DERIS-B: Identificação de Pessoas via Blockchain

Gustavo F. dos Santos, Renata H. S., Reiser, Maurício L. Pilla

[1]Centro de Desenvolvimento Tecnológico – Universidade Federal de Pelotas (UFPEL)
Caixa Postal 96.010-610 – Pelotas – RS – Brasil

{gfdsantos, reiser, pilla}@inf.ufpel.edu.br

Abstract. *A important current problem is directly relationed with digital identification of people. With focus in a decentralized way to develop a digital identification, this work proposes to solve a problem of decentralized networks where users are exclusevely identificated by a public key. Was developed a set of protocol based in a programable blockchain, such Ethereum. As result, we achive a prototype that implements a simplified version of the protocol, a set of unity tests and a short economical analysis of the operational costs of these protocols in the Ethereum.*

Resumo. *Um importante problema atual está relacionado com a identificação digital de pessoas. Com foco em desenvolver uma forma descentralizada de identificação digital, este trabalho se propõe a resolver um problema das redes descentralizadas onde um usuário é identificado exclusivamente por uma chave pública. Para tal foi desenvolvido um conjunto de protocolos baseado em blockchain programável, como o Ethereum. Foi obtido como resultados um protótipo que implementa uma versão simplificada do protocolo proposto, um conjunto de testes unitários e uma breve análise de custos econômicos operacionais destes protocolos no Ethereum.*

1. Introdução

Embora a internet tenha sido desenvolvida para ser uma forma descentralizada de comunicação entre computadores, a web torna-se centralizada cada vez mais com empresas privadas sendo detentoras da maior parte do tráfego de informação na internet [Staltz 2016]. Com o crescimento de serviços privados de identificação, como serviços de e-mail e redes sociais, uma mesma pessoa pode ter múltiplos perfis digitais, falsificando informação.

Embora pessoas sejam livres para criar seus perfis em redes sociais, tornando sua vida digital completamente diferente da sua vida não-digital; esta liberdade também esconde a real identidade de pessoas que propagam o ódio na web. Desta forma neste trabalho buscou-se desenvolver uma forma identificação que não seja atrelada a uma entidade central, que forneça uma única forma de identificação digital e que seja possível alcançar a mesma flexibilidade que sistemas centralizados possuem, como a fácil troca de senhas.

Ao lidar com identificação descentralizada, algumas informações pessoais precisam ser imutáveis para garantir que uma identidade seja preservada, como pares de chaves criptográficas. Toda identidade em sistemas descentralizados é identificada através de um par de chaves, onde uma chave é pública e a outra é privada. A última deve ser mantida em segredo para evitar que outros realizem o roubo da identidade.

Sistemas centralizados como o Google conseguem lidar com a perda de senhas atuando como entidade central, detentora de todos os dados envolvidos no sistema, capaz de prover segurança para uma pessoa trocar a sua chave privada. A entidade central, no entanto, exige a confiança dos usuários, o que nem sempre é possível ou razoável.

O principal objetivo deste trabalho é definir uma forma de identificação via *blockchain*, através de contratos digitais, que oferece uma camada de processamento sobre informação imutável. Neste trabalho foi desenvolvido uma forma de habilitar que uma pessoa seja dona de uma identidade digital e que possa provar ser dona desta identidade, mas com a possibilidade de recuperar a identidade em caso de perda da senha.

A organização deste trabalho é a seguinte: na Seção 2 são apresentadas brevemente as principais tecnologias utilizadas no desenvolvimento do protótipo deste trabalho. A Seção 3 contém a descrição dos protocolos de identificação desenvolvidos. Na Seção 4, são apresentados resultados de custos de operação do protótipo inicial, que implementa os protocolos descritos na Seção 3. Por fim, a Seção 5 contém as discussões sobre este trabalho, bem como projeções de trabalhos futuros.

2. Principais Tecnologias

Neste trabalho foram utilizadas como base, duas tecnologias: Ethereum e IPFS. O Ethereum atua como a unidade de processamento com capacidade de garantir que operações foram realizadas corretamente, pelo fato de ser um computador global descentralizado [Buterin 2014]. O papel do IPFS resume-se a atuar como unidade de armazenamento distribuído [Benet 2014].

A capacidade do IPFS em indexar informação baseado no seu conteúdo é importante para este trabalho pois o armazenamento de informações na *blockchain* é caro em termos econômicos, uma forma de armazenamento descentralizado que garanta a consistência da informação é um tópico importante neste trabalho.

A seguir é discutido alguns tópicos principais sobre as principais tecnologias utilizadas durante o desenvolvimento do trabalho.

2.1. Blockchain

A *blockchain* é uma estrutura de dados em forma de grafo, distribuída ao longo de vários participantes, com a capacidade de manter o consenso sobre a informação armazenada. Um dos resultados do sistema de consenso de uma *blockchain* é uma cadeia de blocos única. Uma representação desta cadeia de blocos pode ser vista na Figura 1, onde o bloco azul é conhecido como bloco *Genesis* e os blocos verdes denotam os blocos válidos da *blockchain*. Todos os demais blocos brancos não existem na ramificação verdadeira e todas as informações armazenadas nestes blocos não são válidas.

