Proposta de Solução para os Desafios de Privacidade, Escalabilidade e Programabilidade na Adoção de uma Rede de Responsabilidade Regulada (RLN) em blockchain

Felipe Martins de Lima (felipe@bywise.org)
Henrique Gomes de Moura (hgmoura@unb.br)
ChainXS

Resumo

A adoção da tecnologia *blockchain*, em uma Rede de Responsabilidade Regulada, possui desafios de privacidade, escalabilidade e programabilidade. Este documento propõe uma solução para esses desafios por meio de uma tecnologia brasileira, a ChainXS Blockchain, fork da Bywise. Através da integração de tecnologias como endereços furtivos, distribuição uniforme de dados, livro razão quimérico e validação paralela de transações, a solução garante a privacidade dos usuários, além de poder lidar com volumes crescentes de transações, conseguindo manter altos níveis de desempenho sem abdicar da auditabilidade por entidades reguladoras. Estas abordagens equilibram a necessidade de conformidade regulatória com a proteção da privacidade, oferecendo uma infraestrutura robusta e eficiente para a adoção massiva de tecnologias *blockchain*.

Palavras-chave: Privacidade, Blockchain, CBDC, Real Digital, RLN.

Abstract

The adoption of blockchain technology, in a Regulated Accountability Network, faces challenges of privacy, scalability, and programmability. This document proposes a solution to these challenges using a Brazilian technology, ChainXS Blockchain, a fork of Bywise. By integrating technologies such as stealth addresses, uniform data distribution, chimeric ledger and parallel validation of transactions, the solution guarantees user privacy, can handle increasing volumes of transactions and can maintain high levels of performance without giving up auditability for regulatory bodies. These approaches balance the need for regulatory compliance with privacy protection, offering a robust and efficient infrastructure for the mass adoption of blockchain technologies.

Keywords: Privacy, Blockchain, CBDC, Real Digital, RLN.

1 Introdução

A adoção da tecnologia *blockchain* permissionada tem revelado benefícios para instituições financeiras e governos, como segurança aprimorada, transparência e programabilidade. No entanto, esses avanços também trazem consigo desafios significativos, especialmente nas áreas de privacidade e escalabilidade. A garantia de privacidade dos usuários e a necessidade de escalar operações para atender a um volume crescente de transações, em uma rede transparente, são obstáculos que limitam a utilização dessa tecnologia em larga escala.

Os desafios enfrentados pelo setor financeiro e governamental não são exclusivos. A tecnologia *blockchain* vem crescendo, também, em empresas privadas que tiveram avanços nos setores imobiliário, setores logístico, setores de saúde e de educação. Segundo a Deloitte, 40% das empresas já implementaram *blockchains*, e os principais desafios enfrentados foram: a dificuldade de implantação do sistema em questão, no que se diz respeito à infraestrutura, e a escassez de profissionais capacitados para o projeto[4].

A tecnologia *blockchain* teve seu início com o *whitepaper* do Bitcoin, em 31 de outubro de 2008[7]. A Bitcoin gerou um sistema auditável, capaz de validar pagamentos online, entre pessoas (usuários), sem a necessidade de uma instituição financeira. Apesar de possuir alguns problemas relacionados como a falta de privacidade, escalabilidade e programabilidade, a Bitcoin foi a primeira criptomoeda *blockchain* que conseguiu criar um ativo legitimamente descentralizado.

A criptomoeda Ethereum adicionou programabilidade na *blockchain* com a EVM (*Ethereum Virtual Machine*) capaz de processar contratos inteligentes, gerar tokens, entre outras funcionalidades[1]. Outras criptomoedas tais como a Monero e a Zcash (ZEC) geraram uma rede que realiza transações de maneira anônima[10]. Em 2015 surge o Projeto Hyperledger, visando criar um *blockchain* voltada para múltiplos segmentos de mercado baseado em EVM[5].

Em 2017, iniciou-se o desenvolvimento da ChainXS[**?**], uma *blockchain* brasileira mantida de código aberto pela Devel Blockchain. Sua proposta inicial era criar uma tecnologia de pagamentos baseada em *blockchain* com sete pilares, dentre eles a escalabilidade, a segurança, a privacidade e a usabilidade. Ao longo dos anos a plataforma foi se especializando em atender o mercado corporativo em múltiplos segmentos, assim como a Hyperledger. Hoje, a ChainXS é uma *blockchain* brasileira construída do zero, focada em ser uma camada de segurança e gestão de dados para empresas. Por ser construída especificamente para esse propósito, hoje ela implementa uma série de funcionalidades que não estão presentes na maioria dos projetos de *blockchain*.

