TECNOLOGIA BLOCKCHAIN:

Fundamentos, Tecnologias de Segurança e Desenvolvimento de Software.

ALEXANDRE MELO BRAGA, MSC. CISSP, CSSLP, PMP
AMBRAGA@CPQD.COM.BR

01 Introdução

Muito tem sido falado sobre Blockchain e criptomoedas. De fato, esta tecnologia tem despertado grande interessante no cenário das Tecnologias da Informação e Comunicação (TIC). Porém, existe muito mais na tecnologia Blockchain que as moedas criptográficas. Este whitepaper aborda, a partir de um ponto de vista tecnológico, dois assuntos do universo Blockchain que possuem diversos desafios e oportunidades: desenvolvimento de aplicações e segurança de sistemas.

A tecnologia Blockchain é considerada [1] um acelerador de inovação na indústria, sendo baseada nas capacidades da 3ª plataforma tecnológica, que é caracterizada pela computação ubíqua (qualquer lugar e hora) e consumida por comunidades colaborativas.

A 1ª plataforma foi caracterizada pelos mainframes e redes de dados, já a 2ª foi a Internet, os PCs e as redes locais [1].

Privacidade, escalabilidade e interoperabilidade são três desafios da tecnologia Blockchain comuns a várias aplicações [1]. Outros desafios são a transferência de dados em grandes volumes, a integração aos sistemas existentes e segurança, que depende em grande parte em como a aplicação blockchain é construída [1].

Este whitepaper consolida informação de várias fontes e estudos recentes, apresentando-os de um ponto de vista diferenciado e voltado para o desenvolvimento de aplicações Blockchain seguras. Este whitepaper está organizado da seguinte forma. A Seção 2 explica os conceitos fundamentais da tecnologia, enquanto a Seção 3 detalha o desenvolvimento de software baseado em Blockchain. A Seção 4 aborda as questões de segurança de sistemas Blockchain e a Seção 5 faz considerações finais. Ainda, o leitor interessando pode consultar as referências bibliográficas na Seção 6.

Este whitepaper faz parte de uma série de cinco planejados pelo CPqD para serem lançados ao longo de 2017. O primeiro, denominado "Tecnologia Blockchain: uma visão geral", está disponível no site do CPqD[1].
Para os próximos meses de 2017, prevê ainda o lançamento dos seguintes whitepapers:

— Tecnologia Blockchain: Aplicações e iniciativas de governo e empresas;

— Tecnologia Blockchain: aplicações em IoT;

— Tecnologia Blockchain: Aplicações no setor elétrico.

---

1 Fonte: Tecnologia Blockchain: uma visão geral. www. https://www.cpqd.-com.br/wp-content/uploads/2017/03/cpqd-whitepaper-blockchain-impresso.pdf. Acessada em julho de 2017.

02 Fundamentos da Tecnologia Blockchain

Esta seção detalha os seguintes conceitos fundamentais e necessários para o entendimento do restante do texto: Blockchain, consenso distribuído, transação, criptografia para Blockchain, encadeamento de blocos, registro da transação, árvores de Merkle e propriedades técnicas do Blockchain.

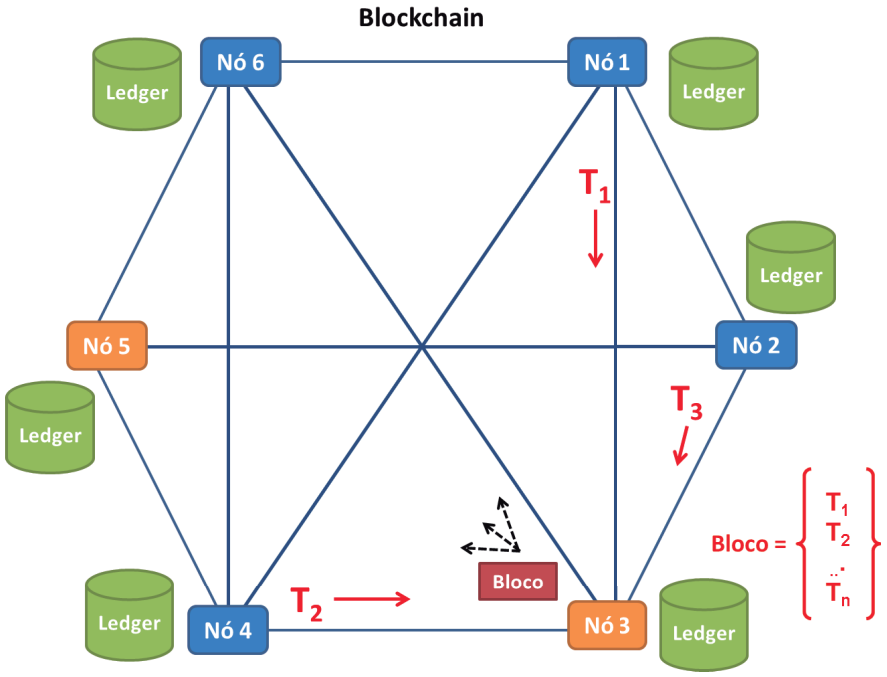

Figura 1 – Blockchain como uma base de dados distribuída.

02.1 O QUE É BLOCKCHAIN:

Em linhas gerais, um blockchain é uma base de dados distribuída e compartilhada pelos nós de um sistema distribuído organizado como uma rede peer-to-peer (P2P), conforme ilustrado na Figura 1.
Qualquer nó desta rede, com os direitos de acesso adequados, pode consultar e modificar a base de dados distribuída.

Os registros desta base de dados são chamados blocos. A base de dados somente aceita a inclusão de blocos novos e nunca a remoção ou modificação de blocos existentes. Por isto, a coleção de blocos é crescente e guarda a história desde a sua criação até o momento da atualização mais recente.

Um blockchain é um ambiente seguro para registro de transações, uma vez que não há adulteração e nem modificação dos registros já feitos.

O blockchain é mantido simultaneamente por todos os nós da rede P2P, não existindo local principal ou preferencial para armazenamento de uma base de dados original. Todo nó tem a sua réplica da base de dados, e todas são mantidas integras, consistentes e sincronizadas pelos protocolos de consenso. Este texto adota o termo ledger para a base de dados (coleção crescente de registros de transações) distribuída e blockchain para o sistema distribuído formado pela ledger distribuída e os nós da rede P2P.

Uma curiosidade é que, de fato, não existe qualquer informação na ledger que se pareça com uma moeda eletrônica (no sentido de uma sequência de bits unicamente identificável, distinguível das demais e transferível), apesar de o termo criptomoeda ser comumente associado ao Blockchain e ao Bitcoin.

02.1.1 **Consenso Distribuído:**

Muito se fala sobre os métodos de consenso. Uma dúvida frequentemente deixada em aberto em muitos textos se refere ao objeto de consenso: os nós da rede P2P decidem consensualmente sobre a ordem em que as transações são registradas na ledger.

