Uma arquitetura para proteger a privacidade dos dados de saúde armazenados na blockchain

Christofer L. Sega* Anubis G. de M. Rossetto †
Valderi R. Q. Leithardt ‡

2022

Resumo

Com o rápido desenvolvimento da tecnologia blockchain nos últimos anos, a sua aplicação em cenários que precisam de privacidade, como a área da saúde, começou a ter um fomento e a ser discutida. Este trabalho apresenta uma arquitetura para garantir a privacidade dos dados relacionados à área da saúde, que são armazenados e compartilhados dentro de uma rede blockchain de maneira descentralizada, através do uso de criptografia com os algoritmos RSA, ECC e AES. Foram realizados testes de avaliação a fim de verificar o impacto da criptografia na arquitetura proposta quanto ao custo, uso de memória e tempo de execução. Os resultados demonstram um impacto, principalmente, sobre o tempo de execução e no aumento do custo de envio dos dados a blockchain, porém, justificável considerando a privacidade e a segurança fornecidas com a arquitetura e a criptografia.

Palavras-chave: Blockchain; Criptografia; Dados da saúde; DApp; Privacidade.

*<christofersega@gmail.com>
†<anubis.rossetto@passofundo.ifsul.edu.br>
‡<valderi@ipportalegre.pt>

1 Introdução

Muitos avanços relacionados a tecnologia blockchain foram consolidados recentemente, destacam-se: o advento da blockchain 2.0, a rede blockchain Ethereum (principal blockchain programável e pública), o Hyperledger Fabric (blockchain privada e permissionada), o aperfeiçoamento dos contratos inteligentes e a utilização da criptografia sobre os dados que trafegam na blockchain, como o Elliptic Curve Cryptography (ECC). Com isso, tornou-se possível desenvolver formas de garantir a privacidade, integridade e controle de acesso dos dados dentro de alguma aplicação específica.

Na blockchain proposta inicialmente por Nakamoto (2008), os dados são publicamente visíveis para todos na rede blockchain. Consequentemente, é importante que essas informações sejam criptografadas antes de serem armazenadas, para assim garantir a confidencialidade dos dados. Desta forma, será possível manter o conteúdo da transação privado, contribuindo para reduzir o risco de vinculação do pseudônimo a identidade real do usuário da blockchain, o que é fundamental para promover o compartilhamento baseado na necessidade de saber (ZHANG; XUE; LIU, 2019).

Para garantir a privacidade, em alguns casos, é necessário fazer uso da criptografia, a qual possui diversas técnicas e algoritmos que podem ser utilizados para implementar a segurança que ela fornece. Outro aspecto que deve ser considerado é a identificação do tipo de criptografia que melhor satisfaz o problema de garantir a segurança dos dados que vão ser armazenados na blockchain. Entre vários exemplos de aplicações e cenários, está o controle do acesso a informações como diagnósticos médicos, resultados de exames e demais informações confidenciais, vitais e sensíveis ao paciente devem ficar sob sua responsabilidade, pois em um ambiente centralizado o paciente não possui controle sobre os dados armazenados, somente a instituição que os armazena. Dessa forma, como apontado em Shi et al. (2020) o usuário não sabe se suas informações podem estar sendo disponbilizadas para, por exemplo, uma companhia de seguro que pode não aceitar realizar certa cobertura devido a essa divulgação indevida de dados.

A motivação para esse projeto teve origem a partir da leitura de trabalhos que abordam o cenário da blockchain, tais como Dasgupta, Shrein e Gupta (2019), Feng et al. (2019), Shi et al. (2020) e Zhang, Xue e Liu (2019). Esses trabalhos apontam os sistemas voltados para área de saúde como um potencial a ser pesquisado, principalmente com a camada de privacidade. Neste sentido, este trabalho apresenta uma arquitetura para garantir a privacidade dos dados relacionados à área da saúde que são armazenados dentro de uma rede blockchain e no InterPlanetary File System (IPFS), através do uso de criptografia com os algoritmos Rivest-Shamir-Adleman (RSA), ECC e Advanced Encryption Standard (AES). Também foram conduzidos testes de avaliação para verificar o impacto da criptografia na arquitetura, considerando os critérios de custo, uso de memória e tempo de execução.

O trabalho está organizado da seguinte forma: na segunda seção são apresentados os fundamentos e ferramentas necessárias para a construção da arquitetura. A terceira seção aborda os trabalhos relacionados de aplicações para a área de saúde. A quarta seção apresenta a arquitetura proposta juntamente com seus componentes.

A quinta seção descreve como foi feita a avaliação da arquitetura com os algoritmos de criptografia e apresenta os resultados obtidos. Por fim, na sexta seção estão as considerações finais e apontamentos sobre os próximos passos da pesquisa.

2 Fundamentos e Tecnologias

Nesta seção é apresentado o embasamento teórico sobre o qual se fundamenta o trabalho e as tecnologias que foram empregadas. Assim, são abordados os conceitos e características da blockchain, os contratos inteligentes, os algoritmos de criptografia, RSA, ECC, AES, a rede blockchain do Ethereum, o IPFS, os frameworks do Truffle Suite (Truffle e Ganache), o MetaMask, o React, o Node.js, a WEB3 (Web 3.0) e a Application Programming Interface (API) do Infura.

2.1 Blockchain

A blockchain foi originalmente introduzida, ou teve maior reconhecimento, quando Nakamoto em seu trabalho propôs um sistema financeiro utilizando a blockchain para registrar todas as transferências da moeda digital Bitcoin de forma segura e confiável (FENG et al., 2019).

Esta tecnologia é como um livro-razão descentralizado, distribuído e imutável, formado por uma coleção de registros que são criptograficamente vinculados, sendo tal coleção mais conhecida como corrente de blocos que armazenam transações ou eventos (HEWA; YLIANTTILA; LIYANAGE, 2021). Este livro-razão é compartilhado com todos os membros participantes (nós) da rede blockchain.

As transações que são realizadas entre os membros de uma blockchain devem ser aprovadas pelos nós mineradores antes de serem confirmadas e adicionadas à rede blockchain. Dessa forma, para iniciar o processo de mineração, a transação é transmitida a todos os nós da rede e os nós que são mineradores vão organizar as transações em um bloco, verificar as transações no bloco e transmitir o bloco e a sua verificação usando um protocolo de consenso, por exemplo o Proof of Work (POW), para obter a aprovação da rede (ZHANG; XUE; LIU, 2019). Quando os demais nós verificarem se todas as transações contidas no bloco são válidas, o bloco pode ser adicionado a blockchain através de uma função hash criptográfica que conecta os blocos da estrutura, onde o hash do bloco n está vinculado ao hash do bloco $n+1$ (OCHÔA et al., 2019).