Este documento destaca soluções para enfrentar os desafios de privacidade e escalabilidade, anteriormente mencionados, particularmente no contexto de uma Rede de Responsabilidade Regulada (RLN). Vale citar que, a tecnologia ChainXS também se destaca nos quesitos de interoperabilidade e facilidade de integração com sistemas legados não baseados em *blockchain*. Com o objetivo de facilitar a programabilidade da *blockchain*, bem como mitigar a falta de recursos humanos, a ChainXS facilitou o processo de criação de contratos inteligentes utilizando JavaScript - linguagem de programação mais utilizada no mundo atualmente - e a integração com chamada REST API. Em outras palavras, o *stack* de linguagens utilizado reduz as barreiras de entrada para pessoas e instituições desenvolverem suas próprias soluções.

2 Privacidade

Alguns dos conceitos fundamentais relacionados à implementação de *blockchains*, com foco na privacidade, foram selecionados para este tópico. Ao final, todas as principais informações foram tabeladas, de modo a destacar diferenças para com a proposta de solução (ChainXS), apresentada neste artigo.

2.1 Livro Razão Distribuído

Em uma *blockchain*, qualquer nó da rede pode ver todas as transações já realizadas. Está característica dá a possibilidade de auditoria, permitindo a qualquer um validar todo o histórico de transações da rede. Por consequência a segurança é aumentada, porém, somada à perda da privacidade dos usuários. Caso se vincule uma carteira a uma identidade real, como uma instituição ou pessoa, seu saldo e suas transações não serão mais privadas.

O livro razão distribuído é historicamente dividido em dois grandes grupos, no que diz respeito aos modelos de *blockchain* aceitos: modelo UTXO e modelo baseado em contas (*Account-Based*). As vantagens e desvantagens de cada modelo podem ser conferidas em referências especializadas[13].

2.2 Modelo de Contas

No caso de *blockchains* permissionadas baseadas em EVM, a privacidade é ainda menor, uma vez que estas são baseadas em um modelo de contas. Isso representa um problema para as instituições financeiras, especialmente se servirem como validadores, uma vez que poderiam potencialmente visualizar as transações e informações das contas de uns e outros. Esta falta de confidencialidade pode prejudicar a confiança entre as instituições participantes e comprometer a segurança geral da rede.

O modelo baseado em contas possui três grandes desvantagens em relação à privacidade. A primeira se relaciona com a dependência entre o resultado de uma transação e o estado de entrada, fato que dificulta o processo de escalabilidade por conta da dificuldade na execução de transações em paralelo - geralmente, as transações que afetam a mesma conta precisarão ser executadas uma após a outra.

A segunda desvantagem está no incentivo à reutilização de endereços, de forma a facilitar a vinculação das transações a um único proprietário. A terceira e última desvantagem, aqui listada, está no volume de dados (requisitos) necessários para vincular uma carteira a um único usuário. Em redes permissionárias os endereços precisam ser previamente autenticados. E no caso de redes baseadas em gás, o usuário precisa enviar saldo para a carteira para depois conseguir utilizá-la.

2.3 Transação de Saída Não Gasta - UTXO

Em *blockchains* sem grande programabilidade como a Monero, cripto-moeda focada em privacidade, as transações são baseadas em UTXO. Estritamente falando, não há contas ou carteiras na camada de protocolo. Em vez disso, as moedas são armazenadas como uma lista de saídas de transações não gastas, ou seja, em UTXOs. Cada UTXO possui uma quantidade e um critério para gastá-lo. As transações são criadas consumindo UTXOs existentes e produzindo novos UTXOs em seu lugar.

No modelo UTXO, os usuários são incentivados a gerar um novo endereço para cada transação recebida. Ao usar um novo endereço a cada vez, é difícil vincular definitivamente moedas diferentes a um único proprietário. Este modelo de privacidade, baseado em endereços furtivos

(*stealth addresses*) [12], possui uma estrutura mais robusta contra vazamentos e contra variados ataques.

2.4 Privacidade na Monero

A Monero é uma cripto-moeda especializada em anonimidade. Utiliza três princípios para garantir a privacidade dos seus usuários. De maneira resumida, ela criptografa os valores transacionados, utiliza endereços furtivos e embaralha os emissores da transação utilizando a assinatura em anel a partir de outras transações pendentes na blockchain. Isso proporciona um embaralhamento, as transações respeitam os princípios de não rastreabilidade e desligabilidade propostos por T. Okamoto e K. Ohta[8].

