Mecanismos de Transparência e Auditoria de Blockchains Permissionadas

Salão de Ferramentas SBSeg 2025 (modalidade código aberto)

Pedro H. Barcha Correia[1], **Otávio Vacari Martins**[1], **Marcos A. Simplicio Jr.**[1]

[1]Laboratório de Arquitetura e Redes de Computadores (LARC)
Universidade de São Paulo (USP) – São Paulo, SP, Brazil

{pedro.correia, otaviovacari, msimplicio}@usp.br

Resumo. *Blockchains permissionadas, como as privadas e consorciadas, são comuns em contextos que demandam controle de acesso. Porém, ao restringir os participantes da rede, diminuem a transparência do sistema. Diante disso, este trabalho propõe uma ferramenta de auditoria de blockchains permissionadas baseada no compartilhamento descentralizado do estado da blockchain via Interplanetary File System (IPFS). A partir de uma implementação utilizando Hyperledger Fabric, foram desenvolvidos mecanismos que permitem que entidades externas à rede blockchain possam identificar e se recuperar de alterações indevidas, ataques de visão fragmentada, uso de contratos inteligentes distintos dos declarados pela rede e execução de transações inválidas.*

1. Introdução

Blockchains são comumente categorizadas de acordo com seu modelo de permissão, que determina como a rede é mantida. Neste sentido, o Instituto Nacional de Padrões e Tecnologia dos Estados Unidos (NIST) reconhece **duas categorias de blockchain**: **(i) Não permissionada**, um modelo de rede aberta no qual qualquer indivíduo pode propor transações, ler a blockchain e, sobretudo, escrever nela (ou seja, publicar blocos); e **(ii) Permissionada**, onde a participação é restrita a um grupo limitado de entidades conhecidas, potencialmente sujeitas a políticas rigorosas de controle de acesso, e apenas aqueles autorizados a participar como nós (i.e., a federação) podem escrever novos blocos [Yaga et al. 2018]. Enquanto o modelo não permissionado é comumente referido como blockchain pública, a categoria permissionada é frequentemente subdividida em blockchain privada (quando controlada por um indivíduo ou organização) e consórcio (quando governada por múltiplos participantes) [Robinson and Brainard 2019, Chen et al. 2023]. Embora redes pioneiras como o Bitcoin [Nakamoto 2008] e o Ethereum [Ethereum 2014] implementem a abordagem pública, o modelo permissionado tem atraído interesse significativo tanto da indústria quanto de organizações governamentais, sendo adotado em grandes projetos como as moedas digitais de Bancos Centrais (CBDCs) [Zhang and Huang 2022]. Essa ampla adoção pode ser justificada pelas vantagens oferecidas pelas redes permissionadas em termos de mecanismos de privacidade, controle de acesso e ganhos em desempenho de rede [Correia et al. 2024].

Apesar de seus benefícios, **blockchains permissionadas apresentam desafios de transparência em relação a redes não permissionadas** [Correia et al. 2024]. Enquanto essas últimas permitem que entidades externas à rede leiam seu conteúdo, o acesso

público para leitura é opcional em blockchains permissionadas [Yaga et al. 2018]. Mesmo quando essa política é adotada, a transparência e a capacidade de auditoria para entidades externas são limitadas. Isso decorre do fato de que, por não serem nós da rede, não conseguem verificar o comportamento dos membros da rede intimamente, como seria possível em blockchains públicas, onde poderiam tornar-se membros. Assim, **ataques realizados pela federação dificilmente são detectáveis ou comprováveis por entidades externas**. Alguns exemplos são a execução de contratos inteligentes maliciosos, a reescrita da blockchain e a proposição e execução de operações inválidas [Kimura et al. 2024].

