OTTx: Transações de Uso Único para Serviços Blockchain

André Defrémont[1], Billy Anderson Pinheiro[2], Alan Veloso [1], Roberto Samarone Araujo[1], Antônio Jorge Abelem[1]

[1]Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará (UFPA)
Caixa Postal 479 – 66.075-110 – Belém – PA – Brasil

[2]Pesquisa e Desenvolvimento – Amachains
66.075-750 – Belém – PA – Brasil

andre.def93@gmail.com, billy@amachains, {aveloso,rsa,abelem}@ufpa.br

Abstract. *In business computer networks, secure information exchange among organizations with diverse objectives is common. Users, identified uniquely, access resources through reliable applications. Integrating applications from external organizations introduces untrusted elements and requires necessitating mechanisms to ensure privacy, integrity, and non-repudiation. This work proposes a One-Time Transactions (OTTx) protocol in Blockchains. It addresses the need for secure transaction authentication, reviews the state of the art, and evaluates OTTx in a permissioned Blockchain. Results show that OTTx ensures security, privacy, integrity, and non-repudiation for transactions involving external identities, with satisfactory performance and low network overhead. This contribution advances Blockchain knowledge by providing an effective solution for transactions through untrusted network participants.*

Resumo. *Em redes de negócios empresariais, a troca segura de informações entre organizações com objetivos diversos é comum. Usuários, identificados de forma única, acessam recursos por meio de aplicativos confiáveis. A integração de aplicativos de organizações externas introduz elementos não confiáveis e requer mecanismos para garantir privacidade, integridade e não repúdio. Este trabalho propõe um protocolo de Transações Únicas (OTTx) em Blockchains. Ele aborda a necessidade de autenticação segura de transações, revisa o estado da arte e avalia o OTTx em uma Blockchain permissionada. Os resultados mostram que o OTTx garante segurança, privacidade, integridade e não repúdio para transações envolvendo identidades externas, com desempenho satisfatório e baixa sobrecarga de rede. Esta contribuição avança o conhecimento em Blockchains, fornecendo uma solução eficaz para transações envolvendo participantes de rede não confiáveis.*

1. Introdução

A tecnologia *Blockchain* tem transformado várias indústrias ao possibilitar transações confiáveis e transparentes sem a necessidade de intermediários. O termo *Blockchain* refere-se a um livro-razão descentralizado e distribuído que registra transações em blocos interligados e criptografados [1]. Essa estrutura oferece maior segurança, responsabilidade e transparência, pois todos os participantes da rede têm acesso às mesmas informações e os registros não podem ser alterados retroativamente sem o consenso da rede.

Em redes *Blockchain*, os mantenedores podem controlar o acesso de forma personalizada, e surge a necessidade de permitir o envio de transações por meio de aplicativos de organizações que não possuem infraestrutura *Blockchain* e se comunicam com a rede através de alguma identidade. Por exemplo, em redes acessadas através de APIs (Interface de Programação de Aplicação) pagas, as organizações que consomem essa API não possuem infraestrutura *Blockchain* própria, mas desejam interagir com a rede usando outras identidades. Para permitir que essas organizações externas forneçam serviços a outros usuários sem manipular suas identidades e mantendo a natureza permissionada da rede, são necessários mecanismos adicionais que garantam a integridade, não repúdio e privacidade dos dados nas transações realizadas por meio de serviços de terceiros que usam suas próprias identidades.

Nesse contexto, a implementação de métodos de autenticação com tokens de uso único, como *One-Time Password* (OTP) [4], em contratos inteligentes [6] [7] [8] e a integração com o envio de transações [5], têm ajudado a fornecer uma solução para o problema de autenticação em *Blockchain*. No entanto, essas abordagens não tratam o envio de transações por meio de terceiros de forma segura.

Este trabalho propõe um protocolo para transações de uso único chamado *OTTx (One-Time Transactions)* para redes *Blockchain*. Uma abordagem específica é adotada, consistindo na criação de um contrato inteligente dedicado às funções de autenticação e na habilitação da comunicação entre contratos inteligentes. O estudo também compara a segurança e a eficiência do protocolo OTTx com outros trabalhos e experimentos relacionados, fornecendo dados relevantes para pesquisas futuras nessa área. Ao destacar o potencial da tecnologia *Blockchain* para superar os desafios associados à autenticação de terceiros em transações, este trabalho busca contribuir para avanços significativos nesse campo.

A principal motivação deste trabalho é a necessidade de habilitar novos cenários em redes *Blockchains* para aplicativos externos, mantendo as características da rede. Além disso, aprimorar os métodos de autenticação para esse tipo de rede é um trabalho contínuo que evolui com a sua adoção. O envio de transações a partir de outras identidades incentiva fornecedores de serviços ao redor da rede e fomenta a adoção desse tipo de tecnologia, através de aplicativos mais competitivos.