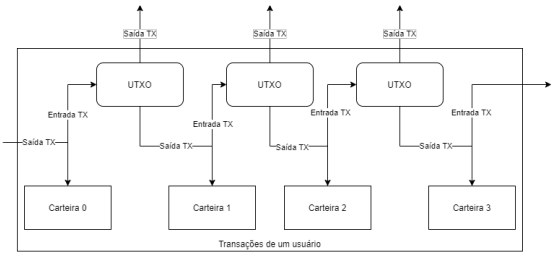
2.5 Privacidade na ChainXS

A utilização de sistemas UTXO em *blockchains* com programabilidade tem sido um grande desafio tecnológico. Com o intuito de permite que os dois tipos de transação operem dentro da mesma rede, a ChainXS utiliza o conceito de Livro Razão Quimérico (*Chimeric Ledgers*)[13].

Este conceito permite a utilização de endereços furtivos e permite que as transações possam ser verificadas trivialmente em paralelo, aumentando a escalabilidade da rede. Isto se deve à natureza apátrida das transações UTXO. As transações não se referem a nenhuma entrada fora dos UTXOs consumidos e das assinaturas correspondentes.

A ChainXS não realiza a criptografia dos valores transacionados, pois ela tenta conciliar auditoria e privacidade. Dessa forma, ela mantém um registro público e aberto dos dados da rede sem expor seus verdadeiros autores. Está apresentado, na Figura 1, uma esquemático representativo da carteira de um usuário ChainXS.

Figura 1: Carteiras de um Usuário ChainXS



A Figura 1 mostra o recebimento de valores através da saída de uma transação e fica armazenado na carteira zero. Quando o usuário realiza o saque, todo o saldo armazenado na carteira zero é gasto, uma parte do saldo é enviando para a carteira um ou mais destinatários e, caso necessário, o saldo restante é armazenado na nova carteira 1 e assim por diante.

Com esta medida, a rede cresce de forma caótica. Ainda que se vincule uma identidade a uma carteira, é improvável rastrear a identidade pela rede, a longo prazo. Ademais, o sistema se mantém auditável e qualquer fraude na rede se torna detectável.

2.6 Comparativo de funcionalidades de privacidade

Segue, na Tabela 1, um comparativo resumido das principais funcionalidades implementadas, referentes à privacidade, por outras *blockchains* e pela ChainXS.

Blockchain	Modelo de Transação	Programabilidade	Privacidade	Auditabilidade	Endereços Furtivos
Bitcoin	UTXO	Limitada	Não permite	Permite	Opcional
Ethereum	Contas	Avançada	Não permite	Permite	Não permite
Hyperledger	Contas	Avançada	Não permite	Permite	Não permite
Monero	UTXO	Limitada	Permite	Não permite	Permite
ChainXS	Quimérico	Avançada	Permite	Permite	Permite

Tabela 1: Comparativo de funcionalidades de privacidade

Neste parágrafo, todos os comentários estão restringidos às *blockchains* apresentadas na Tabela 1. Como se pode ver a ChainXS é a única tecnologia a utilizar o conceito de transação Livro Razão Quimérico. Sua programabilidade é avançada, fato que observamos apenas nas tecnologias Ethereum e Hyperledge. A ChainXS permite privacidade, de maneira similar à Monero (as demais não apresentam este recurso), porém com o aditivo da auditabilidade. Por fim, a ChainXS também trabalha com endereços furtivos, fato que adiciona um nível a mais de segurança à rede.

3 Privacidade em Rede de Responsabilidade Regulada (RLN)

Tendo em mente o sistema proposto no Piloto do Real Digital, foi projetado um sistema, com três partes principais, capaz de utilizar endereços furtivos de maneira eficiente. Tais partes estão apresentadas, e comentadas, nos itens a seguir.

- Real Tokenizado: Nesta parte, o padrão técnico ERC20 (*Ethereum Request for Comment - Standard*

Token) foi utilizado em compatibilidade com endereços furtivos, na rede quimérica ChainXS.

- Contrato STR (*Sponsored Transaction Relayers*): Este contrato controla quais endereços furtivos estão liberados para transacionar valores.

- Oráculo Validador de Endereços (EAO - *Enable Address Oracle*): Este é um sistema de oráculo que coleta o KYC (*Know Your Customer*) juntamente com uma chave pública estendida do usuário. Quando este oráculo detecta um endereço conhecido, com saldo positivo, o oráculo interage com o contrato STR informando os novos endereços, que estão habilitados para transacionar.

