados privados das transações. Em trabalhos posteriores, a figura do "*manager*" é substituída por um protocolo *SMPC*, aprimorando o nível de privacidade. É o caso das soluções *zkHawk* [108] e *Eagle* [131], descritas na Seção 3.3.5, uma vez que não se enquadram mais na categoria de delegação.

Arbitrum. Outro *framework* que utiliza delegação é o *Arbitrum* [132]. Focada em privacidade e escalabilidade, a solução permite que os contratos sejam executados *off-chain*, com apenas o *hash* dos seus estados sendo publicado na *blockchain* para verificação. No *Arbitrum*, cada contrato é representado como uma máquina virtual *(VM)* independente. Ao se criar uma *VM*, é definido um conjunto de "*managers*" responsáveis por avançar seu estado.

Assim como no caso do *Hawk*, a privacidade das transações no *Arbitrum* está associada à confiança nos "*managers*", responsáveis pela execução das funcionalidades das *VMs* e por resguardar os dados das transações, como as entradas dos usuários. Vale destacar que, embora os dados privados não fiquem públicos na *blockchain*, eles permanecem acessíveis aos "*managers*" e às demais partes envolvidas no contrato.

Shadoweth. A solução *Shadoweth* [133] baseia-se na estratégia de delegação para ambientes de execução confiáveis *(TEEs)*. Nessa abordagem, os contratos privados são mantidos *off-chain*, em um armazenamento distribuído baseado em *TEE* – denominado *TEE-DS* – e mantido pelos "*worker nodes*" da solução. Um contrato especial, chamado "*Bounty Contract*", é mantido na *blockchain* pública com a função de orquestrar a implantação, invocação e verificação de contratos privados. A invocação de um contrato privado é feita por meio do "*Bounty Contract*" com os argumentos da chamada criptografados com a chave pública do contrato privado. Essa chave e sua contraparte privada são geradas em um enclave seguro, garantindo que apenas o enclave possa decifrar as informações sensíveis. Esse mecanismo assegura a proteção de todo o processo de execução, incluindo o código do contrato, que permanece oculto até mesmo para os *worker nodes*, já que a computação ocorre exclusivamente dentro do *TEE*. O *Shadoweth* não requer alterações nas *blockchains* existentes e é independente em relação à plataforma *TEE* utilizada.

Ekiden. Em uma abordagem similar ao *Shadoweth*, no *Ekiden* [112] os contratos também são executados *off-chain* utilizando *TEEs*, garantindo que apenas os estados criptografados dos contratos e as "*attestations*" que comprovam a corretude da computação sejam armazenados na *blockchain*. O *Ekiden* se distingue do *Shadoweth* ao não requerer um contrato auxiliar mantido na *blockchain*.

Os autores afirmam que a solução é agnóstica quanto à plataforma *TEE* e à *blockchain*, embora a implementação apresentada no artigo tenha sido baseada no *Intel SGX* [97] e em uma *blockchain* personalizada estendida do *Tendermint* [134]. Nessa configuração, o *Ekiden* obteve desempenho de duas a três ordens de magnitude superior ao *Ethereum*. Apesar de seu alto desempenho, a solução não suporta a invocação de funções entre contratos, o que limita sua utilização em aplicações mais complexas.

Secret Network. O *Secret Network* [135] é um protocolo de código aberto com foco em privacidade baseada em *TEE*, onde as entradas, saídas e estados dos contratos são mantidos criptografados na *blockchain*. No protocolo especificado, a semente de consenso ("*consensus seed*") é gerada originalmente para o nó inicial da rede e distribuída de forma segura para os demais nós validadores. A partir dessa semente, são geradas as chaves que permitem a cifragem e decifragem das entradas de cada transação a ser computada no

ambiente de *TEE*. Ao contrário do *Shadoweth* e *Ekiden*, a *Secret Network* implementa sua própria *blockchain*, baseada no *Cosmos SDK* [136] e atualmente encontra-se em operação [137].

EtherCloak. *EtherCloak* [138] inova ao introduzir um sistema de privacidade hierarquizado, permitindo atribuir um nível de privacidade a cada conta de usuário ou contrato. Isso permite graus variados de proteção, de acordo com os requisitos da aplicação e do usuário. Os quatro níveis de privacidade são:

- 0 (Público): Nenhuma privacidade necessária;
- 1 (Armazenamento – apenas para contas de contrato): Objetos de dados específicos do contrato são protegidos;
- 2 (Estado): O estado da conta, incluindo seu saldo, é privado;
- 3 (Identidade): A conta deve permanecer anônima e não deve poder ser vinculada a nenhuma transação, garantindo resistência a técnicas de *"transaction linking"*.

O *EtherCloak* utiliza o *Intel SGX* [97] e o *Enclave-ready EVM* da *Microsoft* [139] para realizar computações sobre dados privados. Para mitigar riscos relacionados à confiabilidade dos *hosts* de *TEE*, a solução inclui uma funcionalidade de verificação do estado do enclave para garantir a integridade dos estados das contas dentro do *TEE*, bem como um mecanismo de recuperação em caso de falhas. Esse mecanismo detecta falhas nos *hosts* de *TEE* e permite que *hosts* de *backup* recuperem dados perdidos.

Discussão – Soluções Baseadas em Delegação

As soluções baseadas em delegação oferecem recursos significativos de privacidade e, geralmente, bom desempenho, mas também apresentam algumas desvantagens, principalmente devido à dependência da confiança em terceiros. Considerando o *Hawk*, diversos desafios surgem em sua abordagem:

1. É necessário confiar que o "*manager*" não vazará nenhuma informação sensível das transações;
2. Devido à técnica *zkSNARK* utilizada para as *ZKPs*, o processo computacionalmente custoso de "*trusted setup*" não utiliza parâmetros universais e, portanto, é necessário realizá-lo para cada aplicação implementada;
3. O desempenho da solução é bastante prejudicado devido ao alto custo computacional da geração das *ZKPs* pelo "*manager*";

4. O *Hawk*, uma vez que foi pensado de maneira a estender o *Zerocash*, segue o modelo *UTXO*, potencialmente dificultando a implementação de aplicações mais complexas;
5. O *framework* do *Hawk* não é diretamente compatível com *blockchains* existentes, como o *Ethereum*.

O *Arbitrum*, por sua vez, não utiliza *ZKPs* complexos ou outras primitivas criptográficas computacionalmente custosas, uma vez que a interação com os usuários e os contratos *(VMs)* é realizada *off-chain* e os verificadores precisam avaliar apenas o *hash* dos estados das *VMs on-chain*. Assim, a solução apresenta um *overhead* menor para usuários e alcança bom desempenho. Porém, uma vez que as informações sobre as transações são visíveis para todos os "*managers*", o nível de privacidade alcançado pode ser considerado inferior.

As soluções de privacidade baseadas na delegação para *TEE*, além de oferecerem maior privacidade que as *blockchains* tradicionais, permitem obter melhor desempenho e escalabilidade, por realizarem a computação majoritariamente *off-chain* e não envolverem técnicas criptográficas custosas como *ZKPs*. As quatro soluções identificadas dentro dessa categoria [133, 112, 135, 138] fazem uso do *Intel SGX*, embora seus autores aleguem que são agnósticas quanto ao *TEE* utilizado. O ponto de atenção dessa abordagem é o fato de se depositar plena confiança na tecnologia de *TEE* e no seu fabricante. Assim, a exploração de uma vulnerabilidade nesse ambiente pode comprometer a segurança dessas soluções, conforme já mencionado na Seção 2.5.2. O *Secret Network*, por exemplo, já teve sua semente de consenso exposta a vazamento devido a uma vulnerabilidade do *Intel SGX* [140]. Na ocasião, o time técnico da *Secret Network* afirmou ter trabalhado em conjunto com a *Intel* para corrigir a vulnerabilidade [141], mas caso tenha havido vazamento da semente, todas as transações do *Secret Network* ficaram expostas, tendo sua privacidade comprometida.

3.3.5 Soluções Baseadas em *ZKP*

Esta seção descreve as soluções que se baseiam predominantemente na técnica de *ZKP* para prover privacidade.

Zkay. Proposto originalmente em 2019, o *zkay* [109] consiste em uma linguagem de programação de contratos inteligentes que emprega tipos de dados especiais com anotações para marcar informações como privadas. A solução inclui um compilador que interpreta o código e o converte em um contrato em *Solidity* equivalente, preservando a privacidade conforme definido pelo programador. Esse processo envolve armazenar os valores privados criptografados sob a chave pública do proprietário e a utilização de *ZKPs* para garantir

que as modificações de estado do contrato sejam consistentes com as operações realizadas. Embora algumas operações, como a publicação do contrato e a declaração de variáveis, sejam realizadas *on-chain*, outras, como a transformação do contrato e a geração de *ZKPs*, ocorrem *off-chain*.

O *zkay* é compatível com o *Ethereum*, mas possui certas limitações, como a falta de suporte a *loops* com dados privados e operações de divisão, além do uso de funções de criptografia assimétrica consideradas inseguras. Outro ponto de atenção é o custo médio das operações *on-chain*, estimado em 1 milhão de *gas*, o que representa um valor considerável. O *overhead* no desempenho devido às operações criptográficas realizadas *off-chain* também é elevado: a compilação do contrato leva cerca de 5 minutos, enquanto as operações necessárias para efetivar cada transação levam aproximadamente 1 minuto.

O *zkay* v0.2 [142] foi proposto para resolver diversas limitações do *zkay* original e incluiu melhorias como: uso de funções de criptografia seguras, incluindo *RSA*, *ECDH* e *AES*, suporte a novos tipos e funcionalidades na linguagem, além de desempenho aprimorado. Porém, outra limitação do *zkay* que persiste na segunda versão é a impossibilidade de operar com "*foreign values*" – valores pertencentes a outras partes que não o chamador do contrato. Essa limitação é resolvida pelo *Zeestar* [130] que, por utilizar técnicas de criptografia homomórfica, será abordado na próxima seção.

Zapper. O *Zapper* [143] apresenta abordagem diferente das soluções anteriores, porém ainda baseada majoritariamente na técnica de *ZKP*. Nesse *framework*, o desenvolvedor programa os contratos utilizando um *frontend* em *Python* e em seguida o código é compilado para uma linguagem específica (*Zapper Assembly language – Zasm*), que oferece instruções para um processador virtual que executa as transações e operações criptográficas, incluindo a geração de *ZKPs*. Um diferencial importante do *Zapper* é que ele garante tanto a privacidade dos dados da transação quanto o anonimato das partes envolvidas. A plataforma é agnóstica ao *ledger* e ao mecanismo de consenso. Embora o desempenho do *Zapper* seja superior ao do *zkay*, ainda são necessários cerca de 22 segundos para geração de uma nova transação. Por fim, os contratos do *Zapper* possuem limitações importantes que podem dificultar implementações mais complexas: não há suporte a *loops*, *jumps* ou outras estruturas de controle e a chamada de funções entre contratos não é permitida.

Zexe. O *Zexe* [144] apresenta uma importante inovação ao prover um nível mais alto de privacidade, em que a própria computação não é revelada. Nessa solução, os usuários realizam as computações *off-chain* e produzem as transações, atestando sua corretude por meio de *ZKPs*. Assim, as transações ocultam todas as informações sobre as computações e podem ser validadas em tempo constante. O *Zexe* não se trata de um *framework* completo

de contratos inteligentes, mas sim de uma camada adicional que implementa um protocolo que define regras sobre como os registros podem ser operados. Uma vez que a solução adota o modelo *UTXO*, seu uso no contexto de contratos inteligentes mais complexos pode ser desafiador, embora o artigo demonstre a possibilidade de se utilizar a solução para prover privacidade em determinados casos de uso, como aplicações de gerenciamento de ativos digitais e *exchanges* descentralizadas.

VeriZexe. No *Zexe*, as transações têm tamanho pequeno e podem ser verificadas em tempo constante, mas o tempo para sua geração pelos usuários é superior a 50 segundos e pode aumentar de acordo com a complexidade da computação. Outra limitação é a necessidade de se executar um "*trusted setup*" para cada aplicação, devido ao esquema de *ZKP* adotado. A solução *VeriZexe* [145] traz uma inovação importante ao implementar, de maneira eficiente, um único "*setup*" universal e ainda aprimorar o desempenho do *Zexe*, reduzindo expressivamente o tempo de geração das transações, que varia entre 13 e 25 segundos, a depender dos parâmetros.

Kachina. O *Kachina* [146] é um protocolo de contratos inteligentes com preservação de privacidade por meio de *ZKPs*. O protocolo divide o estado do contrato em dois componentes: um estado público compartilhado, mantido *on-chain*, e um estado privado *off-chain*, gerenciado pelos usuários individualmente. Os usuários atualizam seus estados privados e geram *ZKPs* que asseguram a corretude tanto das atualizações do estado privado quanto das transições resultantes no estado público. O artigo introduz o conceito de "*state oracles*", que permitem consultas e atualizações das informações de estado no *ledger* sem expor detalhes privados. Como não é demonstrada a implementação da solução, não é possível avaliar seu desempenho.

ZkHawk. Na solução *zkHawk* [108], proposta cinco anos depois da apresentação do *Hawk*, o "*manager*" é substituído por um protocolo *SMPC* e são utilizadas outras técnicas para geração das *ZKPs* de forma mais eficiente. Porém, o *zkHawk* requer que todas as partes do contrato permaneçam *online* durante as interações para geração das *ZKPs* via *SMPC*. O artigo concentra-se na formalização e especificação do protocolo, mas não apresenta nenhuma implementação, o que dificulta a avaliação de seu desempenho e a compreensão de sua viabilidade prática.

Eagle. O *Eagle* [131] também apresenta uma abordagem que combina *SMPC* com *ZKP* para prover privacidade. Apoiando-se nos fundamentos dos já citados *Hawk* e *zkHawk*, o protocolo estabelecido nesse *framework* inova ao não requerer que os clientes permaneçam *online* durante toda a computação, que é executada por servidores via *SMPC*, sem

requerer a confiança em um "*manager*". Na arquitetura proposta, as mudanças de estado são mantidas *off-chain* por múltiplas rodadas de computação, permitindo assim a execução de contratos de forma contínua, mesmo que os clientes estejam *offline*. No entanto, assim como nos casos do *Kachina* e do *zkHawk*, o artigo sobre o *Eagle* concentra-se na definição e formalização do protocolo, sem abordar aspectos de sua implementação, o que impossibilita a avaliação de sua aplicabilidade e desempenho.

RPSC. A principal inovação do *RPSC* (*Regulatable Privacy-Preserving Smart Contracts*) [147] é o suporte à auditoria por uma entidade reguladora confiável, que pode verificar a legitimidade e, caso necessário, o conteúdo das transações. O *RPSC* é composto por algoritmos para os usuários e o regulador, bem como por um contrato inteligente que valida e persiste as transações na *blockchain*. Primeiramente, o usuário realiza uma operação com dados privados *off-chain* e gera uma transação publicamente verificável usando uma *ZKP*. Em seguida, os dados da transação são criptografados com uma chave simétrica, que, por sua vez, é protegida utilizando a chave pública do regulador. Isso permite que o regulador utilize sua chave privada para verificar a legitimidade da transação e acessar os dados, se necessário. A funcionalidade de auditabilidade, embora seja o principal diferencial do *RPSC*, também introduz sua principal limitação: um aumento significativo no *overhead* para o usuário devido à complexidade adicional das operações criptográficas. Por exemplo, o tempo de geração da *ZKP*, em alguns casos, é mais de duas vezes maior do que no *Hawk* [6] e sete vezes maior do que no zkay [109, 142].

Discussão – Soluções Baseadas em *ZKP*

Por envolverem operações criptográficas computacionalmente intensivas, as soluções baseadas em *ZKP* geralmente buscam realizar as operações *off-chain*, armazenando na *blockchain* apenas informações públicas ou criptografadas. Embora essas soluções ofereçam maior privacidade em comparação com as abordagens baseadas em delegação, elas também apresentam limitações importantes:

1. A necessidade comum de se efetuar um "*trusted setup*" para definição dos parâmetros criptográficos;
2. O alto *overhead* imposto ao usuário, que fica responsável pela geração de *ZKPs* e outras operações criptográficas pesadas;
3. Restrições de funcionalidades dos contratos devido a limitações das técnicas de criptografia utilizadas.

3.3.6 Soluções Baseadas em Criptografia Homomórfica e *ZKP*

A seguir, são detalhadas as soluções que combinam criptografia homomórfica e *ZKP* para aprimorar a privacidade.

Zether. Proposto em 2020, o *Zether* [110] inova ao aliar a técnica de *ZKP* com a criptografia homomórfica aditiva *ElGamal* [148], implementando um modelo "*account-based*" que permite a atualização dos saldos das contas envolvidas nas transações sem revelá-los. A solução introduz uma nova abordagem ao sistema de *ZKP Bulletproofs* [149], denominada Σ-*Bullets*, que torna mais eficiente a geração das provas de conhecimento zero e elimina a necessidade de um "*trusted setup*". Em sua implementação, o *Zether* consiste em um ou mais *tokens Zether* (*ZTH*) gerenciados por contratos inteligentes (*ZSC*) mantidos na *blockchain*. Os usuários podem converter *tokens* em *ZTH* para realizar transações confidenciais. Apesar de ser, em essência, um sistema de pagamentos confidenciais sobre a *blockchain Ethereum* por meio do *token ZTH*, o Zether também pode interoperar com outros contratos inteligentes, permitindo a implementação de várias aplicações que preservam a privacidade.

Um aspecto importante do *Zether* é sua abordagem para evitar o problema de *front-running* por meio do uso de *epochs*, onde o tempo é dividido em blocos configuráveis de x segundos. Durante uma *epoch*, as contas envolvidas em uma transação são bloqueadas, e as transações são processadas apenas no final de cada *epoch*, o que limita o *throughput* da solução. Além disso, os custos transacionais do *Zether* são significativamente altos, com uma única transação podendo ultrapassar 7 milhões de *gas*.

Anonymous Zether. Enquanto o *Zether* assegura a confidencialidade das transações e dos saldos em *ZTH* das contas, o anonimato das partes envolvidas não é garantido. Tal funcionalidade foi mencionada pelos autores do artigo original como uma possível extensão futura para a solução. Essa inovação é oferecida no *Anonymous Zether* [111]. A principal técnica utilizada pela solução é denominada pelos autores como "*Many-out-of-many-proofs*", pois consiste em uma família de extensões da técnica "*One-out-of-many proofs*" [150], similar às assinaturas em anel. Por meio dessa abordagem, a solução permite que, ao efetuar uma transação, o remetente utilize um conjunto de contas (denominado "*anonymity set*"), para esconder sua identidade e a do destinatário. Ao aplicar essa técnica sobre o *Zether*, a solução consegue prover o anonimato das partes envolvidas, assumindo que o "*anonymity set*" seja grande o suficiente. No entanto, esse método exige o bloqueio de todas as contas do "*anonymity set*" durante a *epoch* da transação. A geração de transações leva entre 1 a 6 segundos, demonstrando desempenho superior em comparação com as soluções avaliadas anteriormente. Embora o custo das transações cresça em escala

logarítmica com o tamanho do "*anonymity set*", ele ainda é elevado: uma transação de transferência com o "*anonymity set*" de apenas 2 contas requer cerca de 4,8 milhões de *gas*.

PriDe CT. *"Private Decentralized Confidential Transactions"* (*PriDe CT*) [151] é uma extensão das soluções *Zether* [110] e *Anonymous Zether* [111], permitindo o agrupamento de transações de um remetente para múltiplos destinatários. Isso é alcançado por meio de um processo simplificado de geração de provas e de um "*anonymity set*" reestruturado, onde o endereço do remetente é fixo e qualquer um dos demais membros do conjunto pode ser um destinatário real. Nesse esquema, elimina-se a necessidade de bloquear as contas dos destinatários durante uma *epoch*, permitindo maior concorrência e reduzindo a complexidade e o *overhead* associado às transações. Embora o anonimato do remetente seja sacrificado no *PriDe CT*, os autores argumentam que, no *Anonymous Zether*, a conta do remetente pode ser inferida simplesmente identificando quem invocou o contrato *Zether* para realizar a transação.

Em relação ao desempenho do *PriDe CT*, o artigo relata que ele é mais eficiente do que executar o *Anonymous Zether* de forma ingênua várias vezes, à medida que o número de destinatários em uma transação aumenta. No entanto, os autores não forneceram um repositório público para a implementação da solução.

No mesmo artigo que introduz o *PriDe CT* [151], os autores também propõem o *FUL-Zether*, onde *"FUL"* significa *"Forward Secrecy Until Last Update"* ("Sigilo Progressivo Até a Última Atualização", em tradução livre). Essa funcionalidade é habilitada por um mecanismo no qual o remetente atualiza as chaves do destinatário durante uma transação. Basicamente, o remetente calcula uma cifra de atualização que criptografa um valor aleatório, δ. A nova chave pública do destinatário é então gerada usando δ. O remetente também fornece uma prova de corretude para a atualização. Após receber esta atualização, o destinatário recalcula sua chave privada. Esse processo garante que transações passadas permaneçam seguras, mesmo que as chaves atuais sejam comprometidas.

Zeestar. Dos quatro autores do artigo sobre o *Zeestar* [130], três participaram diretamente dos projetos do *zkay* [109] e *zkay* v0.2 [142]. A solução mantém a abordagem do *zkay* ao permitir que desenvolvedores sem conhecimento avançado em criptografia adicionem restrições de privacidade aos contratos inteligentes por meio de anotações específicas no código. O compilador da solução garante que essas restrições sejam aplicadas corretamente, agora combinando *ZKPs* com criptografia homomórfica. O principal avanço trazido pelo *Zeestar* foi a superação de uma limitação do *zkay*: a incapacidade de operar com "*foreign values*", que inviabilizava a criação de variantes privadas para casos de uso

comuns no *Ethereum*, como carteiras privadas. Essa limitação foi resolvida pelo *Zeestar* por meio da inclusão de operações de criptografia homomórfica aditiva. Assim, mesmo sem conhecer os saldos de outras contas, é possível atualizá-los de forma segura. Apesar das melhorias na eficiência, o desempenho continua sendo um desafio relevante: no *Zeestar*, o tempo médio de processamento das transações *off-chain* é de quase um minuto e os custos *on-chain* permanecem relativamente altos, variando de algumas centenas de milhares a quase 3 milhões de *gas*, dependendo do tipo de operação.

SmartFHE. A solução *smartFHE* [152] se baseia em criptografia totalmente homomórfica *(FHE)* e reduz parcialmente o *overhead* computacional dos usuários ao transferir o processamento de dados e saldos privados para os mineradores. Por utilizar *FHE*, a expressividade da solução é ampliada, permitindo sua aplicação para prover privacidade em casos de uso mais complexos. O *smartFHE* consiste em um *framework* que estende o conjunto padrão de operações do *Ethereum* com novos tipos de transações e recursos criptográficos para permitir operações em contas privadas e dados dos usuários. A solução prevê um protocolo que define as operações a serem feitas pelos usuários, como a geração de *ZKPs* para as transações, e pelos mineradores, que ficam responsáveis por verificar as *ZKPs*, efetuar as computações homomórficas diretamente sobre os dados cifrados e atualizar a *blockchain*.