O mecanismo de funcionamento de *blockchains* dá suporte a moedas criptográficas. O Bitcoin é o primeiro e maior caso de sucesso em moedas criptográficas. O sistema proposto por [Nakamoto 2008] busca uma forma de realizar transações financeiras entre duas partes cuja autenticidade possa ser provada sem o intermédio de um terceiro agente. As transações no Bitcoin são computacionalmente impossíveis de serem revertidas uma vez que são aceitas, mineradas e adicionadas na *blockchain*.

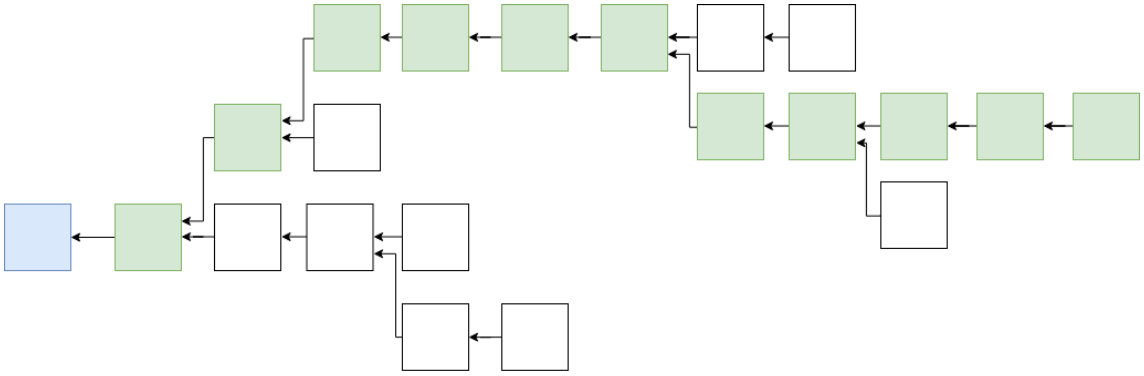

Figura 1. Representação simplificada de uma *blockchain* e suas ramificações.

2.2. Ethereum

Ethereum é uma plataforma genérica de computação onde todas as transações são baseadas em conceitos de máquinas de estado e é implementado sobre a arquitetura de *blockchain* [Wood 2014]. A ideia do Ethereum é juntar conceitos de execução de código, monetização virtual e protocolos que executam na *blockchain* para permitir o desenvolvimento de aplicações que demandam consenso arbitrário, escalabilidade, padronização e características de completude. Ethereum cumpre os requisitos ao adotar uma arquitetura baseada em *blockchain* com uma linguagem de programação Turing-completa embarcada [Buterin 2014].

O Ethereum funciona de acordo com uma máquina de estados baseada em transações definida sob uma arquitetura de *blockchain*. A partir de um estado inicial, computações são realizadas em sequência e o resultado culmina em um estado final de aceitação ou rejeição. A má formatação do conjunto de instruções que servem de entrada a máquina de estados causa uma parada e o estado atual é revertido ao estado inicial.

A especificação do Ethereum não provê uma implementação oficial. Os autores promovem um protocolo bem definido e implementações em diversas tecnologias são encorajadas pela comunidade. Nós Ethereum comunicam-se independente da forma como foram implementados, pois cada nó diferente segue a mesma especificação.

2.2.1. Contratos Digitais

Além de uma plataforma cripto-econômica, o Ethereum possui uma característica de poder executar programas durante o processamento de transações. Programas compilados para executarem na *blockchain* do Ethereum são chamados de contratos digitais [Wood 2014].

Qualquer usuário pode escrever e enviar um contrato digital para a *blockchain*. Um contrato após deposto na *blockchain* pode interagir com usuários e outros contratos digitais, por meio de transações que podem ou não alterar o estado interno do contrato digital.

Atualmente, contratos digitais do Ethereum podem ser implementados em diferentes linguagens de programação de alto nível. Porém, a representação em forma de linguagem de programação deve ser compilada para um código de máquina que é exe-

cutado pela EVM [Wood 2014]. Neste trabalho é utilizado a linguagem de programação Solidity [Dannen 2017] para a implementação dos contratos digitais que compõem a implementação dos protocolos propostos.

2.2.2. Gas

Gas é a unidade de consumo de processamento, usado para controlar a execução de contratos digitais pela EVM. O conceito de Gas impõe que todos os contratos da rede Ethereum sejam sujeitos a taxas. Estas taxas são calculadas a partir da execução do código dos contratos pelos mineradores. Cada transação em Ethereum exige uma quantidade limite de Gas que pode ser consumida, conhecido como *gas limit* e a quantidade de Gas dispendida a cada etapa da execução é calculada utilizando como parâmetro o *gas price*.

O conceito de Gas em Ethereum é equivalente à quantidade de combustível presente em um veículo. Para um veículo percorrer a distância da origem A ao destino B, é necessário uma certa quantidade de combustível, se o combustível acabar durante o trajeto, o veículo fica parado em algum ponto entre A e B. Equivalente no Ethereum, se a execução de um contrato necessitar de uma quantidade maior de Gas que a origem da transação está disposta a pagar, a execução é interrompida e o estado do contrato digital é revertido ao estado inicial.

2.3. IPFS

O *InterPlanetary File System* - IPFS é um sistema de arquivos distribuído *peer-to-peer* que busca conectar todos os dispositivos com o mesmo sistema de arquivos com base em uma estrutura de dados em forma de grafo acíclico chamado de Merkle DAG [Benet 2014].