O trabalho é composto por seis seções. Além da seção introdutória, foi desenvolvido a seção 2, para apresentação dos trabalhos relacionados. Na seção 3, a proposta é descrita em detalhes. Na sequência, há a seção 4 com a avaliação da proposta. Por fim, na seção 5 é feita uma análise comparativa e a seção 6 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Embora os artigos específicos sobre transações autenticadas em *Blockchain* sejam limitados, diversas pesquisas sobre autenticação em *Blockchains* formam uma base para transações autenticadas ou de uso único. Em um estudo [5], foi proposta a introdução de um esquema de Senha de Uso Único baseada em tempo (TOTP) para autenticação de dois fatores (2FA) em *Blockchains Hyperledger Fabric*. No entanto, o foco está principalmente na vulnerabilidade dos tokens de acesso do *Hyperledger* à interceptação. Este trabalho mantém a centralização por meio de um servidor proposto e aborda implementações da

versão descontinuada do *Hyperledger Fabric*. Apesar de implementar transações com tokens, não abrange transações enviadas por outras identidades.

Em [6], os autores propõem um sistema de autenticação/autorização/contabilidade interorganizacional que preserva a privacidade usando *Blockchain*. Este sistema, projetado para uma *Blockchain* pública, armazena senhas codificadas na *Blockchain*. No entanto, o esquema de autorização é limitado a regras de acesso do usuário para funcionalidades de rede, excluindo o acesso de identidades de terceiros aos dados do usuário. O artigo de [7] descreve um método de autenticação baseado em *Blockchain* com uma senha de uso único, mas detalha apenas o método de autenticação com OTP e *Blockchain* permissionada, omitindo a submissão segura de transações de outras identidades.

Por fim, [8] apresenta um protocolo baseado em *microservices* e *Blockchain* para autenticação aprimorada com senha de uso único (MBB-OTP). Embora limitado à autorização de usuários de rede, o uso de segredos compartilhados e vários *microservices* aumenta os custos computacionais e as vulnerabilidades. Além disso, prolonga o processo de autenticação.

O protocolo OTTx se destaca em relação aos trabalhos relacionados, apresentando uma abordagem abrangente e avançada para transações de uso único (*One-Time Transactions - OTTx*) em redes *Blockchain*, não apenas abordando a autenticação, mas também oferecendo uma solução completa para o envio seguro de transações através de identidades de terceiros. Ao garantir privacidade, integridade e não repúdio em todas as fases da transação, o OTTx supera as limitações identificadas nos trabalhos relacionados, que muitas vezes se concentram em aspectos específicos da autenticação. Sua abordagem integrada proporciona uma solução abrangente e coesa, promovendo maior confiança e eficiência nas transações em ambientes *Blockchain*, enquanto os trabalhos revisados frequentemente negligenciam a consideração holística da segurança ao lidar com transações autenticadas em redes descentralizadas.

3. OTTx: One-Time Transactions

O objetivo principal desta proposta é melhorar a segurança nas submissões de transações envolvendo identidades não confiáveis, fortalecendo a privacidade de dados e consolidando as funções de autenticação em um único contrato. A proposta aborda desafios relacionados à privacidade de dados de transações por meio da implementação de um contrato unificado para transações autenticadas. Detalha-se dois aspectos-chave: um contrato singular para a submissão de transações autenticadas e o registro de dados privados. Esses elementos exigem implementações no algoritmo de geração de tokens do usuário, na função de registro de usuário do contrato inteligente, na função de autenticação de transações do contrato e na configuração da rede.

3.1. Visão Geral

O esquema proposto concentra-se em modificações na rede *Blockchain* (Figura 1). Nesta rede *Blockchain* permissionada, o usuário (Cliente/Usuário), previamente cadastrado, gera tokens assinados para transações que serão enviadas por Terceiros Não Confiáveis. Com o token em mãos, o Terceiro Não Confiável envia uma transação com a sua identidade, para a *Blockchain*. A rede se encarrega de validar a transação original e validar a assinatura de ambos os participantes.

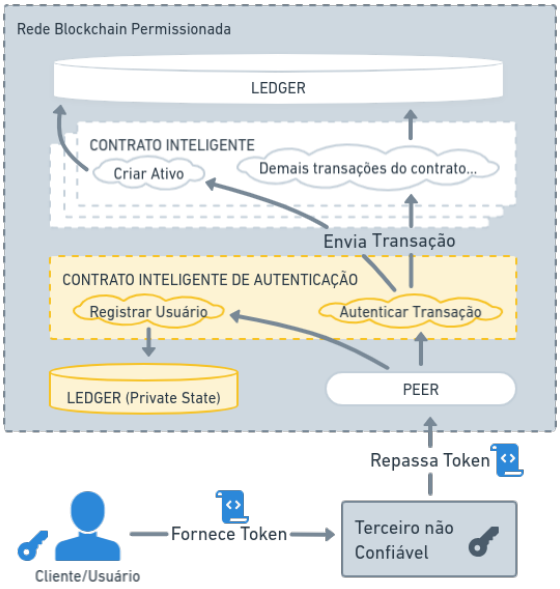

Figura 1. Visão Geral do Protocolo OTTx

A Figura 2 (A) ilustra a sequência de registro do usuário no Contrato Inteligente OTTx. O usuário, possuindo chaves pública e privada, (1) gera o token inicial usando o algoritmo de geração de token e, em seguida, (2) solicita o registro do usuário ao Terceiro Não Confiável, que (3) inicia o registro do usuário no Contrato Inteligente OTTx.