No *smartFHE*, é utilizada a biblioteca *Dalek* [153] para prover uma implementação eficiente do *Bulletproofs* [149] para a geração de *ZKPs*. Embora essa otimização permita reduzir o *overhead* do lado do usuário, a verificação das provas pelos mineradores é computacionalmente custosa, levando entre 2 e 15 segundos, a depender dos parâmetros criptográficos utilizados. Em outras soluções, essa verificação costuma ocorrer em apenas alguns milissegundos. Ainda devido à técnica de *ZKP* adotada, o espaço ocupado por uma transação no *SmartFHE* é considerável, podendo alcançar centenas de *KB* ou até passar de 1 *MB*, o que representa um desafio no contexto de *blockchain*.

fhEVM. A solução *fhEVM* [22] também utiliza criptografia totalmente homomórfica *(FHE)* para prover privacidade, incluindo novas operações por meio de contratos pré-compilados incorporados na Máquina Virtual *Ethereum* (*EVM*) original. Isso permite que os desenvolvedores trabalhem com dados criptografados utilizando a sintaxe tradicional do *Solidity*, sem a necessidade de conhecimento profundo em criptografia. No *fhEVM*, as computações permanecem públicas, enquanto apenas os dados de entrada e saída das transações, assim como os saldos, são criptografados. A criptografia baseia-se em uma chave pública global, sendo que a chave privada associada é distribuída de forma segura entre os validadores da rede, garantindo que nenhum validador individual tenha acesso

completo. Quando um usuário ou contrato autorizado precisa acessar um valor específico (por exemplo, seu saldo), é utilizado um mecanismo de descriptografia por limiar *("threshold decryption")*, exigindo a colaboração de um grupo de validadores para decifrar as informações. Embora o artigo forneça tabelas que mostram a velocidade das operações criptográficas e o tamanho dos textos cifrados de acordo com os tipos de dados, ele não apresenta informações suficientes sobre o desempenho de transações completas. Essa omissão dificulta uma avaliação abrangente do desempenho da solução.

Discussão – Soluções Baseadas em Criptografia Homomórfica e *ZKP*

Assim como as soluções puramente baseadas em *ZKP*, as abordagens que combinam essa técnica com criptografia homomórfica não dependem da confiança em terceiros, garantindo privacidade sem a necessidade de um intermediário confiável. Essas soluções também oferecem maior eficiência, pois permitem operações diretamente sobre saldos privados. Além disso, esquemas que utilizam criptografia homomórfica apresentam ganhos potenciais em expressividade, podendo ser utilizados em aplicações que exigem não apenas a ocultação de valores, mas também operações aritméticas mais complexas sobre informações protegidas.

Apesar dessas vantagens, essas soluções ainda enfrentam certas limitações funcionais e desafios significativos de desempenho. A necessidade de gerar e verificar *ZKPs*, aliada à complexidade computacional das operações homomórficas, resulta em maior *overhead* e custos transacionais elevados, o que pode dificultar sua adoção em cenários com alta demanda por escalabilidade.

3.4 Comparação Qualitativa das Soluções

A partir da análise conceitual das soluções, é possível concluir que não há, atualmente, uma solução de privacidade do tipo "bala de prata", que atenda a todas as aplicações baseadas em *blockchain*. Para cada tipo de aplicação que se pretende implementar, podem ser feitas diferentes avaliações, considerando o nível mínimo de privacidade desejado, os requisitos de desempenho e outras funcionalidades.

3.4.1 Critérios de Comparação

A comparação qualitativa realizada neste trabalho abrange não apenas as funcionalidades de cada solução, mas também aspectos de implementação, como a disponibilidade de repositório público. Neste sentido, os seguintes critérios são considerados:

Recursos de Privacidade e Modelo Adotado:

1. Nível de privacidade: Enquanto algumas soluções proveem privacidade apenas para as entradas e saídas das transações [6, 132, 109, 142, 108, 131], outras proveem anonimato às partes envolvidas [143, 147, 111, 151] e/ou ocultam os saldos das contas [110, 111, 151, 130, 152, 22]. Existem, ainda, soluções em que a própria computação realizada se torna confidencial [133, 112, 135, 144, 145, 146] e, por fim, uma das soluções analisadas [138] permite níveis personalizáveis de privacidade. Os desenvolvedores devem analisar cuidadosamente os requisitos de privacidade de sua aplicação para determinar a abordagem mais adequada. Em geral, quanto maior o nível de privacidade, melhor, mas há casos em que a computação não deve ser ocultada para permitir maior transparência quanto às operações realizadas, ainda que os dados fiquem ocultos.

2. Necessidade de "*trusted setup*": A configuração inicial dos parâmetros de criptografia para a operação de uma solução de privacidade é, em muitos casos, necessária. O aspecto crítico dessa configuração é que ela precisa ser realizada de forma segura, envolvendo apenas entes confiáveis, uma vez que um processo inseguro de "*trusted setup*" pode comprometer a privacidade. Há soluções que não requerem "*trusted setup*" [133, 112, 135, 138, 110, 111, 151, 152, 22], enquanto em outras, a configuração é universal e precisa ser realizada apenas na configuração inicial da solução como um todo [143, 145, 108, 131]. Finalmente, há casos em que é necessário um "*trusted setup*" para cada aplicação a ser implementada sobre a solução de privacidade [6, 132, 109, 142, 144, 147, 130].

3. Confiança em terceiros: Existem soluções [6, 132, 133, 112, 135, 138] que requerem a confiança em entes confiáveis como "*managers*" ou plataformas *TEE* que, se forem maliciosas ou tiverem sua segurança comprometida, podem comprometer a privacidade dos usuários.

4. Modelo adotado: Algumas soluções [6, 143, 144, 145, 108, 131] adotam o modelo *UTXO*, como utilizado no *Bitcoin*, onde os saldos são representados por saídas de transações não gastas e novas transações consomem e geram novos *UTXOs*. Já no modelo "*Account-Based*", adotado no *Ethereum*, o saldo é associado diretamente a cada conta e as transações resultam em alterações imediatas nos saldos do remetente e do destinatário. Assim, para aplicações mais complexas onde a manutenção e atualização eficiente do estado são essenciais, soluções de privacidade que suportem o modelo "*Account-Based*" [132, 133, 112, 135, 138, 109, 142, 146, 147, 110, 111, 151, 130, 152, 22] podem ser mais adequadas.

Desempenho e aplicabilidade:

1. Compatibilidade com *Ethereum*: Como o *Ethereum* é atualmente a plataforma de contratos inteligentes mais expressiva e amplamente utilizada, é desejável que uma solução de privacidade possa ser implementada diretamente sobre essa *blockchain*, garantindo maior compatibilidade e adoção. No entanto, esse requisito não é absoluto para todos os casos de uso, dado que muitas aplicações robustas operam sobre outras *blockchains*, sejam elas públicas ou permissionadas.

2. *Overhead* para o cliente: Em um cenário ideal, é interessante que a privacidade seja implementada sem exigir que o usuário realize computações intensivas, permitindo o uso em dispositivos móveis ou máquinas com *hardware* limitado.

3. Desempenho: É sabido que ao incorporar privacidade às aplicações haverá certo impacto no seu desempenho, devido à sobrecarga associada às operações criptográficas necessárias. Há sempre um balanço entre segurança, desempenho e usabilidade que deve ser avaliado em cada contexto.

4. Custo das transações: O custo transacional é um aspecto importante na escolha da solução de privacidade, especialmente no caso do *Ethereum*, onde operações computacionalmente intensivas podem demandar um alto consumo de *gas*. Aplicações que exigem múltiplas operações para concretizar uma transação ou que envolvem a participação de diversos usuários podem se tornar inviáveis se os custos transacionais forem excessivamente elevados.

5. Existência de repositório: Para viabilizar a avaliação prática da solução, é necessário que seja possível implementá-la e, para tanto, seu código fonte, documentação e outros artefatos devem estar disponíveis em um repositório acessível.

3.4.2 Resultados da Comparação

Nas Tabelas 3.1 e 3.2, as soluções são agrupadas em categorias de acordo com a técnica adotada, e os aspectos descritos anteriormente são avaliados para cada caso, com o objetivo de fornecer uma comparação concisa e objetiva entre as diferentes soluções.

Por meio das tabelas, é possível observar que as soluções baseadas em delegação, com exceção do *Hawk*, permitem alcançar privacidade sem resultar em *overhead* significativo para o usuário, possibilitando um desempenho superior. No entanto, como descrito anteriormente, essas soluções dependem da confiança em entidades específicas ("*managers*" ou *TEE*). Caso essas entidades ajam de forma maliciosa ou tenham sua segurança comprometida, a efetividade da solução é prejudicada.

Por outro lado, soluções baseadas em *ZKP* eliminam a necessidade de confiança em terceiros, mas impõem maior *overhead* ao usuário, prejudicando significativamente o desempenho. Já as soluções que combinam técnicas de *ZKP* com criptografia homomórfica tendem a oferecer um equilíbrio melhor, fornecendo privacidade sem depender de terceiros confiáveis ou de *"trusted setup"* – com exceção do *Zeestar*. Essas soluções, de maneira geral, apresentam menor impacto no desempenho quando comparadas às baseadas exclusivamente em *ZKP*, mas enfrentam como principal obstáculo os altos custos transacionais, especialmente em *blockchains* como o *Ethereum*.

Por fim, ao optar pela adoção de uma solução, é fundamental avaliar sua viabilidade prática, considerando a compatibilidade com *blockchains* relevantes, como o *Ethereum*, e a disponibilidade de um repositório de código público. A Tabela 3.2 destaca a tendência das abordagens baseadas em criptografia homomórfica e *ZKP* de oferecerem suporte ao *Ethereum*, mas também revela que muitas soluções não possuem repositório ativo, dificultando sua implementação.

Tabela 3.1: Soluções de Privacidade – Recursos de Privacidade e Modelo Adotado

Categoria	Solução	Nível de Priv.[1]	Trusted Setup	Confiança em 3os	Modelo
Delegação	Hawk [6]	Transação	Por aplicação	Sim (manager)	UTXO
	Arbitrum [132]	Transação	Por aplicação	Sim (managers)	Account-Based
	Shadoweth [133]	Trans+Comp	Não	Sim (TEE)	Account-Based
	Ekiden [112]	Trans+Comp	Não	Sim (TEE)	Account-Based
	SecretNetwork[135]	Trans+Comp	Não	Sim (TEE)	Account-Based
	EtherCloak [138]	Configurável	Não	Sim (TEE)	Account-Based
ZKP	zkay [109, 142]	Transação	Por aplicação	Não	Account-Based
	Zapper [143]	Trans+Anon	Universal	Não	UTXO
	Zexe [144]	Trans+Comp	Por aplicação	Não	UTXO
	VeriZexe [145]	Trans+Comp	Universal	Não	UTXO
	Kachina [146]	Trans+Comp	Depende do ZKP	Não	Account-Based
	zkHawk [108]	Transação	Universal	Não	UTXO
	Eagle [131]	Transação	Universal	Não	UTXO
	RPSC [147]	Trans+Anon	Por aplicação	Não	Account-Based
HE+ZKP	Zether [110]	Trans+Saldos	Não	Não	Account-Based
	AnonZether [111]	Tr+Sal+Anon	Não	Não	Account-Based
	PriDe CT [151]	Tr+Sal+Anon(Dst)	Não	Não	Account-Based
	Zeestar [130]	Trans+Saldos	Por aplicação	Não	Account-Based
	smartFHE [152]	Trans+Saldos	Não	Não	Account-Based
	fhEVM [22]	Trans+Saldos	Não	Não	Account-Based

[1] Níveis de privacidade:
Transação: dados das transações são confidenciais;
Trans+Comp: transações e computação são confidenciais;
Trans+Anon: transações confidenciais e anonimato das partes envolvidas;
Trans+Saldos: transações e saldos são confidenciais;
Tr+Sal+Anon: confidencialidade das transações e saldos, além do anonimato das partes envolvidas;
Tr+Bal+Anon(Dst): confidencialidade das transações e saldos, além do anonimato do destinatário;
Configurável: usuários e contratos podem possuir níveis configuráveis de privacidade.

Tabela 3.2: Soluções de Privacidade – Desempenho e Aplicabilidade

Categoria	Solução	Comp. c/ Eth	Overhead p/ usuário[1]	Desemp.[2]	Custo em gas[3]	Repositório[4]
Delegação	Hawk [6]	Não	Alto	Baixo	-	Não
	Arbitrum [132]	2ª chain	Baixo	Alto	-	Não-oficial [154]
	Shadoweth [133]	Sim	Baixo	N/A	N/A	Não
	Ekiden [112]	2ª chain	Baixo	Alto	-	Não-oficial [155]
	SecretNetwork[135]	Não	Baixo	N/A	-	Não-oficial [156]
	EtherCloak [138]	Não	Baixo	Alto	-	Não
ZKP	zkay [109, 142]	Sim	Alto	Baixo	$\sim$ 1 milhão	Sim [157]
	Zapper [143]	Não	Alto	Baixo	-	Sim [158]
	Zexe [144]	Não	Alto	Baixo	-	Sim [159]
	VeriZexe [145]	Não	Alto	Moderado	-	Sim [160]
	Kachina [146]	Não	N/A	N/A	-	Não
	zkHawk [108]	Não	Alto	N/A	-	Não
	Eagle [131]	Não	Alto	N/A	-	Não
	RPSC [147]	Sim	Alto	Baixo	$>$ 279 mil	Não
HE+ZKP	Zether [110]	Sim	Moderado	Moderado	$\sim 7,2$ milhões	Não
	AnonZether [111]	Sim	Moderado	Moderado	$>$ 4,8 milhões	Sim [161]
	PriDe CT [151]	Sim	Moderado	Moderado	$>$ 3,8 milhões	Não
	Zeestar [130]	Sim	Alto	Baixo	$>$ 339 mil	Sim [162]
	smartFHE [152]	Não	Moderado	Moderado	-	Não-oficial [163]
	fhEVM [22]	Não	N/A	N/A	-	Sim [164]

Notas:
[1] *Overhead* está relacionado ao tempo médio necessário para as computações a serem feitas pelo usuário: (1) Alto: $t > 10s$; (2) Moderado: $1s \leq t \leq 10s$; (3) Baixo: $t < 1s$.
[2] Desempenho está relacionado ao tempo total de cada transação, em média: (1) Alto: $t < 1s$; (2) Moderado: $1s \leq t \leq 20s$; (3) Baixo: $t > 20s$.
[3] Custo em *gas*: valor médio aproximado de uma única transação, conforme reportado pelos autores.
[4] Repositórios não-oficiais: parecem relacionados à solução, mas não foram mencionados pelos autores.

3.5 Considerações Finais do Capítulo

Neste capítulo, foram identificados os requisitos essenciais de privacidade para aplicações *DeFi*, tomando como base normativos e *frameworks* sobre proteção de dados e segurança. Em seguida, foi conduzida uma análise conceitual e comparativa das soluções de privacidade para *blockchain*, avaliando diferentes critérios como o nível de privacidade oferecido, a dependência da confiança em terceiros, compatibilidade com o *Ethereum*, desempenho e custo transacional.

A comparação das soluções demonstrou que não há uma abordagem única que atenda a todas as necessidades, sendo necessário equilibrar privacidade, eficiência e compatibilidade técnica. No próximo capítulo, essa investigação será aprofundada por meio da implementação prática de duas soluções, permitindo uma avaliação empírica de suas funcionalidades, impactos e desafios.

Capítulo 4

Análise Prática e Panorama Atual da Privacidade em Blockchain

Neste capítulo, são explorados os principais desafios práticos para aprimorar a privacidade em aplicações baseadas em *blockchain*, com foco no contexto de *DeFi*. Para isso, é conduzido um estudo de caso envolvendo a implementação da aplicação *Miles2Coins*, a identificação de seus requisitos de privacidade e usabilidade e sua integração com duas soluções de privacidade analisadas no Capítulo 3: *Anonymous Zether* e *Zeestar*. Os resultados são examinados para avaliar o nível de privacidade alcançado, o impacto no desempenho e os custos transacionais associados. Além disso, os desafios encontrados durante esse processo são identificados e analisados quanto à sua aplicabilidade a outros contextos de *DeFi* e *blockchain*. O capítulo se encerra com uma avaliação do panorama atual da privacidade em aplicações descentralizadas.

O restante deste capítulo está estruturado nas seguintes seções. A Seção 4.1 apresenta a metodologia adotada na etapa prática deste trabalho. Em seguida, a Seção 4.2 descreve as funcionalidades e os requisitos de privacidade da aplicação *Miles2Coins*. A Seção 4.3 detalha o processo de seleção das soluções de privacidade, enquanto a Seção 4.4 discute os desafios previstos na integração e introduz uma premissa fundamental desta etapa. As Seções 4.5 e 4.6 abordam, respectivamente, a integração com *Anonymous Zether* e *Zeestar*. A Seção 4.7 analisa os resultados dos testes, seguida pela Seção 4.8, que avalia os desafios gerais para aprimorar a privacidade em *blockchain*. Na sequência, a Seção 4.9 examina o panorama atual da privacidade nesse contexto. Por fim, a Seção 4.10 traz as considerações finais do capítulo.

4.1 Metodologia – Etapa Prática

A metodologia utilizada nesta parte do trabalho, que envolve o desenvolvimento da aplicação *Miles2Coins* e sua integração com as soluções de privacidade, segue as seguintes etapas:

4.1.1 Implementação da Aplicação *Miles2Coins*

A primeira etapa consiste no desenvolvimento da aplicação *Miles2Coins*, uma plataforma para compra e venda de milhas aéreas tokenizadas por meio de transações *DvP*, concebida como um caso de uso simplificado de *DeFi*.

Justificativa da Abordagem. Ao invés de analisar uma aplicação *DeFi* consolidada, como uma *exchange* descentralizada (por exemplo, *Uniswap* [165] ou *Curve* [166]), optou-se pela implementação da *Miles2Coins*, que oferece um ambiente mais simples e controlado. Isso possibilita uma análise focada nos requisitos de privacidade, sem sobrecarga excessiva de engenharia. Caso uma plataforma *DeFi* amplamente utilizada fosse empregada, a avaliação da privacidade se tornaria mais complexa devido à presença de múltiplos contratos inteligentes, componentes de *front-end* e lógicas especializadas. Além disso, aplicações *DeFi* frequentemente lidam com informações pessoais e financeiras, tornando os desafios de privacidade observados na *Miles2Coins* representativos de outros casos do ecossistema *DeFi*. Assim, os resultados obtidos podem ser extrapolados para um espectro mais amplo de aplicações descentralizadas.

A decisão de utilizar soluções de privacidade já existentes, em vez de desenvolver um novo sistema utilizando *ZKPs*, criptografia homomórfica ou *TEEs*, visa manter um design modular e, idealmente, facilitar a replicação da solução para outros contextos. Além disso, o uso de soluções consolidadas é uma boa prática de segurança, uma vez que tais ferramentas já passaram por testes mais extensivos, reduzindo o risco de vulnerabilidades decorrentes de implementações personalizadas. Por fim, essa abordagem também reflete um cenário realista, no qual um desenvolvedor busca atender a requisitos de privacidade em uma aplicação *DeFi* sem a necessidade de conhecimentos avançados em criptografia.

Esta etapa inclui as seguintes atividades:

1. Definição das funcionalidades principais da aplicação.
2. Implementação dos contratos inteligentes da aplicação e dos *tokens* associados, bem como dos *scripts* para interação com os contratos.
3. Identificação dos requisitos de privacidade e usabilidade.

4.1.2 Seleção de Soluções de Privacidade

Com base na análise teórica do Capítulo 3 e nos requisitos mapeados no passo anterior, deve-se selecionar as soluções para integração. Esse processo considera critérios objetivos, priorizando soluções que atendam aos requisitos de privacidade e, idealmente, não comprometam os aspectos de usabilidade da aplicação.

4.1.3 Integração das Soluções de Privacidade

O objetivo desta etapa é integrar, separadamente, cada solução de privacidade escolhida à aplicação *Miles2Coins*, avaliando de forma prática suas funcionalidades, limitações e os desafios inerentes à integração.

Principais atividades:

1. Implementação dos fluxos de transações na aplicação original, sem privacidade adicional.
2. Análise detalhada da arquitetura e dos aspectos práticos de cada solução.
3. Configuração de um ambiente experimental para cada integração.
4. Integração das soluções à aplicação *Miles2Coins*, incluindo a implementação dos fluxos de transações com a privacidade aprimorada por cada solução.

4.1.4 Avaliação dos Resultados

Para cada solução avaliada, os seguintes aspectos são analisados:

1. Custos transacionais: Medição dos custos referentes aos fluxos de transações antes e após a introdução de cada solução de privacidade.
2. Desempenho: Avaliação do tempo de processamento das transações e do uso de recursos computacionais (RAM e CPU) no ambiente de teste.
3. Nível de privacidade: Análise da privacidade da aplicação final, após cada integração, identificando quais dados foram protegidos e quais permaneceram públicos.
4. Atendimento aos requisitos: Verificação do atendimento aos requisitos previamente mapeados após a implementação das soluções de privacidade.

4.1.5 Identificação dos Desafios e Análise do Panorama Atual

Com base nos resultados obtidos, identificam-se os principais desafios para a privacidade em *blockchain*, permitindo uma reflexão sobre a viabilidade de garantir a privacidade

adequada em aplicações baseadas em *blockchains* públicas. Por fim, traça-se um panorama do estado atual da privacidade em *blockchain*, destacando avanços recentes e desafios remanescentes.

4.2 Aplicação *Miles2Coins*

A aplicação *Miles2Coins* consiste em uma aplicação destinada à compra e venda de milhas aéreas, representadas por meio do *token MilesToken*, utilizando uma *stablecoin* fictícia vinculada ao Real brasileiro, denominada *Surreal*. A arquitetura da *Miles2Coins* inclui um contrato inteligente desenvolvido em *Solidity*, operando em uma rede de teste local do *Ethereum*, e um conjunto de *scripts* implementados em *JavaScript*, que atuam como cliente, facilitando a interação com o contrato na *blockchain*. Para garantir a segurança, a aplicação deverá suportar transações *Delivery vs Payment (DvP)*, assegurando que a negociação seja atômica, ou seja, que ambas as partes concluam suas respectivas etapas para que a operação seja efetivada com sucesso.

4.2.1 Funcionalidades

A aplicação *Miles2Coins* oferece as seguintes funcionalidades principais:

1. Registro de Ofertas (função *placeOffer*): Usuários interessados em negociar milhas podem registrar ofertas, que podem ser:
    - Ofertas de compra: O usuário especifica a quantidade de *MilesTokens* que deseja adquirir e a quantidade de *SurrealTokens* que está disposto a oferecer. Ao criar uma oferta de compra, o contrato *Miles2Coins* é autorizado a gerenciar o valor correspondente em *SurrealTokens*, caso a oferta seja aceita no futuro.
    - Ofertas de venda: O usuário indica a quantidade de *MilesTokens* que deseja vender e o preço por milha em *SurrealTokens*. A criação de uma oferta de venda autoriza o contrato *Miles2Coins* a gerenciar os *MilesTokens* do vendedor, caso a oferta seja aceita posteriormente.
2. Listagem de Ofertas (função *listOffers*): Permite que o usuário consulte as ofertas ativas para encontrar a que mais lhe convém.
3. Aceite de Oferta (função *acceptOffer*): Por meio desta função, o usuário pode aceitar uma oferta específica, desencadeando as seguintes ações pelo contrato *Miles2Coins*:
    - Verificação do saldo de *MilesToken* do vendedor e do saldo de *SurrealTokens* do comprador;

- Caso os saldos sejam suficientes, as transações de transferência dos *tokens* são executadas pelo próprio contrato da aplicação *Miles2Coins* de forma atômica *(DvP)*, garantindo o envio de *MilesTokens* do vendedor para o comprador e de *SurrealTokens* do comprador para o vendedor. Em caso de falha em qualquer etapa, nenhuma transação é concluída.
- Após a conclusão bem-sucedida da transação, a oferta é marcada como inativa.