3.1 Carteiras Determinísticas Hierárquicas

As carteiras determinísticas hierárquicas, introduzidas pela comunidade Bitcoin, suportam diferentes tipos de *blockchains*. Tais carteiras possuem uma estrutura em árvore, na qual cada nó possui uma chave pública e uma chave privada estendida. Qualquer nó pode ter um número qualquer de filhos[11].

A Figura 2 ilustra um exemplo de uma carteira determinística hierárquica. Inicialmente, temos a semente, que é um número aleatório. A partir de desse número, podemos derivar uma chave pública e uma chave privada, que se torna a carteira principal do usuário. Ademais, cada carteira pode derivar a chave pública estendida e a chave privada estendida. As chaves estendidas podem gerar um número qualquer de chaves filhas.

O oráculo com função de EAO pode identificar os endereços furtivos que o usuário utilizará através do sistema de Carteiras Determinísticas Hierárquicas. Ao compartilhar a chave pública estendida com o oráculo, uma única vez, o mesmo não conseguirá assinar transações no nome do usuário, mas, por outro lado, terá acesso a todo o histórico de transações. A partir de um endereço furtivo, gerado por uma chave estendida, também não é possível obter os outros endereços gerados. Logo, apenas duas entidades poderão ter acesso ao histórico de transações, ao saldo e aos dados do autor das transações: o próprio usuário e o oráculo.

A arquitetura de Rede de Responsabilidade Regulada (RLN), proposta neste manuscrito, possibilita a existência de um ou mais oráculos com a função de EAO. Ademais, os próprios agentes financeiros podem hospedar seus próprios oráculos, bem como outras instituições, como, por exemplo, o Banco central, e entidades reguladoras. Uma arquitetura com várias entidades validadoras criará um ambiente descentralizado, onde nenhum agente terá acesso a todos os dados. Um único usuário, poderá também registrar diferentes chaves públicas estendidas em diferentes EAOs.

Figura 2: Carteira Determinística Hierárquica

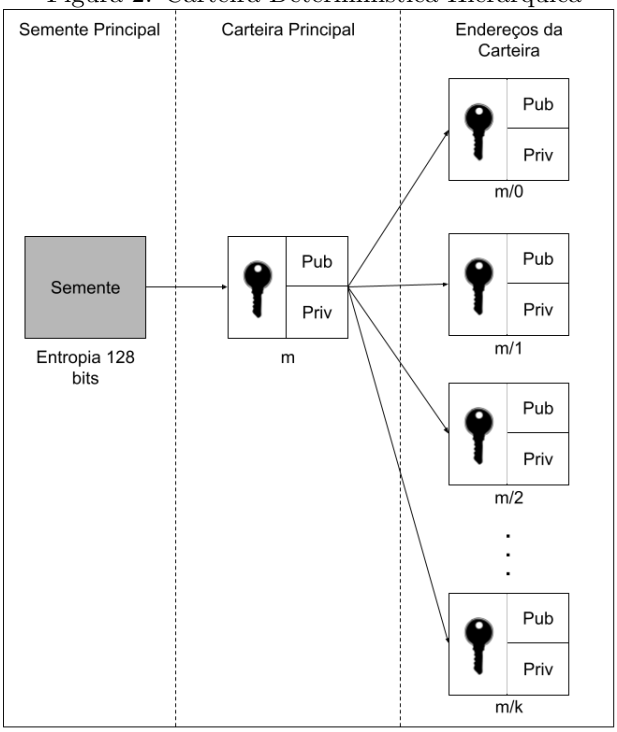
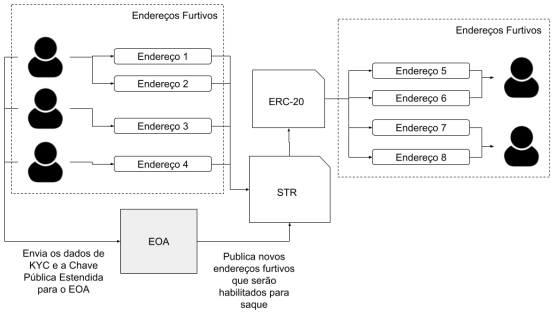


Figura 3: Transação de ativos tokenizados com endereços furtivos



Na Figura 3 temos um exemplo de uma transação de um ativo tokenizado na rede ChainXS. Cada transação possui um ou mais remetentes usando um ou mais endereços furtivos e um ou mais destinatários usando um número ainda maior de endereços furtivos. Todas as entradas se misturam em uma piscina (*pool*) temporária para serem distribuídas na saída aos destinatários - o troco é enviado para novos endereços dos remetentes. Os novos

endereço gerados são, posteriormente, identificados pelo oráculo e liberados para saque.