Dentre as características da blockchain, as mais importantes segundo Hewa, Ylianttila e Liyanage (2021) são:

- Descentralização: concede autoridade para os membros da rede, garantindo redundância em contraste com os sistemas centralizados operados por um terceiro confiável. A descentralização reduz o risco de falhas e acaba melhorando a confiança do serviço com disponibilidade garantida;
- Imutabilidade: os registros de transações no livro-razão, distribuídos entre os nós, são permanentes e inalteráveis. A imutabilidade é uma característica que

difere dos sistemas de banco de dados centralizados. Os registros são resistentes a adulteração computacional com a existência de links criptográficos;

- Link criptográfico: o link criptográfico entre cada registro é classificado em ordem cronológica construindo uma cadeia de integridade pela blockchain. A assinatura digital verifica a integridade de cada registro usando técnicas de hashing e criptografia de chave assimétrica. Violar a integridade do registro do bloco ou da transação acaba tornando o registro e o bloco inválidos.

A segurança da blockchain faz parte dos avanços da criptografia e do design e implementação da blockchain (Bitcoin, Ethereum, etc.). Foram realizadas propostas de blockchains, com o passar do tempo, para melhorar a eficiência da cadeia criptográfica de blocos, por exemplo, incorporar árvores Merkle e colocar vários documentos em um bloco (ZHANG; XUE; LIU, 2019). A blockchain foi construída para garantir diversas características em relação a segurança, como consistência, resistência a adulteração, pseudoanonimato e resistência a ataques de gasto duplo e Distributed Denial of Service (DDoS). Porém, mesmo com o nível atual de segurança que a blockchain consegue prover, em alguns cenários ainda faltam propriedades adicionais de segurança e privacidade (ZHANG; XUE; LIU, 2019).

2.2 Contratos Inteligentes

Os contratos inteligentes podem ser considerados como um programa que é executado quando condições predeterminadas são atendidas (autoexecutável) e que está implantado na blockchain, podendo ser utilizado em serviços financeiros, saúde e governo. É capaz de suportar funções e mecanismos programáveis complexos para automatizar acordos e outros tipos de fluxos (SHI et al., 2020).

Esse tipo de contrato, que pode ser utilizado na blockchain, permite que as partes possam fazer o uso dele para criar terceiros virtuais de confiança que se comportaram de acordo com as regras acordadas entre ambos, dessa forma permitindo a criação de protocolos complexos com um risco de não cumprimento muito baixo (KALODNER et al., 2018).

Na rede blockchain do Ethereum os contratos inteligentes são desenvolvidos formalmente em código de alto nível através do Solidity (linguagem de programação) e são compilados para serem executados pela Máquina Virtual Ethereum (EVM). No conceito do Solidity os contratos inteligentes são um conjunto de códigos, dados, funções e estados, que estão em um endereço específico na rede blockchain do Ethereum (ETHEREUM, 2017).

Para que uma conta interaja com um contrato ou para que ocorra interações entre contratos, deve-se ter o nome e os argumentos da função, assim surge a Application Binary Interface (ABI) que é uma lista das funções e argumentos do contrato organizados no formato de um JavaScript Object Notation (JSON) , assim que ele é compilado. Dessa forma, utiliza-se da ABI para fazer o hash da definição da função e então criar o bytecode EVM necessário para chamar a função (ETHEREUM, 2021).

2.3 Algoritmos de Criptografia

Os algoritmos criptográficos se dividem em dois tipos, os de chave simétrica ou chave privada, que ainda podem ser divididos em algoritmos que operam em um único bit ou em grupos de bits, e os de chave assimétrica ou chave pública.

A criptografia de chave pública foi criada em 1976 quando W. Diffie e M. Hellman, propuseram esta nova ideia, a qual foi seguida por R. L. Rivest, A. Shamir, e L. Adleman, os criadores do algoritmo RSA (OLIVEIRA et al., 2014). Segundo Singh, Khan e Singh (2016), são algoritmos que dependem do uso de uma chave pública e de uma chave privada. A chave pública será livremente distribuída sem comprometer de alguma maneira a chave privada, a qual deve ser mantida em segredo. A chave pública é utilizada para criptografar mensagens de texto simples e verificar assinaturas, já a chave privada é usada para assinar mensagens e descriptografar os textos criptografados para obter as mensagens em texto simples (SINGH; KHAN; SINGH, 2016). A criptografia baseada em curvas elípticas, que é um tipo especial de chave pública, foi inicialmente proposta por Miller e Koblitz no final dos anos 1980, e foi baseada em algoritmos e aplicações de chave pública até então já existentes (OLIVEIRA et al., 2014).

Os algoritmos de chave simétrica vão possuir uma chave (privada) que é igual para ambas as partes que estão trocando informações e deve permanecer em segredo, pois ela será utilizada para criptografar e descriptografar as informações, é um método simples que facilita a criptografia. Entretanto, o problema dessa forma reside no compartilhamento da chave entre as partes, pois se alguém conseguir interceptar essa troca e obter acesso a chave, a pessoa terá acesso para criptografar e descriptografar as informações (OLIVEIRA, 2012).

A criptografia propicia um mecanismo para se conseguir garantir a segurança dos dados nos sistemas de informação atuais. Este trabalho utilizou o algoritmo AES em conjunto com o RSA e o ECC, para assim criar um criptossistema híbrido com objetivo de aumentar a complexidade e força da criptografia. A seguir serão abordados os algoritmos RSA, ECC e AES.

2.3.1 RSA

O algoritmo de criptografia RSA de chave assimétrica tornou-se o padrão para criptografia de chave pública, sendo amplamente utilizado. Sua segurança reside no problema de fatoração de inteiros e o seu processo de descriptografia não é eficiente como seu processo de criptografia (MAHTO; YADAV, 2017). Para uma segurança de dados melhor e mais forte, o RSA acaba necessitando de tamanhos de chave maiores, o que implica em mais sobrecarga sobre os sistemas. Dessa forma, para os sistemas que possuem restrição de memória, o RSA se torna uma segunda opção (MAHTO; YADAV, 2017).

2.3.2 ECC

A segurança do algoritmo de criptografia ECC de chave assimétrica reside no uso das propriedades matemáticas da curva elíptica para realizar o cálculo das chaves

criptográficas (problema do logaritmo discreto sobre curvas elípticas) (MAHTO; YADAV, 2017). É um sistema adequado e promissor para dispositivos que possuem restrição de memória (Smartphone e Smartcards), além de ser utilizado em blockchains, como por exemplo a do Bitcoin que possui o maior valor de mercado atualmente. O ECC consegue manter os níveis de segurança de forma equivalente ao RSA e requer comparativamente menores parâmetros para criptografia e descriptografia do que o RSA (MAHTO; YADAV, 2017).

O algoritmo ECC utiliza da curva elíptica que segundo Mahto e Yadav (2017), uma curva elíptica $\mathbf{C}$ sobre um campo finito primo é definida por uma equação na forma $\mathbf{y^2 = x^3 + ax + b}$ satisfazendo a restrição $\mathbf{4a^3 + 27b^2 \neq 0}$.