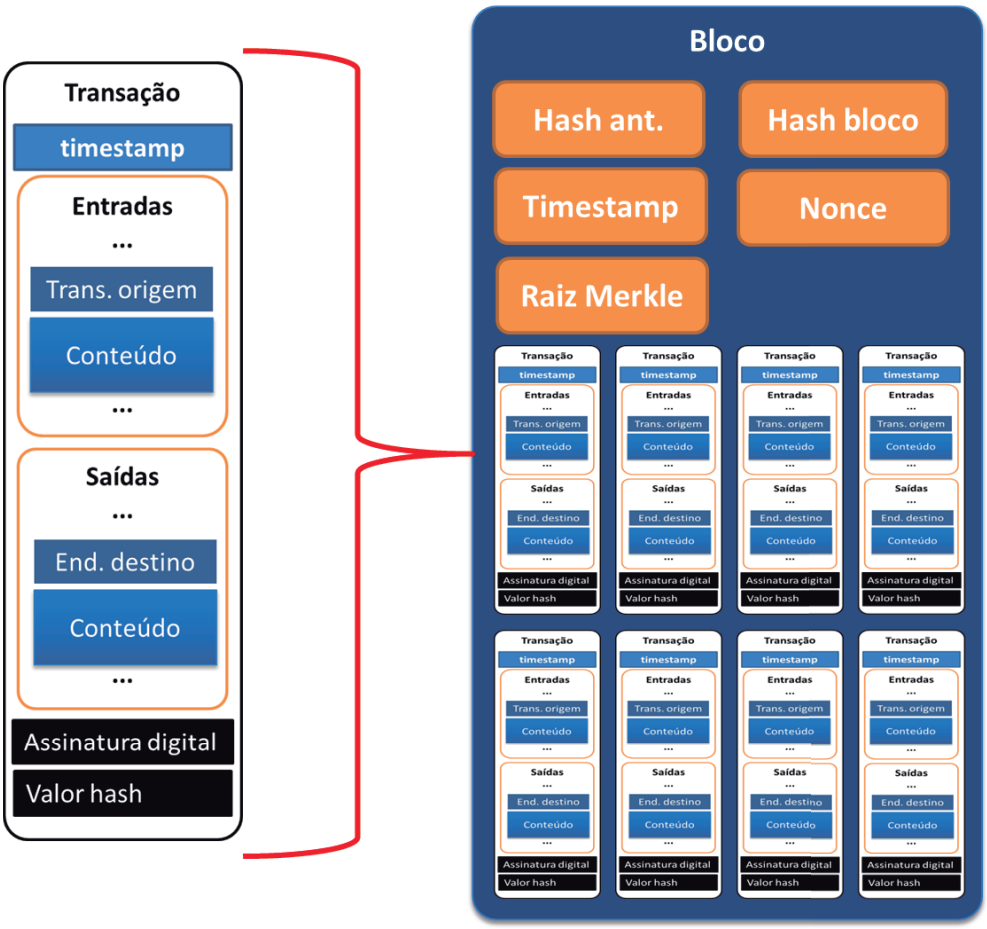

Figura 2 – Transação e Bloco.

Consenso distribuído é um termo da ciência da computação usado na disciplina de sistemas distribuídos e é um aspecto crítico do Blockchain e das criptomoedas. Consenso significa que quase todos (os envolvidos) concordam. Consenso é diferente de unanimidade, uma vez que nem todos tem que concordar, basta que a maioria concorde.

No Blockchain, o consenso ocorre entre os nós da rede P2P por meio de métodos compostos por protocolos específicos e regras bem definidas. Todos os nós da rede P2P são envolvidos de algum modo na tomada de decisão por consenso. Trata-se, portanto, de um grupo (ou comunidade) decidindo em conjunto e de modo confiável.

Opcionalmente, um nó centralizador (validador) pode coletar e propagar o consenso na rede P2P. O resultado de uma realização do protocolo de consenso deve ser confiável (determinístico) para toda execução.

Exemplos de métodos de consenso utilizados em blockchains são os seguintes: o consenso bizantino (caracterizado pela necessidade de 3*n+1 nós na rede P2P para tolerar n divergências no consenso), a prova de trabalho utilizada no Bitcoin para mineração, e a prova de participação utilizada pelo Ethereum.

Do ponto de vista de quem desenvolve aplicações sobre plataformas blockchain modernas, os métodos de consenso são uma funcionalidade, serviço ou configuração a ser habilitada e parametrizada.

Sendo muitas vezes transparente (em termos programáticos) para o desenvolvedor de aplicações.

02.1.2 **Transação:**

A estrutura de dados de uma transação reflete a semântica da aplicação. No caso das criptomoedas, esta estrutura se parece com um balancete contábil de débito e crédito e é composta dos seguintes elementos (Figura 2): um timestamp, o identificador (hash) da transação anterior de onde vem o valor de entrada (pode haver mais de um), o valor de entrada, o valor de saída, o endereço de destino (que vai receber o crédito), e uma assinatura digital feita com a chave privada do debitado[2].

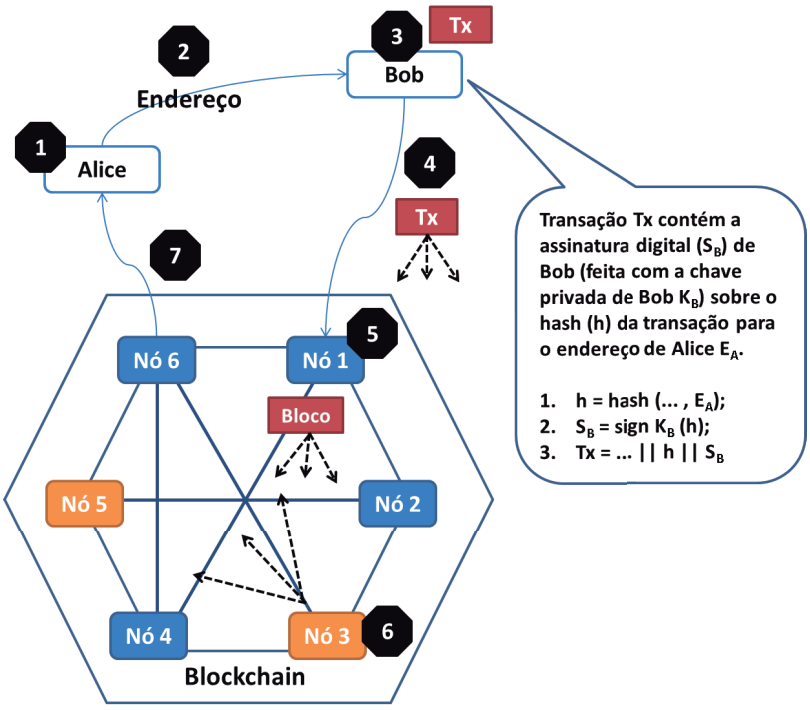

Figura 3 – Passos de uma transação.