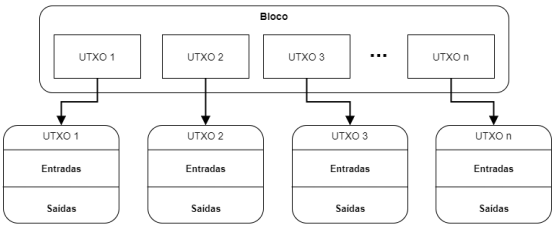
Esta arquitetura tem vários benefícios, dentre os quais podemos citar a privacidade e o baixo tempo de confirmação das transações, sem abrir mão da programabilidade. A privacidade da rede é obtida a partir da não rastreabilidade e da desligabilidade nas transações e nos saldos. Outro benefício, relativo à privacidade, reside na própria arquitetura do sistema, pois a mesma permite que apenas agentes autorizados acessem o histórico de transações, bem como os saldos dos usuários. Em respeito ao tempo de confirmação das transações, a arquitetura do sistema permite que as operações de transferência sejam confirmadas em poucos segundos, assim como o PIX. Porém, os fundos ficam retidos para o destinatário durante um curto período de tempo - cerca de alguns minutos - para, então, serem liberados para a próxima transferência.

Estes benefícios garantem, a todos os integrantes, a possibilidade de auditar a rede. Quanto mais a RLN é utilizada, mais segura e anônima ela se torna, sem prejuízo para as entidades reguladoras, pois as mesmas conseguirão monitorar e extrair informações privativamente. Por fim, a solução proposta torna a rede escalável, devido ao fato de que cada transação pode ser verificada trivialmente, em paralelo, permitindo maior capacidade de processamento de transações com o aumento da quantidade de nós na rede.

4 Escalabilidade

Em uma *blockchain* tradicional, os blocos são auto contidos, tendo todas as transações, *hashes* e assinaturas em um único pacote de dados. Esta estrutura concatena os blocos para formar o livro razão, que mostra o histórico de transação desde o princípio da rede. Este arranjo está esquematizado na Figura 4, a seguir.

Figura 4: Bloco padrão



Um nó validador que deseje gerar um novo bloco deve montá-lo através de um armazenamento compartilhado de transações denominado de *mempool*. Neste ponto da rede ficam armazenadas as transações que ainda não foram adicionadas na *blockchain*.

4.1 As Limitações

A Rede do Bitcoin possui limites bem conhecidos. Um bloco demora em média 10 minutos para entrar na *blockchain*, chegando a possuir até 4000 transações, o que leva a um valor máximo aproximado de 6.67 transações por segundo[9], conforme se vê na Equação 1.

$$\frac{maxTransactions}{blockTime} = \frac{4000}{10 * 60} = 6.67 \qquad (1)$$

Os *forks* e redes baseadas no Bitcoin trazem alterações como o tamanho de bloco e tempo de processamento de bloco para maximizar a escalabilidade. Como exemplo temos a Bitcoin Cash, com um tamanho de bloco de 8 MB e um tamanho médio de transação de 480 bytes, possibilitando uma taxa máxima aproximada de 56 transações por segundo, conforme se vê na Equação 2. Outros desenvolvedores tentaram aprimoramentos semelhantes nesta direção, porém, com resultados bem distantes das 65 mil transações por segundo da gigante de pagamentos VISA.

$$\frac{maxTransactions}{blockTime} = \frac{\frac{8*(1024*1024)}{250}}{10 * 60} = 55.92 \qquad (2)$$

Hoje, existem outros tipos de *blockchains* que chegam a taxas muito altas. Para tanto, elas utilizam outras formas de consenso, tais como a *Proof of Stake* e a *Proof of Authority*. Ambas as estratégias geram muitas dúvidas quando a segurança e, por esta razão, não possuem a massiva validação experimental que a Bitcoin deu ao *Proof of Work*.

Como forma de exemplo, é possível citar a Solana, constituída por uma variação de *Proof of Stake*. Esta *blockchain* possui um tempo de bloco de 400 milissegundos e um tamanho de bloco de até 128 MB. Em teoria, a rede poderia atingir um TPS máximo de 65.000, porém, a rede já passou por várias quedas, ficando indisponível por várias horas. Este problema ocorre, em parte, devido a um problema de atraso durante o processo de propagação dos blocos na rede. Com tempo insuficiente para a propagação a rede pode iniciar um processo de bifurcação na *blockchain*, também conhecido como *fork*.