Os ativos digitais criados especificamente para a *Miles2Coins* são:

- *MilesToken*: Um *token* no padrão *ERC-1155* [78] representando as milhas aéreas;
- *SurrealToken*: Um *token* no padrão *ERC-20* [75] representando a *stablecoin* fictícia.

Para gerenciar esses *tokens*, além do contrato inteligente principal da aplicação *Miles2Coins* (*Miles2Coins.sol*), foram desenvolvidos dois contratos adicionais: *MilesToken.sol* e *SurrealToken.sol*.

O código-fonte e a documentação da aplicação *Miles2Coins* estão disponíveis em [167].

4.2.2 Requisitos de Privacidade e Usabilidade

No design original, a aplicação *Miles2Coins* não oferece recursos de privacidade além daqueles já proporcionados pela *blockchain* subjacente. Isso implica que os usuários são responsáveis por proteger seus pseudônimos e, idealmente, criar novos pares de chaves para cada transação. Além disso, nessa configuração, todas as transações realizadas na plataforma ficam publicamente registradas na *blockchain*, expondo informações como valores negociados, endereços *Ethereum* das contas envolvidas e saldos dos *tokens*. Essa transparência permite a aplicação de técnicas de desanonimização e vinculação de transações, comprometendo a privacidade dos usuários [7, 8, 9]. Entretanto, é importante também considerar os aspectos de usabilidade da aplicação, como a eficiência, o desempenho e a facilidade de uso, especialmente em contextos onde a privacidade adiciona complexidade ao fluxo de interação.

Diante desse cenário, é fundamental identificar os requisitos de privacidade e usabilidade específicos da aplicação *Miles2Coins*, que servirão como base para a seleção e avaliação das soluções de privacidade mais adequadas para esse caso de uso. Os requisitos identificados são os seguintes:

- **R.M2C-1** – Confidencialidade e Anonimato: Considerando que a aplicação lida com operações financeiras, é imprescindível que os dados transacionais sejam confidenciais. Além disso, preservar o anonimato das partes envolvidas é desejável, para minimizar os riscos de exposição.

- **R.M2C-2** – Descentralização e Independência de Terceiros: Soluções que presumem a confiança em terceiros não são recomendadas, já que se deseja uma aplicação plenamente descentralizada, simples e sem dependências externas que possam comprometer sua segurança ou funcionamento.
- **R.M2C-3** – Compatibilidade com *Ethereum* e Modelo *Account-Based*: Para garantir maior expressividade, a aplicação deve ser compatível com o *Ethereum*, a principal plataforma de contratos inteligentes. O modelo *Account-Based* deve ser suportado, pois está alinhado com a arquitetura do *Ethereum* e facilita futuras expansões para casos de uso mais complexos. A adoção do modelo *UTXO* poderia limitar essa compatibilidade e flexibilidade [26, 27].
- **R.M2C-4** – Desempenho e Custos Transacionais Moderados: É desejável que a aplicação não imponha um *overhead* excessivo no lado do cliente, para não prejudicar a experiência do usuário. Embora os custos transacionais sejam um fator relevante, não se espera que representem uma barreira ao uso da aplicação. Caso necessário, esses custos podem ser incorporados como uma taxa de serviço repassada aos usuários da plataforma.

4.3 Escolha das Soluções de Privacidade

Uma vez mapeados os requisitos de privacidade e usabilidade da aplicação *Miles2Coins*, é possível avançar na escolha das soluções de privacidade mais adequadas para integração. Essa seleção considera os requisitos previamente identificados, as funcionalidades das soluções estudadas e comparadas nas Seções 3.3 e 3.4, e a disponibilidade de repositórios ativos contendo documentação e códigos-fonte. O processo de escolha é detalhado a seguir.

1. Eliminação das soluções que dependam da confiança em terceiros: Das vinte soluções estudadas, seis são baseadas em delegação: *Hawk* [6], *Arbitrum* [132], *Shadoweth* [133], *Ekiden* [112], *Secret Network* [135] e *EtherCloak* [138]. Nessa arquitetura, existe a premissa de confiança em terceiros, como *"managers"* ou *TEE*, dificultando o atendimento ao requisito **R.M2C-2** (descentralização e independência de terceiros). Além disso, a falta de repositórios oficiais (vide Tabela 3.2) e a necessidade de ambientes específicos, como o *Intel SGX*, tornam inviável sua avaliação prática. Assim, essas soluções foram descartadas.
2. Garantia de compatibilidade com *Ethereum*: Para atender ao requisito **R.M2C-3**, as soluções precisam operar no modelo *Account-Based* e ser compatíveis com

o *Ethereum*, possibilitando maior expressividade. Apenas seis soluções atendem a esses critérios: *zkay* [109, 142], *RPSC* [147], *Zether* [110], *Anonymous Zether* [111], *PriDe CT* [151] e *Zeestar* [130]. As demais (*Zapper* [143], *Zexe* [144], *VeriZexe* [145], *Kachina* [146], *zkHawk* [108], *smartFHE* [152] e *fhEVM* [22]) foram excluídas.

3. Verificação de repositórios públicos: Das seis soluções remanescentes, apenas *zkay*, *Anonymous Zether* e *Zeestar* possuem repositórios ativos [157, 161, 162], um fator essencial para viabilizar sua avaliação prática. Portanto, as demais soluções (*RPSC*, *Zether* e *PriDe CT*) foram eliminadas.

4. Escolha do *Anonymous Zether*: Essa solução promete atender plenamente ao requisito **R.M2C-1**, oferecendo confidencialidade às transações e saldos, além de anonimato às partes envolvidas. Adicionalmente, ela não depende de terceiros confiáveis nem exige "*trusted setup*", tornando-a uma forte candidata para integração.

5. Escolha do *Zeestar* como evolução do *zkay*: O *Zeestar* foi proposto pelos principais desenvolvedores do *zkay*, e apresentou melhorias significativas ao projeto original, justificando a exclusão do *zkay* desta seleção. Assim como o *Anonymous Zether*, o *Zeestar* atende ao requisito **R.M2C-1** no que tange à confidencialidade das transações e saldos, embora não ofereça anonimato – um item desejável. Outra desvantagem com relação ao *Anonymous Zether* diz respeito ao "*trusted setup*", necessário para cada aplicação implementada com o *Zeestar*. Porém, uma possível vantagem do *Zeestar* está no menor custo transacional em comparação com o *Anonymous Zether*.

6. Análise do atendimento ao requisito **R.M2C-4** (desempenho e custos moderados): A análise conceitual realizada anteriormente indica que as soluções *Anonymous Zether* e *Zeestar* apresentam ganhos de eficiência em relação ao desempenho e aos custos transacionais quando comparadas às suas predecessoras, *Zether* e *zkay*. Embora seu desempenho seja inferior ao das abordagens baseadas em delegação, essas soluções prometem oferecer um nível mais elevado de privacidade ao eliminarem a necessidade de confiar em terceiros, o que justifica sua escolha.

Com base nesse processo, as soluções *Anonymous Zether* e *Zeestar*, ambas baseadas em criptografia homomórfica e *ZKP*, foram selecionadas para a etapa de avaliação prática. Nessa etapa, avalia-se a integração dessas soluções com a aplicação *Miles2Coins*, considerando o nível de privacidade alcançado, a complexidade da integração, o desempenho e os custos transacionais da aplicação final. Primeiramente, é testado o *Anonymous Zether*, seguido pelo *Zeestar*, e os obstáculos enfrentados em cada integração são detalhados.

4.4 Desafios Esperados e Premissa da Integração

Os principais desafios previstos para a integração das soluções de privacidade incluem:

- Impacto no desempenho: Como abordado no Capítulo 3, as soluções baseadas na combinação de criptografia homomórfica e *ZKP*, embora mais eficientes que aquelas exclusivamente baseadas em *ZKP*, ainda possuem impacto significativo no desempenho. No entanto, considerando que a aplicação *Miles2Coins* operaria no *Ethereum*, onde a taxa de transações *("throughput")* é relativamente baixa [168], e dado que os usuários dessa aplicação provavelmente não executariam muitas transações em um curto espaço de tempo, o desempenho não é considerado um fator crítico neste caso.
- Custos transacionais: Sabe-se que operações como verificações de *ZKP* e cálculos de criptografia homomórfica são computacionalmente intensivas, potencialmente resultando em altas taxas de *gas* no *Ethereum* [28, 111]. Avaliar esses custos é essencial para determinar a viabilidade das soluções em cenários reais.
- Complexidade da integração: As soluções de privacidade podem possuir limitações de funcionalidade que restrinjam sua utilização plena ou comprometam o nível de privacidade alcançado. Durante os testes de integração, deve-se avaliar se os obstáculos encontrados podem ser superados com ajustes pontuais ou se a complexidade das modificações requeridas tornaria a integração impraticável.

Uma premissa fundamental deste projeto é evitar mudanças estruturais significativas na aplicação *Miles2Coins* e nas soluções de privacidade. O objetivo é integrar cada solução sem alterar profundamente as funções da aplicação ou adaptar extensivamente o código das soluções para obter melhores resultados em termos de privacidade ou desempenho. Essa premissa, embora possa resultar em uma solução final mais limitada, é importante para avaliar a aplicabilidade das soluções de privacidade em seu estado atual, sem grandes aprimoramentos ou alterações. Além disso, essa abordagem ajuda a evitar o acoplamento excessivo entre a aplicação e a solução de privacidade, garantindo que a aplicação não se torne excessivamente dependente de uma arquitetura específica de privacidade.

4.5 Integração com o *Anonymous Zether*

Esta seção detalha o processo de integração da aplicação *Miles2Coins* com o *Anonymous Zether*.

4.5.1 Ambiente Experimental – *Anonymous Zether*

O ambiente experimental foi configurado localmente em uma máquina com *Windows* 11, utilizando o *Node.js* [169] como ambiente de execução *JavaScript* e o *Hardhat Network* [170] para emular um nó local da *blockchain Ethereum*. Essa ferramenta permite uma execução estável e controlada de testes, sendo amplamente utilizada para desenvolvimento e depuração de contratos inteligentes. Nesse ambiente, foram implantados o contrato principal da aplicação *Miles2coins*, assim como os contratos dos *tokens MilesToken* e *SurrealToken*. Adicionalmente, os contratos do *Anonymous Zether* também foram configurados e implantados nesse ambiente.

Para os testes de desempenho, foi utilizado o *Apache JMeter* [171]. A configuração de *hardware* utilizada inclui uma *CPU Intel Core i7-10750H* com 6 núcleos e 12 *threads* operando a uma frequência de 2,60 *GHz*, 16 *GB* de *RAM* e um *SSD PCIe* de 512 *GB*.

4.5.2 Aspectos práticos do *Anonymous Zether*

Conforme mencionado na Seção 3.3.6, em sua implementação prática, o *Anonymous Zether* é composto por um conjunto de contratos inteligentes, incluindo os *Zether Smart Contracts (ZSCs)* para gerenciamento de *tokens Zether (ZTH)* e vários contratos auxiliares. Adicionalmente, inclui um cliente composto por *scripts JavaScript* responsáveis por realizar operações criptográficas, como geração de pares de chaves e *ZKPs*, e por interagir com os contratos *Zether* que executam na *blockchain*. O sistema emprega criptografia homomórfica aditiva *ElGamal* [148]. Em um caso de uso básico, os usuários convertem seus *tokens* ou criptomoedas em *tokens ZTH* para efetuar transações de forma confidencial e anônima.

Em 2021, Benjamin E. Diamond, autor do artigo que descreve o *Anonymous Zether* [111], disponibilizou a solução como código aberto no *GitHub* [172]. No entanto, esse repositório não recebeu atualizações desde então. Atualmente, o repositório do *Anonymous Zether* [161] é mantido pela *Kaleido*, uma empresa de tecnologia especializada em *blockchain* [173]. Diamond também publicou um artigo [174] descrevendo um contrato de liquidação para viabilizar transações *DvP* entre dois *tokens Zether*. Esse contrato assegura a troca atômica e privada dos *tokens* entre as partes envolvidas. No entanto, o autor não forneceu uma implementação do contrato de liquidação.

Um projeto recente em que o *Anonymous Zether* foi testado é o *Drex* [13], também conhecido como Real Digital, a futura moeda digital (*CBDC*) brasileira. Na fase piloto do projeto, o Banco Central do Brasil e determinadas instituições financeiras participantes do projeto avaliaram diversas soluções de privacidade, incluindo o *Anonymous Zether*. Nesse contexto, foi desenvolvido um contrato de liquidação para permitir transações *DvP*

utilizando *tokens Zether* representando a *CBDC* e outros ativos, como títulos públicos federais.

Neste trabalho, uma vez que o *Anonymous Zether* é empregado para aprimorar a privacidade dos usuários da aplicação *Miles2Coins*, que contempla transações *DvP*, são utilizados artefatos dos seguintes projetos:

- *Anonymous Zether (branch "hardhat")* [161], mantido pela *Kaleido*, que inclui os principais contratos inteligentes da solução e outros artefatos relacionados.
- *Anonymous Zether Client* (*branch "real-digital"*) [175], também mantido pela *Kaleido*, que contém a implementação de um cliente *JavaScript* para interação com os contratos *Zether*.
- Projeto piloto *Drex*, desenvolvido pelo Banco Central do Brasil [176], contendo o contrato de liquidação para transações *DvP* utilizando o *Anonymous Zether*.

A principal limitação do *Anonymous Zether* reside no fato de que sua funcionalidade é restrita a uma classe limitada de contratos inteligentes privados, como trocas de *tokens*, leilões, canais de pagamento, mecanismos de votação e consenso [110]. Considerando que a funcionalidade mais crítica da aplicação *Miles2Coins* é a transação *DvP* na qual os *MilesTokens* são trocados por *SurrealTokens*, essa restrição deve impactar, mas não impedir o uso do *Anonymous Zether* para aprimorar a privacidade.

Nesse cenário, as transações de registro e listagem de ofertas (*placeOffer* e *listOffers*) da aplicação *Miles2Coins* não podem ser integradas ao *Anonymous Zether*, pois não envolvem operações diretas com *tokens*. A integração é relevante apenas para a etapa final da operação *acceptOffer*, onde ocorre a transferência efetiva dos *tokens* entre as partes.

4.5.3 Processo de Integração – *Anonymous Zether*

O principal objetivo da integração da aplicação *Miles2Coins* com o *Anonymous Zether* é garantir que as transações *DvP* sejam realizadas de forma confidencial, ocultando os detalhes das operações e os saldos de *MilesToken* e *SurrealToken*, além de preservar o anonimato das partes envolvidas. Antes de realizar a integração, é interessante observar como ocorre uma transação *DvP* utilizando apenas o *Anonymous Zether* e os *tokens MilesToken* e *SurrealToken* diretamente.

Transação *DvP* no *Anonymous Zether*. O primeiro passo é o *deploy* dos contratos *ZSC* e dos *tokens ZTH*, a ser feito pelo administrador do *Anonymous Zether*. Feito isso, para que os usuários realizem uma transação *DvP* usando o *Anonymous Zether*, eles

precisam interagir com os *scripts* do *Anonymous Zether Client* para realizar as seguintes ações:

1. Criar uma conta no *Zether* (função *newAccount*), que basicamente gera uma carteira *Ethereum* com seu endereço e par de chaves associado, e uma conta confidencial *("shielded account")*, composta por um par de chaves criptográficas *ElGamal* usadas no *Zether*.
2. Registrar a conta confidencial nos contratos *Zether* correspondentes aos *tokens ZTH* que o usuário utilizará (*MilesToken* e *SurrealToken*). Esse passo é necessário para que o *Zether* autorize as operações com os *tokens ZTH*.
3. Financiar o contrato *Zether* (função *fund*) correspondente ao *token* desejado com a quantidade de *tokens* a ser transferida.
4. Criar uma conta *Ethereum* de uso único para a transação *DvP*.
5. Iniciar a transação *DvP* usando a função *startDvP*. Esta função recebe os seguintes parâmetros:
   - *sender*: chave pública da conta confidencial do remetente;
   - *receiver*: chave pública da conta confidencial do destinatário;
   - *amount*: quantidade de *tokens ZTH* a ser transferida;
   - *signer*: conta *Ethereum* de uso único;
   - *zsc*: endereço do contrato inteligente *Zether (ZSC)* correspondente ao *token ZTH* a ser transferido.

   A função *startDvP* gera a *ZKP* no lado do cliente usando a chave privada *ElGamal* associada à conta do usuário, entre outros parâmetros.
6. Confirmar a transação *DvP* usando a função *executeDvP*, que recebe os seguintes parâmetros:
   - *senderEthAddress*: conta *Ethereum* de uso único gerada no passo 4 ou retornada pela função *startDvP* se não for fornecida;
   - *counterpartyEthAddress*: conta *Ethereum* de uso único da contraparte;
   - *proof*: *ZKP* gerada no passo anterior (*startDvP*).

Ressalta-se que ambas as partes envolvidas na transação *DvP* precisam completar todos os passos mencionados. Contudo, o protocolo *Zether* não especifica como as chaves públicas das contas confidenciais e os endereços das contas *Ethereum* são compartilhados

entre as partes. Além disso, o *Zether* utiliza o conceito de *epochs* para dividir o tempo em blocos, com duração padrão de 6 segundos. Durante uma *epoch*, as contas participantes de uma transação são bloqueadas, e as transações são processadas apenas ao final desse intervalo. A *epoch* é um dos parâmetros usados na geração das *ZKPs* do *Zether*. Isso significa que ambas as partes de uma transação precisam sincronizar suas chamadas das funções *startDvP* e *executeDvP* dentro da mesma *epoch*, caso contrário, a verificação das *ZKPs* na etapa de execução da transação *DvP* falhará.

Desafios de integração. Conforme descrito na Seção 4.2.1, na aplicação original, a transação *DvP* é executada pelo contrato inteligente *Miles2Coins* em nome dos usuários, garantindo a execução atômica das transferências. No entanto, ao utilizar o *Anonymous Zether*, o contrato *Miles2Coins* não pode realizar a transação, pois não possui as chaves privadas associadas aos usuários, necessárias para gerar as *ZKPs*.

Assim, é necessário que os próprios usuários executem a transação *DvP* por meio do *Anonymous Zether*. Porém, como já dito, para que os usuários realizem as transações, eles precisam conhecer a chave pública *ElGamal* da conta confidencial e o endereço *Ethereum* de uso único da outra parte, além de se sincronizar para realizar as operações dentro da mesma *epoch*. Portanto, a aplicação *Miles2Coins* precisaria passar por diversas adaptações para viabilizar um mecanismo auxiliar capaz de mediar essa troca de informações entre as partes. Isso exigiria mudanças nas funções existentes, além da criação de novas funções para que as partes troquem as informações necessárias e concordem com um horário para a transação *DvP*, tudo isso de forma segura por meio de outras técnicas criptográficas.

Considerando a premissa de não realizar mudanças estruturais significativas na aplicação *Miles2Coins* e nas soluções de privacidade, decidiu-se que as adaptações mencionadas não seriam implementadas, especialmente para evitar o acoplamento excessivo entre a aplicação e o *Anonymous Zether*. Assim, neste trabalho, assume-se que o contrato *Miles2Coins*, ao receber uma transação de aceite de oferta (função *acceptOffer*), simplesmente emitirá o evento *offerAccepted*. Ao detectar este evento, as partes envolvidas na transação devem utilizar outros meios para negociar os parâmetros da transação *DvP* a ser realizada via *Anonymous Zether*. Essa comunicação poderia ser realizada usando protocolos de mensagens *peer-to-peer* seguros, como o *Waku* [177], ou outros métodos seguros, conforme apropriado.

Fluxo da aplicação *Miles2Coins* com o *Anonymous Zether*. A Figura 4.1 ilustra um fluxo típico de operações na aplicação *Miles2Coins* com a privacidade da transação *DvP* aprimorada por meio do *Anonymous Zether* – indicada pelas operações marcadas com um cadeado. O diagrama mostra a interação entre o Usuário A (Alice) e o Usuário

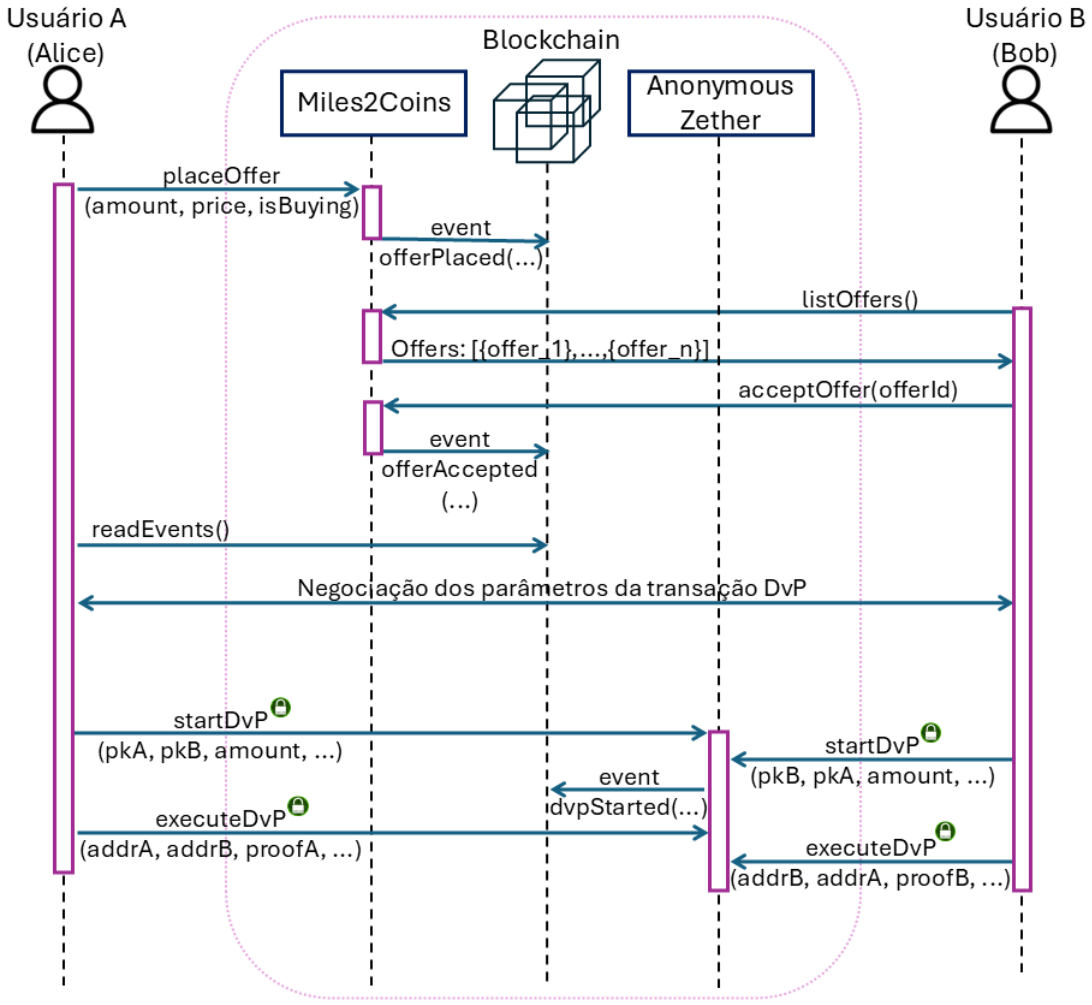

Figura 4.1: Fluxo da aplicação *Miles2Coins* com a transação *DvP* via *Anonymous Zether*

B (Bob) e a *blockchain*, onde a aplicação *Miles2Coins* é usada para criação e aceite de uma oferta, enquanto o *Anonymous Zether* é utilizado para executar a transação *DvP* com privacidade. As transações são realizadas da seguinte forma:

1. Registro de uma oferta:

    - Alice, interessada em comprar ou vender *MilesTokens* em troca de *SurrealTokens*, inicia o processo chamando a função *placeOffer* no contrato *Miles2Coins*. Ela deve especificar a quantidade de *tokens* a ser negociada e o preço, além de indicar se a operação é de compra ou venda.
    - O contrato *Miles2Coins* armazena a oferta e emite o evento *offerPlaced* na *blockchain*, informando o *ID* e outros detalhes da oferta.