O fluxo completo de ações para a realização de uma transação Blockchain fim-a-fim, detalhado a seguir e ilustrado na Figura 3, pode ser divido em quatro partes [3]: preparação dos clientes, registro da transação, consenso e consulta ou confirmação.

Tendo em vista dois usuários/clientes, Alice e Bob, a preparação dos clientes consiste de dois passos:

1 — Alice (Cliente A) cria sua conta (gera/escolhe um endereço);

2 — Alice divulga sua conta (endereço) para Bob (Cliente B).

O registro da transação ocorre em outros dois passos:

3 — Bob forma uma transação e a assina digitalmente para o endereço de A;

4 — Bob propaga a transação entre os nós da rede P2P.

O consenso é transparente para os clientes e ocorre em mais dois passos:

5 — Os nós da rede trabalham para obter o consenso e a transação é incluída em um bloco, de acordo com as regras da transação;

6 — Os nós da rede P2P propagam seu resultado para outros nós, a transação é aceita de acordo com o consenso e passa a fazer parte do Blockchain.

Finalmente, a consulta ou confirmação conclui o fluxo em um passo:

7 — Alice consulta a ledger e entende que sua transação foi aceita.

Muitas vezes, devido à natureza assíncrona da comunicação e ao tempo relativamente longo (para máquinas e não para pessoas) necessário para a realização do consenso, o último passo (confirmação) não é realizado.

Conforme será discutido adiante no texto, muitas fraudes e outros problemas de segurança em transações poderiam ser evitados simplesmente aguardando o tempo necessário e verificando a realização bem sucedida da transação.

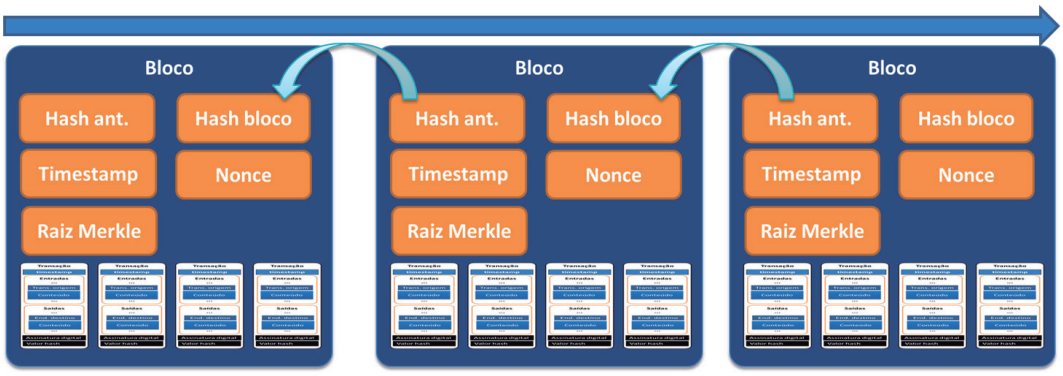

Figura 4 – A cadeia de blocos.

02.1.3 **A Cadeia de Blocos:**

As transações são incluídas em blocos (Figura 2), que estão ordenados em uma cadeia, formando uma estrutura de dados conhecida em computação como lista ligada (Figura 4). O bloco mais recente é a cabeça (head) da cadeia. Cada bloco contém um conjunto de transações e um cabeçalho composto dos seguintes itens: o hash do bloco anterior (ponteiro para o item anterior da lista), um número pseudoaleatório único (nonce), e o hash da raiz da árvore de transações no bloco.

As transações dentro de um bloco estão ordenadas entre si de acordo com uma estrutura em árvore binária baseada em hashes, que é conhecida como Merkle Tree. A Figura 5 ilustra a estrutura da árvore Merkle e a verificação de uma transação. Nesta estrutura, as folhas da árvore são os hashes das transações e os hashes dos pais são calculados com os hashes dos filhos. Por exemplo, os hashes dos ramos imediatos são calculados com os hashes das folhas, os hashes dos ramos intermediários são calculados com os hashes dos ramos imediatos, sucessivamente, até o cálculo do hash da raiz da arvore, que é incluído no bloco.

A estrutura em árvore acelera a operação de verificação se a transação pertence ao bloco, que pode ser feita em log(n) computações de hash, onde n é o tamanho da árvore. A verificação do hash de uma transação só usa o ramo da árvore (Merkle branch) em que a transação está localizada, que é necessário para verificar o hash da transação.

02.2 CRIPTOGRAFIA PARA BLOCKCHAIN:

As soluções comuns de Blockchain usam duas rotinas criptográficas. Primeiro, as funções de resumo criptográfico (vulgo funções de hash) são usadas na geração dos endereços, que consistem de valores hash calculados a partir das chaves públicas. Segundo, as assinaturas digitais utilizadas para garantir a autenticidade e irrefutabilidade das transações.

Funções de hash geram uma sequência de bits, o valor do hash, que é único para o documento de entrada da função. O hash é muito menor que o documento original e geralmente tem um tamanho fixo de dezenas (algumas centenas) de bits. A função de hash é unidirecional porque não é reversível, isto é, não é possível recuperar o documento original a partir da sequência binária do hash. Além disso, idealmente, não existem dois documentos que geram o mesmo valor de hash. Exemplos de funções de hash seguras utilizadas atualmente são o SHA-2 e o SHA-3.

A criptografia assimétrica (de chave pública) para assinatura digital é usada para obter integridade, autenticidade e irrefutabilidade.

Assinatura digital é o resultado de certa operação criptográfica com a chave privada sobre o texto claro. O dono da chave privada pode gerar mensagens

assinadas, que podem ser verificadas por qualquer um que conheça a chave pública correspondente.

O assinante não pode negar a autoria, pois há uma assinatura digital feita com sua chave privada. Por isto, a assinatura é irrefutável. A assinatura pode ser verificada por qualquer um com a chave pública. Mais detalhes sobre criptografia podem ser encontrados nas referências [4].

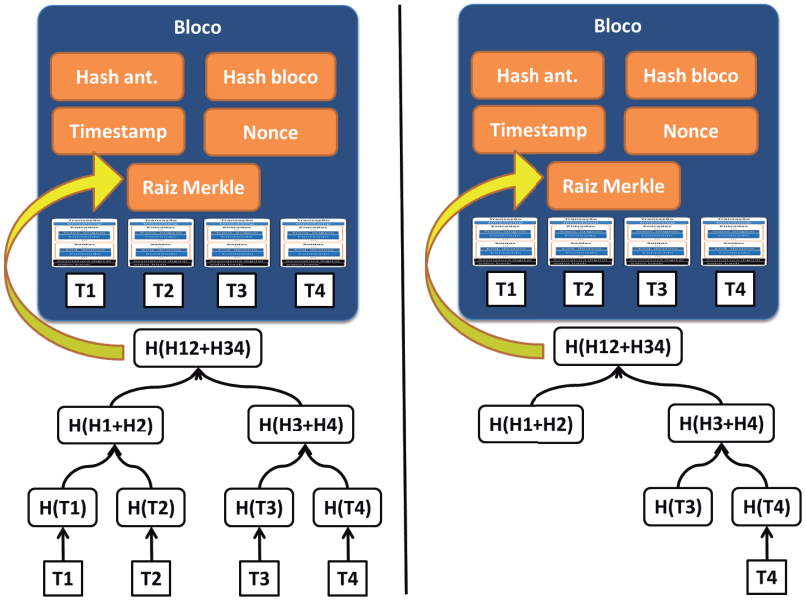

Figura 5 – Árvore Merkle das transações de um bloco (esquerda) e verificação da transação (direita).

assinadas, que podem ser verificadas por qualquer um que conheça a chave pública correspondente.