Os *forks* acontecem quando dois nós geram um bloco válido no concurso, quase simultaneamente. Desta forma, os dois nós transmitirão para os nós vizinhos e a rede se dividirá conforme a sequência de recebimento dos blocos. Estes fenômenos acontecem com uma certa frequência, mas, felizmente, existem algoritmos para decidir qual árvore do livro razão será continuada, sem necessidade

de intervenção. Na prática, esse processo acontece várias vezes por dia em uma *blockchain* com grandes volumes de dados.

Christian Decker e Roger Wattenhofer modelaram a probabilidade de *forks* acontecerem na *blockchain* da Bitcoin[3]. Esta probabilidade está diretamente relacionada com o tamanho do bloco e com o tempo médio entre um bloco e outro. Caso uma *blockchain* tenha blocos muito grandes e tempo médio de concurso muito baixo a rede gerará *forks* mais rápido que os trata, sendo dividida a todo momento em várias *sidechains,* com arvores do livro razão próprias. Também não é ideal que a taxa de surgimento de *forks* seja próxima a de tratamento. Pequenas *forks* podem ser quase inofensivas, mas o problema passa a ser relevante se ocorrer a separação da rede em grandes *forks*, fazendo com que a *blockchain* cresça separadamente[6].

A única solução é a intervenção, de modo que apenas uma arvore do livro razão seja selecionada - as arvores paralelas e suas transações desaparecem como se nunca tivessem existido. Nestes casos, os prejuízos podem ser grandes, pois muito dinheiro acaba sendo circulado paralelamente e, de forma súbita, as transações podem ser revertidas. Infelizmente, nestas situações, todo o dinheiro gasto nestas transações poderá sofrer reversão, junto à *blockchain*, com boa parte não recuperada.

Portanto, problemas de estabilidade e de segurança fazem com que a simples alteração do tempo de entrada de bloco, ou do seu tamanho, seja inviável. O aumento da velocidade da *blockchain*, em duas ou três vezes, poderia, nestas situações, piorar a propagação dos blocos na rede, ou seja, o aumento da velocidade poderia diminuir proporcionalmente a segurança e, consequentemente, colocar a rede em risco de quebra.

Para aumentar a escalabilidade até valores muito altos é preciso fazer mais do que equilibrar variáveis. É necessário rever todos os processos da rede para encontrar desperdícios de poder computacional e reavaliar a necessidade de algumas etapas.

5 Estrutura de bloco e propagação da ChainXS

Um dos processos revistos na ChainXS é a validação completa do bloco antes da transmissão para outros nós. Esta medida é extremamente necessária, pois um *hacker* pode realizar *spam*, gerando uma quantidade enorme de blocos inválidos para congestionar a rede.

Durante a validação, cada nó acaba segurando a informação por um tempo, gerando um atraso. Este atraso aumenta o tempo da propagação do bloco, dando margem para outro bloco válido aparecer e gerar um *fork* na rede, mesmo que temporário. Este fenômeno é chamado de colisão de blocos. Para minimizá-lo, é necessário que a rede propague um bloco muito rapidamente, pois quanto menor for a janela de tempo disponível para colisão, menor será probabilidade da mesma ocorrer.

Para evitar *spam* o bloco precisa ser validado antes de transmitido, mas otimizações podem ser feitas. A estratégia da ChainXS é utilizar a *mempool* da rede para pré-processar as transações. Devido ao uso da *mempool*, em uma *blockchain*, qualquer nó da rede passará a conhecer as transações muito antes de entrarem oficialmente para a *blockchain*. Pôde-se, então, validar todas as transações da *mempool* antes mesmo de se ter um bloco pronto. A *hash* do bloco serve como identificador e, caso necessário, também pode ser utilizada para verificar alguma alteração.

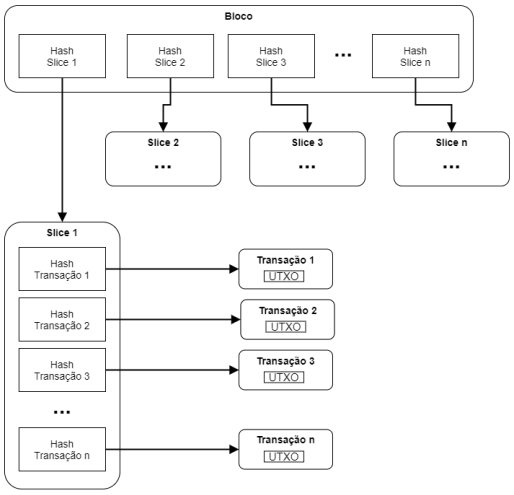
Se apenas as *hashs* das transações forem armazenadas no bloco, criptografadas pelo algoritmo SHA-256, o bloco será composto por hashs com 32 bytes. Se cada bloco possuir 10 MBs então ter-se-ia uma taxa de 546,1 transações por segundo, conforme se vê na Equação 3.