A geração inicial do token envolve informações do usuário, um par de chaves criptográficas (*pk*, *sk*), identificador (*id*) e senha (*pw*). Um algoritmo combina esses dados para gerar um hash criptográfico como token. Além disso, o algoritmo produz o hash do token e uma mensagem assinada (*sigm*) contendo informações relevantes.

Para (2) solicitar o registro do usuário, o usuário informa explicitamente ao terceiro a sua chave pública (*pk*), identificador (*id*), hash do token gerado (*hash(t)*), mensagem assinada (*sigm*) e identificador do terceiro (*idt*). Para evitar tentativas de adulteração, a verificação da mensagem assinada garante que os dados fornecidos correspondam.

No passo (3), o Terceiro Não Confiável chama a Transação de Registro de Usuário do Contrato Inteligente OTTx. A fase de validação e registro começa com (4) verificação da mensagem assinada em relação aos dados passados para o contrato. Se válido, (5) o contrato cria um par de chaves para transações, usado para criptografia assimétrica em transações futuras. Em (6), a chave de gravação, chave pública do usuário e hash do token são armazenados em uma coleção de dados privados acessível apenas a pares e organizações autorizadas. Finalmente, (7) o contrato retorna dados públicos de transação, e (8) o terceiro deve retornar a chave pública para transações ao usuário.

A Figura 2 (B) exibe o diagrama de sequência para autenticação de transações. A sequência começa com (9) o Usuário/Cliente criptografando dados de transação (*txData*) usando a chave de transação (*tpk*) do processo de registro. O Usuário então (10) gera tokens para transações, semelhante ao registro, mas incluindo dados de transação criptografados na mensagem assinada. O Usuário (11) solicita a transação ao Terceiro Não Confiável, passando dados necessários. O terceiro (12) chama a Transação para Au-

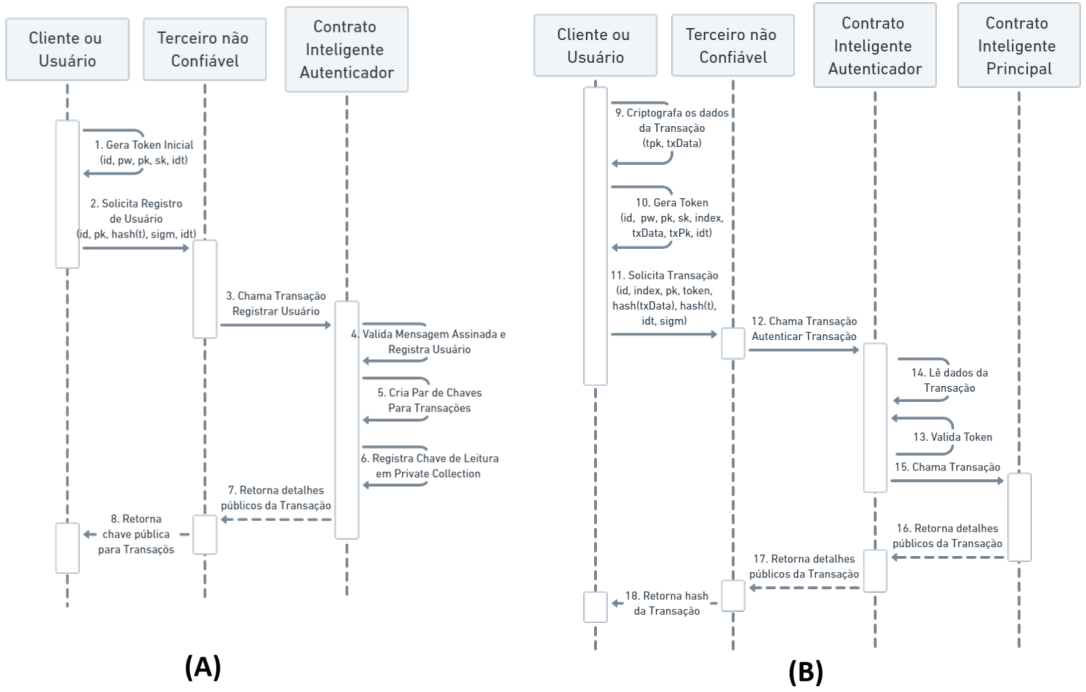

Figura 2. Diagrama de Sequência de Registro de Usuário (A) e Autenticação de Transação (B) do OTTx

tenticar Transações do Contrato Inteligente OTTx para validação do token e gravação de dados em outros contratos inteligentes.

A fase de validação da transação começa com (13) validação de tokens e da mensagem assinada, verificando a existência do identificador do usuário e o índice correto. Se não houver erros, (14) o contrato busca na coleção privada a chave de leitura para descriptografar os dados da transação, que são (15) enviados diretamente para o contrato principal.

O protocolo OTTx foi implementado e testado usando o *Hyperledger Fabric* [10] v2.0 em uma configuração permissionada privada. Desenvolvido em *Node.js*, o protocolo envolve configurações de rede, algoritmos de geração de tokens de cliente/usuário e um contrato inteligente único para autenticação. Organizações que precisam acessar dados privados devem ser designadas durante a criação da rede.