O assinante não pode negar a autoria, pois há uma assinatura digital feita com sua chave privada. Por isto, a assinatura é irrefutável. A assinatura pode ser verificada por qualquer um com a chave pública. Mais detalhes sobre criptografia podem ser encontrados nas referências [4].

02.3 PROPRIEDADES TÉCNICAS DO BLOCKCHAIN

O Blockchain tem uma séria de propriedades técnicas que são geralmente identificadas como benefícios para os negócios baseados nesta tecnologia. A propriedade mais conhecida em geral é a imutabilidade, em que a ledger somente é alterada de modo incremental e por consenso das partes envolvidas. Blocos e transações já incluídos na ledger são imutáveis.

A atualidade se refere à atualização periódica da ledger que sempre ocorre de modo autêntico e legítimo em intervalos curtos de atualização, geralmente próximos do tempo real. Já a irrefutabilidade garante que uma transação realizada, e replicada em todos os nós da rede, não pode mais ser negada pelo seu autor.

A prevenção contra a duplicação de transações garante que não há registro duplo de transações. Esta propriedade é importante no Bitcoin e outras criptomoedas, pois evita gastar o mesmo valor duas vezes, sendo uma proteção contra o ataque de double spending.

Duas propriedades relacionadas são a transparência e a visibilidade pública. Na primeira, todos os nós da rede P2P, assim como os softwares clientes com acesso só para leitura, veem as transações registradas.
Na segunda, todos os nós da rede têm acesso a ledger e podem verificar a sua legitimidade.

Descentralização se refere ao fato de não existir proprietário único da ledger, uma vez que todo nó da rede P2P é coproprietário, mantém a sua réplica da ledger e contribui para atualizar as outras réplicas. A

disponibilidade do Blockchain é geralmente alta porque alguns nós fora do ar não impedem o funcionamento dos outros nós, preservando a capacidade do chegar ao consenso. Vale observar que cada mecanismo de consenso requer uma quantidade mínima de nós disponíveis (operantes e conectados) para que o consenso seja viável.

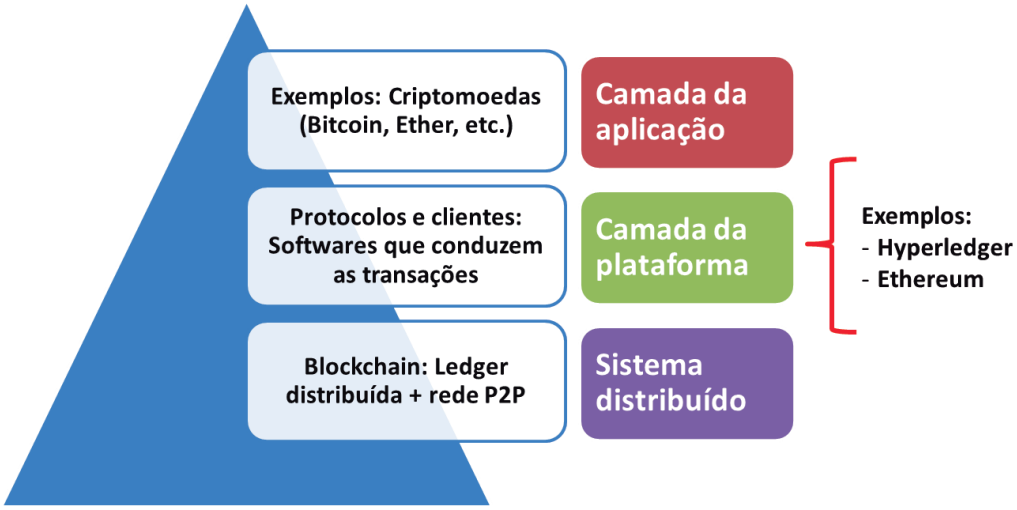

Figura 6 – Camadas tecnológicas do Blockchain.

Finalmente, a desintermediação é uma propriedade emergente da atuação do Blockchain como um conector de sistemas complexos (sistemas de sistemas), geralmente eliminando intermediários artificiais nas integrações entre sistemas e abrindo espaço para a simplificação de processos.

03 Desenvolvimento de Aplicações

Esta seção aborda o desenvolvimento de aplicações em software da tecnologia blockchain e divide o assunto em três partes: a aplicação propriamente, o software cliente de usuário final, e as decisões de projeto de sistemas blockchain.

03.1 A APLICAÇÃO BLOCKCHAIN:

A aplicação Blockchain é abordada por três aspectos relacionados: as camadas tecnológicas fundamentais, a arquitetura em módulos da aplicação e os contratos inteligentes.

03.1.1 Camadas de software:

Em linhas gerais, uma aplicação Blockchain é composta por três camadas [5] ilustradas na Figura 6. A primeira camada é a do sistema distribuído que consiste na infraestrutura fundamental, responsável pela implementação do conceito de “ledger distribuída” e as funcionalidades necessárias para que ele possa ser utilizado, tais como métodos de consenso, armazenamento da ledger e protocolos de comunicação ponto a ponto, é o que normalmente se chama de Blockchain.

A segunda camada contém os serviços de apoio e infraestrutura, que viabilizam o desenvolvimento de aplicações robustas e seguras, relacionados à gestão de chaves criptográficas, integridade e confiabilidade de dados, disponibilidade de nós da rede P2P, rastreabilidade de transações, gestão de identidade, sigilo, privacidade, reputação, entre outros aspectos de

segurança (de acordo com o nicho de aplicações preferencial da aplicação e do Blockchain), sendo geralmente associada à camada de plataforma. Fazem parte desta camada os softwares que conduzem as transações e as plataformas como Hyperledger [6] e Ethereum [7].

A terceira camada é a de aplicação, sendo composta não apenas pela lógica de negócios da aplicação, como também pelos contratos inteligentes, como programas de computador que viabilizam a implementação, dentro de cada um dos nós da rede P2P, de parte das regras de negócio da aplicação. Os contratos inteligentes são discutidos adiante no texto.

As criptomoedas (tais como o Bitcoin) são aplicações sobre plataformas Blockchain e fazem parte desta camada. Esta visão simplificada em três camadas é geralmente elaborada em maior detalhe durante a implementação de sistemas sofisticados.