O IPFS é nomeado desta forma por conta da sua definição: o sistema foi desenvolvido com foco em tecnologias futuras de exploração interplanetária, onde a comunicação entre a origem da informação e o ponto requisitante possui latência na ordem de minutos ou horas. O IPFS permite que uma versão de uma informação - uma parte da Internet, por exemplo - seja armazenada via IPFS em um local e ser atualizada de tempos em tempos com a versão mais atual via versionamento, ou seja, o IPFS permite a comunicação via internet entre diferentes planetas.

Vários projetos utilizam o IPFS como base para a construção de uma internet cujo endereçamento é feito com base no conteúdo e não na origem do conteúdo. Um arquivo pode residir em diferentes computadores, mas é acessado através de uma hash que representa este arquivo.

Diferentemente do protocolo HTTP, onde o conteúdo é endereçado baseado na sua localidade e tem como principal ponto de falha o servidor que armazena o conteúdo, o IPFS provê o endereço baseado no conteúdo [Benet 2014]. Cada arquivo é endereçado por uma hash (*Merkle link*) que representa o arquivo. O mecanismo de endereçamento do IPFS permite que o conteúdo acessado através da hash seja verificável, o que garante a consistência de arquivos.

3. Metodologia e Proposta de Protocolos

Foram desenvolvidos três protocolos diferentes: protocolo de registro de usuário, de troca de chaves onde o usuário possui a chave principal e um protocolo de troca de chaves onde o usuário não possui a sua chave principal. Estes protocolos são organizados em uma arquitetura do tipo Usuário/Autoridade, onde Autoridades atuam somente como usuários de confiança. Qualquer usuário pode ser uma Autoridade e servir de confiança para outros usuários, inclusive um usuário pode confiar nele mesmo.

Neste protocolo, usuários podem deixar de confiar em Autoridades, mas para isso uma pessoa deve confiar em pelo menos duas Autoridades, visto que uma autoridade é necessária para realizar a troca de dados na *blockchain* e qualquer troca de dados, por regra, resulta em um novo contrato digital de usuário. Mais detalhes serão discutidos a seguir.

O sistema criptográfico utilizado para a geração de chaves é o algoritmo de criptografia de curva elíptica - ECC [Kapoor et al. 2008]. O que motivou o uso deste sistema criptográfico é que, chaves assimétricas geradas através do algoritmo de criptografia de curva elíptica precisam de uma quantidade menor e bits para representar chaves cuja segurança é equivalente a um par de chaves gerado pelo algoritmo RSA [Kapoor et al. 2008]. É necessário que cada interação com o Ethereum utilize a menor quantidade de bytes possível.

A organização dos contratos digitais de usuário possui um mecanismo de ligação aos contratos adjacentes, pertencentes ao usuário. Assim, caso um contrato antigo seja acessado, é possível navegar até a versão mais recente do contrato digital de usuário. Na Figura 2 é possível esta estrutura, onde cada contrato digital pertencente a uma mesma identidade possui três endereços: um endereço que aponta para contrato digital anterior, que é escrito no momento da criação do contrato digital e não pode ser reescrito; um endereço que aponta para o contrato digital original de uma identidade, utilizado para o rápido acesso; um endereço que aponta para o próximo contrato, este endereço só pode ser escrito, em uma única vez por uma autoridade de confiança do usuário.

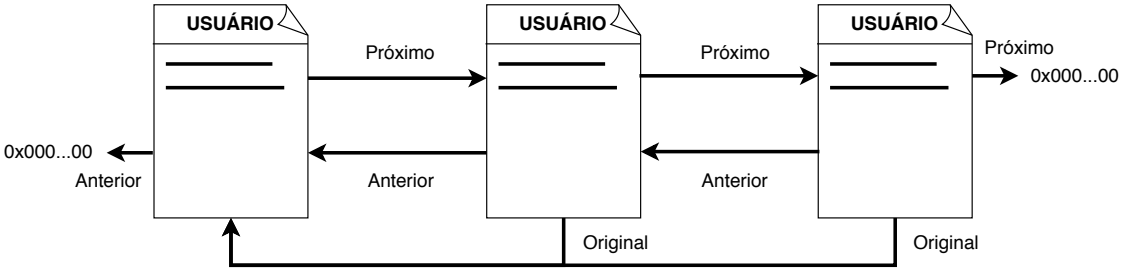

Figura 2. Estrutura de armazenamento de endereços nos contratos de usuários.

3.1. Protocolo de Registro de Usuário

Registrar pessoas deve ser um procedimento que ofereça segurança sem a necessidade de haver confiança em uma empresa privada ou órgão governamental sobre a segurança dos dados dos usuários quanto o acesso não autorizado [Thakur 2017].

No ato de registro a pessoa deve informar dados básicos de identificação, como por exemplo, nome, data e local de nascimento, local de residencia, entre outros. Estas

informações são tratadas como atributos pessoais. Também é informada uma chave pública, que faz parte de um par de chaves assimétrico, geradas no dispositivo que a pessoa estiver utilizado no ato do registro. Em um contexto de sistema autônomo, estes dados básicos e necessários são informadas ao sistema por meio de um aplicativo.