Com o intuito de mitigar esse problema, alguns trabalhos na literatura utilizam blockchains públicas para ancorar (*pin*) informações da blockchain permissionada [Chen et al. 2023, Bhanupriya et al. 2021]. Em [Robinson and Brainard 2019], a fim de permitir a detecção de conluios, os autores propõem ancorar periodicamente o estado de blockchains permissionadas em blockchains públicas. Note que essa abordagem apresenta um custo elevado, já que a maioria das blockchains públicas, como a Ethereum, impõe taxas para a inserção de dados. Outra possibilidade adotada por trabalhos relacionados é o uso do Interplanetary File System (IPFS) – um sistema distribuído de compartilhamento de arquivos – para ancorar dados da blockchain [Kimura et al. 2024, Al-Sarray et al. 2024]. Essa abordagem dispensa o pagamento a terceiros para a adição de dados à rede, uma vez que qualquer usuário pode fazê-lo. No entanto, para garantir que os dados permaneçam disponíveis, é necessário manter um nó ou contratar um serviço de ancoragem (*pinning*), o qual costuma ser mais acessível economicamente que a Ethereum [Pinata 2025].

Apesar de os trabalhos relacionados realizarem um amplo estudo dos possíveis ataques, as soluções apresentadas são voltadas a cenários específicos (e.g., votação eletrônica, venda de ativos digitais). O presente trabalho não só implementa uma solução com as funcionalidades já existentes na literatura, mas também propõe novos mecanismos de auditoria de blockchains permissionadas. Tudo isso, desenvolvido de maneira agnóstica de aplicação, integrado com o Hyperledger Fabric [Androulaki et al. 2018], *framework* de desenvolvimento de blockchains permissionadas. Assim, as seguintes **contribuições deste trabalho**:

(a) Agregação de mecanismos de transparência conhecidos em uma única ferramenta, incluindo: leitura de blocos, reconstrução do encadeamento de hashes da blockchain [Nakamoto 2008] e armazenamento periódico de partes da blockchain no IPFS [Kimura et al. 2024, Al-Sarray et al. 2024];

(b) Novos mecanismos de auditoria: adaptação do mecanismo de IPFS para o formato *append-only logs*; utilização do Interplanetary Name System (IPNS) para facilitar a obtenção dos logs, inclusive por meio de subscrição; e mecanismo de detecção de implantação de contratos inteligentes distintos dos alegados pela rede.

(c) Automação dos mecanismos de auditoria, facilitando a detecção de ataques de modificação do conteúdo da blockchain, ataques de visão fragmentada (*split-view*) e uso de contratos inteligentes ilegítimos em blockchains permissionadas;

(d) Interface para auditoria aprofundada, permitindo que entidades especializadas identifiquem transações ilegítimas e contratos inteligentes com código malicioso que favoreça a federação;

(e) Mecanismo que possibilita a recuperação (*fork*) da blockchain a estados prévios a ataques realizados pela rede;

(f) Integração da ferramenta com o Hyperledger Fabric.

2. Panorâma do cenário e da solução

A fim de se propor uma ferramenta agnóstica de transparência e auditoria de blockchains permissionadas, é necessário primeiramente definir os atores do sistema:

Federação: A federação constitui o núcleo operacional da rede blockchain permissionada. Os nós da federação são responsáveis pela gestão completa dos dados da plataforma e pela execução de contratos inteligentes. Estes contratos definem tanto a estrutura dos dados do sistema (como tokens fungíveis e não-fungíveis) quanto as operações permitidas no ambiente. Em redes permissionadas, o processo de admissão é controlado. Este modelo de governança centralizada da federação fundamenta a necessidade do segundo tipo de stakeholder.

Monitor: O monitor representa uma solução para a desconfiança inerente em sistemas centralizados de blockchain. Este stakeholder possui a capacidade de identificar condutas impróprias sem necessidade de integração direta à rede blockchain. Sua atuação pode ser tanto pontual (e.g., validar uma transação) quanto contínua (e.g., verificar os blocos recentes). Conforme detalhado na Seção 4, os monitores podem detectar: alterações no encadeamento da blockchain, ataques split-view, transações inválidas, presença de vulnerabilidades em contratos inteligentes e uso de contratos distintos dos alegados pela federação.