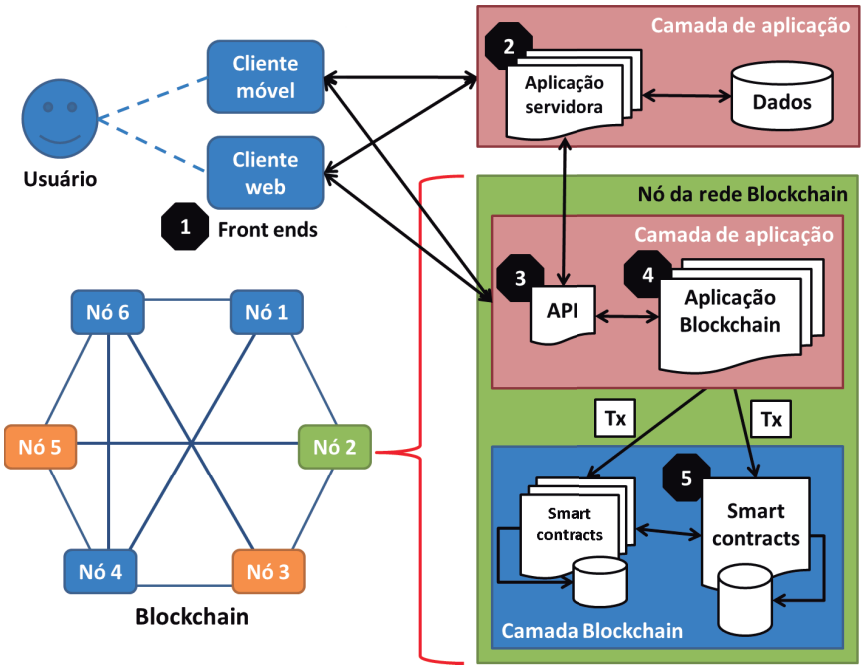

Figura 7 – Estrutura de uma aplicação Blockchain.

03.1.2 Estrutura de uma aplicação Blockchain

Em geral, as aplicações blockchain possuem uma arquitetura de software com cinco módulos bem definidos [8] ilustrados na Figura 7. O primeiro é o cliente ou front-end de usuário final, geralmente associado a aplicativos móveis e interfaces web.

O segundo consiste da aplicação servidora com regras de negócio e dados armazenados fora do blockchain, por meio de plataformas de software tradicionais e bases de dados comuns.

O terceiro módulo é uma camada (API) de integração entre a aplicação servidora (ou outros sistemas legados) e a aplicação blockchain.

O quarto é a aplicação blockchain que manipula (por meio de uma API) a ledger distribuída. Finalmente, o quinto é formado pelos contratos inteligentes como programas de computador implantados e executados em cada um dos nós da rede blockchain.

Em arquiteturas de software complexas, o Blockchain pode atuar como um conector de aplicações [8]. Uma aplicação insere dados, via transações, por um nó da rede P2P. Enquanto outra aplicação coleta, consulta ou recebe dados por outro nó da rede. Além disso, sistemas externos podem interagir com contratos inteligentes ou com nós da rede, simplificando a integração entre sistemas complexos (sistemas de sistemas).

Esta arquitetura de software reforça a ideia de que o blockchain pode ser entendido como um conector sofisticado de sistemas distribuídos grandes e complexos. Tais como, por exemplo, aqueles compostos por cadeias de valor longas ou redes de aglomerados de objetos inteligentes na internet das coisas.

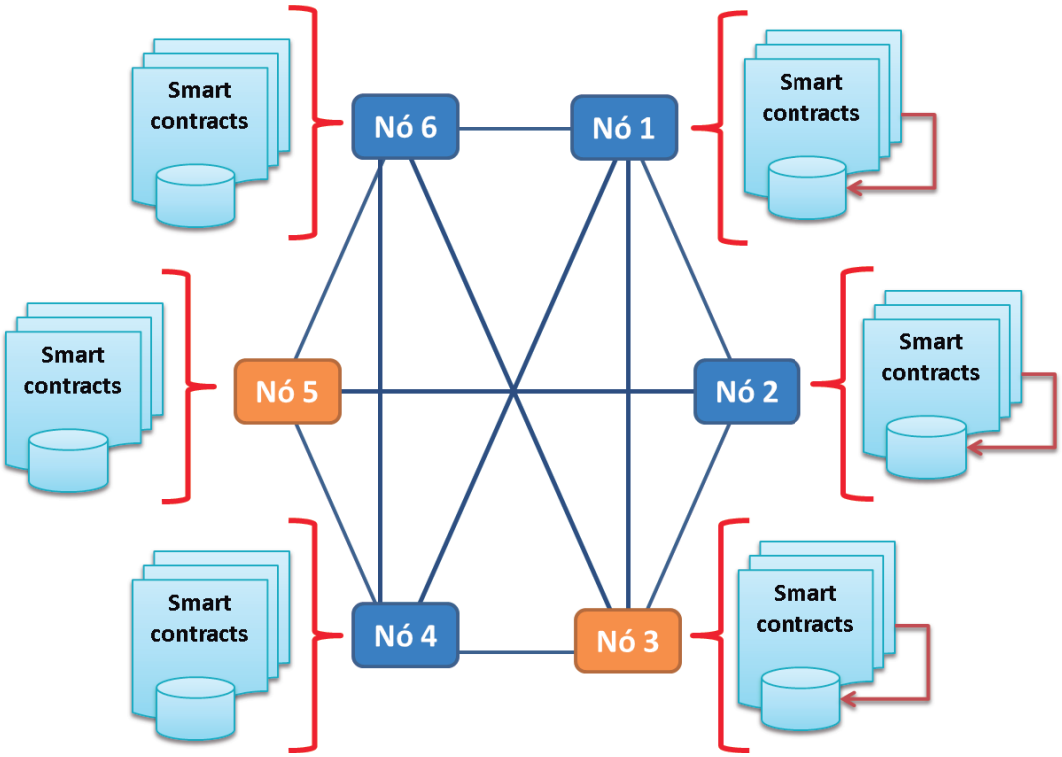

Figura 8 – Contratos inteligentes (Smart Contracts, Chaincodes).

03.1.3 Smart Contracts (Chaincode)

Blockchain e contratos inteligentes (chaincodes) são tecnologias complexas que possuem vários benefícios quando usados isoladamente. A combinação destas tecnologias complexas, em vários contextos de aplicação, levanta novas preocupações com segurança relacionadas não apenas às tecnologias isoladas em seus casos de uso comuns, mas também emergentes das

interações desconhecidas e até inesperadas entre estas tecnologias para resolver casos de uso incomuns (inovadores) em novas situações.

Os contratos inteligentes resolvem questões que necessitam de acordos com mínima confiança entre as partes participantes de um sistema distribuído [9], [10].

Conforme ilustrado na Figura 8, os contratos inteligentes são programas de computador (escritos em linguagens de programação gerais ou específicas) que podem ser corretamente executados por uma rede de nós mutuamente suspeitos de uma rede P2P, sem que seja necessária uma entidade externa confiável para mediação do acordo [9], [10].