2.3.3 AES

O algoritmo de criptografia AES de chave simétrica foi desenvolvido em 1998 por Joan Daemen e Vincent Rijmen, permite um tamanho de bloco de dados fixo de 128 bits e suporta tamanho de chave de 128, 192, e 256 bits, além de qualquer combinação de dados (PATIL et al., 2016). Segundo Oliveira (2012) é um dos algoritmos de chave simétrica mais populares, sendo adotado como padrão pelo governo dos Estados Unidos, e considerado o substituto do Data Encryption Standard (DES), devido sua rapidez, fácil execução e pouca exigência de memória.

Sendo a cifra de bloco de chave simétrica mais utilizada dentro da segurança de computadores, principalmente pela sua padronização pelo NIST e também por todas as cripto análises publicadas sobre este algoritmo, conseguindo resistir a diversos tipos de ataques (SARAIVA et al., 2019). Assim, torna-se uma escolha ideal para a criptografia de dados de maior volume, devido a sua performance, podendo ser combinado com a segurança de um algoritmo de chave assimétrica.

2.4 Rede Blockchain do Ethereum

Na blockchain do Ethereum existe a EVM, que é um computador canônico, o qual todos os membros da rede Ethereum concordam. Cada nó da rede mantém uma cópia do estado deste computador e qualquer participante pode transmitir uma solicitação de transação para que este computador execute (ETHEREUM, 2022). Quando uma solicitação é realizada, os outros participantes da rede verificam, validam e executam o cálculo, o qual causa uma mudança de estado da EVM que será confirmada e propagada pela rede (ETHEREUM, 2022).

A rede do Ethereum é uma rede blockchain não-permissionada, ou seja, possui acesso aberto sem a necessidade de autenticação ou a existência de uma entidade central. Assim qualquer usuário pode criar uma carteira e fazer parte da rede, podendo se tornar um nó minerador e tentar validar os blocos em troca de uma taxa paga em Ether (ETH), token da rede Ethereum (WÜST; GERVAIS, 2018).

Os mecanismos criptográficos da rede vão garantir que as transações sejam verificadas como válidas e então adicionadas na blockchain, não podendo ser alteradas posteriormente devido a dificuldade de realizar alguma adulteração, além disso também garantem que todas as transações sejam assinadas e executadas com as

devidas permissões (ETHEREUM, 2022). O Ethereum utiliza o mecanismo de consenso POW, desse modo o usuário que deseja adicionar um novo bloco à cadeia deve resolver um problema computacional difícil que vai requerer muito poder de computação, esse processamento que é conhecido como mineração, uma tentativa e erro de força bruta, onde quem calcular com sucesso o resultado será recompensado em ETH (ETHEREUM, 2022).

O Ethereum é a primeira plataforma de blockchain de código aberto que oferece uma linguagem completa para os desenvolvedores implantarem seus contratos e suas *Decentralized Applications* (DApp) ao custo de uma taxa paga à rede no momento da migração do contrato para a blockchain (SHI et al., 2020).

2.5 IPFS

O IPFS é um sistema de arquivos distribuído (peer-to-peer) para armazenar, compartilhar e acessar arquivos, sites, aplicativos e dados com objetivo de tornar a web atualizável, resiliente e mais aberta (IPFS, 2022).

De acordo com sua documentação, existem três princípios fundamentais para compreender o IPFS (IPFS, 2022):

- Identificação única por meio de endereçamento de conteúdo: os endereços de acesso as informações são baseados no conteúdo e não na sua localização, assim quando é adicionado um novo arquivo no IPFS esse arquivo é dividido em pedaços chamados "IPFS's objects" (protegidos por criptografia, sistema de hash e assinatura digital) que vão possuir um Content Identifier (CID), o qual é um registro permanente do arquivo, além disso os "IPFS's objects" guardam até 256kb de dados e também podem conter links para outros objetos que estão na rede;
- Vinculação de conteúdo por meio de Directed Acyclic Graphs (DAG): a estruturada do IPFS é um grande DAG, especificamente um Merkle DAG onde cada nó tem um identificador que é o resultado do hash do conteúdo do nó. A representação de seu conteúdo é feita em blocos e cada bloco tem uma raiz Merkle, isso significa que diferentes partes do arquivo podem vir de diferentes fontes. As Merkle DAGs só podem ser construídas a partir das folhas, cada nó em um Merkle DAG é a raiz de um (sub) Merkle DAG em si e os nós na Merkle DAG são imutáveis;
- Descoberta de conteúdo por meio de Distributed Hash Tables (DHT): o IPFS usa uma tabela de hash distribuída, que é uma tabela dividida entre todos os pares da rede descentralizada, para armazenar as chaves e valores. O projeto "libp2p" é a parte do IPFS que fornece o DHT e controla os pares que se conectam e conversam entre si. Utiliza-se o DHT para encontrar a localização atual dos pares que estão armazenando o conteúdo que está sendo buscado.

Para os desenvolvedores blockchain, o endereçamento de conteúdo IPFS permite o armazenamento de grandes arquivos fora de uma blockchain específica, pois ao invés de armazenar o arquivo na blockchain, que pode ser custoso, pode-se

armazenar apenas os links imutáveis e permanentes, que vão referenciar o arquivo armazenado dentro do IPFS. Desta forma, possibilita o carimbo de data e hora e a proteção do conteúdo sem ter que adicionar os dados de maior volume na própria blockchain (IPFS, 2022).

2.6 Ferramentas Utilizadas

Para o desenvolvimento da solução foram utilizadas as ferramentas e as tecnologias relacionadas a seguir:

- Truffle Suite: o Ganache, o Truffle e o Drizzle formam a coleção de ferramentas conhecida como Truffle Suite. No trabalho é utilizado do Ganache que simula uma blockchain privada para realização de testes e do Truffle que permite compilar e migrar os contratos para uma rede blockchain e também realizar testes de envio de informações (TRUFFLESUITE, 2022);
- MetaMask: é uma carteira criptografada (digital) e um gateway para aplicativos da blockchain, possibilita que os usuários gerenciem suas contas, chaves e tokens de várias maneiras, incluindo carteiras de hardware, enquanto os isola do contexto do site (METAMASK, 2021);
- React: uma biblioteca JavaScript declarativa, código mais previsível e simples de depurar e é baseada em componentes para construir interfaces de usuário (REACT, 2022);
- Node.js: um runtime JavaScript assíncrono e orientado a eventos, projetado para criar aplicativos de rede escaláveis (NODE.JS, 2022);
- WEB 3.0: possui a descentralização em seu núcleo, diferente da WEB 1.0 e WEB 2.0, além de trazer algumas características adicionais como: ser verificável, autogovernado, sem permissão, distribuído e robusto entre outros recursos (DABIT, 2021);
- API do Infura: fornece uma infraestrutura para DApps de maneira fácil e rápida. Através do Infura os desenvolvedores podem se conectar ao Ethereum e ao IPFS via HTTPS e WebSocket com tempos de resposta e disponibilidade de serviço satisfatórios (INFURA, 2022).

3 Trabalhos Relacionados

O trabalho de Aguiar (2021) aborda essa temática de forma bem diferente da proposta do presente trabalho, com o desenvolvimento de um framework blockchain utilizando do Hyperledger Fabric, uma blockchain privada e permissionada, junto com métodos de anonimização para preservar a privacidade durante o compartilhamento de dados de saúde. Caracteriza-se como uma abordagem diferente pela escolha da rede blockchain e a utilização de métodos de anonimização como o K-Anonimato.