2.1. Append-only logs

Este trabalho utiliza uma blockchain permissionada com acesso de escrita restrito, porém acesso de leitura aberto, permitindo que qualquer interessado solicite intervalos específicos de blocos, atuando, portanto, como monitor. Isso viabiliza a verificação de dados e operações, mas não impede que a federação forneça dados forjados. Para mitigar esse problema, adotam-se *append-only logs* (ou logs transparentes), inspirados no esquema de transparência de certificados da Google [Google Open Source nd]. **A federação publica o estado da blockchain periodicamente (e.g., diariamente) no InterPlanetary File System (IPFS)**, uma rede de compartilhamento de arquivos baseada em comunicação peer-to-peer (P2P), na qual os participantes podem distribuir conteúdo de forma resistente à censura [Benet 2014]. O armazenamento desses "*checkpoints*" da blockchain no IPFS assegura a disponibilidade dos dados e permite que outras entidades os forneçam caso a federação deixe de fazê-lo. Além disso, essa arquitetura desencoraja condutas indevidas como a reescrita do encadeamento da blockchain, uma vez que seria possível detectar e denunciar tais ataques ao comparar o estado atual dos dados (fornecido pela federação) com aqueles salvos no IPFS. **A publicação periódica no IPFS contém:**

Cauda da blockchain: o último bloco da blockchain. Se a federação modificar qualquer bloco (e os subsequentes, para manter a consistência da cadeia de hashes), um monitor pode detectar a alteração comparando os blocos modificados com os originais, armazenados no IPFS.

World state: um resumo do estado da blockchain. Em um cenário de criptomoedas, por exemplo, seria semelhante ao balanço das carteiras. A adição desses *snapshots* da

blockchain ao IPFS viabiliza a recuperação desses estados. Assim, caso a federação atue inadequadamente e os monitores identifiquem tal comportamento, permite-se a recriação (ou *fork*) da blockchain a partir de um estado anterior à fraude, possivelmente sob gestão de uma nova (e mais confiável) federação.

CID da publicação anterior: cada publicação no IPFS possui um Content Identifier (CID), identificador que corresponde ao hash do seu conteúdo. Ao adicionar o CID da publicação anterior no IPFS (campo *prev-cid*), estabelece-se um encadeamento de hashes, criando-se efetivamente *append-only logs*. Caso a federação publique um *prev-cid* que não corresponda à publicação anterior, a política de *append-only* é violada. Este ataque pode ser facilmente detectado por monitores que mantenham localmente o encadeamento original ou o consultem via serviço de *pinning*.

Assinatura digital: antes de realizar a publicação, a federação confere o conteúdo acima e o assina com suas chaves privadas. Isso impede que a rede negue a autoria das publicações e permite que os monitores comprovem má conduta.

Certificados digitais: comprovam a autenticidade das assinaturas digitais. Os documentos devem identificar as entidades correspondentes e ser assinados por uma Autoridade Certificadora confiável.

Para realizar as publicações, utiliza-se InterPlanetary Name System (IPNS), protocolo que vincula o CID da publicação a uma chave pública (no caso, de um dos membros da federação) [IPFS nd]. Assim, os monitores podem utilizar essa chave como endereço para localizar a publicação mais recente e se inscrever para receber automaticamente novos logs (modelo *publish-subscribe*). Vale ressaltar que apenas o membro da federação que possui a chave privada correspondente à chave pública é capaz de vincular conteúdo do IPFS a esse endereço.

2.2. Portal de transparência

Além do mecanismo de *logs*, este trabalho também propõe um portal de transparência, com o qual **os monitores podem auditar a plataforma, solicitando à federação:**

Blocos: é possível requisitar blocos para: **(a)** Procurar por transações específicas (por exemplo, um usuário do sistema atuando como monitor, confirmando que seus dados estão presentes na blockchain); **(b)** Procurar por transações inválidas, aprovadas de forma ilegítima pela federação; e **(c)** Verificar a integridade do encadeamento, comparando o conteúdo da blockchain com os *append-only logs* e validando a cadeia de hashes. Para isso, a ferramenta permite que o monitor calcule localmente os hashes dos blocos solicitados e os compare com os contidos nos blocos retornados pela federação. Idealmente, todos os nós da rede deveriam retornar os blocos solicitados, e não apenas um dos membros da federação. Isso permite a identificação de divergências nos dados, potencialmente causadas por comportamento malicioso dentro da rede.