Diz-se que os nós da rede são mutuamente suspeitos por que eles não precisam confiar incondicionalmente uns nos outros, uma vez que podem ser competidores ou até mesmo adversários.

O programa executável (binário) do contrato inteligente é implantado e executado nos nós da rede P2P de um Blockchain e sua execução correta é imposta (garantida) pelo consenso, que é composto por regras e protocolo.

O chaincode da plataforma Hyperledger, escrito na linguagem Go, e o Contract da Ethereum Virtual Machine (EVM), escrito na linguagem Solidity, são dois exemplos de contratos inteligentes.

03.1.4 O Cliente Blockchain (eWallet) e a gestão de chaves criptográficas

Também chamado de eWallet (carteira eletrônica) no jargão das

criptomoedas (e do Bitcoin), o software cliente blockchain é a aplicação, para usuário final, mais comum com os blockchains atuais, uma vez que existem em quantidade muito maior que os nós da rede P2P [2], [5], [11], [12].
Uma eWallet típica implementa mecanismos para armazenamento seguro de chaves criptográficas, assinatura digital de transações, encriptação de transações e transmissão segura de dados [13], [14].

A função principal de uma eWallet é a gestão de uma coleção de chaves privadas para a manipulação correta dos ativos de valor da aplicação [13], [14]. Ativos associados a cada uma das chaves.

A eWallet é personalizada para lógica da aplicação e adota metáforas específicas do negócio, que não estão necessariamente relacionadas às criptomoedas. Comumente, a eWallet não precisa participar do consenso, uma vez que não é um nó da rede P2P, mas é crítica para a proteção dos dados da conta do usuário final.

O Blockchain oferece uma grande oportunidade para a popularização da criptografia assimétrica (de chave pública). Grosso modo, a complexidade de utilização destes sistemas criptográficos de chave pública pelos usuários finais tem sido a principal barreira para adoção em larga escala das criptomoedas e outros sistemas baseados em Blockchain [13], [14].

Há inovações em usabilidade da gestão de chaves, boa parte delas está em buscar metáforas adequadas e abstrações claras, que ofereçam o uso transparente da criptografia [13], [14]. De modo geral, cabe aos desenvolvedores de aplicações entender que há mais no mundo Blockchain

que só criptomoedas e evitar sempre propor mais uma criptomoeda, mesmo em contextos onde esta metáfora não é adequada.

03.1.5 Decisões de projeto para aplicações blockchain

Os arquitetos, projetistas e desenvolvedores de aplicações baseadas na tecnologia Blockchain devem tomar uma série de decisões de projeto relacionadas não apenas a arquitetura do Blockchain, mas também a arquitetura da aplicação e como ela usa um Blockchain [8]. Estas decisões são explicadas a seguir.

Algumas decisões de projeto do Blockchain estão relacionadas aos mecanismos que afetam a velocidade de processamento de transações, tais como as seguintes:

— Tamanho do bloco: quantas transações fazem parte do bloco;

— Transações fora do blockchain: transações sobre dados armazenados fora da ledger, mas que deve ser vinculadas às transações na ledger;

— Tamanho das transações: sobre quais dados as transações são aplicadas;

— Quantas assinaturas por transação: uma única assinatura de quem origina a transação, duas assinaturas de origem e destino, ou até várias assinaturas de origem, destino e autorização;

— Protocolo P2P escalável, por exemplo, protocolos com mensagens leves e de baixa frequência, ou protocolos que garantam atualização rápida de réplicas em redes P2P grandes.

Outra decisão de projeto do Blockchain está relacionada ao mecanismo de consenso. Existem várias opções disponíveis para mecanismo de consenso, algumas das mais conhecidas são as seguintes: prova de trabalho (o método

adotado pelo Bitcoin), participação, consenso bizantino, e até nenhum (neste caso qualquer transação que entra na rede P2P é incluída na cadeia de blocos, o que pode resultar em inconsistências severas).

Geralmente, os requisitos da aplicação exigem que algum método de consenso seja escolhido. A não utilização de métodos de consenso põe em dúvida a necessidade do Blockchain.

Do lado da aplicação que usará Blockchain, as decisões de projeto mais importantes estão relacionadas aos seguintes assuntos:

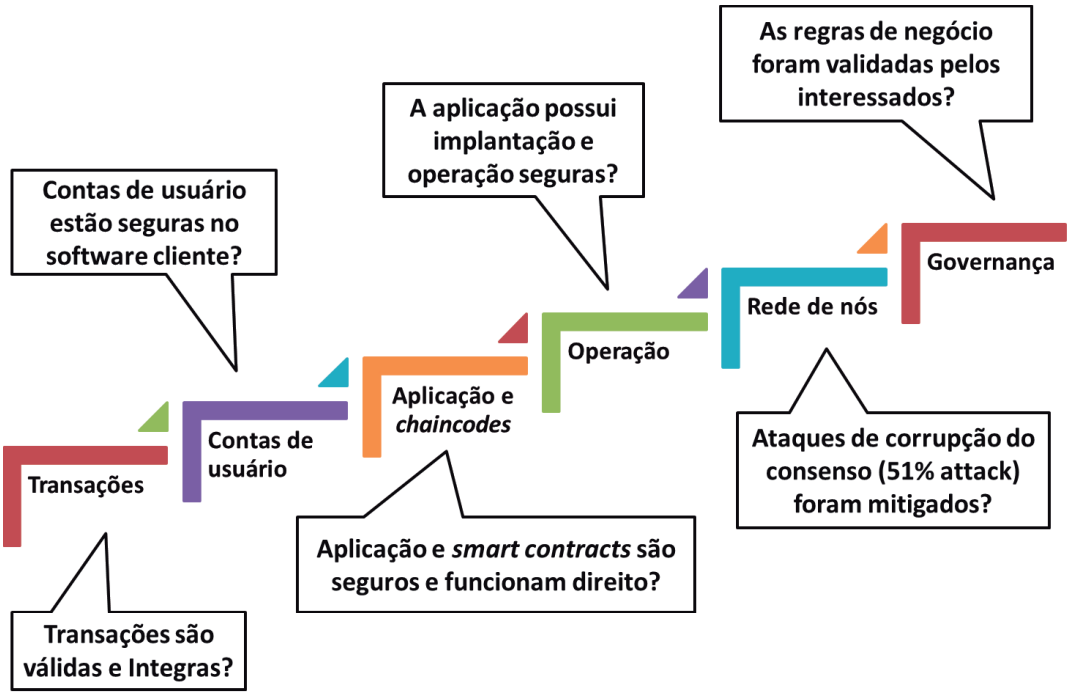

Figura 9 – Camadas de segurança da tecnologia Blockchain.

— Escopo de armazenamento dos dados, que podem estar armazenados no Blockchain (in-chain) ou fora do Blockchain (off-chain), em bases de dados convencionais.

— Acesso da aplicação aos nós de rede P2P, que pode ser pública (com acesso e irrestrito da aplicação aos nós), privada (os nós não podem ser

acessados diretamente e nem livremente) ou híbrida. O caso híbrido ocorre, por exemplo, quando eWallets só de consulta têm acesso irrestrito aos nós, mas as aplicações que registram transações somente o fazem por nós específicos.