2. Listagem das ofertas ativas:

    - Bob, interessado em visualizar as ofertas disponíveis, chama a função *listOffers*.

- Bob recebe uma lista com todas as ofertas ativas, incluindo a registrada por Alice.

3. Aceite de uma oferta:

   - Após analisar as ofertas, Bob aceita a oferta de Alice chamando a função *acceptOffer* e informando o *ID* da oferta. Isso dispara o evento *offerAccepted* na *blockchain*.
   - Alice monitora os eventos e detecta que sua oferta foi aceita.

4. Negociação dos parâmetros da transação *DvP*:

   - Conforme discutido anteriormente, Alice e Bob devem se comunicar para negociar os parâmetros e o horário exato da transação *DvP*.

5. Início da transação *DvP*:

   - Uma vez acordados os parâmetros da transação, Alice inicia o processo chamando a função *startDvP* na aplicação *Anonymous Zether*. Isso inclui fornecer parâmetros como as chaves públicas *ElGamal* dos usuários, o valor da transação e a geração da *ZKP* que assegura a validade da transação.
   - Aproximadamente ao mesmo tempo, Bob também chama a função *startDvP* para iniciar sua parte da transação *DvP*.
   - Essas ações são registradas na *blockchain* por meio do evento *dvpStarted*.

6. Conclusão da transação *DvP*:

   - Alice e Bob chamam a função *executeDvP*, informando seus endereços *Ethereum* de uso único e as *ZKPs* geradas na transação *startDvP*.
   - Neste momento, os *tokens* são trocados de forma confidencial usando o *Anonymous Zether*.
   - Apesar de nesta etapa não ser registrado um evento, o próprio registro da transação com sucesso na *blockchain* já permite sua confirmação e a torna irreversível.

Ressalta-se que as atividades de *deploy* dos contratos e de *minting* dos *tokens* para os usuários não estão representadas na figura, visando maior clareza. Essas operações são realizadas previamente, fora do fluxo padrão da aplicação *Miles2Coins*.

Implementação. Este trabalho envolve a implementação do fluxo de transações descrito, considerando um cenário específico em que Alice atua como vendedora de *MilesTokens*, enquanto Bob os adquire utilizando *SurrealTokens*. Inicialmente, para avaliar o desempenho e os custos transacionais da versão original da *Miles2Coins*, os passos 1 a 3 do fluxo foram implementados diretamente na aplicação, sem mecanismos de privacidade. Nesse cenário, quando Alice chama a função *placeOffer* para vender *MilesTokens*, o contrato inteligente da aplicação recebe sua aprovação para realizar a transferência dos *tokens* em seu nome. Da mesma forma, quando Bob aceita a oferta, o contrato também recebe sua aprovação para transferir seus *SurrealTokens*. Isso permite que a função *acceptOffer*, ao ser invocada por Bob, execute a transação *DvP* em nome dos usuários. Como a privacidade não está ativada nesse fluxo, todos os detalhes das transações permanecem publicamente visíveis. Na sequência, foi implementado um segundo fluxo utilizando o *Anonymous Zether* para garantir a privacidade nos passos 5 e 6, que correspondem especificamente à transação *DvP*.

A documentação completa de ambos os fluxos está disponível no diretório *"examples"* dentro do repositório da aplicação *Miles2Coins* [167]. Além disso, o *Anonymous Zether* e o *Anonymous Zether Client*, com pequenas adaptações realizadas, foram incorporados como módulos no mesmo repositório, tendo seus códigos-fonte e documentações localizados na pasta *"external"*.

4.6 Integração com o *Zeestar*

Esta seção detalha o processo de integração da aplicação *Miles2Coins* com o *Zeestar*.

4.6.1 Ambiente Experimental – *Zeestar*

Diferentemente do *Anonymous Zether*, que baseia-se apenas em contratos inteligentes e *Javascript*, o *Zeestar* utiliza majoritariamente *scripts Python*. Assim, optou-se por baixar o repositório do *Zeestar* e instalá-lo em um ambiente *Docker* executando no *Docker Desktop* para *Windows* [178]. A máquina foi a mesma utilizada nos testes com o *Anonymous Zether*, portanto as especificações são as mesmas que constam em 4.5.1, com exceção para a memória *RAM*, que ficou limitada a 8 *GB* no contêiner *Docker* em que a solução foi executada. Ressalta-se que, embora os ambientes não sejam idênticos, devido às particularidades de cada solução, ainda assim foi possível obter estimativas representativas de desempenho.

4.6.2 Aspectos práticos do *Zeestar*

Além da documentação que consta no repositório principal do *Zeestar* [162], há um tutorial disponibilizado pelos autores [179] que inclui um contrato de exemplo e os comandos para compilá-lo e rodá-lo utilizando o *backend eth-tester* [180] para emular a *blockchain Ethereum*. O *eth-tester* permite executar comandos de maneira interativa. Outro caminho para testar o *Zeestar* seria executar um cliente *standalone Ethereum* usando *Ganache* [181] ou *Hardhat* [170] e implementar os contratos nesse nó local.

O *Zeestar* manteve a abordagem original do *zkay*, apresentando-se como uma linguagem que utiliza anotações de privacidade para especificar quem pode acessar determinados dados. Essas anotações seguem o formato $\tau@\alpha$, em que τ representa o tipo do dado e α define quem pode acessá-lo. As opções para α são:

1. *me*: Apenas o chamador atual pode acessar o dado.
2. *all*: Dado público acessível por todos.
3. Variável de estado: Um proprietário fixo e público (ex.: um endereço específico).
4. Chave de mapeamento: Proprietário baseado na chave do mapeamento (ex.: *mapping(address!x => uint@x)*).

Essas anotações protegem dados sensíveis ao restringir a leitura aos proprietários designados, mas é importante destacar que qualquer usuário pode alterar os dados, mesmo que estejam marcados como privados.

Adicionalmente, o *Zeestar* introduz a utilização de *tags* no formato $< \mu >$, onde $\mu \in \{+, \}$. A tag $< + >$ especifica que uma variável será cifrada com um esquema de criptografia homomórfica aditiva, permitindo a realização de operações de adição e subtração diretamente sobre os dados criptografados, sem necessidade de revelar seu conteúdo. Por outro lado, quando a *tag* está ausente ou definida como $<>$, aplica-se um esquema de criptografia não homomórfica ao dado.

4.6.3 Processo de Integração – *Zeestar*

Para implementar a aplicação *Miles2Coins* com a privacidade proporcionada pela solução *Zeestar* (ou *zkay* 0.3), é necessário adaptar seu código para a linguagem *zkay*, incluindo as anotações de privacidade para os dados. Nesse sentido, foi elaborado o contrato *miles2coins.zkay*.

Diferentemente do *Anonymous Zether*, o *Zeestar* não se propõe a prover anonimato às partes envolvidas nas transações. Assim, o objetivo principal desta parte do trabalho seria manter a confidencialidade dos saldos dos usuários. Para isso, as funções de *mint*, usadas

para gerar *tokens* para os usuários, e as transações de leitura e escrita no saldo foram configuradas para operar com valores privados ao usuário, conforme ilustrado abaixo:

- Função para criação *("mint")* de milhas :

```
function mintMiles(address to, uint32@me<+> amount) public {
    require(me == owner);
    miles_balance[to] += reveal(amount, to);
}
```

- Função para obtenção do saldo de milhas:

```
function getMilesBalance() public view returns (uint32@me<+>) {
    return miles_balance[me];
}
```

Destaca-se o uso da anotação *@me<+>* para indicar que as variáveis *amount* e *miles_balance[x]* devem ser cifradas utilizando criptografia homomórfica. A função *reveal* é utilizada no *Zeestar* para tornar uma variável pública (caso se utilize *"all"* como segundo argumento) ou para recriptografar o dado apenas para acesso de determinado destinatário. No caso da função *mintMiles*, a variável *"amount"* é recriptografada apenas para o endereço *"to"*, ou seja, para o usuário que está recebendo os *tokens*.

Desafios de integração. O *Zeestar* apresentou uma limitação relevante que se tornou evidente apenas durante este experimento: a impossibilidade de contratos privados chamarem funções externas de outros contratos. Essa restrição impede, por exemplo, que sejam implementados contratos separados para a aplicação *Miles2Coins* e os *tokens MilesToken* e *SurrealToken*. Para contornar essa limitação e viabilizar os testes, foi necessário consolidar todos esses elementos em um único contrato inteligente. Assim, o contrato *miles2coins.zkay* foi elaborado de forma a incorporar tanto as operações da aplicação *Miles2Coins* quanto as principais funcionalidades dos *tokens*.

Durante a implementação, constatou-se que, para evitar que terceiros inferissem os saldos a partir das transações de compra e venda de *MilesTokens*, os valores dessas transações precisariam ser mantidos privados. No entanto, a arquitetura do *Zeestar* demonstrou limitações importantes nesse aspecto, sobretudo pela dificuldade de manipular valores privados pertencentes a múltiplos usuários. Como a linguagem *zkay* exige que as somas homomórficas ocorram sempre no mesmo domínio de privacidade, se os valores transacionados fossem privados ao criador da oferta ou ao próprio contrato, seria inevitável torná-los públicos para efetuar as operações. A opção de reclassificá-los exclusivamente

para o usuário que aceita a oferta também não é viável, pois esse usuário não pode chamar a função *reveal* para dados que não são de sua propriedade. Essas limitações inviabilizam um fluxo em que o valor transacionado permaneça totalmente oculto para terceiros.

Fluxo da aplicação *Miles2Coins* com o *Zeestar*. A Figura 4.2 ilustra o fluxo da aplicação *Miles2Coins* utilizando o *Zeestar*. Nesse contexto, as funcionalidades da aplicação são mantidas, mas agora com os saldos dos usuários confidenciais, conforme indicado pelos cadeados na figura. Destaca-se que, nesse caso, o contrato *Miles2Coins* mantém em custódia temporária os *tokens* necessários para garantir a conclusão bem-sucedida da transação *DvP* no momento de aceite da oferta.

No fluxo ilustrado, as seguintes transações são realizadas:

1. Alice chama a função *placeOffer* para registrar uma oferta de venda de milhas, identificada pela variável $isBuying = 0$
    - Os *MilesTokens* de Alice são transferidos para o contrato como garantia e associados à oferta realizada.
2. Bob invoca *listOffers* para verificar as ofertas disponíveis.
    - Bob recebe uma lista das ofertas ativas, incluindo a registrada por Alice.
3. Bob aceita a oferta de Alice por meio da função *acceptOffer*.
    - O valor total em *SurrealTokens* ($total\_cost = amount \times price$) é transferido ao contrato.
    - Em seguida, o contrato realiza as transferências dos *tokens* para Alice e Bob, atualizando os saldos de *MilesTokens* e *SurrealTokens* conforme apropriado.

Para maior clareza do diagrama, transações como o *deploy* dos contratos, *minting* dos *tokens* e outras operações menos relevantes não são exibidas.

Implementação. De forma análoga ao processo realizado para o *Anonymous Zether*, o fluxo de transações apresentado na Figura 4.2 foi implementado utilizando o *Zeestar* para aprimoramento da privacidade. Nesse caso, os contratos da aplicação *Miles2Coins* e dos *tokens* foram adaptados para a linguagem *zkay* e unificados em um único contrato, devido à restrição de chamadas a funções externas, conforme descrito na Seção 4.6.3. Foram adicionadas as devidas anotações de privacidade para possibilitar o *"minting"* confidencial dos *tokens* e a manutenção de saldos privados.

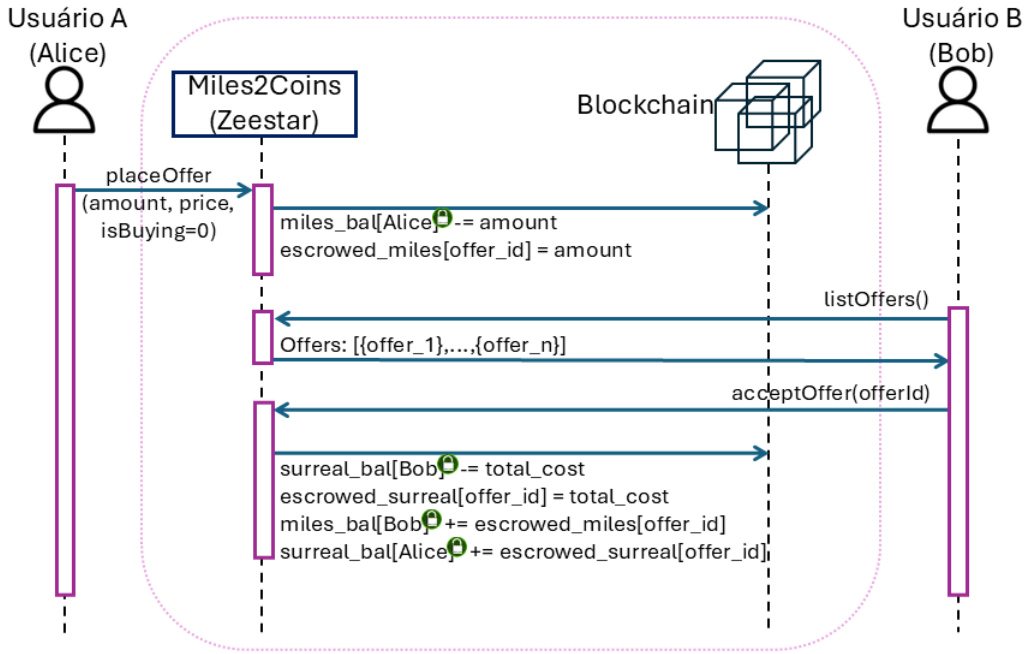

Figura 4.2: Fluxo da aplicação *Miles2Coins* com o *Zeestar*

A documentação completa desse fluxo, juntamente com o código-fonte do contrato *miles2coins.zkay*, está disponível no diretório *"examples"* do repositório da aplicação *Miles2Coins* [167]. Também foi realizado um *fork* do repositório original do *Zeestar (zkay v0.3)* [182].

4.7 Resultados Experimentais

Esta seção apresenta a avaliação dos resultados da integração da aplicação *Miles2Coins* com as soluções *Anonymous Zether* e *Zeestar*, visando responder às seguintes questões-chave:

1. *Qual o custo transacional total para executar um fluxo completo de operações na aplicação antes e depois da integração com as soluções de privacidade?*
2. *Qual é o impacto no desempenho da aplicação devido ao aprimoramento de privacidade com cada solução?*
3. *Em que medida a privacidade é aprimorada após a integração da aplicação com cada solução?*
4. *Em que nível os requisitos mapeados nas seções 3.2.3 e 4.2.2 são atendidos?*

Tabela 4.1: Custos Transacionais – *Miles2Coins* Original

Usuário	**Transação**	**Contrato**	**Gas**	**US$[2]**
Admin	Deploy	SurrealToken	654.180	$ 24,86
Admin	Deploy	MilesToken	1.119.477	$ 42,54
Admin	Deploy	Miles2Coins	910.807	$ 34,61
Admin	Mint MilesTokens	MilesToken	52.534	$ 2,00
Admin	Mint SurrealTokens	SurrealToken	70.614	$ 2,68
A	placeOffer	Miles2Coins	211.406	$ 8,03
B	listOffers	Miles2Coins	0[1]	-
B	acceptOffer	Miles2Coins	151.319	$ 5,75

Uso Total de Gas		
Admin	2.807.612	$ 106,69
Usuário A	211.406	$ 8,03
Usuário B	151.319	$ 5,75

Notas:
[1] *listOffers* não utiliza nenhum *gas* porque é executada localmente.
[2] O preço do *gas* e a taxa de câmbio *ETH/USD* são calculados com base na média dos últimos 90 dias, considerando dados até 28 de fevereiro de 2025 [183, 184]:
$1\,gas = 11,758\,gwei = 11,758 \times 10^{-9}\,ETH$ / $1\,ETH =$ *US$*$3.231,68$

O código-fonte e a documentação da aplicação *Miles2Coins*, bem como os *scripts*, contratos, documentos e outros artefatos referentes à integração com as soluções *Anonymous Zether* e *Zeestar* estão disponíveis em [167].

4.7.1 Custos Transacionais

Os custos transacionais são um fator crítico para aplicações implementadas em *blockchain*, em especial no *Ethereum*, onde as tarifas de *gas* podem atingir valores elevados, dependendo do caso. Quanto mais operações determinada transação realiza *on-chain*, maior é seu custo. Adicionalmente, operações que demandam muitos recursos, como *CPU* e armazenamento, geralmente incorrem em custos mais elevados. Um desafio recorrente para várias soluções de privacidade, em especial aquelas que fazem uso de *ZKPs*, é o custo das operações criptográficas realizadas *on-chain*.

Portanto, é essencial analisar os custos transacionais da aplicação *Miles2Coins*, juntamente com os custos adicionais resultantes da incorporação de privacidade por meio das soluções *Anonymous Zether* e *Zeestar*. A Tabela 4.1 mostra os custos em *gas* associados às transações da aplicação *Miles2Coins* original, sem aprimoramentos na privacidade, enquanto as tabelas 4.2 e 4.3 revelam os custos referentes às transações utilizando o *Anonymous Zether* e *Zeestar*, respectivamente.

Tabela 4.2: Custos Transacionais – *Anonymous Zether*

Usuário	**Transação**	**Contrato**	**Gas**	**US$[3]**
Admin	Deploy ZSC1 (SurrealToken)	ZSC1	2.085.169	$ 79,23
	Deploy ZSC2 (MilesToken)	ZSC2	2.213.277	$ 84,10
	Deploy DvpZSC	DvpZSC	1.600.353	$ 60,81
	Deploy - Contratos Auxiliares	5 Contratos no Total	8.508.558	$ 323,31
A	Criar conta Zether	- (local) [1]	0	$ -
B	Criar conta Zether	- (local) [1]	0	$ -
A	Registrar em ambos os ZSCs	ZSC1/ZSC2	450.734	$ 17,13
B	Registrar em ambos os ZSCs	ZSC1/ZSC2	416.486	$ 15,83
A	Aprovar ZSC2 para transferir MilesTokens	MilesToken	46.177	$ 1,75
A	Fund ZSC2	ZSC2	207.318	$ 7,88
B	Aprovar ZSC1 para transferir SurrealTokens	SurrealToken	46.223	$ 1,76
B	Fund ZSC1	ZSC1	205.711	$ 7,82
A	startDvP	DvpZSC	441.697	$ 16,78
B	startDvP	DvpZSC	441.687	$ 16,78
A	executeDvP	DvpZSC	1.537.887	$ 58,44
B	executeDvP	DvpZSC	8.427.527[2]	$ 320,23

Uso Total de Gas			
Usuário	**Gas**	**US$[3]**	**Comparação Miles2Coins**
Admin	14.407.357	$ 547,46	5,1x
Usuário A	2.683.813	$ 101,98	12,7x
Usuário B	9.537.634	$ 362,42	63,0x

Notas:

[1] As contas do *Zether* ("*shielded accounts*") são criadas localmente pelo cliente.

[2] Como o Usuário B é o segundo a chamar *executeDvP*, operações adicionais são realizadas *on-chain*. Estas incluem verificações de *ZKP* e atualizações de saldo, levando a um consumo maior de *gas*.

[3] O preço do *gas* e a taxa de câmbio *ETH/USD* são calculados com base na média dos últimos 90 dias, considerando dados até 28 de fevereiro de 2025 [183, 184]: $1\,gas = 11,758\,gwei = 11,758 \times 10^{-9}\,ETH\,/\,1\,ETH = US\$3.231,68$

Tabela 4.3: Custos Transacionais – *Zeestar*

Usuário	Transação	Contrato	Gas	US$[2]
Admin	Criação de Usuários e Deploy de Contratos	Miles2Coins e 6 Contratos Auxiliares	6.656.045	$ 252,92
Admin	Mint MilesTokens (Usuário A)	Miles2Coins	358.940	$ 13,64
Admin	Mint SurrealTokens (Usuário B)	Miles2Coins	359.023	$ 13,64
A	placeOffer	Miles2Coins	417.149	$ 15,85
B	listOffers	Miles2Coins	0[1]	-
B	acceptOffer	Miles2Coins	527.992	$ 20,06

Uso Total de Gas			
Usuário	**Gas**	**US$[2]**	**Comparação Miles2Coins**
Admin	7.374.008	$ 280,20	2,6x
Usuário A	417.149	$ 15,85	1,9x
Usuário B	527.992	$ 20,06	3,6x

Notas:
[1] *listOffers* não utiliza nenhum *gas* porque é executada localmente.
[2] O preço do *gas* e a taxa de câmbio *ETH/USD* são calculados com base na média dos últimos 90 dias, considerando dados até 28 de fevereiro de 2025 [183, 184]:
$1\,gas = 11,758\,gwei = 11,758 \times 10^{-9}\,ETH$ / $1\,ETH = US\$3.231,68$

Custos – *Miles2Coins*. Na aplicação *Miles2Coins* original, a implantação dos contratos da aplicação e dos *tokens*, bem como o *minting* inicial de *tokens* para os usuários, consumiu um total de 2.807.612 *gas*, resultando em um custo estimado de aproximadamente US$ 106,69. Embora esse valor seja relativamente elevado, ele corresponde a um custo único, tornando-o aceitável dentro do contexto analisado. Já as transações realizadas pelos usuários A e B tiveram custos de US$ 8,03 e US$ 5,75, respectivamente. Esses valores também podem ser considerados razoáveis, diante da alta cotação atual do *ETH* e, consequentemente, do *gas*.