Contrato inteligente: um link para um repositório contendo o código-fonte dos contratos inteligentes que definem as operações permitidas no sistema. Isso possibilita a verificação da lógica de negócios e permite que os monitores verifiquem a existência de código malicioso, como *backdoors*. Além disso, o portal de transparência permite que o monitor calcule o hash do código-fonte localmente e o compare com o hash de implantação contido na blockchain. Isso fornece certo grau de confiança de que o contrato

disponibilizado é o mesmo executado pela federação (veja a Seção 4 para mais detalhes).

Conteúdo dos *append-only logs*: permite que monitores recuperem publicações no IPFS de forma simples. A seção também contém o endereço IPNS, útil para monitores que desejem receber automaticamente o conteúdo dos logs.

World state: apesar de ser publicado periodicamente no IPFS, também pode ser obtido através do portal de transparência, por comodidade ao monitor.

Note que o **portal de transparência visa facilitar e automatizar a auditoria** da blockchain. A maioria dos mecanismos propostos é executada de maneira automática: a verificação do encadeamento de hashes da blockchain, o confrontamento dos blocos com o conteúdo do IPFS, a vistoria das assinaturas dos logs e a verificação do contrato implantado. Apesar dessa facilitação, o arcabouço proposto também permite auditorias mais aprofundadas, por monitores especializados que desejem, por exemplo, buscar *backdoors* no contrato inteligente e identificar transações inválidas na blockchain (vide a Seção 4).

3. Implementação

A fim de avaliar a factibilidade da solução proposta, foi implementada uma plataforma digital que simula um ecossistema de finanças descentralizadas (DeFi). Nela, usuários podem se cadastrar, criando uma carteira digital e, assim, podendo propor a emissão de criptomoedas (tokens fungíveis) para a federação, a qual aprova imediatamente todas as requisições. O intuito dessa aplicação simples é criar um cenário onde monitores possam auditar o sistema utilizando IPFS e o portal de transparência, descritos na Seção 2.

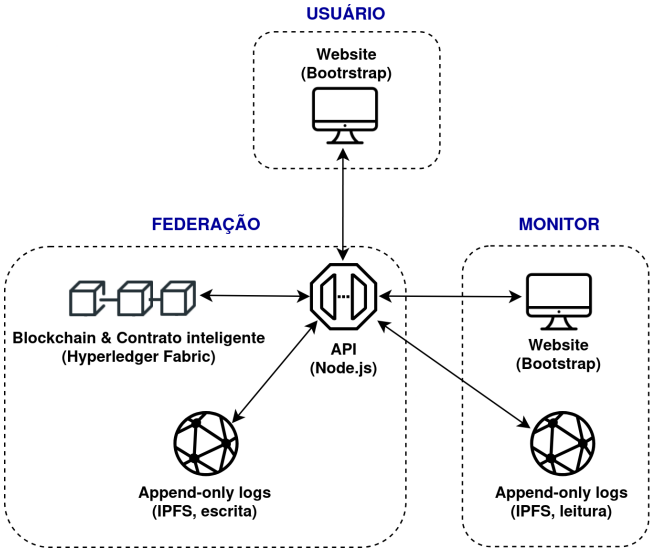

Figura 1. Arquitetura da implementação. As flechas indicam comunicação entre dois componentes

As funcionalidades de cadastro, emissão de tokens e o portal de transparência foram agregadas em uma página *web*. Sua comunicação com a blockchain, desenvolvida em Hyperledger Fabric 2.5, é feita através de uma API que utiliza Express.js, conforme ilustrado na Figura 1. Como o intuito do trabalho não é avaliar o desempenho da rede blockchain, mas sim os mecanismos de transparência propostos, emprega-se uma topologia de rede simples, com apenas uma organização executando e ordenando as transações.

Além disso, utiliza-se uma implementação local do IPFS, que não se conecta à sua *mainnet*, o que agiliza o envio de dados para o IPFS e não depende da gestão de portas no computador e em roteadores para seu funcionamento. Com essa configuração simples, é possível executar a prova de conceito com poucos recursos computacionais (4Gb/2vCPU) em uma máquina virtual, disponível abertamente, junto ao código-fonte da ferramenta e sua documentação[1].