Outra informação necessária é o cadastro de pelo menos um dispositivo de segurança, onde dispositivos de segurança são parte da identificação digital de uma pessoa. Cada dispositivo de segurança possui um par de chaves atrelado, a chave pública desde dispositivo é armazenado publicamente no contrato digital do usuário. A chave privada é criptografada e armazenada localmente no dispositivo seguro. No momento atual, considera-se dispositivos seguros apenas computadores, *smartphones* e *tablets*, pois estes dispositivos possuem uma camada de sistema operacional que provê as ferramentas para o armazenamento seguro da chave privada. Por exemplo, *smartphones* podem encriptar a chave privada através do AES [Daemen and Rijmen 2001], onde a chave simétrica pode ser a impressão digital do dono do *smartphone*.

A comunicação entre pessoas e autoridades deve ser realizada sob um canal de comunicação que oferece criptografia ponta a ponta de mensagens. No diagrama presente na Figura 3 é possível ver o fluxo do processo de registro de contrato digital de usuário perante a um contrato digital de autoridade, onde esta autoridade já é uma autoridade de confiança para o usuário.

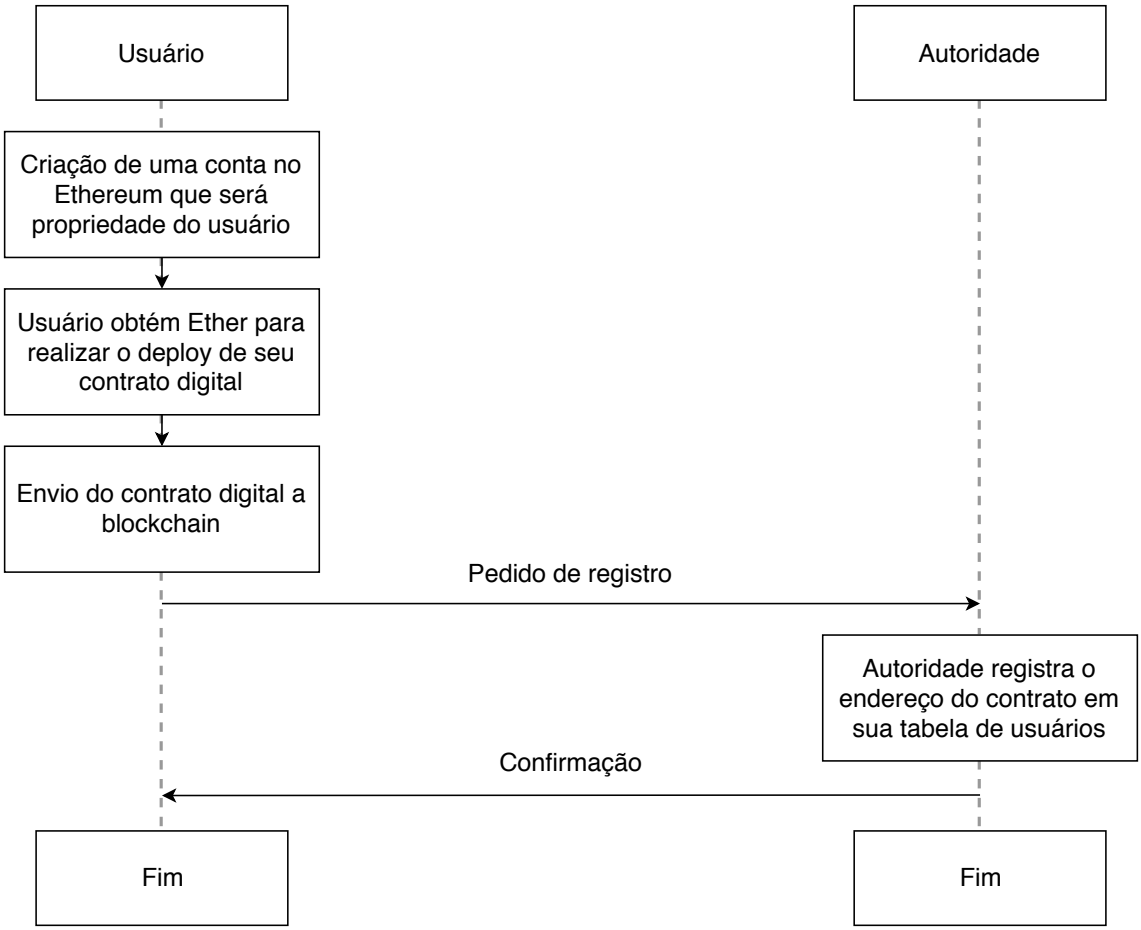

Figura 3. Protocolo de registro de uma pessoa em uma autoridade

Uma conta de usuário é criada na *blockchain* do Ethereum através da chave privada do usuário. Uma conta no Ethereum dá a possibilidade de assinar transações, o que é suficiente para provar que um usuário fez uma determinada transação. A autoridade envia uma quantidade de Ether à conta do usuário para que seja possível o envio do contrato digital do usuário a *blockchain*.

Os atributos de usuário são codificados em um formato JSON e adicionado no IPFS, que retorna um CID - uma hash que identifica um conteúdo. O conteúdo adicionado ao IPFS é informado a autoridade que replica a informação entre as demais autoridades. Os atributos do usuário são armazenados pelo usuário, pelas autoridades ou por qualquer usuário que acessar os atributos de um usuário. A Figura 4 mostra como o processo de armazenamento de atributos no IPFS ocorre. Um objeto é serializado no formato JSON, depois é convertido em um *buffer*, então é adicionado ao IPFS, os blocos deste objeto são armazenados localmente e uma *hash* que representa este objeto é criada.