3.2. Geração de Tokens

Para o Usuário, são necessários algoritmos para geração de tokens. Essas funções são baseadas nos trabalhos de [12] e [7], utilizando as funções de hash SHA256 e o algoritmo de assinatura RSA-SHA256. A implementação e teste do lado do cliente foram realizados em *JavaScript*, usando o ambiente de execução *Node.js* e os pacotes *crypto*[1] e *keypair*[2].

```
1  if index == 0 then
2        token = hash({ id, index, pw, sk })
3        hashToken = hash(token)
```

[1]https://www.npmjs.com/package/crypto-js
[2]https://www.npmjs.com/package/keypair

```
4        message = { id, pk, hashToken }
5        signedMessage = sign(sk, message);
6        return { token, hashToken, signedMessage }
7   else
8       nextToken = hash({ id, index, pw, sk })
9       hashToken = hash(nextToken);
10      hashTxData = publicEncrypt(pkt, txData);
11      message = { id, pk, hashToken, hashTxData, idt };
12      signedMessage = sign(sk, message);
13      token = hash({ id, index: index - 1, pw, sk });
14      return { token, hashToken, signedMessage };
15  end if
```

Algoritmo 1. Algoritmo de Geração de Tokens OTTx

Para gerar tokens, o usuário deve sempre passar o valor atual do índice como parâmetro para que os tokens corretos sejam gerados. Se o Usuário ainda não tiver um registro, ele deve passar o valor zero (0) para o índice para criar o token inicial. Portanto, a função de geração de token mostrada no Algoritmo 1 começa verificando o valor do índice para decidir se deve gerar um token inicial (índice = 0) ou não. Para tokens iniciais, a função gera um hash (SHA256 [13]) combinando o identificador do usuário (*id*), índice (*index*), senha (*pw*) e chave privada (*sk*). Após gerar esse primeiro hash (token), a função gera um novo hash (*hashToken*), desta vez passando o hash anterior como parâmetro. Assim, o segundo hash gerado (*hashToken*) será armazenado na *Blockchain* para ser usado como verificador na próxima solicitação e deve ser atualizado pelo próximo hash na próxima chamada. Além desses dois hashes, a função gera uma mensagem assinada (*signedMessage*) combinando o identificador do usuário (*id*), chave pública (*pk*) e o hash do token (*hashToken*) usando o algoritmo de assinatura RSA-SHA256 [14]. A mensagem assinada será usada para manter a integridade das informações publicamente passadas para o contrato inteligente.

Para valores de índice maiores que zero (0), a função de geração de token mostrada no Algoritmo 1 também gera, como descrito anteriormente, um novo token (*nextToken*), o hash deste novo token (*hashToken*) e a mensagem assinada (signedMessage). No entanto, desta vez, o hash é gerado usando criptografia assimétrica [11] (publicEncrypt) dos dados da transação (*hashTxData*), usando a chave pública para transações (*pkt*) gerada na transação de registro. Os dados da transação criptografados (*hashTxData*) são incluídos na mensagem assinada juntamente com o identificador do terceiro não confiável (*idt*). Dessa forma, o terceiro não confiável não pode ler os dados da transação, e o token não pode ser enviado por outra identidade, prevenindo ataques de replay.

3.3. Contrato Inteligente

Outra contribuição deste trabalho é a implementação de um contrato inteligente personalizado para o processo de autenticação de transações. Para alcançar isso, o contrato deve incluir funções para registro de usuário e validação de token, além de facilitar a comunicação com outros contratos.

O Algoritmo 2 apresenta o algoritmo para a função de registro de usuário no contrato inteligente de autenticação. A função recebe o identificador do usuário (*id*), sua chave pública (*pk*), o hash do token inicial (*hashToken*), a mensagem assinada e o

identificador do terceiro não confiável (*idt*). Com esses dados, o contrato verifica se o identificador do usuário já existe. Se não existir, verifica a validade da mensagem assinada com a chave pública do usuário (*V(pk, signedMessage, message)*). Se validado, o contrato gera novas chaves para a criptografia assimétrica dos dados da transação (generateKeyPairSync), e a chave privada é salva com segurança usando a funcionalidade do *Hyperledger Fabric* para criar coleções de dados privados, juntamente com o identificador do usuário (*id*), chave pública (*pk*), chave pública da transação (*txPublicKey*) e chave privada da transação (*txPrivateKey*). Posteriormente, o contrato registra o identificador do usuário (*id*), chave pública (*pk*), hash do token (*hashToken*), índice (*index*) definido como um (1) e a chave pública da transação (*txPublicKey*) na *Blockchain*. Os dados salvos privadamente não são expostos a nenhuma parte fora das organizações autorizadas no momento da criação da rede. Com esses dados registrados, o usuário pode começar a enviar transações para autenticação no contrato.

```
1  Input: id, pk, hashToken, signedMessage, idt
2  Output: true or false
3  if id exists then
4      return false
5  else
6      message = { id, pk, hashToken, idt}
7      if V(pk, signedMessage, message) === true then
8          txPublicKey, txPrivateKey = generateKeyPairSync()
9          save private {id, pk, txPublicKey, txPrivateKey}
10         save {id, pk, hashToken, index = 1, txPublicKey}
11         return true
12     else
13         return false
14     end if
15 end if
```