Custos – *Anonymous Zether*. O custo aferido para as transações do *Anonymous Zether*, por sua vez, foi significativamente mais alto. A implantação dos *tokens* privados e dos demais contratos necessários demandou mais de 14 milhões de *gas*, equivalente a US$ 547,46. Embora o elevado custo já tenha sido relatado nos artigos do *Zether* [110] e do *Anonymous Zether* [111], nenhum deles incluiu uma análise específica de transações *DvP*. No caso testado neste trabalho, o custo total das operações realizadas pelos usuários foi de US$ 464,40. Essas cifras demonstram que a utilização do *Anonymous Zether* no *Ethereum* para esse caso de uso seria inviável.

Custos – *Zeestar*. Embora a solução *Zeestar* tenha apresentado custos mais baixos em comparação ao *Anonymous Zether*, o valor médio a ser desembolsado pelos usuários foi cerca de 2,6 vezes superior ao das transações realizadas na aplicação *Miles2Coins* original. Esse aumento significativo representa um obstáculo importante para a adoção da solução em cenários reais.

Considerações sobre os custos. Implementar privacidade *on-chain* no *Ethereum* traz grandes desafios com relação ao custo. Como as soluções testadas e muitas outras dependem fortemente de operações criptográficas, seria ideal que essas operações fossem oferecidas como *precompiles* – contratos pré-compilados que incluem funções otimizadas e com menor custo de execução [3]. No entanto, a inclusão de novos *precompiles* exige consenso da comunidade, pois representam mudanças fundamentais no protocolo. Devido aos altos custos de *gas* e à volatilidade do *ETH*, implementar privacidade no *Ethereum* pode não ser viável economicamente até que operações criptográficas sejam providas pela rede de forma significativamente mais barata.

4.7.2 Desempenho

O impacto no desempenho das soluções de privacidade pode ser bastante elevado, devido às operações criptográficas necessárias. Nesta avaliação, foi aferida a utilização de *CPU* e *RAM*, bem como o tempo necessário para efetuar as transações da aplicação *Miles2Coins* sem privacidade adicional. Posteriormente, as mesmas métricas foram analisadas para a transação *DvP* realizada via *Anonymous Zether*. Por fim, foi avaliado o desempenho do fluxo completo da aplicação implementada via *Zeestar*. Todas as métricas foram obtidas com base na média de 10 execuções de cada teste avaliado, a fim de garantir maior consistência nos resultados.

Desempenho – *Miles2Coins*. A Tabela 4.4 mostra o desempenho de um fluxo completo da aplicação *Miles2Coins* original, compreendendo as seguintes transações: *placeOffer* (usuário A), *listOffers* (usuário B) e *acceptOffer* (usuário B), que inclui a transação *DvP*. A tabela fornece dois conjuntos de dados: um para uma única execução do fluxo completo e outro para 100 execuções consecutivas do mesmo fluxo. Observa-se que a execução de 100 fluxos completos, totalizando 300 transações, leva apenas 2,4 segundos, atingindo uma taxa superior a 7,3 mil transações por minuto.

Desempenho – *Anonymous Zether*. Para avaliar especificamente o desempenho da transação *DvP* utilizando o *Anonymous Zether*, até 32 contas foram criadas, cada uma recebendo 10.000 *SurrealTokens* e 10.000 *MilesTokens*, e esses *tokens* foram usados para

Tabela 4.4: Desempenho do *Miles2Coins*

Nº de Fluxos Completos	**Nº de Transações**	**Tempo (ms)**	**Tx/Min**	**Uso de CPU**	**Uso de RAM (MB)**
1	3	401	448,88	17%	42,19
100	300	2.436	7.389,16	25%	84,22

Tabela 4.5: Desempenho da Transação *DvP* via *Anonymous Zether*

Usuários	**Txs DvP**	**Tempo (ms)**	**Tx/Min**	**Uso de CPU (médio/máx.)**	**Uso de RAM (MB)**
2	1	11.202	5,36	8%/20%	272,03
4	2	20.128	5,96	12%/43%	403,76
8	4	35.336	6.79	16%/76%	667,89
16	8	67.154	7,15	16%/96%	1237.14
32	16	132.271	7.26	16%/96%	2289,80

financiar seus respectivos contratos *Zether – ZSC1*, representando a versão privada do *SurrealToken*, e *ZSC2*, representando o *MilesToken* privado. Um *script* foi desenvolvido para receber um par específico de usuários como parâmetro e executar as quatro etapas sequenciais necessárias para completar a transação *DvP* entre eles:

1. *startDvP* para o usuário A, que venderia *MilesTokens (tokens ZSC2).*
2. *startDvP* para o usuário B, que faria a compra, transferindo *SurrealTokens (tokens ZSC1).*
3. *executeDvP* para o usuário A, confirmando sua parte da transação *DvP.*
4. *executeDvP* para o usuário B, confirmando sua parte e concluindo a transação *DvP.*

Com essa configuração, o *JMeter* foi utilizado para executar o *script* mencionado e realizar as transações *DvP* em paralelo entre vários pares de usuários simultaneamente. Inicialmente, apenas dois usuários foram simulados (uma única transação *DvP*), depois o número foi dobrado até atingir 32, resultando na execução paralela de 16 transações *DvP*. Os dados de desempenho avaliados para o *Anonymous Zether* constam na Tabela 4.5.

Como se pode observar, o *Anonymous Zether* introduziu um *overhead* significativo no desempenho nas transações *DvP*. Uma única transação usando o *Zether* levou mais de 11 segundos, quase 28 vezes mais que os 401 milissegundos para o fluxo completo – incluindo a transação *DvP* – na aplicação *Miles2Coins* original. O processamento de dezesseis transações pelo *Zether* gerou um consumo de mais de 2 *GB* de *RAM* e levou mais de 2 minutos, resultando em uma taxa de transferência de apenas 7,26 transações por minuto.

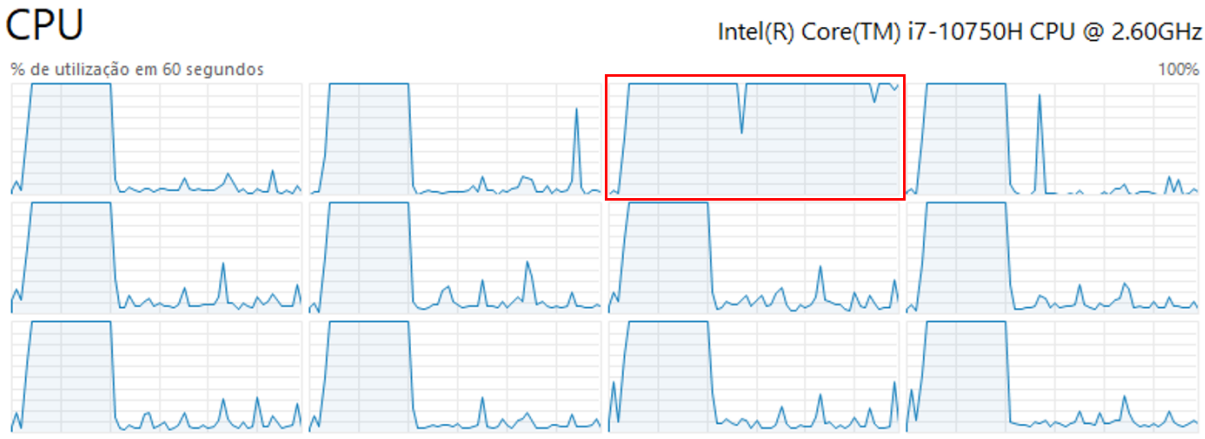

Figura 4.3: Uso de *CPU* no *Anonymous Zether* – 16 Transações *DvP*

É importante ressaltar que, para este experimento, a ferramenta *Hardhat Network* foi configurada para minerar transações automaticamente, de modo que a taxa de transferência da rede *blockchain* não afetasse os resultados dos testes. Embora a máquina experimental possua 6 núcleos e 12 *threads*, o processo do nó da *Hardhat Network* não suporta *multithreading*. Assim, o *JMeter* foi eficaz apenas para o processamento simultâneo de operações executadas localmente pelo código *JavaScript*, como a geração de *ZKP*s. As operações *on-chain* realizadas pelos contratos *Zether*, como a verificação de *ZKP*s e atualização de saldos criptografados, sobrecarregaram significativamente o desempenho devido à ausência de *multithreading*. Isso pode ser confirmado na Figura 4.3, que demonstra que os 12 processadores lógicos foram ocupados nos primeiros 20 segundos aproximadamente, após os quais apenas o processador destacado continuou com o processamento restante. Embora os experimentos tenham sido realizados em uma única máquina, o uso de múltiplas máquinas (por exemplo, uma para cada usuário) ajudaria apenas a paralelizar tarefas executadas localmente. Isso não afetaria as operações na *blockchain*, pois todas as transações devem ser executadas sequencialmente e na mesma ordem por cada nó da rede.

Como mencionado anteriormente, o *Anonymous Zether* utiliza o conceito de *epochs* para dividir o tempo em blocos configuráveis, com tamanho padrão de 6 segundos. No entanto, devido à complexidade das transações *DvP*, durante os testes, a *epoch* precisou ser estendida para 120 segundos – ou até 240 segundos no teste envolvendo 32 usuários – para evitar erros na verificação de *ZKP*s, pois essas provas utilizam a *epoch* como um de seus parâmetros de entrada. Como resultado, após o processamento das transações, os novos saldos das contas envolvidas só eram atualizados ao final da *epoch* configurada, o que poderia levar quase quatro minutos no pior cenário.

Tabela 4.6: Desempenho da Solução *Zeestar*

Nº de Fluxos Completos	**Nº de Transações**	**Tempo (segundos)**	**Tx/Min**	**Uso de CPU (médio/máx.)**	**Uso de RAM (MB)**
1	3	93	1,94	18,7%/97,5%	879,9
10	30	953	1,89	17,2%/100%	946,9

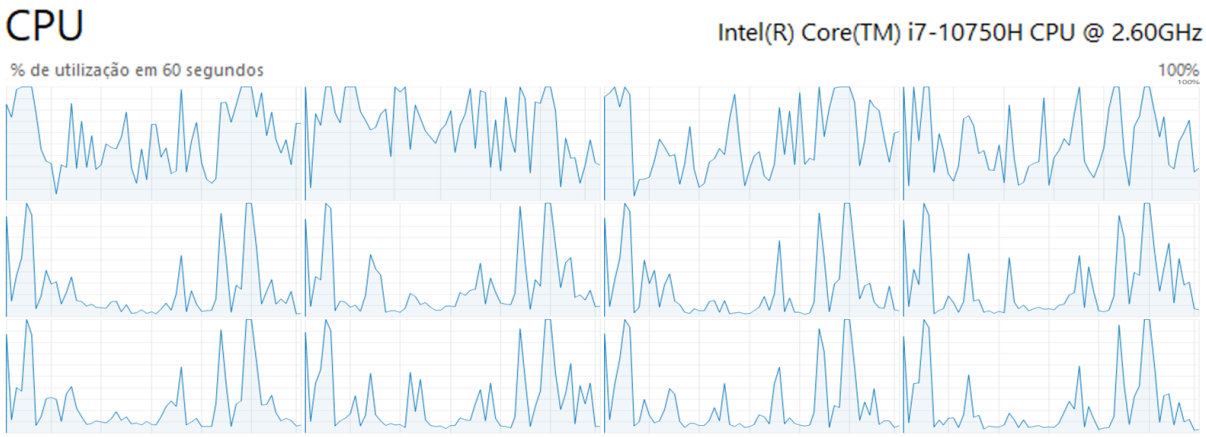

Figura 4.4: Uso de *CPU* no *Zeestar* – 10 Fluxos

Desempenho – *Zeestar*. Para mensurar a performance da solução *Zeestar*, foi configurado um *script Python* para realizar as seguintes operações:

1. Criação dos usuários;
2. Implantação dos contratos;
3. *Mint* de *tokens* para os usuários;
4. *placeOffer* (Usuário A);
5. *listOffers* (Usuário B);
6. *acceptOffer* (Usuário B).

O *script* foi elaborado de maneira a registrar os horários iniciais e o tempo gasto em cada operação. Para medir a *CPU* e a memória *RAM* utilizada durante os testes, foi utilizado o comando *top*, nativo do *Linux* executado no ambiente *Docker*. As operações iniciais (itens 1 a 3) foram executadas isoladamente, pois não fazem parte do fluxo normal da aplicação, que abrange apenas as transações 4 a 6, conforme descrito na Seção 4.6.3. A Tabela 4.6 apresenta os dados de desempenho obtidos durante a execução das operações para 1 e 10 fluxos da aplicação, correspondendo à realização sequencial das transações 4 a 6, repetindo-as de 1 a 10 vezes.

A execução de um único fluxo da aplicação levou aproximadamente 93 segundos, resultando em uma taxa inferior a duas transações por minuto, um desempenho inferior ao obtido com o *Anonymous Zether*. É importante ressaltar que, durante os testes, o *Zeestar* apresentou erros que impediram sua execução em um nó *Ethereum* padrão via *Ganache* (que, em tese, deveria ser suportado [179]) ou *Hardhat*. Para esclarecer essa limitação, foi feito contato com Samuel Steffen, um dos principais desenvolvedores da solução, que confirmou que esse problema já havia sido reportado [185] e que, de fato, a versão atual da solução não oferecia suporte a esses *backends*. Dessa forma, o único *backend* suportado foi o *eth-tester* [180], que permite apenas a execução sequencial dos comandos. Consequentemente, não houve melhora no desempenho ao realizar 10 fluxos, uma vez que não havia suporte para o envio de múltiplas transações em paralelo. De todo modo, dado que as transações no *Ethereum* também são processadas sequencialmente pelos nós, apenas as operações realizadas *off-chain* poderiam ser otimizadas, limitando os ganhos potenciais da paralelização no desempenho geral.

A Figura 4.4 apresenta o gráfico de utilização da *CPU* da máquina física durante os 60 segundos finais do fluxo de execução da aplicação utilizando o *Zeestar*. Os picos observados no gráfico evidenciam o uso intensivo de todos os núcleos da *CPU* em momentos específicos, especialmente durante as transações *placeOffer* (nos primeiros 10 segundos) e *acceptOffer* (nos 20 segundos finais). Esses picos refletem o processamento intensivo necessário para operações complexas, como a geração de *ZKP*s e o uso de criptografia homomórfica.

Observou-se um impacto significativo no desempenho da aplicação ao incorporar privacidade utilizando ambas as soluções testadas. Embora esse fator não represente um impedimento específico para a aplicação *Miles2Coins*, em outros cenários que exijam maior *throughput* ou aplicações que possuam um perfil mais intensivo em número de transações, a adoção dessas soluções poderia tornar-se inviável.

4.7.3 Nível de Privacidade Alcançado

Obter informações sobre transações no *Ethereum* ou em outras *blockchains* sem recursos adicionais de privacidade é simples. Os valores transacionados, juntamente com os endereços do remetente e destinatário, podem ser acessados diretamente dos dados brutos da transação. Com a *Application Binary Interface (ABI)* de um contrato, que define suas funções e estruturas de dados, é possível decodificar as transações para entender exatamente quais funções foram invocadas, os parâmetros usados e os resultados gerados. Geralmente, obter a *ABI* de determinado contrato é trivial, uma vez que muitos desenvolvedores publicam as *ABIs* para deixar transparentes as funcionalidades dos contratos [186]. Portanto, no contexto da aplicação *Miles2Coins* original, é possível determinar

com facilidade quais contas realizaram ou aceitaram ofertas, assim como as quantidades de *MilesTokens* trocadas por *SurrealTokens* e vice-versa.

Nível de Privacidade – *Anonymous Zether*. Uma camada adicional de privacidade é implementada ao se realizar a parte final do fluxo, que envolve transações *DvP*, através do *Anonymous Zether*. Essas transações tornam-se confidenciais, impossibilitando a determinação dos *tokens* trocados e das quantidades envolvidas. No entanto, como as transações *placeOffer* e *acceptOffer* permanecem desprotegidas, ainda é possível fazer inferências sobre a quantidade de *tokens* que cada conta possui, embora sem certeza sobre quais negociações foram efetivamente concluídas.

Além de tornar as transações *DvP* confidenciais, o *Anonymous Zether* inclui a funcionalidade de *"anonymity sets"*, conforme descrito na Seção 3.3.6, para ocultar as contas do remetente e do destinatário. Infelizmente, essa funcionalidade não pôde ser testada neste trabalho. O motivo é que, em sua implementação atual, o código da função *startDvP* do cliente *Anonymous Zether* [175] não prevê o recebimento do parâmetro *"decoys"*, utilizado para criar o conjunto de anonimato. Após modificar o código da função para aceitar esse parâmetro, surgiram outros problemas: a desserialização do vetor *"decoys"* não estava sendo realizada corretamente e, ao corrigir isso, ocorreu um erro na geração das *ZKP*s. Isso indica que a funcionalidade não está totalmente operacional na versão atual do *Anonymous Zether*. Consequentemente, decidiu-se não prosseguir com a correção desses problemas devido à premissa de não realizar alterações profundas na solução, conforme descrito na Seção 4.4. Embora o recurso de *"anonymity set"* não tenha sido utilizado, o uso de contas *Ethereum* de uso único para cada transação ainda adicionou algum nível de anonimato. De todo modo, os testes de desempenho apresentados aqui fornecem uma linha de base para avaliar o *overhead* mínimo da solução, uma vez que o uso de *"anonymity sets"* impactaria ainda mais o desempenho, como demonstrado no artigo do *Anonymous Zether* [111].

Nível de Privacidade – *Zeestar*. Ao utilizar o *Zeestar*, foi possível garantir a confidencialidade das transações de *"minting"* dos *tokens* e das operações de escrita e leitura dos saldos privados de *MilesToken* e *SurrealToken* dos usuários. A solução permitiria manter privados os valores das ofertas, mas essa abordagem tornaria inviável a listagem das ofertas disponíveis. Isso ocorre porque o usuário que está fazendo a consulta precisaria ter as informações reveladas apenas para si, mas o *Zeestar* não suporta essa reclassificação seletiva, conforme limitação descrita na Seção 4.6.3. Assim, não faria sentido criar ofertas de forma sigilosa para posteriormente publicá-las indistintamente. Portanto, os dados das transações *placeOffer* e *acceptOffer* foram mantidos públicos, de forma similar ao que

foi feito nos testes da solução *Anonymous Zether*. Dessa forma, apesar dos saldos serem privados, nesse caso, eles também poderiam ser estimados através da análise do histórico de transações da aplicação.

4.7.4 Atendimento aos Requisitos Mapeados

É amplamente reconhecido que não existe 100% de segurança em sistemas, e o mesmo princípio se aplica à privacidade. Embora tenham sido identificados desafios que limitaram parcialmente sua efetividade, as soluções *Anonymous Zether* e *Zeestar* foram capazes de aprimorar, em alguma medida, a privacidade da aplicação *Miles2Coins*. Diante disso, é essencial avaliar até que ponto a aplicação final – após a integração com cada solução – atende aos requisitos de privacidade e usabilidade da aplicação *Miles2Coins*, conforme mapeamento realizado na Seção 4.2.2:

- **R.M2C-1** – Confidencialidade e Anonimato: Conforme discutido na Seção 4.7.3, as integrações com *Anonymous Zether* e *Zeestar* não garantem privacidade total, pois determinadas transações, como *placeOffer* e *acceptOffer*, permanecem visíveis na *blockchain* sem mecanismos adicionais de criptografia ou anonimização. Assim, este requisito foi apenas parcialmente atendido.
- **R.M2C-2** – Descentralização e Independência de Terceiros: Esse critério foi determinante para a escolha das soluções adotadas. Ambas utilizam apenas *ZKPs* e criptografia homomórfica, sem delegação de computação para terceiros, garantindo o atendimento desse requisito.
- **R.M2C-3** – Compatibilidade com *Ethereum* e Modelo *Account-Based*: Esse requisito foi atendido, uma vez que ambas as soluções são compatíveis com *Ethereum* e adotam o modelo indicado. No entanto, vale destacar que, na versão atual, o *Zeestar* não pôde ser implementado diretamente em um nó *Ethereum*, sendo restrito a um ambiente de testes com o *backend eth-tester*, conforme detalhado na Seção 4.7.2.
- **R.M2C-4** – Desempenho e Custos Transacionais Moderados: Ambas as soluções impactaram significativamente o desempenho e os custos transacionais. Embora a degradação do desempenho não inviabilize o uso da aplicação, ela compromete a experiência do usuário. Já os altos custos tornam a implementação da aplicação final inviável em *Ethereum*. Portanto, este requisito não foi atendido.

A seguir, avalia-se o atendimento aos requisitos de privacidade para *DeFi*, conforme definidos na Seção 3.2.3:

- **R.DeFi-1** – *Privacy by Design*: Esse princípio orienta que a privacidade deve ser incorporada desde a concepção do sistema. Arquiteturalmente, a aplicação *Miles2Coins* foi projetada considerando esse aspecto, atendendo ao requisito nesse nível. No entanto, caso fosse implantada sem mitigar os desafios identificados, esse princípio não seria plenamente respeitado na prática.
- **R.DeFi-2** – Consentimento: Esse requisito depende da implementação da própria aplicação e não é diretamente influenciado pelas soluções de privacidade adotadas. Para garantir conformidade, a aplicação poderia exigir uma transação inicial na qual o usuário fornecesse consentimento explícito para o processamento de suas operações e movimentações de *tokens*.
- **R.DeFi-3** – Minimização da Coleta de Dados: Assim como no caso do item anterior, esse requisito está mais relacionado ao desenho da aplicação do que às soluções de privacidade utilizadas. Na *Miles2Coins*, apenas os dados estritamente necessários para registro e aceite de propostas são coletados, atendendo a esse critério.
- **R.DeFi-4** – Proteção dos Dados Pessoais e Confidencialidade das Transações: Esse requisito está diretamente relacionado ao **R.M2C-1** e, conforme detalhado anteriormente, foi atendido apenas parcialmente.
- **R.DeFi-5** – Controle por Parte do Usuário: Esse requisito representa um dos maiores desafios em aplicações baseadas em *blockchain*, pois a imutabilidade da rede impede a modificação ou exclusão de registros. Na aplicação *Miles2Coins*, mesmo que um usuário possa desativar uma oferta, ela permanecerá registrada na *blockchain*, ainda que com o status de inativa. Essa limitação decorre da própria natureza imutável da *blockchain* e não pode ser resolvida de forma simples por uma solução de privacidade. Portanto, esse requisito não foi atendido.
- **R.DeFi-6** – Auditoria e Responsabilização: Em blockchains públicas sem mecanismos de privacidade, a auditabilidade é garantida, pois todas as transações são verificáveis por qualquer nó. No entanto, ao conferir confidencialidade às transações *DvP*, saldos e operações de *minting*, a integração com as soluções de privacidade compromete esse requisito. Para garantir a responsabilização, a *Miles2Coins* poderia ser gerida por uma entidade responsável pela gestão e tratamento de incidentes de segurança e privacidade. Dessa forma, esse requisito foi atendido apenas parcialmente.

Mesmo após a integração com as soluções de privacidade, os requisitos **R.M2C-1**, **R.M2C-4**, **R.DeFi-4**, **R.DeFi-5** e **R.DeFi-6** permanecem parcialmente ou totalmente não atendidos. Um desafio recorrente observado é a dificuldade de equilibrar proteção

dos dados e auditabilidade, um dilema comum à maioria das soluções de privacidade analisadas. Técnicas como *ZKPs* e *TEEs* permitem validar transações sem expor seus dados, mas restringem o acesso a informações privadas apenas ao proprietário. Uma exceção a esse padrão é a solução *RPSC* [147], que permite auditoria das transações por um regulador previamente autorizado. No entanto, essa solução apresenta limitações ainda mais graves de desempenho, além de não possuir um repositório público disponível, conforme discutido nas seções 3.3.5 e 4.3.