Já o trabalho de Omar et al. (2019) que traz a MediBchain, possui certa similaridade pois busca deixar o controle do dados em nuvem sob responsabilidade

do paciente/usuário, fazendo uso da criptografia do algoritmo ECC para alcançar o pseudoanonimato e a privacidade. Os autores não apresentam a utilização de algum outro algoritmo de criptografia, nem o uso do IPFS para diminuir o custo de armazenamento na blockchain, bem como não mencionam a possibilidade de permissão dos arquivos. É apresentada uma análise aprofundada do custo e do protocolo do sistema desenvolvido.

No trabalho de Gan et al. (2020) é proposto um controle de acesso, onde os pacientes serão supervisores de seus dados médicos e as instituições podem acessar seus dados sem autorização, mas pode-se revogar o acesso a qualquer momento. É considerado no trabalho um sistema de incentivo onde os pacientes que compartilharem seus dados recebem uma recompensa conforme um critério avaliativo.

Em Mendonça et al. (2021) é abordada a concessão e revogação de registros eletrônicos de saúde armazenados em um banco de dados externo, com um contrato inteligente na blockchain responsável pelo controle do acesso e outro dos dados. Foi conduzida uma simulação para observar o tempo de espera das solicitações realizadas pelas DApps. Já a arquitetura do presente trabalho também aborda um sistema para as permissões sobre os arquivos armazenados, que funciona de forma diferente, pois o acesso é dado conforme um intervalo de tempo definido, o controle também funciona através de um contrato inteligente, e os testes realizados sobre essa implementação tem o foco, diferente dos demais trabalhos, de ver o impacto da criptografia sobre a arquitetura conforme os critérios de tempo de execução, uso de memória e o aumento do custo de envio das informações para a blockchain.

Por fim, o trabalho de Shahnaz, Qamar e Khalid (2019) apresenta um framework para o setor de saúde para realizar o armazenamento seguro de registros eletrônicos na blockchain com regras de acesso, utilizando o IPFS para armazenar os arquivos. Entretanto, os autores do trabalho apresentam uma abordagem pouco voltada para a privacidade e a criptografia dos dados para garantir a segurança, já que a segurança do sistema apresentado depende da própria segurança fornecida pelas tecnologias utilizadas, diferentemente deste trabalho que visa garantir a privacidade com a adição de um esquema a mais de criptografia. Também são realizados testes pelos autores simulando um cenário com vários usuários utilizando de diferentes funções do framework, onde são avaliados três critérios: o tempo de execução, taxa de transferência e a latência.

4 Arquitetura da Solução

Com o objetivo de realizar a proteção da privacidade dos dados de saúde e verificar o desempenho das técnicas de criptografia, foi projetada uma arquitetura de aplicação que permita ao usuário acessar e se comunicar com a blockchain do Ethereum e o IPFS, assim obtendo seus dados relacionados à saúde que foram armazenados de forma segura através de criptografia. O usuário pode acessar a aplicação através do acesso ao MetaMask adicionando uma carteira digital. Os usuários da aplicação podem efetuar buscas por seus documentos armazenados e compartilhar documentos. A persistência dos dados de maior volume é feita através do IPFS.

Foram definidos os requisitos funcionais e não funcionais para a solução proposta. Como requisitos funcionais, temos:

- Realizar Login: a aplicação só estará disponível após login do usuário pelo MetaMask;
- Informar chaves de criptografia: toda vez que o usuário logar na aplicação serão geradas suas chaves (pública e privada); a aplicação possui um espaço para que o usuário informe suas chaves; na primeira vez que acessar a aplicação pode guardar suas chaves da forma que preferir, e no próximo login pode informar as suas chaves;
- Armazenar informações: o usuário pode enviar informações a serem armazenadas no IPFS, de forma segura através da criptografia AES, o qual gera um hash para se ter acesso ao conteúdo na rede, esse hash e a chave privada do AES, gerada aleatoriamente, serão armazenados de forma criptografada na blockchain do Ethereum, através de um dos algoritmos de chave assimétrica (ECC ou RSA), para garantir a privacidade e o controle do acesso aos dados;
- Confirmar ou rejeitar transações: o usuário, após enviar uma requisição para armazenar informações ou uma permissão nos respectivos contratos, pode aceitar ou rejeitar a transação através do MetaMask;
- Buscar informações: o usuário pode buscar as informações que possui armazenadas no IPFS através do hash armazenado na blockchain, de acordo com o seu usuário logado;
- Conceder permissão: o usuário A pode conceder permissão de acesso a um arquivo que possui armazenado no IPFS para um usuário B, através de algumas informações como o hash armazenado na blockchain, o tempo específico que a permissão vai ter de validade, o endereço do usuário B e a chave pública do usuário B;
- Visualizar arquivos com permissão: um usuário que teve uma permissão concedida por outro usuário a um arquivo, pode ter acesso para visualizar esse arquivo dentro do tempo que foi estipulado pelo usuário que cedeu a permissão.

Já com relação aos requisitos não funcionais, foram elencados:

- DApp: a aplicação possuirá as características de uma aplicação descentralizada;
- Armazenamento da aplicação: a aplicação não armazena nenhum dado de forma permanente e centralizada, apenas temporariamente;
- Segurança dos dados: os arquivos enviados ao IPFS serão criptografados pelo algoritmo AES. O hash, que identifica o local dos dados, e a chave privada do AES (utilizada para criptografar o arquivo) serão criptografados utilizando o RSA ou o ECC, antes de serem armazenados na blockchain, assim permitindo que o usuário consiga acessar novamente os seus dados de forma segura. A chave privada compartilhada na permissão será criptografada pelo AES.

A Figura 1 mostra a arquitetura da solução com os fluxos de troca de informações entre os seis componentes. A seguir são descritos os componentes da arquitetura.