Algoritmo 2. Algoritmo de Registro de Usuário no Contrato Inteligente OTTx

O Algoritmo 3 ilustra a implementação da função para autenticação de token no contrato inteligente. Esta função é responsável por verificar a validade dos tokens, atualizar os dados do usuário e invocar a transação original solicitada pelo usuário. A função verifica se o identificador do usuário existe, se o hash do token enviado (*hash(token)*) é igual ao armazenado na *Blockchain* (*HASHTOKEN*), verifica a mensagem assinada e garante o índice de token correto. Se todas as validações passarem, o contrato obtém a chave privada da transação do usuário (*TXPRIVATEKEY*) e descriptografa os dados da transação para enviá-los para o contrato original solicitado usando a função para invocar outros contratos na rede *Hyperledger Fabric* (*invokeChaincode*). Finalmente, o contrato atualiza os dados do usuário registrados na *Blockchain* para incluir o hash do próximo token (*hashToken*) e incrementa o índice em uma unidade.

```
1  Input: index, id, pk, token, hashToken, hashTxData, signedMessage,
       idt
2  Output: true or false
3  if id exists then
4      get {HASHTOKEN, INDEX} where ID = id
5      message = {id, pk, hashTxData, hashToken, idt}
6      if HASHTOKEN = hash(token) and verifyMessage(pk, signedMessage,
           message) === true then
```

```
 7           if INDEX === index
 8               get private {TXPRIVATEKEY} where ID = id
 9               txData = privateDecrypt(txPrivateKey, hashTxData)
10               invokeChaincode(txData.chaincode, txData.args, txData.
                     channel)
11               update {id, pk, nextToken, index, txPublicKey}
12               return true
13           else
14               return false
15           end if
16       else
17           return false
18       end if
19   else
20       return false
21   end if
```

Algoritmo 3. Algoritmo de Transações no Contrato Inteligente OTTx

4. Avaliação da Solução Proposta

A avaliação abrange aspectos de segurança e eficiência da solução. Além disso, melhorias relacionadas a trabalhos anteriores sobre autenticação são listadas e discutidas. As avaliações demonstram que a proposta apresentada por este trabalho tornou a solução mais competitiva do que as anteriores, além de mais segura e versátil para vários cenários.

Esta seção visa analisar a eficácia da solução proposta em relação a diferentes tipos de ataques e desafios de segurança. A avaliação está dividida em cinco subseções, cada uma abordando um aspecto específico da solução. A primeira subseção, apresenta o modelo adversário que engloba as limitações da análise de segurança. A segunda subseção, ”Resistência a Ataques de Repetição”, explora como a solução lida com ataques que tentam reutilizar tokens em identidades ou solicitações subsequentes. A terceira subseção, ”Resistência a Ataques de Força Bruta”, discute como a criptografia assimétrica e o processo de geração de hash de tokens protegem contra ataques que tentam adivinhar a chave privada. A quarta subseção, ”Falsificação de Token”, examina a capacidade da solução de evitar a falsificação de tokens, considerando as propriedades de segurança de funções de hash e criptografia. A quinta subseção, ”Confidencialidade de Dados em Transações”, aborda a proteção dos dados do usuário por meio de criptografia assimétrica e acesso restrito às chaves privadas. Finalmente, a sexta subseção, ”Recadastramento de Identidade”, explora a implementação de um contrato centralizado para registro e autenticação, garantindo a singularidade do identificador e prevenindo ataques de recadastramento.

4.1. Modelo Adversário

O protocolo proposto assume que um adversário tem acesso à rede *Blockchain* e pode obter informações armazenadas, incluindo o hash de dados de transações privadas em tokens usados e não usados. No entanto, o adversário não pode comprometer ou obter a chave privada armazenada de forma privada no cliente de maneira segura. Para garantir a integridade das mensagens e evitar ataques de intermediários, as comunicações são protegidas por SSL/TLS [15]. O adversário também não pode modificar dados de transações armazenados na *Blockchain*, mas seu objetivo é ler ou modificar dados de transações solicitados antes que sejam validados pelos nós da *Blockchain*.

A proposta visa garantir a privacidade dos dados da transação, resistência a ataques de força bruta, ausência de vazamento de dados de transações na comunicação entre contratos, dados de transações à prova de violação e resistência a ataques de dicionário em dados privados. Esses critérios serão usados para avaliar a solução proposta e medir sua eficácia contra potenciais ataques adversários.

Este trabalho utilizou o *framework Hyperledger Fabric v2.0* para implementar a rede e conduzir testes apenas na configuração de uma rede *Blockchains* permissionada privada. Também não explorou o uso de chaves privadas diferentes para dados privados em cada transação, e não descreve soluções para o cenário em que o usuário não recebe ou perde as chaves de acesso.