Ressalta-se que os requisitos de privacidade da aplicação *Miles2Coins* podem ser considerados mais simples e menos restritivos do que aqueles de aplicações *DeFi* mais complexas, como as *exchanges* descentralizadas, e de *CBDCs*. Isso reforça a complexidade de garantir a conformidade de aplicações baseadas em *blockchains* públicas com normativos de proteção de dados, como a LGPD [40] e a GDPR [39]. Encontrar o equilíbrio entre privacidade e transparência continua sendo um dos maiores obstáculos nesse contexto.

4.8 Desafios para a Privacidade em *Blockchain*

Esta seção examina os principais desafios identificados no aprimoramento da privacidade na aplicação *Miles2Coins*. Além disso, discute a viabilidade da privacidade em *blockchains* públicas de forma mais ampla, dado que as dificuldades observadas neste estudo se estendem não apenas ao cenário de aplicações *DeFi*, mas a todo o ecossistema de *blockchain*.

4.8.1 Desafios Práticos no Aprimoramento da Privacidade

Escolha da Solução Adequada

O desafio inicial surge ao selecionar a solução de privacidade a ser utilizada, uma vez que cada abordagem possui seus próprios benefícios e limitações, que podem influenciar diretamente sua viabilidade de implementação. Sabe–se que não existe uma solução única que ofereça máxima privacidade para todos os tipos de aplicações sem compromissos relevantes.

Neste trabalho, o *Anonymous Zether* foi escolhido por sua abordagem equilibrada, combinando um desempenho moderado com um nível satisfatório de privacidade, sem demandar do desenvolvedor um conhecimento aprofundado em criptografia. Esperava-se que a solução não apenas proporcionasse transações confidenciais e mantivesse os saldos criptografados, mas também garantisse anonimato às partes envolvidas, o que não se verificou na prática. Ademais, o experimento revelou certas limitações que comprometem sua aplicabilidade em cenários reais. De maneira similar, a solução *Zeestar* também foi

selecionada devido à sua maior eficiência e ao nível mais elevado de privacidade esperado em comparação com as soluções exclusivamente baseadas em *ZKP*, incluindo o *zkay*, a partir do qual evoluiu. Porém, essa solução também apresentou limitações de funcionalidade que podem restringir seu uso prático. Além disso, ambas as soluções analisadas impactaram significativamente o desempenho e os custos transacionais.

Apesar de os obstáculos encontrados terem sido relacionados a características e limitações específicas das soluções testadas, é bastante provável que problemas similares seriam enfrentados caso outras soluções fossem avaliadas de forma prática.

Limitações e Dificuldades de Integração

Assim como outras soluções de privacidade, *Anonymous Zether* e *Zeestar* não são programas simples que podem ser instalados e utilizados diretamente, sem a necessidade de personalização. É necessário entender as funcionalidades e deficiências de cada solução, mas a documentação curta e incompleta das soluções não facilita essa tarefa. Os artigos sobre as soluções [111, 130] e os documentos disponíveis em seus repositórios [161, 162] não esclarecem claramente quais são as limitações atuais de cada uma.

A principal limitação do *Anonymous Zether* é a incapacidade de adicionar privacidade a qualquer função de um contrato inteligente, restringindo–se a transferências de *tokens* e outros casos de uso simples. Além disso, embora exista um artigo do autor da solução descrevendo os detalhes para a realização de transações *DvP* privadas [174], essa implementação não está disponível nos repositórios do *Anonymous Zether* [161, 175] – existe apenas no repositório do projeto *Drex* [176], conforme mencionado na Seção 4.5. Outro aspecto relevante é que o *Zether* não suporta todos os tipos de *tokens*, apenas os fungíveis – por exemplo, *ERC–20* [75] ou *ERC–1155* [78]. *Tokens* não fungíveis, como o *ERC–721* [77], não podem ser transferidos de forma privada pela solução, pois suas características únicas seriam perdidas em sua representação como *tokens Zether (ZTH)*. Como a aplicação *Miles2Coins* utiliza apenas *tokens* fungíveis, isso não representa um problema neste caso. Entretanto, para aplicações que precisam lidar com outros tipos de *tokens*, essa limitação impediria o uso do *Zether* para aprimorar a privacidade. Finalmente, a baixa modularidade do *Anonymous Zether* dificulta significativamente sua integração com a aplicação *Miles2Coins* de forma transparente para o usuário final.

Dentre as limitações do *Zeestar*, destacam–se a falta de suporte a chamadas de funções externas de outros contratos e a impossibilidade de revelar valores a usuários específicos, de forma seletiva. Enquanto a primeira deficiência impediu a implementação dos contratos dos *tokens* de forma independente da aplicação, a segunda resultou na sensível diminuição do nível de privacidade alcançado pela aplicação final, já que não foi possível manter o sigilo das ofertas. Por fim, o problema na implantação dos contratos desenvolvidos em um

nó *standalone Ethereum*, que foi confirmado com o desenvolvedor da solução, atualmente impossibilita seu uso em um ambiente de produção. Para resolver essa falha, segundo o desenvolvedor, seria necessário um grande esforço de desenvolvimento na correção do código do *Zeestar*.

Consequentemente, para uma integração completa da aplicação *Miles2Coins* com ambas as soluções de privacidade, seriam necessárias mudanças profundas tanto na aplicação quanto nas próprias soluções, o que resultaria em um sistema fortemente acoplado, com menor flexibilidade e limitada possibilidade de replicação em outros casos de uso.

Tokens Privados *vs. Tokens* Públicos

Como destacado por Benedikt Bünz, um dos autores do artigo original do *Zether*, para maximizar a privacidade, os usuários deveriam utilizar *tokens Zether* em todas as suas operações [187]. No entanto, isso é inviável, a menos que o *Zether* seja amplamente adotado. Assim, a integração do *Zether* apenas com a aplicação *Miles2Coins* e os *tokens* envolvidos apresenta uma fragilidade: como os *tokens* originais – *MilesToken* e *SurrealToken* – permaneceriam públicos tanto antes quanto no momento de financiamento dos contratos *Zether* associados, qualquer pessoa poderia inferir o saldo de uma conta específica. Por exemplo, se um usuário possuir 100.000 *MilesTokens* e quiser vendê-los por meio da aplicação *Miles2Coins* integrada ao *Anonymous Zether*, ele precisará primeiro financiar o contrato *ZSC*, tornando essa transação inicial pública na *blockchain*. Apenas as transações subsequentes permanecerão confidenciais. O mesmo problema ocorre na conversão dos *tokens Zether* de volta para os *tokens* originais, pois o valor trocado será visível publicamente. Essa limitação só poderia ser superada se os *tokens* fossem criados diretamente no *Zether*. No entanto, em uma situação real, isso exigiria que as companhias aéreas e o gestor do *SurrealToken* adotassem o *Zether* como base para seus *tokens*, algo pouco viável na prática.

No caso do *Zeestar*, ocorre uma situação oposta, mas que resulta em um problema similar. Caso os *tokens MilesToken* e *SurrealToken* fossem públicos, eles teriam contratos externos que proporcionariam funções como *minting* e transferência. Porém, essas funções não poderiam ser acionadas por aplicações implementadas via *Zeestar*, devido à limitação já citada. Assim, em um cenário de produção, as companhias aéreas precisariam manter o *MilesToken* diretamente vinculado à aplicação *Miles2Coins* desde a sua criação. O mesmo teria que ser feito com o *SurrealToken*, o que seria claramente inviável, caso ele fosse uma *stablecoin* real, utilizada em múltiplas aplicações.

Custos de Transação e Impacto no Desempenho

Apesar de ambas as soluções serem compatíveis com o *Ethereum*, o uso do *Anonymous Zether* ou do *Zeestar* para efetuar transações privadas nesta *blockchain* se mostrou potencialmente inviável, tanto neste como em outros casos de uso, devido aos altos custos de *gas*, conforme mostrado na Seção 4.7.1. Consequentemente, caso a aplicação *Miles2Coins* fosse integrada às soluções estudadas em um ambiente de produção, deveria ser considerada a utilização de *blockchains* alternativas, que operem com taxas de transação mais baixas, como a *Polygon PoS* [188] ou a *BNB Smart Chain* [57], ou que possuam taxas configuráveis, como as *blockchains* permissionadas – por exemplo, o *Hyperledger Fabric* [50].

O impacto no desempenho também foi elevado no caso das duas soluções avaliadas e, embora não tenha sido um fator completamente limitante para a aplicação *Miles2Coins*, em cenários com requisitos de desempenho mais rigorosos, essas soluções provavelmente não poderiam ser adotadas.

Escopo de Privacidade Reduzido

Como descrito na Seção 4.7.3, não foi possível incorporar privacidade às fases de criação e aceite de ofertas da aplicação *Miles2Coins*. Portanto, embora outros usuários da *blockchain* não possam ver a transação *DvP* (utilizando o *Anonymous Zether*) nem as transações de *minting* e os saldos dos usuários (utilizando o *Zeestar*), eles poderão visualizar as ofertas existentes e as que foram aceitas, possibilitando que qualquer usuário infira que as contas associadas a essas transações possuíam as quantidades de *tokens* indicadas. No caso do *Anonymous Zether*, como o recurso de *"anonymity sets"* não estava funcionando na versão atual da solução, também não foi possível prover anonimato para as partes envolvidas nas transações. Assim, conclui-se que o grau de privacidade adicional proporcionado pela integração das soluções *Anonymous Zether* e *Zeestar* com a aplicação *Miles2Coins* foi abaixo do esperado.

Ao considerar os desafios enfrentados no processo de integração e o nível limitado de privacidade alcançado na aplicação final, conclui-se que, em um contexto real, a adoção dessas soluções seria pouco viável para esse caso de uso específico. Mesmo que outras aplicações possam se beneficiar de forma mais eficaz das soluções analisadas, os custos operacionais no *Ethereum* e o elevado impacto no desempenho podem tornar sua implementação desafiadora na maioria dos cenários.

4.8.2 Viabilidade da Privacidade em *Blockchains* Públicas

Dada a privacidade limitada alcançada com a incorporação das soluções *Anonymous Zether* e *Zeestar*, é importante explorar como a privacidade poderia ser estendida para

outras transações na aplicação *Miles2Coins*. Essa discussão fornece uma visão sobre o paradoxo da privacidade em um ambiente público e distribuído das *blockchains*. A questão-chave é: *é possível oferecer privacidade robusta em blockchains públicas?*

Se a aplicação *Miles2Coins* seguisse uma abordagem tradicional, baseada na *web*, ela provavelmente estaria hospedada em um site protegido com criptografia *TLS*. Nesse contexto, as funções *placeOffer* e *acceptOffer* estariam acessíveis dentro da área autenticada da aplicação. Considere Alice e Bob, que desejam vender e comprar milhas aéreas, respectivamente. Após fazer *login*, Alice poderia registrar uma oferta de venda, acionando a reserva automática de suas milhas por meio de uma integração do sistema com *APIs* de companhias aéreas parceiras. Essa e outras ofertas ativas poderiam então ser visualizadas por outros usuários, mas a identidade do vendedor permaneceria oculta. Somente os *IDs* das ofertas, a quantidade de milhas e o preço seriam visíveis. Quando Bob aceitasse uma oferta de venda, ele seria redirecionado para uma página de pagamento segura, e a aplicação gerenciaria a transação, garantindo que Alice recebesse o pagamento, e Bob, as milhas. Nesse modelo centralizado, assumindo que ambos os usuários confiem na plataforma *Miles2Coins* e em seu operador, a privacidade dos usuários seria preservada, pois os detalhes da transação estariam acessíveis apenas para as partes envolvidas e para a própria plataforma. Os dados das ofertas também poderiam ser criptografados diretamente no banco de dados, impedindo que até mesmo administradores internos acessassem detalhes sensíveis das transações.

Na versão da aplicação *Miles2Coins* baseada em *blockchain*, a implementação de criptografia nos dados das ofertas apresenta desafios significativos. Permitir que os usuários acessem informações criptografadas das ofertas sem expor dados sensíveis exigiria mecanismos adicionais de comunicação. Se essa comunicação for realizada *on-chain*, ela poderia ser exposta ou se tornar extremamente custosa. Por exemplo, se Alice criptografar os detalhes de sua oferta com uma chave simétrica, como Bob acessaria o conteúdo? Se Alice compartilhar a chave de criptografia por meio de uma transação *on-chain*, a chave ficaria visível para todos os nós da rede, comprometendo a confidencialidade. Da mesma forma, a criptografia baseada em contratos inteligentes falha porque a lógica dos contratos é executada de forma transparente, expondo o texto em claro durante a decifragem. O uso de *ZKP*s para registro de ofertas, onde Alice prova que a oferta é válida e que ela possui os *tokens* necessários, tornaria a listagem de ofertas impraticável, pois os compradores interessados não teriam acesso aos dados reais das ofertas. Além disso, se apenas os vendedores tivessem acesso aos seus próprios dados de oferta, os compradores precisariam se comunicar diretamente com todos os vendedores *off-chain* para obter informações sobre as ofertas disponíveis. De maneira similar, a utilização de assinaturas em anel, embora permita ofuscar pseudônimos, exige grandes conjuntos de anonimato para evitar a desano-

nimização probabilística, adicionando complexidade operacional sem garantir privacidade absoluta. Cada uma dessas abordagens introduz obstáculos significativos que não podem ser resolvidos sem comprometer a confidencialidade dos dados, degradar a usabilidade ou aumentar substancialmente a complexidade e os custos transacionais.

Este estudo demonstra que alcançar privacidade robusta em uma *blockchain* pública exclusivamente por meio de mecanismos *on-chain* é altamente desafiador em muitos cenários. Embora determinados casos de uso, como transferências de criptomoedas [16, 15], leilões [110] e votação [189], demonstrem sucessos específicos em privacidade, não há uma solução "bala de prata" que funcione para qualquer aplicação sem limitações ou compromissos relevantes. Esses compromissos incluem:

1. Interação *off-chain*: A maioria das técnicas de privacidade não podem ser aplicadas inteiramente *on-chain* sem expor dados intermediários para todos os nós. Soluções como *ZKP*s frequentemente exigem computação no lado do cliente e gerenciamento de chaves *off-chain*. Da mesma forma, soluções baseadas em delegação para *TEE* dependem de ambientes *off-chain* para lidar com os cálculos privados.

2. Comprometimento parcial da privacidade: Se o objetivo for manter todos os detalhes da transação, saldos dos usuários e lógica de negócios ocultos, soluções puramente *on-chain* enfrentam limitações inerentes, como mostrado neste estudo. Por outro lado, soluções baseadas em delegação exigiriam que os usuários confiassem em *TEEs* ou *"managers"*, comprometendo parcialmente sua privacidade.

3. Centralização: Técnicas como *SMPC* e abordagens baseadas em delegação dependem de um subconjunto de nós ou de uma infraestrutura especializada para processar as transações com segurança. Além de reduzir a privacidade, isso introduz certo nível de centralização, o que conflita com a essência totalmente descentralizada das *blockchains* públicas.

Portanto, embora o aprimoramento da privacidade seja possível por meio de diversas técnicas, as *blockchains* públicas atualmente não possuem métodos diretos e puramente *on-chain* para proporcionar níveis robustos de privacidade. Assim, os desenvolvedores precisam combinar múltiplas técnicas, aceitar alguma exposição parcial de dados ou incorporar componentes *off-chain* para proteger informações sensíveis. O objetivo deve ser encontrar uma abordagem que proporcione um melhor equilíbrio entre privacidade, desempenho e usabilidade, o que, por si só, já representa um desafio significativo.

4.9 Panorama Atual da Privacidade em *Blockchain*

Nesta seção, apresenta-se um panorama atualizado da privacidade no ecossistema de *blockchain*. Destaca-se que os números apresentados aqui foram obtidos em 28 de fevereiro de 2025. O ecossistema de *blockchain* tem se mantido em constante evolução ao longo dos últimos anos. No intervalo de 2023 a 2025, novas *blockchains* surgiram [58, 190], importantes atualizações em redes já estabelecidas foram implementadas [191, 192] e diversas soluções de privacidade foram propostas [147, 151, 138, 22]. No cenário de *DeFi*, o *Ethereum* continua como a *blockchain* de camada 1 mais popular, com aproximadamente US$ 51 bilhões de *Total Value Locked (TVL)* – valor total depositado em contratos inteligentes, indicando a liquidez total da rede. Isso equivale a cerca de 52% do *TVL* total nas centenas de *blockchains* existentes [193]. Paralelamente, as soluções de camada 2 vêm ganhando cada vez mais espaço, impulsionadas sobretudo pela escalabilidade superior e pela redução dos custos de transação. Nesse contexto, destacam-se as *blockchains Base* [58], criada em 2024 pela *DEX Coinbase*, com cerca de US$ 2,8 bilhões de *TVL*, e *Arbitrum* [194], com US$ 2,6 bilhões de TVL [193].

No entanto, apesar da tecnologia de *blockchain* avançar rapidamente, as preocupações com privacidade nesse ecossistema – em especial no contexto de *DeFi* – permanecem. Isso ocorre porque, até o momento, nenhuma das *blockchains* de maior adesão, seja de camada 1 ou 2, fornece suporte padrão a transações privadas. É importante ressaltar que o *Arbitrum*, apesar de ter surgido a partir da solução homônima estudada no Capítulo 3 deste trabalho, não manteve nenhuma das funcionalidades de privacidade previstas no seu artigo acadêmico original, focando apenas em escalabilidade e redução de custos. Dentre as soluções aqui estudadas, apenas a *Secret Network* [135] foi implementada em produção e manteve o foco em privacidade. Porém, essa *blockchain* jamais teve grande adoção, e hoje possui um *TVL* de apenas US$ 8,8 milhões, ou seja, apenas 0,01% do mercado *DeFi* [195].

Atualmente, existem outros projetos em andamento que visam incorporar privacidade em *blockchain* que merecem destaque. Alguns deles já estão em produção, outros ainda em fase de concepção para se tornarem operacionais:

- *Manta Network* [196]: Focada em escalabilidade, redução de custos e privacidade, a *Manta Network* é uma *blockchain* modular composta por duas camadas interoperáveis. *Manta Atlantic* é uma *blockchain* de camada 1 no ecossistema *Polkadot* [197], com suporte a endereços e *tokens* privados, além de outros recursos baseados em *ZKP*s. *Manta Pacific* [18] atua como uma solução de camada 2 para *Ethereum*, utilizando *zkEVM* para permitir a criação de aplicações privadas utilizando *ZKP*s diretamente na *EVM*. Desde seu lançamento em setembro de 2023, a adoção da

Manta Network ainda está em estágio inicial, com um *TVL* de pouco mais de US$ 45 milhões [198].

- *Oasis Protocol* [199]: Visando versatilidade, escalabilidade e privacidade, a arquitetura do *Oasis Protocol* consiste em uma camada de consenso e outra de computação, que permite a operação de múltiplos ambientes de execução paralelos, denominados *ParaTimes*, para execução de aplicações especializadas. Um desses *ParaTimes* é o *Oasis Sapphire* [200], que fornece uma *EVM* confidencial baseada em *TEE* para executar contratos inteligentes privados, assegurando que os dados sensíveis permaneçam ocultos mesmo dos validadores da rede. Apesar de estar em operação desde 2020, o *Oasis Sapphire* não obteve grande adesão: o *TVL* atual das aplicações executando sobre o *Sapphire* é de cerca de US$1,02 milhões [201].

- *Midnight* [202]: O projeto *Midnight* foi desenvolvido com base no protocolo *Kachina*, estudado na Seção 3.3.5 deste trabalho. A solução oferece proteção de dados programável por meio de contratos inteligentes escritos em uma linguagem baseada em *TypeScript*, que inclui primitivas adicionais para marcar dados como privados ou públicos, sem exigir que os desenvolvedores possuam conhecimento avançado em criptografia. No *Midnight*, os dados privados são mantidos pelos usuários *off-chain*, e sua corretude é atestada por meio de *ZKP*s. A solução visa facilitar a implementação de casos de uso como identidade digital, tokenização de ativos e votação segura. Atualmente, o *Midnight* encontra-se em estágio de desenvolvimento, com uma *"testnet"* disponível para que desenvolvedores explorem a plataforma e forneçam feedbacks para futuros aprimoramentos antes da implementação da *"mainnet"*, cuja data de lançamento ainda não foi definida.

- *Aztec* [19]: Projetado como uma solução de camada 2 para *Ethereum*, o *Aztec* também se baseia em *ZKP*s para prover privacidade. Iniciado em 2018, o *Aztec* implementou sua *"devnet"* apenas em agosto de 2024. Atualmente, o projeto tem se concentrado no desenvolvimento de novas funcionalidades e no aprimoramento de sua linguagem *Noir*. De forma similar à linguagem *zkay* estudada neste trabalho, a *Noir* foi criada para simplificar o desenvolvimento de contratos inteligentes, permitindo combinar dados públicos e privados para proporcionar privacidade seletiva sem exigir conhecimentos avançados em criptografia por parte dos desenvolvedores.

- *Polygon Nightfall* [203]: Desenvolvida em parceria com a *Ernst & Young (EY)*, a *Polygon Nightfall* foi uma solução de camada 2 que combinava *Optimistic Rollups* [204] e *ZKP*s para a realização eficiente de transações confidenciais. O público-alvo do projeto *Nightfall* eram empresas que desejavam utilizar a *blockchain* em suas aplicações, mas que necessitavam de privacidade. Após o lançamento da versão beta

da *"mainnet"* em 2022, não houve novas atualizações sobre o projeto, e a solução atualmente não está listada no site principal da *Polygon*, indicando que deve ter sido descontinuada.

Outra abordagem que tem sido adotada para casos de uso com requisitos de privacidade, principalmente por empresas e instituições públicas, é a utilização de *blockchains* permissionadas com recursos de privacidade, onde se destacam os exemplos a seguir:

- *Hyperledger Fabric* [205]: O projeto *Hyperledger* foi lançado em dezembro de 2015 pela *Linux Foundation*, com o apoio de grandes empresas de tecnologia como *IBM*, *Intel*, *Red Hat* e *Consensys*. Dentro desse ecossistema, foi desenvolvido o *Hyperledger Fabric*, uma *blockchain* permissionada voltada para aplicações empresariais. Sua arquitetura modular oferece flexibilidade e suporte a diversos mecanismos de consenso, bem como a integração com múltiplas linguagens de programação para contratos inteligentes. Um dos diferenciais do *Fabric* é a capacidade de criar canais privados *("private channels")*, permitindo que dados sejam compartilhados exclusivamente entre participantes autorizados da rede. Além disso, ele oferece recursos adicionais de privacidade, como coleções privadas de dados, *ACLs* e suporte à criptografia dos dados antes de seu registro no *ledger* [206].
- *Hyperledger Besu* [207]: Também desenvolvido sob o guarda-chuva do projeto *Hyperledger*, o *Besu* é uma implementação de *Ethereum* que suporta redes públicas ou permissionadas e pode ser utilizado por consórcios que demandam governança estruturada e controle aprimorado. No modelo permissionado, essa *blockchain* suporta o mecanismo de consenso *Proof of Authority (PoA)*, no qual apenas validadores pré-selecionados têm a responsabilidade de validar os blocos da cadeia. O principal recurso de privacidade do *Besu* são os grupos privados *("privacy groups")*, que permitem que as transações sejam armazenadas apenas nos nós participantes de cada grupo, enquanto seus *hashes* são publicados no *ledger* público para garantir integridade e verificabilidade.
- *R3 Corda* [208]: Apesar de ter sido inicialmente projetado para o setor financeiro, o *Corda* pode ser utilizado em outros setores. Essa *blockchain* permissionada se diferencia pelo seu modelo de transações *P2P*, onde os dados são compartilhados apenas entre as partes envolvidas, ao invés de serem replicados em um *ledger* global. A validação das transações é feita por nós notários *("notary nodes")*, sem acesso aos dados subjacentes. Assim como o *Hyperledger Fabric*, o *Corda* também oferece controle granular de acesso, permitindo que apenas informações estritamente necessárias sejam compartilhadas.