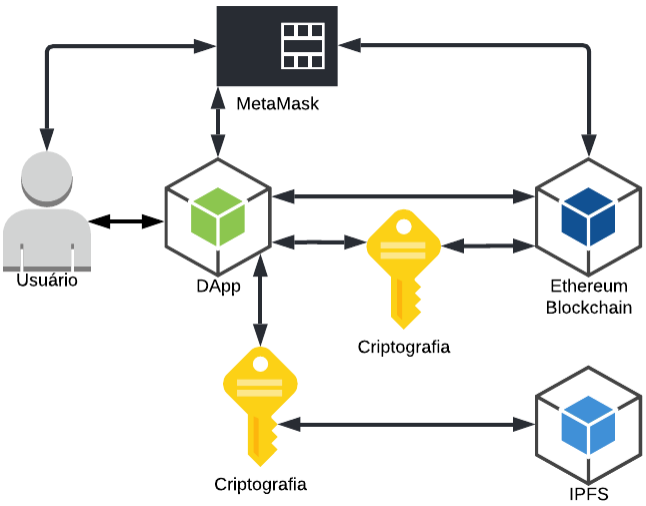
- Usuário: vai realizar o acesso a aplicação, enviar e receber dados (informações, arquivos e permissões), além de confirmar as transações e pagar a taxa da persistência desses dados na blockchain;
- DApp: responsável por se comunicar com a interface da carteira digital (MetaMask), com a blockchain (Ethereum) enviando e recebendo os dados, com o banco descentralizado (IPFS) enviando os arquivos e tratando o hash de retorno, com o usuário recebendo e mostrando os dados requisitados, e por fim, realizar a criptografia/descriptografia do hash, da chave privada do AES, do arquivo e da chave privada compartilhada na concessão de permissão;
- Interface de Carteira Digital (MetaMask): responsável por gerenciar as transações realizadas na DApp, solicitando a confirmação do usuário e debitando a taxa para a persistência dos dados na blockchain;
- Blockchain (Ethereum): responsável por guardar os contratos inteligentes que regem como as informações pertinentes do arquivo e da permissão serão armazenadas, como por exemplo, o hash que é utilizado para se ter acesso ao arquivo no IPFS, o endereço do usuário que realizou uma transação e o endereço do usuário a qual está sendo cedida a permissão, realizando dessa forma o link do respectivo usuário a determinada informação dentro dos contratos inteligentes;
- Sistema de arquivos descentralizado (IPFS): responsável por receber os arquivos criptografados da DApp, armazená-los e devolver um respectivo hash único que indica onde se encontra essa determinada informação na rede do IPFS;
- Criptografia: esse é o componente chave para garantir a segurança e privacidade dos dados armazenados na arquitetura proposta, pois através dos algoritmos de criptografia RSA ou ECC a aplicação criptografa o hash e a chave privada do AES antes de enviar essas informações para a blockchain; Com o AES é feita a criptografia do arquivo e da chave privada compartilhada na concessão da permissão. Dessa forma, somente o usuário que possui acesso a chave privada para descriptografar esse hash e a chave privada do AES, que vai conseguir ter acesso às informações armazenadas. Este processo, garante que as informações na blockchain e no IPFS fiquem seguras.

4.1 Funcionamento da Arquitetura

O diagrama de atividades da Figura 2 busca demonstrar a interação que ocorre entre os componentes para o envio de um arquivo.

Inicialmente, o usuário abre a DApp e é necessário que ele realize o login através do MetaMask para conseguir acessar as funcionalidades da DApp. Após o

Figura 1 – Esquema da arquitetura



Fonte: Do autor, 2022

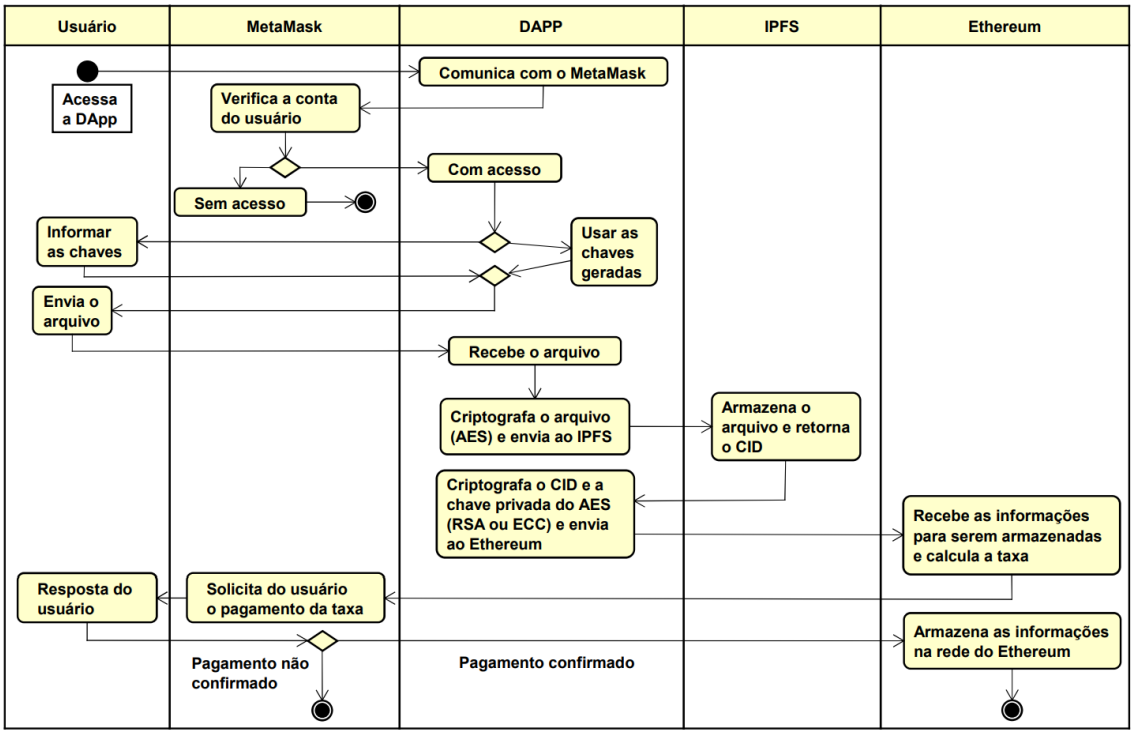
login realizado, a DApp vai utilizar as chaves criptográficas informadas pelo usuário ou as chaves criptográficas geradas pela própria DApp para criptografar as informações. Na sequência o usuário seleciona o arquivo para envio, a DApp criptografa o arquivo com o algoritmo AES e o envia para ser armazenado no IPFS, o qual retorna o CID, hash que identifica onde se encontra o arquivo do usuário na rede do IPFS. Dessa forma, a DApp criptografa a chave privada do AES, utilizada para criptografar o arquivo do usuário, e o CID do IPFS com um dos algoritmos (RSA ou ECC). Assim com as informações sensíveis protegidas, elas podem ser armazenadas na blockchain do Ethereum, o qual solicita ao usuário, através do MetaMask, o pagamento da taxa de processamento para que os dados sejam armazenados, o usuário então pode rejeitar e cancelar o processo ou realizar o pagamento e ter os dados armazenados.

O funcionamento das permissões é semelhante ao funcionamento do envio de um arquivo, entretanto não é necessário ser enviado outro arquivo ao IPFS, pois é feito o compartilhamento das informações necessárias para que o usuário B consiga ter acesso ao arquivo de um usuário A, como por exemplo o CID e a chave privada do AES. Assim, o usuário A pode enviar uma permissão através do fornecimento de algumas informações, como a chave pública do usuário B, o endereço da carteira digital do usuário B e o tempo de validade da permissão definido pelo usuário A.

O front-end e o back-end da DApp foram desenvolvidos com JavaScript, HTML5, React e o Node.js, e a parte do contrato inteligente em Solidity. O código[1] para a construção da DApp foi adaptado a partir de revisões feitas nas documentações das tecnologias descritas, alguns projetos de código aberto descentralizados desenvolvidos por McCubbin (2022) e também pesquisas nas documentações de bibliotecas de criptografia. Mais detalhes sobre a implementação da arquitetura estão

[1] Código disponível em: https://github.com/SegaBR/DApp

Figura 2 – Diagrama de atividades da arquitetura



Fonte: Do autor, 2022

disponíveis no relatório técnico [2].