4.2. Resistência a Ataques de Repetição

Ataques de repetição podem ocorrer quando o atacante tem acesso a tokens usados ou não usados, tentando reutilizá-los em outras identidades ou solicitações subsequentes. No entanto, a solução proposta resiste a esses ataques devido à verificação da mensagem assinada no contrato (linhas 5-6 do Algoritmo 3), que contém o identificador do terceiro não confiável (*idt*). Como o atacante não possui a chave privada de assinatura, ele não pode adulterar a mensagem. Tokens usados não são validados pela *Blockchain*, pois o contrato inteligente espera o hash do próximo token (*hashToken*) e a mensagem assinada correspondente (*sigm*) para cada verificação (linhas 5-6 do Algoritmo 3).

4.3. Resistência a Ataques de Força Bruta

A criptografia assimétrica garante que derivar a chave privada da chave pública e da mensagem assinada seja computacionalmente inviável. A combinação do uso da senha (*pw*) e da chave privada (*sk*) para gerar hashes de token armazenados na *Blockchain* (*hashToken* = *hash(hash(id,índice,pw,sk))*) torna computacionalmente inviável a tentativa de adivinhar a chave privada, garantindo resistência a ataques de força bruta (linhas 4-5 e 10-11 do Algoritmo 1).

4.4. Falsificação de Token

Para falsificar um token, o adversário precisa passar pelas verificações do contrato, satisfazendo o hash atual do token (hash(token)) e a verificação da mensagem assinada (*verifyMessage(pk, signedMessage, message)*) (linhas 5-6 do Algoritmo 3). No entanto, devido às características de segurança da criptografia assimétrica e da função de hash, o adversário não pode deduzir o token necessário para gerar o hash, assim como a chave privada (*sk*) para criar a mensagem assinada. Supondo que seja impraticável alterar valores mantidos pelos mantenedores da rede *Blockchain*, se o adversário tentar falsificar o próximo token adquirindo maliciosamente um token válido e alterando o valor do hash do próximo token (*hashToken*), o contrato inteligente não validará a solicitação. Isso ocorre porque a mensagem assinada também contém o hash do próximo token, tornando impossível gerar a mensagem assinada sem a chave privada do usuário, tornando o sistema resistente a ataques de falsificação.

4.5. Confidencialidade de Dados em Transações

Os dados de transações do usuário são criptografados de forma assimétrica (linha 12 do Algoritmo 1) com chaves geradas exclusivamente para esse fim (linha 8 do Algoritmo 2).

Para obter os dados de transações do usuário, o adversário pode tentar adquirir a chave privada usada para descriptografar esses dados. No entanto, a chave privada é armazenada em uma coleção de dados privados da rede *Blockchain* (txPrivateKey) e garante que apenas pares de organizações pré-configuradas possam acessar esses dados (linha 9 do Algoritmo 2). A descriptografia desses dados ocorre apenas dentro do contrato inteligente, e os dados são transmitidos de forma transitória para evitar a inclusão em blocos da *Blockchain*, tornando desafiador vazar essas informações.

4.6. Recadastramento de Identidade

Um adversário pode se passar pela identidade do usuário em contratos nos quais o usuário ainda não se registrou e assumir a posse de transações desse contrato usando o identificador do usuário. A proposta deste trabalho implementa um contrato centralizado apenas para métodos de registro e autenticação, centralizando os registros de identificadores e chaves públicas, impedindo registros de usuários com o mesmo identificador se já estiverem registrados no contrato (linha 3 do Algoritmo 2). Para tornar o sistema imune a ataques de recadastramento, a proposta implementa a comunicação de contrato para contrato, pois o contrato deve chamar o contrato solicitado na solicitação original de transação (linha 10 do Algoritmo 3).

5. Análise Comparativa

A Tabela 1 oferece uma análise comparativa do protocolo OTTx proposto e outros trabalhos relevantes, destacando características-chave e distinções.

Na autenticação de token, tanto OTTx, [5], [7] e [8] abordam essa funcionalidade, enquanto [6] não o faz, ressaltando a alinhamento do OTTx com os avanços na autenticação de token.

A submissão autenticada de transações, uma característica crucial, é suportada pelo OTTx e por [5], distinguindo-os de outros trabalhos que não especificam esse aspecto, o que aprimora a segurança e confiabilidade do sistema.

Assegurar resistência à falsificação de token é fundamental em sistemas de autenticação. OTTx, [5], [7] e [8] abordam essa preocupação, enquanto [6] não o faz.

Uma vantagem notável do OTTx é a eliminação da necessidade de uma terceira parte totalmente confiável, compartilhada por [6], [7] e [8], mas não por [5]. Essa ausência reduz a dependência de terceiros, aprimorando a segurança da autenticação.

A capacidade de garantir o não-repúdio de dados inseridos por outras identidades é uma característica única do OTTx, proporcionando benefícios adicionais de rastreabilidade e responsabilidade.

A redução da dependência na Autoridade Certificadora (CA) é uma vantagem compartilhada pelo OTTx e por [8], aprimorando a robustez e confiabilidade do sistema.

A resistência a ataques de reutilização, abordada pelo OTTx, [5], [7] e [8], protege contra ataques de repetição e reutilização de informações de autenticação. [6] não aborda essa preocupação.

O uso de um único contrato para autenticação pelo OTTx, não encontrado em outros trabalhos, simplifica o processo, oferecendo eficiência e centralização.