- *Quorum* [209]: Desenvolvido inicialmente pela *JP Morgan*, adquirido posteriormente pela *Consensys* e agora mantido pela *Kaleido*, o *Quorum* é uma implementação de *Ethereum* voltada para redes permissionadas, com foco em aplicações empresariais que exigem privacidade e desempenho. O *Quorum* utiliza o gerenciador de privacidade *Tessera* [210] para realizar transações protegidas por criptografia de ponta a ponta, garantindo que os dados sejam acessíveis apenas aos participantes autorizados, enquanto mantém a integridade e verificabilidade no *ledger*.

4.9.1 Discussão sobre o Panorama

A partir da análise do panorama atual da privacidade em *blockchain*, conclui-se que, apesar da constante evolução tecnológica, o cenário ainda apresenta grandes desafios no que diz respeito à privacidade. Os altos custos transacionais e o baixo desempenho do *Ethereum* impulsionaram o surgimento de um mercado significativo de *blockchains* de camada 2 voltadas para resolver essas limitações. No entanto, iniciativas com foco real em privacidade no âmbito de *blockchains* públicas ainda são escassas, e os projetos existentes encontram-se em estágios iniciais ou enfrentam baixa adesão. Apesar dos conhecidos desafios de privacidade, as aplicações *DeFi* continuam em expansão: das 24,6 milhões de carteiras ativas diárias – *Unique Active Wallets (UAW)* – registradas ao final de 2024, 32% estavam associadas a aplicações *DeFi* [211].

Embora as *blockchains* permissionadas sejam alvo de críticas por não preservarem a característica fundamental de descentralização das *blockchains* públicas, elas se destacam como uma alternativa viável para casos de uso corporativos e institucionais que demandam conformidade com regulamentações de proteção de dados, como a LGPD e a GDPR [212]. Essas redes permitem maior controle e governança, além de oferecerem mecanismos adicionais de privacidade, como canais privados com criptografia de ponta a ponta. Contudo, no contexto das *blockchains* públicas, as soluções existentes ainda apresentam limitações que restringem sua adoção em larga escala.

4.10 Considerações Finais do Capítulo

Este capítulo apresentou a avaliação empírica das soluções *Anonymous Zether* e *Zeestar* por meio de sua integração com a aplicação *Miles2Coins*. Os resultados demonstraram o aprimoramento da privacidade proporcionado pelas soluções testadas, mas também evidenciaram desafios técnicos que impactariam sua adoção em cenários reais. Os obstáculos encontrados refletem dificuldades mais amplas enfrentadas pelo ecossistema de *blockchain*. A análise do panorama atual reforçou a necessidade de soluções que ofereçam um balanço entre privacidade, transparência, eficiência e usabilidade.

Capítulo 5

Conclusões

Neste trabalho de mestrado, foi conduzido um estudo abrangente e atualizado sobre as soluções de privacidade para *blockchain*, abordando suas estratégias, vantagens e limitações. No total, 20 soluções foram examinadas por meio de uma análise conceitual e de uma comparação direta, considerando critérios como o nível de privacidade oferecido e o impacto no desempenho. Além disso, a integração das soluções *Anonymous Zether* e *Zeestar* com a aplicação *Miles2Coins* possibilitou testar essas tecnologias na prática, avaliando sua eficácia, custos transacionais e desempenho.

Os desafios encontrados na integração dessas soluções podem ser extrapolados para outros casos de uso em cenários de *DeFi* e aplicações descentralizadas em geral, evidenciando a complexidade de se alcançar privacidade robusta em *blockchains* públicas. Essas limitações reforçam a necessidade de avanços contínuos em pesquisas e inovações nas camadas de protocolo voltadas à privacidade. Até que tais avanços se concretizem, os desenvolvedores enfrentarão um cenário de difíceis concessões, no qual será preciso equilibrar requisitos de privacidade com aspectos como transparência, usabilidade e desempenho.

O panorama atual da privacidade em *blockchain* revela que, apesar dos avanços constantes, essa questão ainda ocupa um papel secundário. Enquanto novas *blockchains* continuam a ser desenvolvidas e inúmeras aplicações descentralizadas ganham destaque, as soluções voltadas à privacidade ainda são raras e, na maioria dos casos, estão em estágios iniciais de desenvolvimento. Como resultado, informações sensíveis dos usuários, incluindo dados pessoais, de saúde ou financeiros, continuam sendo expostas diariamente nas *blockchains* públicas. Essa realidade evidencia que a conformidade com regulamentações de privacidade, como a LGPD e a GDPR, está longe de ser alcançada nesse ecossistema.

É crucial que a comunidade — composta por desenvolvedores, usuários, reguladores e outros atores — intensifique os esforços para tratar a privacidade como um requisito fundamental no contexto de *blockchain*, promovendo o desenvolvimento de novas técnicas e soluções robustas, integradas e amplamente suportadas pelas plataformas. Embora o

tema esteja em constante evolução e as questões de privacidade ainda não tenham sido plenamente resolvidas, espera-se que este trabalho contribua para evidenciar o estágio atual das técnicas e soluções voltadas à privacidade, bem como os desafios remanescentes nesse cenário. Com o avanço contínuo das técnicas criptográficas e de outras abordagens para privacidade, aliado ao amadurecimento do ecossistema de *blockchain*, espera-se que soluções de privacidade cada vez mais robustas e eficientes sejam desenvolvidas. Somente assim será possível viabilizar um futuro em que aplicações descentralizadas conciliem inovação, segurança e proteção efetiva da privacidade dos usuários.

Referências

[1] Nakamoto, Satoshi: *Bitcoin: A peer-to-peer electronic cash system*. Bitcoin, 2008. `https://bitcoin.org/bitcoin.pdf`. xi, 1, 10, 11

[2] Buterin, Vitalik: *A next-generation smart contract and decentralized application platform*. Ethereum project white paper, 2014. 1, 2, 11, 15

[3] Wood, Gavin *et al.*: *Ethereum: A secure decentralised generalised transaction ledger*. Ethereum project yellow paper, 2014. 1, 2, 11, 77

[4] Ivanov, Nikolay, Chenning Li, Qiben Yan, Zhiyuan Sun, Zhichao Cao e Xiapu Luo: *Security Defense For Smart Contracts: A Comprehensive Survey*. ACM Computing Surveys, 55(14s):1–37, 2023. 1, 11, 22

[5] Sunny, Farhana Akter, Petr Hajek, Michal Munk, Mohammad Zoynul Abedin, Md. Shahriare Satu, Md. Iftekharul Alam Efat e Md. Jahidul Islam: *A Systematic Review of Blockchain Applications*. IEEE Access, páginas 59155–59177, 2022. 1

[6] Kosba, Ahmed, Andrew Miller, Elaine Shi, Zikai Wen e Charalampos Papamanthou: *Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts*. Em *2016 IEEE Symposium on Security and Privacy (SP)*, páginas 839–858. IEEE, 2016. 1, 3, 25, 40, 46, 51, 53, 54, 60

[7] Bernal Bernabe, Jorge, Jose Luis Canovas, Jose L. Hernandez-Ramos, Rafael Torres Moreno e Antonio Skarmeta: *Privacy-Preserving Solutions for Blockchain: Review and Challenges*. IEEE Access, 7:164908–164940, 2019. 1, 3, 18, 19, 23, 59

[8] Biryukov, Alex e Sergei Tikhomirov: *Deanonymization and Linkability of Cryptocurrency Transactions Based on Network Analysis*. Em *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, páginas 172–184. IEEE, 2019. 1, 18, 59

[9] Androulaki, Elli, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer e Srdjan Capkun: *Evaluating User Privacy in Bitcoin*. Em *Financial Cryptography and Data Security*, volume 7859, páginas 34–51. Springer Berlin Heidelberg, 2013. 1, 18, 59

[10] Jensen, Johannes Rude, Victor Von Wachter e Omri Ross: *An introduction to decentralized finance (defi)*. Complex Systems Informatics and Modeling Quarterly, 26:46–54, 2021. 2

[11] Atlantic Council: *Central bank digital currency tracker - alantic council*, 2024. `https://www.atlanticcouncil.org/cbdctracker/`. 2

[12] Jahan, Sarwat, Elena Loukoianova, Evan Papageorgiou, Natasha X Che, Ankita Goel, Mike Li, Umang Rawat, Yong Sarah Zhou e Ankita Goel: *Towards central bank digital currencies in asia and the pacific: Results of a regional survey*, 2022. https://www.imf.org/en/Publications/fintech-notes/Issues/2022/09/27/Towards-Central-Bank-Digital-Currencies-in-Asia-and-the-Pacific-Results-of-a-Regional-Survey-523914. 2

[13] Banco Central do Brasil: *Drex - Real Digital*. https://www.bcb.gov.br/estabilidadefinanceira/drex. 2, 63

[14] Ben Sasson, Eli, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer e Madars Virza: *Zerocash: Decentralized Anonymous Payments from Bitcoin*. Em *2014 IEEE Symposium on Security and Privacy*, páginas 459–474. IEEE, 2014. 3, 37, 40

[15] Zcash: *Zcash: Privacy-protecting digital currency*, 2024. https://z.cash/. 3, 37, 90

[16] Noether, Shen: *Ring Confidential Transactions*. Cryptology ePrint Archive, 2015. 3, 22, 37, 90

[17] Valenta, Martin e Philipp Sandner: *Comparison of Ethereum, Hyperledger Fabric and Corda*. Frankfurt School Blockchain Center, 2017. 3

[18] Manta Network: *Manta Pacific*, 2025. https://pacific.manta.network/. 3, 16, 91

[19] Aztec Labs: *Aztec | The Privacy-first Layer 2 on Ethereum*, 2025. https://aztec.network/. 3, 92

[20] Belchior, Rafael, André Vasconcelos, Sérgio Guerreiro e Miguel Correia: *A Survey on Blockchain Interoperability: Past, Present, and Future Trends*. ACM Computing Surveys, 54(8):1–41, 2022. 3

[21] Belles-Munoz, Marta, Jordi Baylina, Vanesa Daza e Jose L. Munoz-Tapia: *New Privacy Practices for Blockchain Software*. IEEE Software, 39(3):43–49, 2022. 3, 19

[22] Dahl, Morten, Clément Danjou, Daniel Demmler, Tore Frederiksen, Petar Ivanov, Marc Joye e Dragos Rotaru: *fhEVM: Confidential EVM Smart Contracts using Fully Homomorphic Encryption*, 2023. https://github.com/zama-ai/fhevm/blob/main/fhevm-whitepaper.pdf. 3, 49, 51, 53, 54, 61, 91

[23] Zhao, Shijun, Qianying Zhang, Yu Qin, Wei Feng e Dengguo Feng: *SecTEE: A Software-based Approach to Secure Enclave Architecture Using TEE*. Em *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, páginas 1723–1740. ACM, 2019. 3, 20

[24] Zhang, Rui, Rui Xue e Ling Liu: *Security and Privacy on Blockchain*. ACM Computing Surveys, 52(3):1–34, julho 2019. 3, 18, 21, 22

[25] Maxwell, Gregory: *Coinjoin: Bitcoin privacy for the real world*, 2013. https://bitcointalk.org/?topic=279249. 3, 21

[26] ConsenSys, Inc.: *Thoughts on utxos by vitalik buterin*. https://medium.com/@Consensys/thoughts-on-utxo-by-vitalik-buterin-2bb782c67e53. 3, 14, 60

[27] Clifford, Jordan: *Intro to blockchain: Utxo vs account based | by jordan clifford | medium*, 2019. https://jcliff.medium.com/intro-to-blockchain-utxo-vs-account-based-89b9a01cd4f5. 3, 14, 60

[28] Almashaqbeh, Ghada e Ravital Solomon: *Sok: Privacy-preserving computing in the blockchain era*. Em *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, páginas 124–139. IEEE, 2022. 3, 20, 21, 23, 62

[29] Tanenbaum, Andrew S. e Maarten van Steen: *Distributed systems: principles and paradigms*. Pearson, Prentice Hall, second edition edição, 2007. 6, 7

[30] Coulouris, George F, Jean Dollimore e Tim Kindberg: *Distributed systems: concepts and design*. Pearson Education, fifth edition edição, 2012. 6, 7

[31] Avizienis, A., J. C. Laprie, B. Randell e C. Landwehr: *Basic concepts and taxonomy of dependable and secure computing*. IEEE Transactions on Dependable and Secure Computing, 1(1):11–33, 2004. 7

[32] National Institute of Standards and Technology: *Security glossary - dsrc - nist*. https://csrc.nist.gov/glossary/term/security. 8

[33] Gabinete de Segurança Institucional da Presidência da República: *Glossário de Segurança da Informação - GSI/PR*. https://www.gov.br/gsi/pt-br/ssic/glossario-de-seguranca-da-informacao-1. 8

[34] Nogueira, Michele: *Afinal, o que é cibersegurança?* https://horizontes.sbc.org.br/index.php/2023/07/afinal-o-que-e-ciberseguranca/. 8

[35] Pratt, Mary K.: *Emerging cyber threats in 2023 from ai to quantum to data poisoning*. https://www.csoonline.com/article/651125/emerging-cyber-threats-in-2023-from-ai-to-quantum-to-data-poisoning.html. 8

[36] Dixon, William e Nicole Eagan: *3 ways ai will change the nature of cyber attacks*. https://www.weforum.org/agenda/2019/06/ai-is-powering-a-new-generation-of-cyberattack-its-also-our-best-defence/. 8

[37] Guembe, Blessing, Ambrose Azeta, Sanjay Misra, Victor Chukwudi Osamor, Luis Fernandez-Sanz e Vera Pospelova: *The Emerging Threat of Ai-driven Cyber Attacks: A Review*. Applied Artificial Intelligence, 36(1), 2022. 8

[38] Godoy, Claudio Luiz Bueno de: *Privacidade*. https://enciclopediajuridica.pucsp.br/verbete/474/edicao-1/privacidade. 9

[39] lex.europa.eu eur: *Regulation - 2016/679 - EN - gdpr - EUR-Lex*. https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng. 9, 23, 29, 31, 85

[40] Planalto.gov.br: *L13709*. https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. 9, 19, 29, 30, 34, 85

[41] National Institute of Standards and Technology: *Cybersecurity framework*. https://www.nist.gov/cyberframework. 9, 29, 32

[42] National Institute of Standards and Technology: *Security and Privacy Controls for Information Systems and Organizations*. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf. 9, 32

[43] National Institute of Standards and Technology: *Privacy Framework | NIST*. https://www.nist.gov/privacy-framework/privacy-framework. 9, 29, 32

[44] Center for Internet Security: *CIS Critical Security Controls*. https://www.cisecurity.org/controls. 9, 29, 32

[45] Center for Internet Security: *CIS Controls v8 Privacy Companion Guide*. https://www.cisecurity.org/insights/white-papers/cis-controls-v8-privacy-companion-guide. 9, 29, 32

[46] Portal gov.br: *Framework de Privacidade e Segurança da Informação*. https://www.gov.br/governodigital/pt-br/privacidade-e-seguranca/framework-guias-e-modelos. 10, 32

[47] Machado, Tereza Cristina: *A segurança dos Dados no Privacy by Design e Privacy by Default*. https://www.jusbrasil.com.br/artigos/a-seguranca-dos-dados-no-privacy-by-design-e-privacy-by-default/1671594853. 10

[48] Chen, Yourong, Hao Chen, Yang Zhang, Meng Han, Madhuri Siddula e Zhipeng Cai: *A survey on blockchain systems: Attacks, defenses, and privacy preservation*. High-Confidence Computing, 2(2):100048, 2022. 10

[49] Modani, Mahak, Satyam Patidar e Sushma Verma: *A Methodological Review on Applications of Blockchain Technology and its Limitations*. Information Management and Computer Science, 4(1):01–05, 2021. 10

[50] Cachin, Christian *et al.*: *Architecture of the hyperledger blockchain fabric*. Em *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, 2016. 11, 88

[51] Dragonchain Foundation, Inc.: *Dragonchain foundation*. https://dragonchain.org/. 11

[52] Solidity Team: *Solidity programming language*. https://soliditylang.org/. 11, 15

[53] DAppRadar: *2024 Overview: The Most Popular Projects on DappRadar*. https://dappradar.com/blog/2024-overview-the-most-popular-projects-on-dappradar. 12

[54] CPQD: *Solução de identidade digital descentralizada do CPQD pode ser integrada também à rede Ethereum*. https://www.cpqd.com.br/noticias/solucao-de-identidade-digital-descentralizada-do-cpqd-pode-ser-integrada-tambem-a-rede-ethereum/. 12

[55] Defillama: *Defillama - DeFi Dashboard*, 2025. https://defillama.com/. 12

[56] Solana: *Web3 infrastructure for everyone | solana*, 2024. https://solana.com/. 12, 16

[57] bnbchain.org: *BNB Smart Chain (BSC): Bring Smart Contracts to BNB Chain*, 2024. https://www.bnbchain.org/en/bnb-smart-chain. 12, 88

[58] base.org: *Base*, 2024. https://www.base.org/. 12, 16, 91

[59] Alchemy Insights Inc.: *The 12 most important web3 programming languages (2023)*. https://www.alchemy.com/overviews/web3-programming-languages. 12

[60] Richards, Tony: *The Future of Payments: Cryptocurrencies, Stablecoins or Central Bank Digital Currencies?*, 2021. hhttps://www.rba.gov.au/speeches/2021/pdf/sp-so-2021-11-18.pdf. 12, 13

[61] Bitcoin: *Bitcoin - open source p2p money*, 2024. https://bitcoin.org/en/. 12, 16

[62] Ethereum Foundation: *Ethereum.org*. https://ethereum.org/. 12, 16

[63] Cardano: *Cardano | what is ada*, 2024. https://cardano.org/what-is-ada/. 12

[64] Binance: *What is bnb? | binance academy*, 2018. https://academy.binance.com/en/articles/what-is-bnb. 12

[65] Bolt, Wilko, Vera Lubbersen e Peter Wierts: *Getting the Balance Right: Crypto, Stablecoin and CBDC*. De Nederlandsche Bank Working Paper, 2022. 12

[66] Tether: *Tether*, 2024. https://tether.to/en/. 12

[67] Circle Internet Financial: *Usdc | digital dollars backed 1:1 with usd | circle*, 2024. https://www.circle.com/en/usdc. 12

[68] Paxos Trust Company: *Pax gold - paxos*, 2024. https://paxos.com/paxgold/. 12

[69] eNaira.gov.ng: *enaira - Same Naira, more possibilities*, 2024. https://enaira.gov.ng/. 13

[70] SandDollar: *Digital Bahamian Dollar Sand Dollar*, 2024. https://www.sanddollar.bs/. 13

[71] Adrian, Tobias, Dong He, Tommaso Mancini-Griffoli e Tao Sun: *Central bank digital currency development enters the next phase*, 2023. https://www.imf.org/en/Blogs/Articles/2023/11/20/central-bank-digital-currency-development-enters-the-next-phase. 13

[72] Lisi, Andrea, Andrea De Salve, Paolo Mori, Laura Ricci e Samuel Fabrizi: *Rewarding reviews with tokens: An Ethereum-based approach*. Future Generation Computer Systems, 120:36–54, 2021. 13

[73] Musharraf, Mohammad: *Ethereum Token Approvals Explained*, 2024. https://www.ledger.com/academy/ethereum-token-approvals-explained. 13

[74] Ali, Muddasar e Sikha Bagui: *Introduction to NFTs: The Future of Digital Collectibles*. International Journal of Advanced Computer Science and Applications, 12(10), 2021. 13

[75] Vogelsteller, Fabian e Vitalik Buterin: *Erc-20: Token standard*, 2015. https://eips.ethereum.org/EIPS/eip-20. 13, 59, 86

[76] Idelberger, Florian e Péter Mezei: *Non-fungible tokens*. Internet Policy Review, 11(2), 2022. 13

[77] Entriken, William, Dieter Shirley, Jacob Evans e Nastassia Sachs: *Erc-721: Non-fungible token standard*, 2018. https://eips.ethereum.org/EIPS/eip-721. 13, 86

[78] Radomski, Witek, Andrew Cooke, Philippe Castonguay, James Therien, Eric Binet e Ronan Sandford: *Erc-1155: Multi Token Standard*, 2015. https://eips.ethereum.org/EIPS/eip-1155. 13, 59, 86

[79] Sandner, Philipp: *Will Blockchain Replace Clearinghouses? A Case Of DVP Post-Trade Settlement*, 2020. https://www.forbes.com/sites/philippsandner/2020/12/02/will-blockchain-replace-clearinghouses-a-case-of-dvp-post-trade-settlement/. 14

[80] Chakravarty, Manuel, James Chapman, Kenneth MacKenzie, Orestis Melkonian, Michael Peyton Jones e Philip Wadler: *The Extended UTXO Model - IOHK Research*, 2020. https://iohk.io/en/research/library/papers/the-extended-utxo-model/. 14

[81] Binance Academy: *Off-Chain | Binance Academy*, 2024. https://academy.binance.com/en/glossary/off-chain. 15

[82] Gudgeon, Lewis, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry e Arthur Gervais: *SoK: Layer-Two Blockchain Protocols*. Em *Financial Cryptography and Data Security*, volume 12059, páginas 201–226. Springer International Publishing, 2020. 15

[83] Optimism Foundation: *Home | Optimism.io*, 2024. https://www.optimism.io/. 16

[84] Bui, Van Cuong, Sheng Wen, Jiangshan Yu, Xin Xia, Mohammad Sayad Haghighi e Yang Xiang: *Evaluating Upgradable Smart Contract*. Em *2021 IEEE International Conference on Blockchain (Blockchain)*, páginas 252–256. IEEE, 2021. 16

[85] Atzei, Nicola, Massimo Bartoletti e Tiziana Cimoli: *A survey of attacks on Ethereum smart contracts*. Em *Principles of Security and Trust: 6th International Conference*, páginas 164–186. Springer Berlin Heidelberg, 2017. 16, 24

[86] Choi, Jaeseung, Doyeon Kim, Soomin Kim, Gustavo Grieco, Alex Groce e Sang Kil Cha: *SMARTIAN: Enhancing Smart Contract Fuzzing with Static and Dynamic Data-Flow Analyses*. Em *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, páginas 227–239. IEEE, 2021. 17, 25