5 Avaliação da Arquitetura

Foi conduzida uma avaliação da arquitetura buscando verificar o impacto da utilização da criptografia com os dois algoritmos: RSA e ECC. O objetivo é analisar o desempenho de toda a arquitetura proposta em conjunto com o AES, considerando a implicação da criptografia: a) do hash que o IPFS retorna para a aplicação, b) da chave privada do AES utilizada na criptografia do arquivo pelo algoritmo AES, e também c) da chave privada do AES utilizada pelo algoritmo AES na criptografia da chave privada do outro usuário. Essas informações criptografadas são armazenadas na blockchain.

Os algoritmos foram avaliados quanto aos critérios de tempo de execução, uso de memória e a influência no aumento do custo da taxa de envio das informações à rede blockchain. A avaliação não considera o tempo de envio dos arquivos para o IPFS, pois muitas variáveis, como conexão de internet, banda, tráfego de rede, entre outros fatores instáveis, podem afetar e deixar os resultados imprecisos.

O ECC é um algoritmo amplamente utilizado nos sistemas de blockchain, como por exemplo a rede do Bitcoin e do Ethereum. O RSA é um algoritmo que se tornou base e foi amplamente utilizado para transmissão de dados de forma

[2] Relatório Técnico disponível em: https://doi.org/10.48550/arXiv.2207.09919

segura em vários sistemas (SUMA; BOUHMALA; WANG, 2021). Assim, a avaliação desses dois algoritmos assimétricos, que são utilizados amplamente nos sistemas atuais, tem como escopo mostrar como eles vão performar dentro da arquitetura proposta, devido principalmente à característica de tamanho de chave necessária. Segundo recomendações da National Institute of Standards and Technology (NIST), o tamanho de chave de 3072 bits para o RSA se equipara a chave 256 bits para o ECC em questão de segurança, fornecendo um nível de 128 bits de segurança, que é o ideal indicado para sistemas além do ano de 2030 (BARKER; ROGINSKY, 2019). Assim o trabalho adota estes tamanhos de chaves. Para o ECC é feito o uso do Elliptic Curve Integrated Encryption Scheme (ECIES)[3] e da curva secp256k1[4] que é a mesma utilizada pela rede do Bitcoin e do Ethereum.

Os mesmos parâmetros equivalentes de teste foram utilizados para cada algoritmo e foram enviados três tamanhos diferentes de arquivos, um arquivo de 200KB (representando um arquivo de laudo de exame simples), 2MB (representando um arquivo de laudo mais complexo ou de imagem simples) e 30MB (representando um arquivo de imagem mais complexo).

Os critérios avaliados para cada algoritmo, bem como para a abordagem sem criptografia, são:

- Tempo de execução: medido o tempo do início do envio de um arquivo ou permissão até o final do processo, sendo realizado através do framework do Truffle que permite realizar testes de envio de informações para a blockchain e da interface do Performance[5] que disponibiliza informações de desempenho relacionadas ao tempo;
- Uso de memória: através de recursos nativos do próprio Node.js que fornece informações sobre o consumo de memória junto com o framework do Truffle;
- Impacto da criptografia no custo da taxa: foram utilizadas as informações fornecidas pelo Ganache e MetaMask que mostram o "gas" utilizado para realizar a transação.

O ambiente para a realização dos testes é formado pelo Ganache, criando uma simulação da blockchain do Ethereum de forma local sem conexão com a rede principal ou de testes, e o Truffle para a execução dos testes locais, assim obtendo as informações de memória, tempo de execução e o custo aproximado. As configurações da máquina utilizada para o processamento dos testes são: Intel(R) Core(TM) i5-8400 CPU - 2.80GHz, 8 GB de memória RAM e sistema operacional Windows 10 Pro.

[3] ECIES: esquema de criptografia híbrido que combina criptografia assimétrica baseada em ECC com simétrica para fornecer a criptografia de dados através da chave privada e chave pública ECC correspondente.

[4] secp256k1: é uma curva elíptica definida pela curva $y^2 = x^3 + 7$ *sobre o corpo finito* $p = 2^{256} - 2^{32} - 977$.

[5] Performance: interface que suporta medições de latência do lado do cliente em aplicações, presente também no Node.js, permite utilizar do método "performance.now()" representando o tempo com números de ponto flutuante com precisão de até microssegundos.

Antes da execução dos testes, simulando a interação do usuário com a aplicação, foi realizada a migração dos contratos para a blockchain teste do Ganache. Os testes se baseiam no envio de um arquivo ou no envio de uma permissão de um arquivo, onde é feito o tratamento das informações e a criptografia, além de ser realizada a confirmação e validação das informações enviadas para a blockchain.

5.1 Resultados da Avaliação

A partir dos testes se obteve os resultados quanto aos três critérios de avaliação, os quais podem ser observados nas Tabelas 1, 2 e 3 para cada tamanho de arquivo enviado 200KB, 2MB e 30MB, respectivamente e na Tabela 4, os resultados sobre o envio da permissão.

Nos resultados do primeiro teste para o arquivo de 200KB, apresentados na Tabela 1, podemos observar que com relação ao uso de memória, obteve um acréscimo de apenas 1% das abordagens com criptografia em relação a sem criptografia, representando um aumento médio de 6,73MB, o uso de memória do RSA e do ECC foi semelhante. Em relação ao aumento do tempo de execução, os valores são bem maiores, tanto o RSA quanto o ECC tiveram um aumento de mais de 50% em relação à abordagem sem criptografia, sendo o maior aumento do RSA. O custo teve um aumento de mais de 250%, com o RSA tendo uma diferença de 36,94% maior no custo em comparação com o ECC.

Tabela 1 – Resultados do envio do arquivo de 200KB

	Sem criptografia	**RSA**	**ECC**
Tempo de Execução (ms)	2977,280	5310,352	4675,219
Uso de memória (MB)	618,168	625,351	624,461
Aumento do custo de envio (ETH)	0,00765303	0,03170854875	0,02888082

Fonte: Do autor, 2022

Nos resultados do segundo teste para o arquivo de 2MB, apresentados na Tabela 2, podemos observar que o uso de memória atingiu um acréscimo semelhante se comparado com os resultados do primeiro teste (1.2%), representando 7,5MB, o uso de memória do RSA e do ECC foi semelhante. Em relação ao aumento do tempo de execução, tanto o RSA quanto o ECC tiveram um aumento de mais de 50% em relação à abordagem sem criptografia, sendo o maior aumento do RSA com 78,24%. O custo também teve um aumento próximo dos resultados da Tabela 1.