— Quantidade de ledgers, que pode ocorrer em ledger única ou multi-ledger;

— Validação de transações: só com dados da aplicação ou com dados internos e externos. Em qualquer caso, a transação deve ser determinística.

— Permissionamento (de acesso) à ledger, que pode ser controlado ou livre. No caso controlado, o acesso também é identificado, autenticado e autorizado. Por isto, não há o anonimato que caracteriza o acesso livre.

04 Aspectos de Segurança

Esta seção analisa cinco aspectos de segurança da tecnologia Blockchain: segurança em camadas, vulnerabilidades mais comuns, privacidade e anonimato, segurança de contratos inteligentes, e ataques específicos contra criptomoedas.

04.1 CAMADAS DE SEGURANÇA DE UM BLOCKCHAIN

O Blockchain e as aplicações construídas com ele devem adotar a segurança em camadas. Há seis camadas de segurança a serem consideradas em uma aplicação Blockchain. Estas camadas são o resultado da compilação de boas práticas presentes em livros texto da área e estão ilustradas na Figura 9.

A camada fundamental é a segurança da transação. Requisito mínimo sem o qual o Blockchain não faz sentido. O Blockchain deve validar as transações com confiança e previsibilidade ao final do consenso. O consenso vai confirmar a finalidade e a imutabilidade de transação.

Trata-se de proteções sintática e estrutural para as transações e os blocos que as contém. Estas proteções não impedem fraudes semânticas associadas à lógica da aplicação.

A segunda camada oferece segurança da conta de usuário. A conta do usuário é geralmente gerenciada pelo próprio usuário em aplicativos de uso pessoal (eWallets). Muitas vezes, a proteção da conta do usuário é confundida com a segurança do software cliente.

Esta camada de segurança é influenciada por dois fatores: a conscientização dos usuários no uso seguro da tecnologia, e a implementação correta dos mecanismos de segurança para dispositivos móveis e sistemas web.

A terceira camada contempla a segurança da aplicação e dos chaincodes. Fazem parte desta camada as boas práticas de desenvolvimento seguro de software, incluindo a codificação segura de smart contracts e a definição de requisitos de segurança, avaliação de arquitetura e testes de segurança da aplicação.

A quarta camada atende a segurança de implantação e de operação da aplicação. Fazem parte desta camada os testes de aceitação e homologação da aplicação e dos chaincodes antes de implantação em produção. Uma vez no ambiente de produção, a aplicação deve ser monitorada para detecção de anomalias de funcionamento e comportamento. Monitoramentos avançados podem até detectar fraudes.

A quinta camada cobre a segurança da rede P2P de seus nós. Nesta camada, os mecanismos de proteção tradicionais das redes de computadores (tais como sistemas de firewall, IDS, IPS, etc.) podem ser aplicados para proteção dos nós da rede P2P do Blockchain. Além disso, proteções específicas devem ser aplicadas para a segurança do protocolo de comunicação e de consenso. Ainda, deve ser observada a quantidade mínima necessária de nós disponíveis para garantir o consenso.

A sexta camada de segurança se refere à governança da aplicação e do Blockchain. Esta camada abriga aquelas decisões sobre a estrutura e projeto

do Blockchain, discutidas na seção anterior, que afetam o funcionamento com segurança, incluindo ainda controles antifraude, auditoria, privacidade e até conformidade a normas padrões específicos do nicho de aplicação.

04.2 DEFEITOS COMUNS EM SISTEMAS BLOCKCHAIN

Um estudo bastante recente [15] identificou os defeitos de software (bugs) mais comuns em sistemas Blockchain, que são listados a seguir em ordem decrescente, do mais frequente para o menos frequente: semânticos (na lógica da aplicação), ambiente e configurações, interface gráfica, concorrência, build, segurança, alocação de memória, desempenho, compatibilidade, e bifurcação da ledger.

No geral, defeitos semânticos também afetam a segurança das aplicações, uma vez que diferenças entre requisitos e a intenção do programador podem facilitar a fraude (como no ataque DAO). Um exemplo deste tipo de defeito é a transação para endereços inexistentes (possível no Bitcoin).

Os defeitos de segurança não têm novidades e estão relacionados às ocorrências específicas de vulnerabilidades conhecidas, tais como as seguintes: overflow de inteiros no timestamp de um bloco causado por minerador malicioso, ataques por canal lateral de tempo (timing attack), viabilizado pelo modo com as senhas são comparadas na autenticação, diversas vulnerabilidades de SSL/TLS relacionadas à validação incompleta de certificados que facilitam ataques como o padding oracle e BEAST.

04.3 PRIVACIDADE E PSEUDO-ANONIMATO

No Blockchain tradicional, a privacidade é limitada por dois aspectos.

O primeiro é o pseudo-anonimato da transação e o segundo é o fato de que todas as transações estão em claro, isto é, sem criptografia para sigilo.

Usa-se o termo pseudo-anonimato em vez de anonimato verdadeiro por que a análise das correlações entre transações, os endereços de destino e outros metadados derivados da lógica da aplicação podem facilitar a revelação da identidade do usuário.

Por exemplo [1], no Bitcoin, a partir da análise das transações de origem e dos endereços de destino das transações, é possível identificar as movimentações financeiras entre endereços específicos, reconhecendo padrões de relacionamento entre usuários do Bitcoin.

A exploração de vulnerabilidades em carteiras eletrônicas, pode tanto facilitar o roubo de criptomoedas, como também revelar a identidade do usuário. Por exemplo [16], em eWallets de Bitcoin que utilizam o algoritmo criptográfico ECDSA para assinar transações, cada assinatura requer um número aleatório único e imprevisível.

Porém, defeitos de segurança em algumas destas eWallets, na geração e utilização de números pseudoaleatórios ruins, podem resultar na revelação de chave privada a partir da geração de assinaturas digitais ECDSA repetidas, facilitando o roubo de Bitcoins e o rastreamento de transações assinadas com estas chaves inseguras.

04.4 SEGURANÇA DE SMART CONTRACTS

A segurança de smart contracts é analisada por dois aspectos: defeitos de

segurança comuns a várias plataformas de smart contracts e vulnerabilidades específicas à Ethereum, a plataforma de contratos inteligentes mais utilizada atualmente.

04.4.1 Bugs de segurança em smart contracts:

Estudos recentes ([17], [18]) já catalogaram defeitos de segurança (vulnerabilidades) em contratos inteligentes. Quatro vulnerabilidades são mais conhecidas: a dependência da ordem de transações, a dependência do carimbo de tempo, o tratamento defeituoso de exceções e a vulnerabilidade de código reentrante.