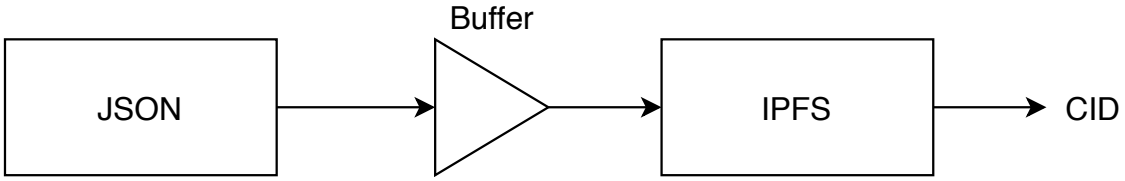

Figura 4. Fluxograma do armazenamento de um objeto no IPFS.

O CID dos atributos do usuário é adicionado ao contrato digital do usuário, assim como alguns atributos públicos para o acesso mais rápido, como o nome do usuário e chave pública. O contrato digital do usuário funciona tanto como ferramenta de identificação mas também como cache de informações que devem ser acessadas com maior rapidez.

Após o envio do contrato digital de usuário a *blockchain*, o usuário comunica a autoridade seu endereço de conta e seu contrato digital. Quando um registro é feito, ou seja, uma transação no Ethereum é minerada, validada e adicionada em um bloco, este contrato digital é identificado como contrato origem. No protocolo de registro de usuário não é necessário autenticação.

3.2. Protocolo de Autenticação

Como pode ser visto na Figura 5, uma autoridade, depois que recebe um pedido de autenticação, gera um segredo e este segredo é criptografado com a chave pública da pessoa que requisita a autenticação. Um pedido de autenticação é uma mensagem passada por um canal seguro que liga diretamente usuário e autoridade. A forma como qual este canal é implementado não consta na especificação do DERIS-B, contudo é proposto que o protocolo Whisper [Mohanty 2018] seja utilizado para esta tarefa. O segredo é um conjunto de informação com tamanho de 4096 bits, gerado pseudo-aleatoriamente durante o processo de autenticação e conhecido apenas pela autoridade.

A chave pública desta pessoa é armazenada publicamente em seu contrato digital, para o fácil acesso durante provas de identidade. Então o segredo criptografado é adicionado ao IPFS e o CID, que identifica esta porção de informação é enviado ao usuário através de um canal de comunicação seguro, que deverá provar que possui a chave privada correspondente descriptografando a informação.