[87] LCX Team: *Smart Contracts Security Challenges Explained*. `https://www.lcx.com/smart-contracts-security-challenges-explained/`. 17

[88] Cointelegraph: *Parity Multisig Wallet Hacked, or How Come?* `https://cointelegraph.com/news/parity-multisig-wallet-hacked-or-how-come`. 17

[89] Brent, Lexi, Anton Jurisevic, Michael Kong, Eric Liu, Francois Gauthier, Vincent Gramoli, Ralph Holz e Bernhard Scholz: *Vandal: A Scalable Security Analysis Framework for Smart Contracts*, 2018. 17

[90] Ober, Micha, Stefan Katzenbeisser e Kay Hamacher: *Structure and Anonymity of the Bitcoin Transaction Graph*. Future Internet, 5(2):237–250, 2013. 18

[91] Fleder, Michael, Michael S. Kester e Sudeep Pillai: *Bitcoin Transaction Graph Analysis*. Arxiv.org, 1502.01657:1–8, 2015. 18

[92] Dingledine, Roger, Nick Mathewson e Paul Syverson: *Tor: The Second-Generation Onion Router*. Em *Proceedings of the 13th USENIX Security Symposium*, páginas 303–319. USENIX, 2004. 18

[93] Henry, Ryan, Amir Herzberg e Aniket Kate: *Blockchain Access Privacy: Challenges and Directions*. IEEE Security & Privacy, 16(4):38–45, 2018. 18

[94] Kuhn, D Richard *et al.*: *A data structure for integrity protection with erasure capability*. NIST Cybersecurity Whitepaper, 2022. 19

[95] Stein, Björn, Konstantin Kuznecov, Sangseop Lee e Jürgen Müller: *A public blockchain solution permitting secure storage and deletion of private data—draft*. Lition Foundation, 2018. 19

[96] Oasis Labs Team: *Towards an open-source secure enclave*. `https://medium.com/oasislabs/towards-an-open-source-secure-enclave-659ac27b871a`. 20

[97] Intel Corporation: *Intel software guard extensions*. `https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html`. 20, 24, 41, 42

[98] Advanced Micro Devices, Inc: *AMD Secure Encrypted Virtualization (SEV) | AMD*, 2023. `https://www.amd.com/en/developer/sev.html`. 20

[99] Arm Limited: *TrustZone for Cortex-A - Arm*, 2024. `https://www.arm.com/technologies/trustzone-for-cortex-a`. 20

[100] Fei, Shufan, Zheng Yan, Wenxiu Ding e Haomeng Xie: *Security Vulnerabilities of SGX and Countermeasures: A Survey*. ACM Computing Surveys, 54(6):1–36, 2021. 21

[101] Muñoz, Antonio, Ruben Ríos, Rodrigo Román e Javier López: *A survey on the (in)security of trusted execution environments*. Computers & Security, 129:103180, 2023. 21

[102] Chaliasos, Stefanos, Marcos Antonios Charalambous, Liyi Zhou, Rafaila Galanopoulou, Arthur Gervais, Dimitris Mitropoulos e Ben Livshits: *Smart Contract and DeFi Security: Insights from Tool Evaluations and Practitioner Surveys*, 2023. 22

[103] Wang, Yajing, Jingsha He, Nafei Zhu, Yuzi Yi, Qingqing Zhang, Hongyu Song e Ruixin Xue: *Security enhancement technologies for smart contracts in the blockchain: A survey*. Transactions on Emerging Telecommunications Technologies, 32(12):e4341, 2021. 23

[104] Qi, Huayi, Minghui Xu, Dongxiao Yu e Xiuzhen Cheng: *SoK: Privacy-preserving smart contract*. High-Confidence Computing, 4(1):100183, 2024. 23

[105] Luu, Loi, Duc Hiep Chu, Hrishi Olickel, Prateek Saxena e Aquinas Hobor: *Making Smart Contracts Smarter*. Em *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, páginas 254–269. ACM, 2016. 24

[106] Zhang, Hengyan, Weizhe Zhang, Yuming Feng e Yang Liu: *SVScanner: Detecting smart contract vulnerabilities via deep semantic extraction*. Journal of Information Security and Applications, 75:103484, 2023. 25

[107] Zhang, Mengya, Xiaokuan Zhang, Yinqian Zhang e Zhiqiang Lin: *TXSPECTOR: Uncovering Attacks in Ethereum from Transactions*. Em *Proceedings of the 29th USENIX Security Symposium*. USENIX, 2020. 25

[108] Banerjee, Aritra, Michael Clear e Hitesh Tewari: *zkHawk: Practical Private Smart Contracts from MPC-based Hawk*. Em *2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, páginas 245–248, 2021. 26, 40, 45, 51, 53, 54, 61

[109] Steffen, Samuel, Benjamin Bichsel, Mario Gersbach, Noa Melchior, Petar Tsankov e Martin Vechev: *zkay: Specifying and Enforcing Data Privacy in Smart Contracts*. Em *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, páginas 1759–1776. ACM, 2019. 26, 43, 46, 48, 51, 53, 54, 61

[110] Bünz, Benedikt, Shashank Agrawal, Mahdi Zamani e Dan Boneh: *Zether: Towards Privacy in a Smart Contract World*. Em Bonneau, Joseph e Nadia Heninger (editores): *Financial Cryptography and Data Security*, volume 12059, páginas 423–443. Springer International Publishing, 2020. 26, 47, 48, 51, 53, 54, 61, 64, 76, 90

[111] Diamond, Benjamin E.: *Many-out-of-Many Proofs and Applications to Anonymous Zether*. Em *2021 IEEE Symposium on Security and Privacy (SP)*, páginas 1800–1817. IEEE, 2021. 26, 38, 47, 48, 51, 53, 54, 61, 62, 63, 76, 82, 86

[112] Cheng, Raymond, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah Johnson, Ari Juels, Andrew Miller e Dawn Song: *Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts*. Em *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, páginas 185–200. IEEE, 2019. 26, 41, 43, 51, 53, 54, 60

[113] IEEE: *IEEE Xplore*. https://ieeexplore.ieee.org/. 29

[114] Association for Computing Machinery: *ACM Digital Library*. https://dl.acm.org/. 29

[115] Google: *Google Scholar*. https://scholar.google.com/. 29

[116] justice.gov: *Office of Privacy and Civil Liberties | Privacy Act of 1974*, 1974. https://www.justice.gov/opcl/privacy-act-1974. 31

[117] Services, U.S. Department of Health and Human: *HIPAA Home | HHS.gov*, 1996. https://www.hhs.gov/hipaa/index.html. 31

[118] Federal Trade Commission: *Gramm-Leach-Bliley Act | Federal Trade Commission*, 1999. https://www.ftc.gov/legal-library/browse/statutes/gramm-leach-bliley-act. 31

[119] The White House: *Executive Order - Establishment of the Federal Privacy Council | whitehouse.gov*, 2016. https://obamawhitehouse.archives.gov/the-press-office/2016/02/09/executive-order-establishment-federal-privacy-council. 31

[120] FPC.gov: *Fair Information Practice Principles (FIPPs) | FPC.gov*, 2016. https://www.fpc.gov/resources/fipps/. 32

[121] oag.ca.gov: *California Consumer Privacy Act (CCPA)*, 2023. https://oag.ca.gov/privacy/ccpa. 32

[122] Commonwealth of Virginia: *Virginia Customer Data Protection Act*, 2023. https://law.lis.virginia.gov/vacodefull/title59.1/chapter53/. 32

[123] Forbes.com: *U.S. Data Privacy Protection Laws: A Comprehensive Guide*, 2023. https://www.forbes.com/sites/conormurray/2023/04/21/us-data-privacy-protection-laws-a-comprehensive-guide/. 32

[124] Klosowski, Thorin: *The State of Consumer Data Privacy Laws in the US (And Why It Matters) | Wirecutter*, 2021. https://www.nytimes.com/wirecutter/blog/state-of-privacy-laws-in-us/. 32

[125] Dicether: *Flip a coin - dicether*, 2023. https://dicether.com/games/flipACoin. 33

[126] Aave: *Aave*, 2025. https://aave.com/. 33

[127] Daian, Philip, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach e Ari Juels: *Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges*, 2019. 33

[128] Mollajafari, Sepideh e Kamal Bechkoum: *Blockchain Technology and Related Security Risks: Towards a Seven-Layer Perspective and Taxonomy.* Sustainability, 15(18):13401, 2023. 36

[129] Narula, Neha, Willy Vasquez e Madars Virza: *zkLedger: Privacy-Preserving Auditing for Distributed Ledgers.* Em *Proceedings of the 29th USENIX Security Symposium.* USENIX, 2018. 37

[130] Steffen, Samuel, Benjamin Bichsel, Roger Baumgartner e Martin Vechev: *ZeeStar: Private Smart Contracts by Homomorphic Encryption and Zero-knowledge Proofs.* Em *2022 IEEE Symposium on Security and Privacy (SP)*, páginas 179–197. IEEE, 2022. 38, 44, 48, 51, 53, 54, 61, 86

[131] Baum, Carsten, James Hsin yu Chiang, Bernardo David e Tore Kasper Frederiksen: *Eagle: Efficient Privacy Preserving Smart Contracts.* Em *International Conference on Financial Cryptography and Data Security*, páginas 270–288. Springer Nature Switzerland, 2023. 40, 45, 51, 53, 54

[132] Kalodner, Harry, Steven Goldfeder, Xiaoqi Chen, S Matthew Weinberg e Edward W Felten: *Arbitrum: Scalable, private smart contracts.* Em *Proceedings of the 27th USENIX Security Symposium.* USENIX, 2018. 40, 51, 53, 54, 60

[133] Yuan, Rui, Yu Bin Xia, Hai Bo Chen, Bin Yu Zang e Jan Xie: *ShadowEth: Private Smart Contract on Public Blockchain.* Journal of Computer Science and Technology, 33(3):542–556, 2018. 41, 43, 51, 53, 54, 60

[134] Tendermint Inc.: *Tendermint: Consensus without mining.* `https://tendermint.com/static/docs/tendermint.pdf`. 41

[135] Secret Network: *Secret network: A privacy-preserving secret contract & dapp platform.* `https://scrt.network/graypaper`. 41, 43, 51, 53, 54, 60, 91

[136] Kwon, Jae e Ethan Buchman: *Cosmos whitepaper.* `https://v1.cosmos.network/resources/whitepaper/en`. 42

[137] Secret Network: *Secret network.* `https://scrt.network/`. 42

[138] Luo, Xinyi, Kaiping Xue, Zhuo Xu, Mingrui Ai, Jianan Hong, Xianchao Zhang, Qibin Sun e Jun Lu: *EtherCloak: Enabling Multi-Level and Customized Privacy on Account-Model Blockchains.* IEEE Transactions on Dependable and Secure Computing, 2024. 42, 43, 51, 53, 54, 60, 91

[139] Microsoft: *GitHub - microsoft/eEVM*, 2023. `https://github.com/microsoft/eEVM`. 42

[140] *Sgx.fail.* `https://sgx.fail/`. 43

[141] Secret Network: *Notice: Successful resolution of xapic vulnerability.* `https://scrt.network/blog/notice-successful-resolution-of-xapic-vulnerability`. 43

[142] Baumann, Nick, Samuel Steffen, Benjamin Bichsel, Petar Tsankov e Martin Vechev: *zkay v0.2: Practical Data Privacy for Smart Contracts.* Arxiv.org, 2020. 44, 46, 48, 51, 53, 54, 61

[143] Steffen, Samuel, Benjamin Bichsel e Martin Vechev: *Zapper: Smart Contracts with Data and Identity Privacy.* Em *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, páginas 2735–2749. ACM, 2022. 44, 51, 53, 54, 61

[144] Bowe, Sean, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra e Howard Wu: *Zexe: Enabling Decentralized Private Computation.* Em *2020 IEEE Symposium on Security and Privacy (SP)*, páginas 947–964. IEEE, 2020. 44, 51, 53, 54, 61

[145] Xiong, Alex Luoyuan, Binyi Chen, Zhenfei Zhang, Benedikt Bünz, Ben Fisch, Fernando Krell e Philippe Camacho: *VeriZexe: Decentralized Private Computation with Universal Setup.* Em *Proceedings of the 32th USENIX Security Symposium.* USENIX, 2023. 45, 51, 53, 54, 61

[146] Kerber, Thomas, Aggelos Kiayias e Markulf Kohlweiss: *Kachina – Foundations of Private Smart Contracts.* Em *2021 IEEE 34th Computer Security Foundations Symposium (CSF)*, páginas 1–16, Dubrovnik, Croatia, 2021. IEEE. 45, 51, 53, 54, 61

[147] Jiang, Zoe L., Min Xie, Hanlin Chen, Yijian Pan, Jiazhuo Lyu, Man Ho Au, Junbin Fang, Yang Liu e Xuan Wang: *RPSC: Regulatable Privacy-Preserving Smart Contracts on Account-Based Blockchain.* IEEE Transactions on Network Science and Engineering, 11(5):4822–4835, 2024. 46, 51, 53, 54, 61, 85, 91

[148] Elgamal, Taher: *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.* IEEE transactions on information theory, 31(4):469–472, 1985. 47, 63

[149] Bunz, Benedikt, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille e Greg Maxwell: *Bulletproofs: Short Proofs for Confidential Transactions and More.* Em *2018 IEEE Symposium on Security and Privacy (SP)*, páginas 315–334. IEEE, 2018. 47, 49

[150] Groth, Jens e Markulf Kohlweiss: *One-Out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin.* Em Oswald, Elisabeth e Marc Fischlin (editores): *Advances in Cryptology - EUROCRYPT 2015*, volume 9057, páginas 253–280. Springer Berlin Heidelberg, 2015. 47

[151] Guo, Y., H. Karthikeyan, A. Polychorniadou e C. Huussin: *PriDe CT: Towards Public Consensus, Private Transactions, and Forward Secrecy in Decentralized Payments.* Em *2024 IEEE Symposium on Security and Privacy (S&P)*, páginas 183–183, Los Alamitos, CA, USA, 2024. IEEE Computer Society. 48, 51, 53, 54, 61, 91

[152] Solomon, Ravital, Rick Weber e Ghada Almashaqbeh: *smartFHE: Privacy-Preserving Smart Contracts from Fully Homomorphic Encryption*. Em *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, páginas 309–331. IEEE, 2023. 49, 51, 53, 54, 61

[153] Valence, Henry de, Cathie Yun e Oleg Andreev: *dalek-cryptography/bulletproofs: A pure-Rust implementation of Bulletproofs using Ristretto*, 2023. https://github.com/dalek-cryptography/bulletproofs. 49

[154] Labs, Offchain: *Github - offchainlabs/arbitrum*, 2023. https://github.com/OffchainLabs/arbitrum. 54

[155] Ekiden: *Github - ekiden/ekiden*, 2018. https://github.com/ekiden/ekiden. 54

[156] Labs, SCRT: *Github - scrtlabs/secretnetwork*, 2024. https://github.com/scrtlabs/SecretNetwork. 54

[157] ETH - SRI Lab: *Github - eth-sri/zkay*, 2022. https://github.com/eth-sri/zkay. 54, 61

[158] ETH - SRI Lab: *Github - eth-sri/zapper*, 2022. https://github.com/eth-sri/zapper. 54

[159] SCIPR Lab: *Github - scipr-lab/zexe*, 2024. https://github.com/scipr-lab/zexe. 54

[160] Espresso Systems: *Github - expressosystems/veri-zexe*, 2023. https://github.com/EspressoSystems/veri-zexe. 54

[161] Kaleido: *Github - kaleido-io/anonymous-zether: A private payment system for ethereum-based blockchains*, 2024. https://github.com/kaleido-io/anonymous-zether/. 54, 61, 63, 64, 86

[162] ETH - SRI Lab: *Github - eth-sri/zkay at sp2022*, 2022. https://github.com/eth-sri/zkay/tree/sp2022. 54, 61, 70, 86

[163] Sunscreen Tech: *Github - sunscreen-tech/sunscreen*, 2024. https://github.com/Sunscreen-tech/Sunscreen. 54

[164] ZAMA.ai: *Github - zama-ai/fhevm: A Solidity library for interacting with an fhEVM blockchain*, 2024. https://github.com/zama-ai/fhevm/. 54

[165] Uniswap: *Home | uniswap protocol*, 2023. https://uniswap.org/. 56

[166] Curve: *Pools - Curve*, 2025. https://curve.fi/dex/. 56

[167] Almendra, Daniel: *dalmendra/Miles2Coins*, 2024. https://github.com/dalmendra/Miles2Coins/. 59, 69, 73, 74

[168] Szaniecki, Yuri: *The Dencun upgrade is live: Ethereum's evolution continues at full throttle*, 2024. https://hashdex.com/en-US/insights/the-dencun-upgrade-is-live-ethereum-s-evolution-continues-at-full-throttle. 62

[169] OpenJS Foundation: *Node.js – Run JavaScript Everywhere*, 2024. https://nodejs.org/. 63

[170] Nomic Foundation: *Hardhat Network | Ethereum development environment for professionals by nomic foundation*, 2024. https://hardhat.org/hardhat-network/docs/overview. 63, 70

[171] Apache Software Foundation: *Apache JMeter*, 2024. https://jmeter.apache.org/. 63

[172] Diamond, Benjamin: *benediamond/anonymous-zether: A private payment system for ethereum-based blockchains, with no trusted setup.*, 2021. https://github.com/benediamond/anonymous-zether/. 63

[173] Kaleido, Inc: *Kaleido: Enterprise-Grade Blockchain & Digital Asset Platform*, 2024. https://www.kaleido.io/. 63

[174] Diamond, Benjamin E.: *Anonymous Zether: Infrastructure*, 2023. https://github.com/kaleido-io/anonymous-zether/blob/hardhat/docs/Infrastructure.pdf. 63, 86

[175] Kaleido: *Github - kaleido-io/anonymous-zether-client: A client implementation for anonymous zether with a rest interface*, 2024. https://github.com/kaleido-io/anonymous-zether-client/. 64, 82, 86

[176] Banco Central do Brasil: *GitHub - bacen/pilotord-kit-onboarding*, 2024. https://github.com/bacen/pilotord-kit-onboarding/. 64, 86

[177] Waku.org: *Waku is Uncompromising Web3 Communication at Scale | Waku*, 2024. https://waku.org/. 66

[178] Docker Inc.: *Docker: Accelerated Container Application Development*, 2024. https://www.docker.com/. 69

[179] Baumann, N., R. Baumgartner, B. Bichsel, S. Steffen e SRI Lab ETH Zurich: *Tutorial - zkay documentation*, 2021. https://eth-sri.github.io/zkay/tutorial.html. 70, 81

[180] The Ethereum Foundation: *ethereum/eth-tester: Tool suite for testing ethereum applications*, 2023. https://github.com/ethereum/eth-tester. 70, 81

[181] Truffle Suite: *Ganache - Truffle Suite*, 2023. https://www.trufflesuite.com/ganache/. 70

[182] Almendra, Daniel: *zeestar/eval-sp2022 at master - dalmendra/zeestar*, 2025. https://github.com/dalmendra/zeestar/tree/master/eval-sp2022. 73

[183] Etherscan: *Ethereum average gas price chart | etherscan*, 2024. https://etherscan.io/chart/gasprice. 74, 75, 76

[184] Etherscan: *Ether daily price (usd) chart | etherscan*, 2024. https://etherscan.io/chart/etherprice. 74, 75, 76

[185] *Problem when deploying to ganache - Issue #8*, 2023. https://github.com/eth-sri/zkay/issues/8. 81

[186] Pierro, Giuseppe Antonio, Roberto Tonelli e Michele Marchesi: *Smart-Corpus: an Organized Repository of Ethereum Smart Contracts Source Code and Metrics*, 2020. 81

[187] ZKProof Standards: *Zether: Towards privacy in a smart contract world*, 2019. https://www.youtube.com/watch?v=Nm0tn0vH194. 87

[188] Polygon Labs: *PoS - Polygon Knowledge Layer*, 2024. https://docs.polygon.technology/pos/overview/. 88

[189] Zhang, Wenbin, Yuan Yuan, Yanyan Hu, Shaohua Huang, Shengjiao Cao, Anuj Chopra e Sheng Huang: *A Privacy-Preserving Voting Protocol on Blockchain*. Em *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, páginas 401–408. IEEE, 2018. 90

[190] Linea: *Linea: the home network for the world*, 2024. https://linea.build/. 91

[191] Ethereum.org: *Cancun-Deneb (Dencun) FAQ*, 2024. https://ethereum.org/en/roadmap/dencun/. 91

[192] Ethereum.org: *Overview - Ordinal Theory Handbook*, 2023. https://docs.ordinals.com/overview.html. 91

[193] Defillama: *All Chains TVL - Defillama*, 2025. https://defillama.com/chains. 91

[194] *Arbitrum - The Future of Ethereum*. https://arbitrum.io/. 91

[195] CoinGecko.com: *Secret Blockchain: Coins, NFTs, Exchanges & More | CoinGecko*, 2025. https://www.coingecko.com/en/chains/secret. 91

[196] Manta Network: *Manta Network | The Modular Blockchain for ZK Applications*, 2025. https://manta.network/. 91

[197] Polkadot: *Polkadot | The secure, powerful core of Web3*, 2025. https://polkadot.com/. 91

[198] Defillama: *Manta - Defillama*, 2025. https://defillama.com/chain/Manta. 92

[199] Oasis Protocol Foundation: *Oasis*, 2025. https://oasisprotocol.org/. 92

[200] Oasis Protocol Foundation: *Oasis Sapphire*, 2025. https://oasisprotocol.org/sapphire. 92

[201] Defillama: *Oasis Sapphire - Defillama*, 2025. https://defillama.com/chain/Oasis%20Sapphire. 92

[202] Midnight: *Midnight | Empowering Data Protection Apps*, 2025. https://midnight.network/. 92

[203] Polygon Labs: *Introducing Polygon Nightfall Mainnet*, 2022. https://polygon.technology/blog/introducing-polygon-nightfall-mainnet-decentralized-private-transactions-for-enterprise. 92

[204] Ethereum Foundation: *Optimistic Rollups*, 2025. https://ethereum.org/en/developers/docs/scaling/optimistic-rollups/. 92

[205] LF Decentralized Trust: *Hyperledger Fabric*, 2025. https://www.lfdecentralizedtrust.org/projects/fabric. 93

[206] Hyperledger: *Hyperledger Fabric Docs*, 2025. https://hyperledger-fabric.readthedocs.io/en/latest/fabric_model.html#privacy. 93

[207] Hyperledger: *Welcome | Besu Documentation*, 2025. https://besu.hyperledger.org/. 93

[208] R3: *Corda - R3*, 2025. https://r3.com/corda/. 93

[209] Kaleido, Inc.: *Quorum Blockchain*, 2025. https://www.kaleido.io/blockchain-platform/quorum. 94

[210] ConsenSys, Inc.: *Tessera Private Transaction Manager*, 2024. https://docs.tessera.consensys.io/overview. 94

[211] DappRaddar: *Dapp Industry Report - 2024 Overview*, 2025. https://dappradar.com/blog/dapp-industry-report-2024-overview. 94

[212] Poelman, Michelle e Sarfraz Iqbal: *Investigating the Compliance of the GDPR: Processing Personal Data On A Blockchain.* Em *2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)*, páginas 38–44. IEEE, 2021. 94