4. Análise da ferramenta

Esta ferramenta propõe a publicação de *append-only logs* da blockchain no IPFS periodicamente. Isto pode ocorrer diariamente ou a cada hora caso sejam gerados muitos dados na blockchain e, portanto, sejam necessários mais *checkpoints* do seu estado. Devido a essa janela temporal, o mecanismo de IPFS não deve introduzir gargalos de escrita ao sistema. O mesmo vale para gargalos de leitura, cujo pior cenário é quando um monitor opta por receber as publicações diárias (modelo *publish-subscribe*, via IPNS, detalhado na Seção 2.1). Diante disso, a análise e discussão a seguir sobre os resultados obtidos com a implementação da ferramenta são feitas da perspectiva do monitor, visando compreender sua capacidade de detecção de ataques executados por um ou mais membros da federação. Para isso, **considera-se que o monitor, ao solicitar blocos à federação, realiza essa requisição a todos os seus nós,** podendo, portanto, encontrar divergências entre as respostas em casos de ataques que não envolvam todos os membros.

Tabela 1. Alguns ataques que um ou mais nós da federação podem realizar e como os monitores podem detectá-los. Ataque marcados com (*) requerem monitores especializados para serem detectados, enquanto os demais são detectados automaticamente pela ferramenta.

Ataque (pela federação)	Descrição	Detecção (pelo monitor)
Modificação discreta	A federação modifica o conteúdo da blockchain	Recalcular a cadeia de hashes
Cadeia alternativa	A federação modifica o conteúdo da blockchain e ajusta a cadeia de hashes	Comparar a blockchain com os *append-only logs*
Visão fragmentada (*split-view*)	A federação exibe versões diferentes de um mesmo bloco para usuários distintos	Comparar a blockchain com os *append-only logs*
Transação inválida (*)	A federação aprova uma transação não compatível com o contrato inteligente	Simular a transação utilizando o respectivo contrato inteligente
Contrato inteligente alternativo	A federação implanta um contrato inteligente diferente do alegado	Validar o hash do contrato inteligente na blockchain
Código malicioso no contrato inteligente (*)	A federação implementa mecanismos que podem favorecê-la	Inspecionar o código-fonte

A federação pode executar um ataque de **modificação discreta** ao criar, alterar ou excluir indevidamente um bloco. Seria possível, por exemplo, atualizar um bloco passado adicionando transações que emitem tokens silenciosamente para membros da rede. No entanto, neste caso, o bloco subsequente ainda referenciaria o hash do bloco original, não modificado. Assim, um monitor poderia identificar esse comportamento malicioso ao solicitar ambos os blocos e reconstruir a cadeia de hashes. No entanto, para contornar essa detecção, a federação poderia tentar recalcular e atualizar o valor do hash do bloco alterado, bem como dos blocos subsequentes, criando assim um **encadeamento de blocos alternativo**. Esse ataque permitiria, inclusive, a exclusão de blocos inteiros e a criação

[1] https://github.com/Carbon-21/fabric-transparency

de novos blocos com transações forjadas. Todavia, esse comportamento também é detectável. Ao comparar o estado atual da blockchain com os *append-only logs* publicados na rede IPFS, que conservam cópias de blocos anteriores, torna-se trivial concluir que ao menos um bloco foi alterado. Este mecanismo também permite a detecção de ataques de **visão fragmentada** (*split-view*), pois caso dado bloco seja mostrado de maneira distinta a diferentes usuários, a divergência torna-se evidente ao compará-lo com o conteúdo canônico, presente no IPFS. Nesses cenários de divergência, um monitor pode provar que o ataque ocorreu, uma vez que os registros dos logs são assinados pelos nós da rede.