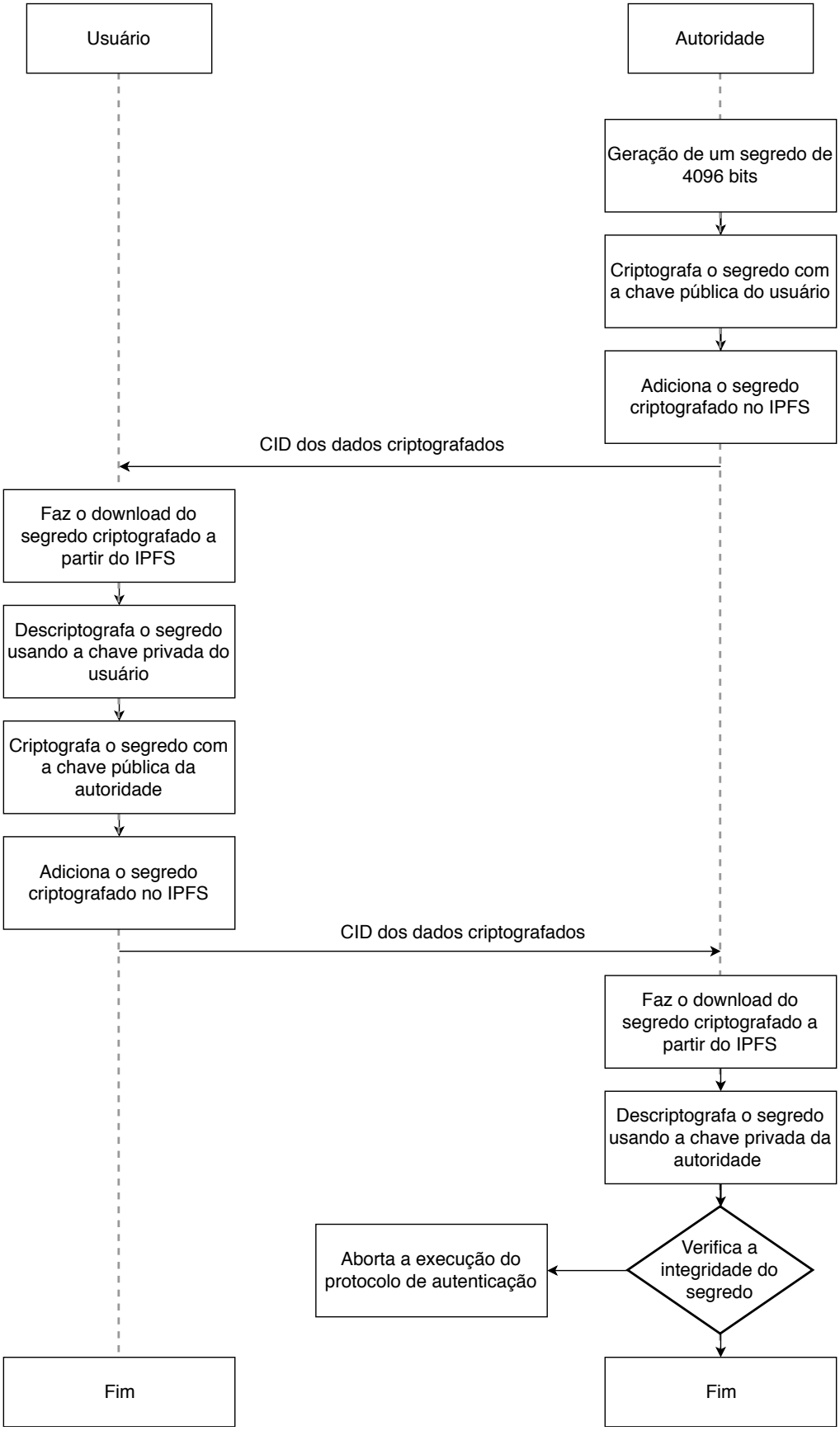

Figura 5. Protocolo de autenticação com chave privada.

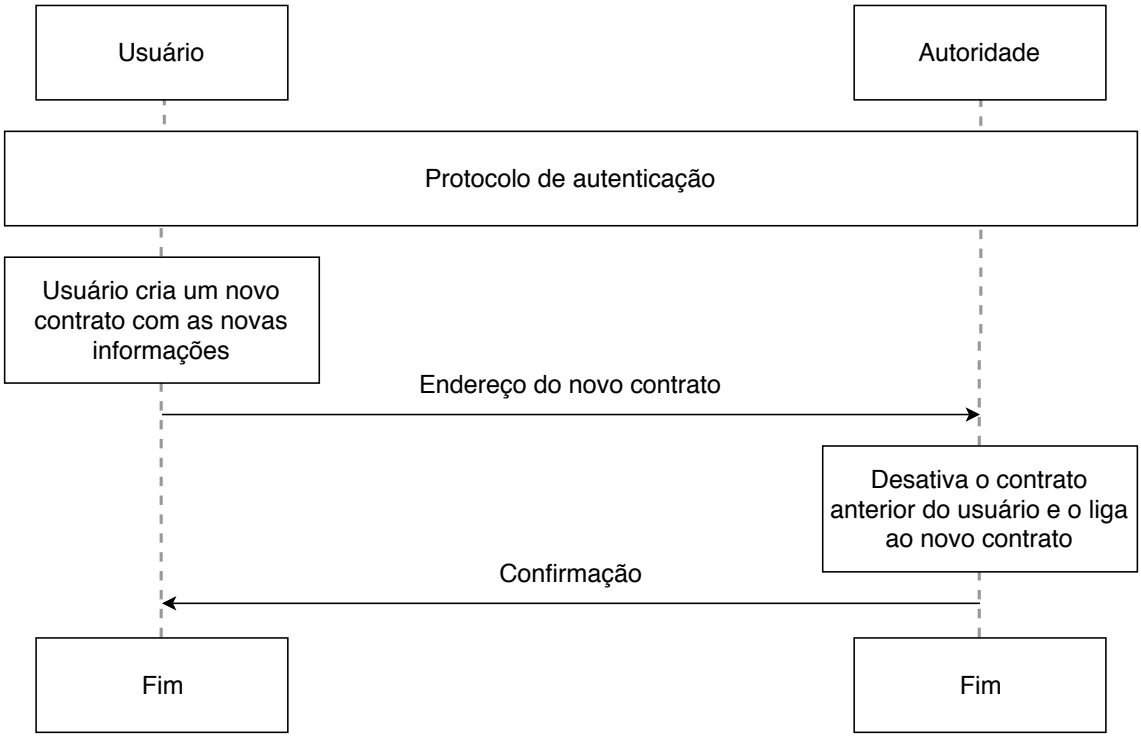

Figura 6. Protocolo de troca de dados.

Em seguida a informação descriptografada é criptografada novamente através da chave pública da autoridade, presente em seu contrato digital, então é adicionada ao IPFS e o CID referente a esta nova informação é enviada à autoridade correspondente, que deve provar possuir a chave privada corresponde a chave pública armazenada publicamente em seu contrato digital de autoridade.

3.3. Protocolo de Troca de Dados com Chave Privada

O protocolo de troca de dados foi proposto para dar suporte ao mecanismo de troca de senhas de usuários. Entretanto, em um universo regido por uma *blockchain*, algumas regras devem ser mantidas. Um diagrama simplificado sobre o protocolo de troca de dados pode ser visto na Figura 6.

Caso um usuário com porte da sua chave privada deseje alterar esta chave privada, ele comunica a autoridade este desejo. A autoridade gera um segredo, que é uma quantidade de informação, e encripta esta informação com a chave pública atual do usuário, acessível a partir de seu contrato digital. Após, a informação criptografada é enviada ao usuário. O usuário utiliza sua chave privada atual para descriptografar o segredo, em seguida ele encripta o segredo novamente, mas com a chave pública da autoridade, que é acessível a partir do contrato digital da autoridade.