Por fim, o OTTx aborda a privacidade de dados de transações, um aspecto não abordado por outros trabalhos.

Em resumo, a análise comparativa destaca as vantagens distintas do OTTx. Sua combinação de recursos o coloca em uma posição favorável como um protocolo de autenticação seguro e eficiente.

Os testes compararão o tempo de autenticação de transações na *Blockchain* com e sem os métodos propostos. A avaliação determinará se a proposta impacta significativamente a eficiência do sistema. Dados e gráficos que representam o tempo de execução de diferentes abordagens serão apresentados, juntamente com uma análise comparativa com trabalhos relacionados. Os resultados destacarão as contribuições da solução proposta para o desempenho e eficiência do sistema.

A Figura 3 (A) compara as propostas de [8] e [7] com a abordagem do OTTx. Notavelmente, os resultados de tempo de execução de [7] são semelhantes, mas ele não lida com dados de transações como parte do processo de autenticação. O tempo de execução total significativamente maior de [8] é atribuído a múltiplos serviços intermediários.

A avaliação utilizou um processador *Intel® Core™ i5-7300HQ CPU @ 2.50GHz × 4, 16 GB* de *RAM* executando *Ubuntu Server* 22.04. Os testes simularam processos de transação com *scripts Nodejs*, considerando um intervalo de bloco de dois (2) segundos e transações de 1000 Bytes.

A Figura 3 (B) mostra a comparação entre o OTTx proposto e uma rede sem transações autenticadas. Observa-se uma pequena perda de desempenho, em torno de 100 milissegundos, mas não é significativa.

Em uma avaliação mais abrangente, é importante considerar limitações, como testes em um ambiente controlado. Trabalhos futuros podem abordar desafios de implantação no mundo real e variações nos tempos de resposta de hardware e rede.

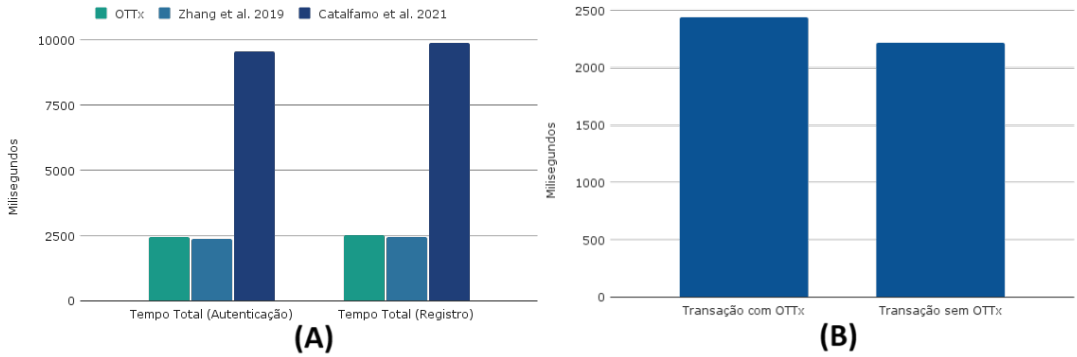

Figura 3. Comparação de Tempo de Execução das Propostas (A) e de Transações com e sem o protocolo OTTx (B)

6. Conclusão e Trabalhos Futuros

Esta seção resume as descobertas do protocolo OTTx proposto, um método seguro e eficiente para submissão de transações em redes *Blockchain*. O OTTx garante segurança, controle de acesso à rede e autenticação eficiente de transações, eliminando a dependência de terceiros não confiáveis.

	OTTx (Proposta)	Park et al. 2018 [5]	Lu et al. 2018 [6]	Zhang et al. 2019 [7]	Catalfamo et al. 2021 [8]
Autenticação com Tokens	X	X		X	X
Envio de Transações com Autenticação	X	X			
Resistência a falsificação de Tokens	X	X		X	X
Nenhum terceiro totalmente confiável é necessário	X		X	X	
Não repúdio de dados inseridos através de outras identidades	X				
Menor dependência da CA	X				X
Resistência a ataques de reutilização	X			X	X
Contrato único para Autenticação	X				
Privacidade dos dados da Transação	X				
Blockchain abordada	HL Fabric v2.x	HL Fabric v1.x	Ethereum	HL Fabric v2.x	HL Fabric v2.x and Ethereum

Tabela 1. Comparação entre a proposta e os trabalhos relacionados (Autor)

O protocolo OTTx apresenta uma abordagem segura e eficiente para o envio de transações por meio de identidades de terceiros em redes *Blockchain*. Ele aborda desafios relacionados à segurança, controle de acesso à rede e autenticação de transações, fornecendo um método simples e seguro para enviar transações autenticadas. Por meio de componentes implementados, incluindo métodos de autenticação, protocolos de contratos inteligentes e geração segura de tokens, o protocolo permite que identidades de terceiros enviem transações para outras identidades.

A implementação do protocolo demonstrou viabilidade e eficácia, comprovando sua eficiência em comparação com outros métodos de autenticação *Blockchain*. Avaliações preliminares indicam eficiência comparável ou superior, com um tempo de transação adicional mínimo (aproximadamente 100 ms). Isso destaca o equilíbrio entre desempenho e segurança, enfatizando a viabilidade técnica do sistema e seu valor tangível na segurança de ambientes sensíveis a transações.