$$\frac{maxTransactions}{blockTime} = \frac{\frac{8*(1024*1024)}{32}}{10*60} = 546.1 \qquad (3)$$

Quando um bloco é propagado, um nó verifica se as *hashs* das transações estão na *mempool* marcadas como válidas, reduzindo quase todo o atraso da propagação do bloco causado por uma validação subsequente. Esta adição é extremamente eficiente, com um ganho de velocidade de quase 10 vezes em relação a Bitcoin Cash. Entretanto, ainda é muito pequena em relação ao número de transações por segundo das grandes empresas de cartões.

A ChainXS utiliza, em sua arquitetura de bloco, o algoritmo chamado de Distribuição Uniforme de Dados (Uniform Data Distribution - UDD). Este algoritmo é formado por transações e *slices*, como mostrado na Figura 5, a seguir. Nesta arquitetura, o bloco é fragmentado em pequenos pacotes contendo as transações. Estes pequenos pacotes de transação são chamados de fatias ou *slices* e carregam as *hashs* das transações.

Figura 5: Blocos, *slices* e transações.



Uma nova *mempool* é adicionada à *blockchain* para conter as *slices*, que também são pré-validadas antes da propagação das mesmas, assim como as transações. Da mesma forma, na propagação de um bloco, a etapa de validação das fatias passa a ser desnecessária.

Esta estrutura é multiplicativa, permitindo um número muito maior de transações por bloco. Se a ChainXS vier a possuir *slices* de 1 MB e blocos com máximo de 10 MB, a mesma retornará um valor absurdamente alto de quase 17 milhões de transações por segundo - a quantidade máxima de transações por segundo é quadrática, conforme se vê na Equação 4.

$$\frac{maxTrans}{blockTime} = \frac{10 * ((\frac{1024^2}{32})^2)}{10 * 60} \approx 1.79 * 10^7 \qquad (4)$$

Tais valores altos de transações por segundo são devido à natureza quadrática da estratificação do bloco, associado a pré-validação das *slices* e das transações. Antagonicamente, se estes elementos fossem revalidados antes da transmissão do bloco, a propagação seria absurdamente lenta, gerando *forks* na rede.

Outra alteração foi realizada sobre as limitações dos blocos e das *slices*. Na ChainXS, estas limitações são feitas em número de transações e não sobre o tamanho em MBs, já que se é armazenado *hashs* de tamanho fixo e não transações/*slices* inteiras.

Também é esperado que com *slices* de 32.768 transações (1 MB) ocorresse uma ou duas *slices* por bloco, no começo da rede. Este fenômeno pode gerar vulnerabilidade para ataques ou práticas desonestas.

Para evitar poucas *slices* em um bloco, os mesmos foram divididos em regiões, conforme apresentado na Tabela 2. Cada região possui um máximo de transações por *slice*, algo semelhante aos épicos do *roadmap*, porém com os IDs das *slices* ao invés de IDs dos blocos na *ledger*.

Região	Começo	Fim	Transações por Slice
1	0	100	10
2	100	1.000	100
3	1.000	10.000	1.000
4	10.000	100.000	10.000
5	100.000	600.000	100.000

Tabela 2: Regiões do bloco ChainXS.

Para se utilizar as *slices* que contenham muitas transações é necessário, primeiramente, preencher as *slices* menores. Dificilmente, um bloco terá menos de 1.000 transações, o que é suficiente para preencher as primeiras 100 *slices*. A quantidade de transações por bloco, neste modelo, pode ultrapassar 60 bilhões - são mais de 100 milhões de transações por segundo.

5.1 Escalabilidade por processamento paralelo

A estrutura de propagação de blocos é muito similar ao protocolo de Torrent, pois opera com a fragmentação, a validação e a distribuição dos dados de forma descentralizada e paralelizada. As limitações de rede são praticamente eliminadas e o processamento da rede passa a ser limitado pelo poder computacional de cada nó. Devido à natureza dos *slices* e ao sistema de transações, baseado em UTXO, é possível paralelizar o processamento da rede em diferentes nós, aumentando, assim, o poder de processamento geral da rede. Dessa maneira, a capacidade de transações por segundo que a rede é capaz e operar, crescerá a medida que mais nós validadores forem sendo conectados à rede.