Com o segredo criptografado com a chave pública da autoridade, o usuário envia esta quantidade de informação a autoridade que aplica sua chave privada para descriptografar o conteúdo do dado enviado pelo usuário e, então, realiza a checagem se a informação descriptografada é igual ao segredo inicialmente criado para a operação.

Se a integridade do segredo for mantida neste processo, esta é uma prova de que o usuário que diz ser dono de um determinado contrato, é realmente dono do contrato e

está apto a submeter uma nova chave pública.

A última participação do usuário no protocolo envolve gerar uma chave pública a partir da nova chave privada. A nova chave pública é enviada para a autoridade que cria um novo contrato baseado nas informações armazenadas no contrato de usuário anterior. Este conjunto de informações alimenta no novo contrato do usuário.

Após criar o contrato do usuário, a autoridade atualiza o contrato antigo, apontando ele para o novo contrato. Depois de desabilitado um contrato não pode mais realizar operações, ou seja, torna-se somente de leitura, para que seja possível provar que um usuário realizou uma assinatura de um documento em um determinado momento.

O último passo deste protocolo consiste na tarefa da autoridade em atualizar sua tabela interna de contratos, atualizando o valor da chave, que é o endereço do primeiro contrato do usuário, para o endereço do novo contrato do usuário.

3.4. Protocolo de Troca de Dados sem Chave Privada

Descobrir chaves privadas de criptografia assimétrica, com tamanhos razoáveis é uma tarefa muito difícil, considerando a arquitetura e competência dos computadores atuais [Kapoor et al. 2008]. Portanto, um espião precisará estar com a posse de um dispositivo de segurança que o usuário cadastrou e este dispositivo deve estar ativo.

Quando uma pessoa perde um dispositivo de segurança, ela deve desativar a validade da chave pública deste dispositivo, a pessoa só poderá realizar esta operação com posse de sua chave privada, comunicando-se com uma autoridade de confiança.

Quando um pedido de troca de senha é feito, sem a possibilidade de o usuário assinar digitalmente as transações, o protocolo deve acionar os dispositivos seguros, solicitando que algum destes dispositivos realize a assinatura do pedido de troca de chave pública. Caso o usuário tenha múltiplos dispositivos cadastrados, o primeiro dispositivo que responder a requisição da autoridade será utilizado ao longo do procedimento.

A desativação de um dispositivo de segurança pode ser feita a partir de outro dispositivo seguro ou através da chave privada principal da pessoa. A desativação consiste na remoção da chave pública do dispositivo, do contrato do usuário. Como chaves públicas são valores estáticos em contratos digitais, a desativação de um dispositivo seguro culmina em um novo contrato digital de usuário.

4. Resultados

Este trabalho tem como resultados um conjunto de protocolos de identificação de pessoas, um protótipo e testes unitários. O protótipo desenvolvido é composto por contratos digitais desenvolvidos com a linguagem de programação Solidity e uma biblioteca escrita em JavaScript, que pode ser utilizada através do NodeJS.

Os contratos digitais de usuário e autoridade implementados neste trabalho não levam em conta a presença de dispositivos seguros, que é uma característica fundamental presente nos protocolos propostos. Embora tenha sido mencionado na seção anterior uma forma de comunicação via *blockchain* através do protocolo Whisper, não foi implementada a comunicação entre autoridades e usuários, que é necessária no processo de autenticação. Em vez disto foram realizados testes unitários em cada método presente

nos contratos digitais. Os protocolos foram testados manualmente através da IDE Remix, onde toda a comunicação entre Usuário e Autoridade foi simulada.

Foi realizada uma bateria de testes unitários sobre os métodos dos contratos digitais de Usuário e Autoridade, que contabilizaram o processamento das operações. A Tabela 1 mostra resultados do custo de cada execução de método na *blockchain* no contrato digital de usuário.

Método	Custo	Custo (dólar)
construtor	1896137	$0.77647
registerToAuthority	44611	$0.01827
unregisterFromAuthority	14648	$0.00613
isRegisteredByAuthority	23375	$0.00957
signData	47373	$0.01940
isSigned	25961	$0.01063
getPublicKey	25547	$0.01045
getCID	23119	$0.00946
getOwner	21942	$0.00891
getOriginalContract	21920	$0.00897
getNextContract	21964	$0.00899
getLastContract	21986	$0.00901
disableAndLinkToNew	48840	$0.02001

Tabela 1. Custo por operações no contrato digital de usuários.

Já a Tabela 2 mostra o custo da execução de cada método no contrato digital de autoridade. Nota-se que contratos de autoridades são muito mais simples, pois a única função é realizar a desativação e ligação de contratos digitais.

Método	Custo	Custo (dólar)
construtor	620213	$0.25397
registerUser	44962	$0.01841
changeUserLatestContract	31507	$0.01291

Tabela 2. Custo por operações no contrato digital de autoridades.

Os valores de custo encontrados na Tabela 1 e na Tabela 2 são medidos em Wei e são constantes para qualquer chamada efetuada. O valor correspondente em dólar norte americano mostrado nas duas tabelas foi obtido através do site Eth Gas Station[1] com o valor de *gas price* em 2.1GWei.