A principal contribuição está em permitir que aplicativos externos interajam de forma confiável e eficiente com a rede *Blockchain* por meio de diversas identidades, garantindo integridade, privacidade e não repúdio.

Embora resultados significativos tenham sido alcançados, é crucial reconhecer limitações e desafios para a implementação no mundo real. Pesquisas futuras devem explorar possibilidades e abordar os seguintes tópicos:

- Aprimorar a segurança e flexibilidade do sistema envolve desenvolver mecanismos para que os usuários gerenciem chaves criptográficas, oferecendo maior controle sobre a confidencialidade da transação.
- Desenvolver métodos eficientes para registrar e sincronizar informações de autenticação de transações, incluindo registros de uso de tokens, índices de autenticação, identificadores de identidades de terceiros e carimbos de data/hora dos tokens. Isso garante transparência e confiabilidade no OTTx.

Essas direções de pesquisa oferecem oportunidades para aprimorar e expandir o protocolo OTTx, abordando desafios específicos e explorando novas possibilidades. O progresso nessas áreas fortalecerá a segurança e confiabilidade das submissões autenticadas de transações em redes *Blockchain*, tanto permissionadas quanto não permissionadas.

Agradecimentos

Este trabalho foi realizado com o apoio da Amazônia Blockchain Solutions (Amachains), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), da Rede Nacional de Ensino e Pesquisa (RNP) e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), projeto 2023/00811-0, projeto 2020/04031-1 e projeto 2018/23097-3.

Referências

[1] Satoshi Nakamoto, ”Bitcoin: A Peer-to-Peer Electronic Cash System,”March 2009. Available at: https://bitcoin.org/bitcoin.pdf.

[2] A.-D Liu, X.-H Du, N. Wang, S.-Z Li, ”Research Progress of Blockchain Technology and Its Application in Information Security,”Journal of Software, vol. 29, pp. 2092-2115, July 2018. DOI: 10.13328/j.cnki.jos.005589.

[3] Shrivas, M. K., Yeboah, T. (2019). ”The disruptive blockchain: types, platforms, and applications.”Texila International Journal of Academic Research, 3, 17-39.

[4] Leslie Lamport, ”Password Authentication with Insecure Communication,”Communications of the ACM, vol. 24, no. 11, pp. 770–772, November 1981. DOI: 10.1145/358790.358797.

[5] Woo-Suk Park, Dong-Yeop Hwang, Ki-Hyung Kim, ”A TOTP-Based Two Factor Authentication Scheme for Hyperledger Fabric Blockchain,”in Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), 2018. DOI: 10.1109/ICUFN.2018.8436784.

[6] Peggy Joy Lu, Lo-Yao Yeh, Jiun-Long Huang, ”An Privacy-Preserving Cross-Organizational Authentication/Authorization/Accounting System Using Blockchain Technology,”in Proceedings of the 2018 IEEE International Conference on Communications (ICC), 2018. DOI: 10.1109/ICC.2018.8422733.

[7] Mingli Zhang, Liming Wang, Jing Yang, ”A Blockchain-Based Authentication Method with One-Time Password,”in Proceedings of the 2019 IEEE 38th International

Performance Computing and Communications Conference (IPCCC), 2019. DOI: 10.1109/IPCCC47392.2019.8958754.

[8] Alessio Catalfamo, Armando Ruggeri, Antonio Celesti, Maria Fazio, Massimo Villari, ”A Microservices and Blockchain Based One Time Password (MBB-OTP) Protocol for Security-Enhanced Authentication,”in Proceedings of the 2021 IEEE Symposium on Computers and Communications (ISCC), 2021. DOI: 10.1109/ISCC53001.2021.9631479.

[9] Christian Cachin et al., ”Architecture of the Hyperledger Blockchain Fabric,”in Proceedings of the Workshop on distributed cryptocurrencies and consensus ledgers, Chicago, IL, 2016.

[10] Elli Androulaki et al., ”Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,”Proceedings of the Thirteenth EuroSys Conference, EuroSys ’18, Porto, Portugal, 2018. DOI: 10.1145/3190508.3190538.

[11] Gustavus J. Simmons, ”Symmetric and asymmetric encryption,”ACM Computing Surveys (CSUR), vol. 11, no. 4, pp. 305-330, 1979.

[12] Chang-Seop Park, ”One-time password based on hash chain without shared secret and re-registration,”Computers and Security, 2018. DOI: 10.1016/j.cose.2018.02.010.

[13] Quynh H. Dang, ”Secure hash standard,”Quynh H. Dang, 2015.

[14] Devrim Unal, Abdulla Al-Ali, Ferhat Ozgur Catak, Mohammad Hammoudeh, ”A secure and efficient Internet of Things cloud encryption scheme with forensics investigation compatibility based on identity-based encryption,”Future Generation Computer Systems, 2021. DOI: https://doi.org/10.1016/j.future.2021.06.050.

[15] Rolf Oppliger, ”SSL and TLS: Theory and Practice,”Artech House, 2016.