Tabela 2 – Resultados do envio do arquivo de 2MB

	Sem criptografia	**RSA**	**ECC**
Tempo de Execução (ms)	3055,597	5446,377	4764,175
Uso de memória (MB)	625,159	633,413	632,037952
Aumento do custo de envio (ETH)	0,00742803	0,031695981428	0,02843082

Fonte: Do autor, 2022

Os resultados do terceiro teste para o arquivo de 30MB, apresentados na Tabela 3, apontam um acréscimo significativo com relação ao uso de memória, mais de 5%, sendo o maior aumento entre todos os testes realizados, das abordagens com criptografia em relação a sem criptografia, representando aproximadamente 42MB. Já o aumento entre o uso de memória do RSA e do ECC permaneceu semelhante. O aumento do tempo de execução continuou constante, o RSA com 76% e o ECC com 57% em relação à abordagem sem criptografia, sendo o maior aumento do RSA com uma diferença de 19% comparado com o ECC. O aumento do custo continuou constante de acordo com os resultados da Tabela 1 e 2.

Tabela 3 – Resultados do envio do arquivo de 30MB

	Sem criptografia	**RSA**	**ECC**
Tempo de Execução (ms)	3205,593	5658,640	5049,206
Uso de memória (MB)	682,622	725,151	724,832
Aumento do custo de envio (ETH)	0,00742995	0,031745358	0,02843082

Fonte: Do autor, 2022

Nos resultados do quarto teste, para o envio de uma permissão para um arquivo, apresentados na Tabela 4, pode-se observar que o uso de memória teve um pequeno acréscimo de 1% das abordagens com criptografia em relação a sem criptografia, representando aproximadamente 7,5MB, com o uso de memória do RSA e do ECC sendo bem próximas. Já o tempo de execução teve um aumento mais expressivo, sendo de 122,86% para o RSA e 103,96% para o ECC em relação à abordagem sem criptografia, demonstrando uma diferença entre os dois de 18,9%. O custo do envio da permissão foi próximo do custo do envio de arquivo, com o ECC apresentando um aumento de 280,62%, uma diferença de 42,01% em comparação com o aumento de custo que o RSA apresentou de 322,24%.

Tabela 4 – Resultados do envio de uma permissão de um arquivo

	Sem criptografia	**RSA**	**ECC**
Tempo de Execução (ms)	2548,37	5679,21	5197,53
Uso de memória (MB)	619,23	626,87	626,76
Aumento do custo de envio (ETH)	0,007992006	0,033745266	0,03038832

Fonte: Do autor, 2022

A diferença de aumento do custo entre o RSA e o ECC no envio dos dados é devido ao tamanho da chave do RSA que é relativamente maior em comparação com a implementação do ECC em questão de bytes, gerando dados criptografados com uma quantidade maior de bytes, o que acaba impactando no custo que está atrelado a quantidade de informação enviada para a blockchain.

Analisando os resultados obtidos com os diferentes arquivos (200KB, 2MB e 30MB), ficou evidenciado que não ocorreu uma diferença expressiva, sendo esta compatível com o tamanho do arquivo. Destaca-se que quanto ao custo não teve

nenhum aumento relacionado diretamente com a diferença do tamanho dos arquivos, pois sua variação se deve a fatores como por exemplo: aumento de dados enviados, preço da taxa de "gas" da rede, ocupação da rede.

Mesmo com o uso de memória apresentando uma diferença pouco significativa entre os algoritmos RSA e ECC, é possível observar que o ECC apresentou valores de tempo de execução e de aumento do custo de envio mais satisfatórios, se demonstrando o algoritmo com o melhor potencial para ser utilizado dentro de uma arquitetura voltada para a blockchain.

6 Considerações Finais

Este artigo apresentou uma arquitetura que permite o gerenciamento do armazenamento dos dados de saúde do usuário de forma descentralizada através da blockchain e utilizando da criptografia com o intuito de garantir a privacidade dos dados, além de colocar o controle desses dados sob responsabilidade do usuário. Com relação a criptografia foram implementadas duas abordagens com uso dos algoritmos RSA e ECC a fim de verificar o impacto na arquitetura. Já o algoritmo AES foi usado para criptografar os arquivos enviados para a plataforma IPFS.

O desenvolvimento da arquitetura mediante ao conjunto das tecnologias e ferramentas descritas e com a aplicação da criptografia sobre os dados sensíveis, pode ser utilizada e aplicada dentro de cenários reais, com um custo de implementação inicial baixo devido ao uso de uma blockchain pública.

O trabalho avaliou as técnicas de criptografia utilizadas, buscando fornecer um paralelo entre dois dos principais algoritmos de chave assimétrica, podendo auxiliar na escolha de uma das técnicas de criptografia dependendo da aplicação que está sendo construída.

Através dos resultados dos testes realizados com os dois algoritmos, pode-se observar que a criptografia teve um impacto sobre os critérios avaliados, principalmente, em relação ao custo, com um aumento acima de 250%, e ao tempo de execução, sendo o RSA o algoritmo com o maior aumento acima de 70%. Já em relação a memória a diferença foi menos significativa entre os algoritmos, ficando o aumento do uso de memória atrelado ao tamanho do arquivo enviado. O impacto que a criptografia teve sobre a arquitetura é válido e justificável devido a toda a privacidade e a segurança sobre os dados que são fornecidas pela combinação da arquitetura com a criptografia.

Em trabalhos futuros, considera-se a realização de testes e coleta de dados para a avaliação do desempenho de outros algoritmos de criptografia dentro do mesmo cenário, e em cenários utilizando de outras blockchains como base. Também pode ser verificada a possibilidade de se utilizar na arquitetura, em conjunto com a criptografia, métodos de anonimização, como por exemplo, o k-anonimato e a privacidade diferencial, com o intuito de aumentar a privacidade do usuário.

An architectural to protect the privacy of health data stored in the blockchain

Christofer L. Sega[‖] Anubis G. de M. Rossetto [**]
Valderi R. Q. Leithardt [††]

2022

Abstract

With the rapid development of blockchain technology in recent years, its application in scenarios that need privacy, such as health area, began to be encouraged and discussed. This work presents an architecture to guarantee the privacy of data related to the health area, which are stored and shared within a blockchain network in a decentralized way, through the use of cryptography with the RSA, ECC and AES algorithms. Evaluation tests were carried out in order to verify the impact of cryptography on the proposed architecture in terms of cost, memory usage and execution time. The results demonstrate an impact, mainly, on the execution time and on the increase in the cost of sending data to the blockchain, however, justifiable considering the privacy and security provided with the architecture and encryption.

Keywords: Blockchain; Cryptography; DApp; Health data; Privacy.

Referências

AGUIAR, E. J. d. *Um framework baseado em blockchain para preservar a privacidade no compartilhamento de dados de saúde*. Dissertação (Mestrado) — Universidade de

[‖]<christofersega@gmail.com>
[**]<anubis.rossetto@passofundo.ifsul.edu.br>
[††]<valderi@ipportalegre.pt>

São Paulo, 2021. Disponível em: <https://www.teses.usp.br/teses/disponiveis/55/55134/tde-16022021-152627/en.php>. Citado na página 8.