Os valores de custos são medidos em cima do custo total para realizar a transação, o que engloba tanto o custo de cada *bytecode* executado pela EVM referente a chamada do método do contrato, quanto o custo para a execução, validação e escrita da transação no livro de registro.

[1] https://ethgasstation.info/calculatorTxV.php

5. Conclusões e Trabalhos Futuros

Blockchain é um mecanismo que não desempenha bem o papel de unidade de processamento, pois necessita de um mecanismo de consenso distribuído. Este mecanismo de consenso interfere diretamente na taxa de execução de transações. *Blockchains* também não são utilizadas como unidade de armazenamento, pois o armazenamento de informação extra é muito caro em termos econômicos. Embora exista um sistema de troca de mensagens no Ethereum, chamado de Whisper, não explorado neste trabalho, estudos sobre o desempenho da *blockchain* para realizar a comunicação por troca de mensagens são necessários.

A partir da necessidade do uso de um armazenamento externo, o sistema de indexação que o IPFS fornece encaixa-se perfeitamente no requisito deste trabalho. Atributos de usuário são armazenados no IPFS, reduzindo a necessidade de armazenamento de somente 256 bits na *blockchain*. Segundo os resultados obtidos, quanto menor a quantidade de informação a ser armazenada na *blockchain*, mais barato é o custo por transação ao manter o *gas price* constante.

Blockchains em geral oferecem uma capacidade de segurança e resiliência que sistemas centralizados não podem entregar, a menos que pessoas tenham confiança sem precedentes em empresas privadas.

O registro público de informações que a *blockchain* oferece em conjunto com a capacidade do Ethereum em realizar computação na *blockchain* foram fundamentais para a realização deste trabalho. Foi possível propor, implementar e testar uma forma de identificação de pessoas baseado em atributos públicos acessíveis via chamadas RPC.

Pessoas mudam de nome, movem-se para outra cidade, trocam de parceiros, trocam chaves privadas - o que implica em trocar as chaves públicas também e, no sistema desenvolvido, cujo protótipo foi testado unitariamente, mostra que é possível este tipo de operação com base em *blockchain* programável, como o Ethereum.

Nos protocolos apresentados neste trabalho, perfis de usuários são atrelados a contratos digitais, que são atrelados a endereços no Ethereum. Porém há um terceiro elemento: as autoridades. Autoridades são contratos controlados por entidades confiáveis. A adição do terceiro elemento permite que troca de informações ocorra com segurança e possibilita o acompanhamento do histórico de uma pessoa ao longo dos diferentes contratos digitais que a representam. Embora uma pessoa tenha utilizado uma chave privada em um determinado momento e, agora utilize uma nova chave privada, é possível provar que esta mesma pessoa assinou documentos e concordou com transações no passado, embora estas transações tenham ocorrido a partir de outro endereço de contrato digital de usuário e outro par de chaves.

Os próximos passos serão a validação completa quanto a aplicabilidade segura do sistema desenvolvido. Também busca-se a implementação completa dos protocolos propostos através de aplicações utilizáveis por pessoas, estas aplicações serão acessíveis através de pacotes instaláveis, tanto em *smartphones* quanto em computadores - os dispositivos de confiança. Também é um objetivo o desenvolvimento de um aplicativo web, para o acesso independente de sistema operacional.

Outro objetivo futuro é a avaliação quanto o comportamento dos protocolos de-

senvolvidos neste trabalho em diferentes sistemas de consenso. A abordagem seguida é genérica e aplicável sob qualquer sistema de consenso que exista no Ethereum, mas talvez possa ser otimizado para o uso em redes que executam Prova de Autoridade como consenso.

Referências

Benet, J. (2014). Ipfs - content addressed, versioned, p2p file system. Disponível em http://arxiv.org/abs/1407.3561. Acesso em: 20 dez. 2018.

Buterin, V. (2014). A next-generation smart contract and decentralized application platform. Disponível em: http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf. Acesso em: 20 dez. 2018.

Daemen, J. and Rijmen, V. (2001). Rijndael, the advanced encryption standard. *Dr. Dobb's Journal*, 26(3):137–139.

Dannen, C. (2017). *Introducing Ethereum and Solidity*. Apress, Brooklyn, NY.

Kapoor, V., Abraham, V. S., and Singh, R. (2008). Elliptic curve cryptography. *Ubiquity*, 2008(May):7:1–7:8.

Mohanty, D. (2018). Ethereum architecture. In *Ethereum for Architects and Developers*, pages 37–54. Springer.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Disponível em: http://bitcoin.org/bitcoin.pdf. Acesso em: 20 dez. 2018.

Staltz, A. (2016). The web began dying in 2014, here's how. Disponível em: https://staltz.com/the-web-began-dying-in-2014-heres-how.html. Acesso em: 20 dez. 2018.

Thakur, M. (2017). Authentication, authorization and accounting with ethereum blockchain. Disponível em: https://helda.helsinki.fi/bitstream/handle/10138/228842/aaa-ethereum-blockchain.pdf?sequence=2isAllowed=y. Acesso em: 20 dez. 2018.

Wood, G. (2014). Ethereum a secure decentralised generalised transaction ledger. Disponível em: https://gavwood.com/paper.pdf. Acesso em: 20 dez. 2018.