Nos ataques mencionados, a federação forja **transações válidas** segundo o contrato inteligente implantado na rede Fabric. Contudo, os nós da rede também podem executar um ataque de transação inválida, no qual dados são escritos na blockchain sem seguir as regras previamente definidas no contrato inteligente. Assim, uma função de transferência de tokens poderia ser criada, por exemplo, para movimentar tokens de carteiras arbitrárias. Para detectar esse tipo de ataque, o monitor deve validar as transações simulando-as localmente, utilizando o mesmo contrato inteligente de código aberto implantado na rede [Kimura et al. 2024]. Em seguida, os resultados da simulação devem ser comparados com aqueles fornecidos pela federação. Ressalta-se que, no Hyperledger Fabric e em outras soluções de blockchain permissionada, os participantes da rede são identificáveis e assinam suas operações. Assim, caso uma transação inválida seja detectada por um monitor, todas as entidades envolvidas podem ser rastreadas e suas ações são irretratáveis, já que são assinadas digitalmente. Desse modo, tanto o proponente da transação quanto os nós responsáveis por sua execução (chamados *endorsers*, no Fabric) podem ser responsabilizados, por exemplo, por meio de sua exclusão do sistema. É importante destacar que, para mitigar esse tipo de ataque, a política de endosso no Fabric pode ser configurada de forma que a maioria dos *endorsers* precise executar cada transação, em vez de apenas um. Dessa forma, em redes com múltiplos *endorsers*, seria necessário haver conluio entre eles para que transações inválidas fossem efetivamente registradas.

No Hyperledger Fabric, após a aprovação de um novo contrato inteligente pela rede, um evento contendo o hash do contrato é registrado na blockchain. Esse mecanismo permite a detecção de ataques de **contratos alternativos**, nos quais a federação implanta um programa diferente daquele declarado. Considerando que o contrato alegado é de código aberto, monitores podem calcular seu hash e compará-lo com o valor registrado na blockchain. Também é possível verificar na blockchain se o contrato não foi substituído por outro posteriormente. Note que caso a federação, em conluio, registre um hash distinto do equivalente ao contrato implantado de fato, configura-se um ataque similar ao de transações inválidas, i.e., são executadas operações distintas das permitidas. Conforme descrito acima, este ataque é detectável.

Ainda sobre os benefícios de contratos inteligentes de código aberto, é necessário mencionar que eles proporcionam maior confiança aos usuários, uma vez que permitem a compreensão mais aprofundada da lógica de negócios do sistema. Combinando isso com as evidências acima de que o programa é o único em execução, é possível averiguar que a federação não inseriu **código malicioso** que a favoreça. A confiança no sistema torna-se ainda maior quando a rede é composta por diversos nós e a política para aprovação de novos contratos exige maioria ou unanimidade. Nesse cenário, caso algum nó proponha a implantação de um contrato malicioso, os demais participantes podem vetar a proposta.

No entanto, deve-se mencionar que, embora ofereça garantias contra ataques da federação, um contrato de código aberto pode representar maior risco de ataques provenientes de entidades externas. A ampla disponibilização do software permite que agentes mal-intencionados o analisem em busca de vulnerabilidades. Consequentemente, ataques de dia zero — cujo impacto é desconhecido *a priori* — podem ser facilitados, uma vez que os potenciais atacantes não precisam recorrer a técnicas de engenharia reversa para compreender os mecanismos do contrato. Para mitigar esse problema, recomenda-se instituir consultas públicas ao código, utilizar ferramentas de detecção de vulnerabilidades em contratos inteligentes como o Slither [Feist et al. 2019] e o Mythril [Consensys Diligence 2025] e, possivelmente, contratar empresas especializadas em auditoria desse tipo de software para avaliá-lo antes de sua implantação.

5. Conclusão e demonstração

Blockchains permissionadas apresentam vantagens nativas em relação ao modelo público, possibilitando, por exemplo, controle de acesso. Em contrapartida, apenas membros autorizados conseguem acompanhar intimamente o conteúdo da blockchain e o estado global da rede. Embora tais informações possam ser consultadas e verificadas por entidades externas (caso a política de leitura seja aberta), argumenta-se que esse modelo resulta em um sistema menos transparente quando comparado a alternativas baseadas em blockchains públicas. Para evitar esse tipo de crítica e facilitar o monitoramento independente e a replicação do sistema por entidades externas, este trabalho propõe a publicação periódica do estado da blockchain no IPFS. A partir deste mecanismo, é possível detectar e responsabilizar a federação por modificações indevidas no conteúdo da blockchain, além de permitir sua reconstrução (*forks*) a partir de estados prévios a ataques. A ferramenta proposta, desenvolvida com Hyperledger Fabric, implementa uma interface que automatiza os mecanismos propostos, facilitando o processo de auditoria e seu entendimento. Além disso, permite que monitores especializados façam análises mais aprofundadas, podendo detectar a execução de transações inválidas e o uso de contratos inteligentes distintos dos declarados pela federação. Assim, a solução suporta elevado grau de transparência e auditoria de governança e, apesar de não prevenir ataques, comportamentos maliciosos por parte da federação tendem a ser desencorajados diante da existência de mecanismos capazes de detectar tais eventos e identificar seus autores.