6 Prova de Conceito

Uma PoC (Proof of Concept) foi implementada para validar os conceitos de privacidade e escalabilidade da ar-

quitetura apresentada. A PoC foi realizada em um notebook de baixo poder computacional, dotado de um processador i7 de 6º geração com 3,5 GHz e 8 GB de RAM. Embora a capacidade teórica da ChainXS seja superior a 1 milhão de transações por segundo, a escalabilidade alcançada dessa PoC, na ChainXS, foi de 100 transações por segundo. Este número pode ser aumentando facilmente, realizando otimizações de código e utilizando *hardwares com* alto poder computacional. No quesito privacidade, todos os objetivos propostos aqui foram alcançados.

O recorde de escalabilidade da ChainXS é de 1000 transações por segundo. Este número vem aumentando ano a ano, devido ao amadurecimento e às otimização do código. Para efeito de comparação, em 2023, o PIX realizou cerca de 150 milhões de transações diárias, ou cerca de 1736 transações por segundo, em média[2]. Os valores de pico podem facilmente ultrapassar essa média. Os resultados obtidos até o momento indicam, diante das possibilidades de novas otimizações, que a ChainXS é viável para criação de um RLN.

7 Conclusão

A transparência e a auditabilidade são benefícios inerentes ao livro razão distribuído, porém, elas frequentemente resultam na perda de privacidade para os usuários. Quando carteiras são vinculadas a identidades reais, as transações e os saldos tornam-se públicos, comprometendo a confidencialidade. Modelos baseados em contas, tais como os utilizados por *blockchains* permissionadas baseadas em EVM, apresentam desafios adicionais para a privacidade, por permitirem a visualização de transações e saldos entre instituições participantes. Em contraste, o modelo UTXO, exemplificado pela Monero, utiliza endereços furtivos e técnicas tais como a assinatura em anel, para garantir o anonimato e a segurança, dificultando a rastreabilidade das transações. Entretanto, essa estratégia acarreta em prejuízos, no sentido da programabilidade e da auditoria da rede.

A ChainXS Blockchain propõe uma solução inovadora ao combinar transações UTXO com programabilidade, utilizando o conceito de Livro Razão Quimérico e a distribuição uniforme de dados. Esta abordagem permite a verificação paralela de transações, aumentando a escalabilidade sem sacrificar a privacidade. Adicionalmente, a utilização de carteiras determinísticas hierárquicas, em uma RLN, e de agentes validadores oferece uma estrutura robusta para privacidade e auditabilidade, permitindo que apenas os próprios usuários, bem como os agentes autorizados, acessem o histórico de transações. Esta arquitetura descentralizada incentiva a utilização da rede por vários validadores e utilizadores, garantindo à mesma auditabilidade, segurança, e promovendo um equilíbrio entre privacidade e transparência.

References

[1] Vitalik Buterin. Ethereum white paper: A next generation smart contract and decentralized application platform, 2013.

[2] Banco Central. Estatísticas do pix. https://www.bcb.gov.br/estabilidadefinanceira/estatisticaspix. Acessado em maio de 2024.

[3] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. *IEEE P2P 2013 Proceedings*, 2013.

[4] Deloitte's. 2020 global blockchain survey. https://www2.deloitte.com/content/dam/insights/us/articles/6608_2020-global-blockchain-survey/DI_CIR%202020%20global%20blockchain%20survey.pdf. Acessado em maio de 2024.

[5] Linux Foundation. Hyperledger whitepaper. https://blockchainlab.com/pdf/Hyperledger%20Whitepaper.pdf, 2015.

[6] Bitcoin Github. March 2013 chain fork post-mortem. https://github.com/bitcoin/bips/blob/master/bip-0050.mediawiki. Acessado em outubro de 2020.

[7] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[8] Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash, 1991.

[9] Erlend Solberg Thorsrud. Long-term bitcoin scalability. https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2562793/19811_FULLTEXT.pdf, 2018. Acessado em outubro de 2020.

[10] Nicolas van Saberhagen. Cryptonote v 2.0. https://cryptonote.org/whitepaper.pdf, 2013.

[11] Pieter Wuille. Hierarchical deterministic wallets. https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki. Acessado em maio de 2024.

[12] Gary Yu. Blockchain stealth address schemes. Cryptology ePrint Archive, Paper 2020/548, 2020. https://eprint.iacr.org/2020/548.

[13] Joachim Zahnentferner. Chimeric ledgers: Translating and unifying utxo-basedand account-based cryptocurrencies. Cryptology ePrint Archive, Paper 2018/262, 2018. https://eprint.iacr.org/2018/262.