BARKER, E.; ROGINSKY, A. Transitioning the use of cryptographic algorithms and key lengths. *NIST Special Publication 800-131A Revision 2*, 2019. Disponível em: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf>. Citado na página 14.

DABIT, N. *Contract ABI Specification*. 2021. Disponível em: <https://www.freecodecamp.org/news/what-is-web3/>. Acesso em: 10 jan 2022. Citado na página 8.

DASGUPTA, D.; SHREIN, J. M.; GUPTA, K. D. A survey of blockchain from security perspective. *Journal of Banking and Financial Technology*, Springer, v. 3, n. 1, p. 1–17, 2019. Disponível em: <https://link.springer.com/article/10.1007/s42786-018-00002-6>. Citado na página 2.

ETHEREUM. *Introdução aos Smart Contracts*. 2017. Disponível em: <https://solidity-portuguese.readthedocs.io/pt/latest/introduction-to-smart-contracts.html>. Acesso em: 10 jan 2022. Citado na página 4.

ETHEREUM. *Contract ABI Specification*. 2021. Disponível em: <https://docs.soliditylang.org/en/develop/abi-spec.html>. Acesso em: 10 jan 2022. Citado na página 4.

ETHEREUM. *Intro To Ethereum*. 2022. Disponível em: <https://ethereum.org/en/developers/docs/>. Acesso em: 10 jan 2022. Citado 2 vezes nas páginas 6 e 7.

FENG, Q. et al. A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications*, v. 126, p. 45–58, 2019. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804518303485>. Citado 2 vezes nas páginas 2 e 3.

GAN, C. et al. Blockchain-based access control scheme with incentive mechanism for ehealth systems: patient as supervisor. *Multimedia Tools and Applications*, Springer, p. 1–17, 2020. Disponível em: <https://link.springer.com/article/10.1007/s11042-020-09322-6#citeas>. Citado na página 9.

HEWA, T.; YLIANTTILA, M.; LIYANAGE, M. Survey on blockchain based smart contracts: Applications, opportunities and challenges. *Journal of Network and Computer Applications*, Elsevier, v. 177, p. 102857, 2021. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804520303234>. Citado na página 3.

INFURA. *The Infura Ethereum API*. 2022. Disponível em: <https://infura.io/product/ethereum>. Acesso em: 10 jan 2022. Citado na página 8.

IPFS. *How IPFS works*. 2022. Disponível em: <https://ipfs.io/#how>. Acesso em: 10 jan 2022. Citado 2 vezes nas páginas 7 e 8.

KALODNER, H. et al. Arbitrum: Scalable, private smart contracts. In: *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, 2018. p. 1353–1370. ISBN 978-1-939133-04-5. Disponível em: <https://www.usenix.org/conference/usenixsecurity18/presentation/kalodner>. Citado na página 4.

MAHTO, D.; YADAV, D. K. Rsa and ecc: a comparative analysis. *International journal of applied engineering research*, v. 12, n. 19, p. 9053–9061, 2017. Disponível em: <https://www.ripublication.com/ijaer17/ijaerv12n19_140.pdf>. Citado 2 vezes nas páginas 5 e 6.

MCCUBBIN, G. *DAPP University*. 2022. Disponível em: <https://github.com/dappuniversity>. Acesso em: 10 jan 2022. Citado na página 12.

MENDONÇA, R. et al. Tratamento de concessão e revogação de acesso a registros eletrônicos de saúde em blockchain. In: *Anais do IV Workshop em Blockchain: Teoria, Tecnologias e Aplicações*. Porto Alegre, RS, Brasil: SBC, 2021. p. 100–113. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/wblockchain/article/view/17133>. Citado na página 9.

METAMASK. *Introduction*. 2021. Disponível em: <https://docs.metamask.io/guide/>. Acesso em: 10 jan 2022. Citado na página 8.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, p. 9, 2008. Disponível em: <https://bitcoin.org/bitcoin.pdf>. Citado na página 2.

NODE.JS. *About Node.js*. 2022. Disponível em: <https://nodejs.org/en/about/>. Acesso em: 10 jan 2022. Citado na página 8.

OCHÔA, I. S. et al. Fakechain: A blockchain architecture to ensure trust in social media networks. In: _____. [S.l.: s.n.], 2019. p. 105–118. ISBN 978-3-030-29237-9. Citado na página 3.

OLIVEIRA, P. Roberto de et al. Energy consumption analysis of the cryptographic key generation process of rsa and ecc algorithms in embedded systems. *IEEE Latin America Transactions*, v. 12, n. 6, p. 1141–1148, 2014. Citado na página 5.

OLIVEIRA, R. Criptografia simétrica e assimétrica: os principais algoritmos de cifragem. *Revista Segurança Digital*, v. 5, p. 11–24, 03 2012. Citado 2 vezes nas páginas 5 e 6.

OMAR, A. A. et al. Privacy-friendly platform for healthcare data in cloud based on blockchain environment. *Future Generation Computer Systems*, v. 95, p. 511–521, 2019. ISSN 0167-739X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X18314201>. Citado na página 8.

PATIL, P. et al. A comprehensive evaluation of cryptographic algorithms: Des, 3des, aes, rsa and blowfish. *Procedia Computer Science*, v. 78, p. 617–624, 2016. ISSN 1877-0509. 1st International Conference on Information Security Privacy 2015. Disponível

em: <https://www.sciencedirect.com/science/article/pii/S1877050916001101>. Citado na página 6.

REACT. *React*. 2022. Disponível em: <https://pt-br.reactjs.org/>. Acesso em: 10 jan 2022. Citado na página 8.

SARAIVA, D. A. et al. Prisec: Comparison of symmetric key algorithms for iot devices. *Sensors*, MDPI, v. 19, n. 19, p. 4312, 2019. Citado na página 6.

SHAHNAZ, A.; QAMAR, U.; KHALID, A. Using blockchain for electronic health records. *IEEE Access*, v. 7, p. 147782–147795, 2019. Citado na página 9.

SHI, S. et al. Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey. *Computers Security*, v. 97, p. 101966, 2020. ISSN 0167-4048. Disponível em: <https://www.sciencedirect.com/science/article/pii/S016740482030239X>. Citado 3 vezes nas páginas 2, 4 e 7.

SINGH, S. R.; KHAN, A. K.; SINGH, S. R. Performance evaluation of rsa and elliptic curve cryptography. In: *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. [S.l.: s.n.], 2016. p. 302–306. Citado na página 5.

SUMA, V.; BOUHMALA, N.; WANG, H. *Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020*. [S.l.]: Springer, 2021. 837-844 p. Citado na página 14.

TRUFFLESUITE. *Home*. 2022. Disponível em: <https://trufflesuite.com/index.html>. Acesso em: 10 jan 2022. Citado na página 8.

WÜST, K.; GERVAIS, A. Do you need a blockchain? In: IEEE. *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. [S.l.], 2018. p. 45–54. Citado na página 6.

ZHANG, R.; XUE, R.; LIU, L. Security and privacy on blockchain. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 52, n. 3, jul 2019. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3316481>. Citado 3 vezes nas páginas 2, 3 e 4.