Dentre os trabalhos futuros, destaca-se o compartilhamento da ferramenta e sua documentação com a comunidade do Hyperledger Fabric. Caso haja interesse na integração da solução ao ecossistema do *framework*, as modificações necessárias para essa incorporação serão executadas.

Demonstração. A ferramenta será apresentada através do *notebook* do autor. A sessão começará com uma breve introdução ao problema, seguida pela demonstração do sistema sob a perspectiva do usuário, que também atuará como monitor. Para isso, será criada uma transação na blockchain, a qual será auditada junto às demais informações disponíveis no portal de transparência e no IPFS. Cada mecanismo de transparência será explicado conforme é demonstrado, bem como os ataques que permite detectar.

Agradecimentos. Este trabalho foi apoiado pela University Blockchain Research Initiative da Ripple e pela CAPES (bolsa 88887.112904/2025-00).

Referências

Al-Sarray, A. M., Hamdani, T. M., and Alimi, A. M. (2024). Decentralized distribution for secure GAN using IPFS with the Hyperledger blockchain. In *2024 IEEE ATSIP*, volume 1, pages 110–115.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., and Manevich, Y. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. In *Proc. of the 13th EuroSys Conference*, pages 1–15.

Benet, J. (2014). IPFS - Content addressed, versioned, P2P file system. *ArXiv Preprint arXiv:1407.3561*.

Bhanupriya, P., Gauni, S., Kalimuthu, K., and Manimegalai, C. (2021). A modified hybrid blockchain framework for secured data transaction. In *Journal of Physics: Conference Series*, volume 1964.

Chen, Z., Lu, Z., and Chen, J. (2023). Cross-chain trusted information match scheme with privacy-preserving and auditability. In *BlockTEA*, pages 94–114.

Consensys Diligence (2025). Repositório do Mythril. https://github.com/ConsenSysDiligence/mythril.

Correia, P. H. B., Marques, M. A., Simplicio, M. A., Ermlivitch, L., Miers, C. C., and Pillon, M. A. (2024). Comparative analysis of permissioned blockchains: Cosmos, hyperledger fabric, quorum, and xrpl. In *2024 IEEE Blockchain*, pages 464–469.

Ethereum (2014). White paper. https://tinyurl.com/msz6c8bs.

Feist, J., Grieco, G., and Groce, A. (2019). Slither: a static analysis framework for smart contracts. in 2019 ieee/acm 2nd international workshop on emerging trends in software engineering for blockchain (wetseb).

Google Open Source (n.d.). Certificate transparency. https://tinyurl.com/mtu8vz8d.

IPFS (n.d.). IPNS. https://docs.ipfs.tech/concepts/ipns/.

Kimura, L. T., Shiraishi, F. K., Andrade, E. R., Carvalho, T. C., and Simplicio, M. A. (2024). Amazon biobank: Assessing the implementation of a blockchain-based genomic database. *IEEE Access*.

Nakamoto, S. (2008). Bitcoin whitepaper. bitcoin.org/bitcoin.pdf.

Pinata (2025). Pricing. https://pinata.cloud/pricing.

Robinson, P. and Brainard, J. (2019). Anonymous state pinning for private blockchains. In *2019 IEEE TrustCom/BigDataSE*, pages 827–834.

Yaga, D., Mell, P., Roby, N., and Scarfone, K. (2018). NIST IR 8202: Blockchain technology overview.

Zhang, T. and Huang, Z. (2022). Blockchain and central bank digital currency. *ICT Express*, 8(2):264–270.