— Dependência da ordem de transações: a ordem em que transações são executadas por um contrato inteligente pode alterar o resultado final deste chaincode. A vulnerabilidade TOD (Transaction-Ordering Dependence) ocorre quando um nó malicioso altera a ordem em que transações são executadas por um contrato. Por exemplo, em transações de compra e venda com criptomoedas, sabendo que o preço vai baixar, um operador malicioso processa primeiro as transações de pagamento com valor alto (processa o máximo que puder, antecipadamente, antes do preço baixar), deixando para depois que o preço baixar (o que é inevitável), o processamento de poucas com pagamento mais baixo.

— Dependência do carimbo de tempo: há contratos que usam o carimbo de tempo da transação (timestamp) como gatilho ou condição para alguma operação crítica. Por exemplo, timestamp é utilizado como semente de PRNG. Assim sendo, um nó malicioso que monta o bloco pode escolher um timestamp válido, porém enviesado, comprometendo a transação ou até a segurança da chave privada do usuário.

— Tratamento de exceções malfeito: quando um contrato inteligente aciona (chama) outro, ele deve estar preparado para o caso excepcional do contrato chamado não terminar sua execução corretamente. Se o término anormal não é tratado com a atenção devida, falhas no contrato chamador podem acorrer e até ser exploradas em ataques e fraudes. Em um exemplo simples, um contrato de crédito, que faz transferência de valores entre origem e

destino, não trata erro no contrato de débito e credita o valor na conta de destino, apesar do erro no débito, sem debitar da conta de origem.

— Vulnerabilidade em código reentrante: neste defeito, dois contratos mutuamente dependentes acessam estados intermediários (inseguros por que possivelmente inconsistentes) um do outro. Se o primeiro contrato toma decisões de negócios com base em estados intermediários inconsistentes, a decisão tomada é incorreta. Esta situação pode ser explorada em fraudes e outros ataques.

04.4.2 **Tipos de vulnerabilidades em smart contracts Ethereum:**

Estudos recentes ([17], [18]) categorizam as vulnerabilidades mais comuns em uma das plataformas Blockchain mais utilizadas atualmente, o Ethereum. Existem, em geral, três categorias de vulnerabilidades: as da linguagem de programação Solidity, as da máquina virtual Ethereum (Ethereum Virtual Machine - EVM), e aquelas associadas ao Blockchain Ethereum.

As vulnerabilidades da linguagem de programação Solidity são todas exploráveis em ataques que roubam ether (a criptomoeda Ethereum) de contratos. Em relação ao tratamento de exceções malfeito, 28% dos contratos não validam o retorno de funções. Ainda, campos privados de contratos podem ter seus valores públicos na ledger. Além disso, há a casos de vulnerabilidade em código reentrante.

Há vulnerabilidades descobertas nos binários dos contratos inteligentes (bytecodes) que são implantados nos nós da rede P2P e executados pela EVM. Contratos já implantados são imutáveis, pois são vinculados a transações no Blockchain, então bugs são difíceis de corrigir e a recuperação pode ser drástica, por exemplo, com uma bifurcação (hard fork) na ledger.

Em relação às vulnerabilidades associadas à implementação do Blockchain Ethereum e como ela gerencia contratos inteligentes, há casos de dependência da ordem de transações e dependência do carimbo de tempo.

04.5 ATAQUES ESPECÍFICOS CONTRA CRIPTOMOEDAS E O BITCOIN

O conhecimento e a análise de ataques contra o Bitcoin são instrutivos para evitar casos semelhantes em construções análogas. A maioria dos ataques se refere ao gasto repetido ou duplicado de Bitcoins (double spending).

Quatro ataques bem documentados [2] são descritos nos próximos parágrafos: o ataque 51%, o ataque de competição (race attack), o ataque do minerador malicioso (Finney Attack) e o spam ou enxurrada de transações (Transaction Spamming/flooding).

Sejam duas transações duplicadas que usam o mesmo valor de origem. A primeira transação a entrar no Blockchain é considerada válida e a outra é descartada. Um atacante poderoso poderia substituir uma transação que já entrou na Blockchain por outra que usa o mesmo valor de origem.

No ataque de 51% (51%+ attack), para substituir uma transação em um bloco, o atacante deve minerar de novo blocos anteriores e acompanhar o passo de geração de blocos novos. Para tal, a capacidade de computação de valores hash do atacante (hash rate) deve ser maior que a da rede P2P. Por isto, este ataque só é viável para um atacante que tem domínio da rede P2P, isto é, ele controla mais de 50% dos nós (ou da capacidade de hash) da rede.

Este atacante poderoso pode sequestrar a rede, se recusando a minerar

blocos de outros mineradores. Por isto, este também é um ataque contra outros mineradores.

No ataque de competição (race attack), transações duplicadas em nós diferentes da rede P2P, causam a falsa impressão de double-spending devido a latência da rede e às diferenças de tempo de propagação de blocos a partir de nós próximos ou de nós distantes.

O double spending aparente só existe até que a transação entre em um bloco e tal bloco alcance os nós da rede P2P. Os nós mais próximos percebem a duplicação antes dos nós mais distantes.

No ataque do minerador malicioso (Finney attack), o atacante minera um bloco em segredo com uma transação sua para si mesmo (autotransação).

A transação não é difundida na rede P2P. Antes de liberar o bloco secreto, o atacante divulga outra transação duplicada tendo como destino uma vítima, que aceita o pagamento sem confirmar o bloco (uma prática ruim).

Então, o atacante libera o bloco secreto antes que outro minerador ache bloco com a transação de pagamento para a vítima, passando o seu pagamento na frente. Daí o double-spending.

O ataque de spam ou enxurrada de transações (Transaction spamming/flooding) é um ataque de negação de serviço (Denial of Service - DoS) contra a rede P2P de um Blockchain. Neste ataque, uma enxurrada de autotransações inibe que o nós atacados processem outras transações.

Análises indicam que este ataque não é viável no Bitcoin, principalmente, por três motivos: apenas poucas transações grátis são permitidas em um bloco, a taxa de serviço do minerador encarece o ataque, e as transações de valor muito baixo são descartadas.

Porém, o ataque pode ser viável em outros blockchains com mecanismos de incentivo diferentes para o consenso.

04 Considerações Finais

Este whitepaper abordou, a partir da análise de estudos recentes, dois assuntos importantes do universo Blockchain: desenvolvimento de aplicações e segurança de software. O objetivo foi o de fomentar o desenvolvimento de aplicações blockchain seguras.

Blockchain é uma tecnologia em que a prática parece estar à frente de teoria em vários aspectos [10]. Blockchain (e as criptomoedas como o Bitcoin) está na interseção entre segurança de software, sistemas distribuídos e sistemas dinâmicos [10] (como os sistemas econômicos).

O nível de descentralização obtido com Blockchain era considerado inatingível na teoria [9]. Porém a abordagem de que pequenas falhas são aceitas pelo sistema de consenso viabiliza a tecnologia na prática [9].

Há diversas oportunidades para a inovação tecnológica em aplicações desta tecnologia, dentre as quais estão as plataformas para desintermediação, o armazenamento de dados globalmente distribuído, e as transações de semânticas específicas [10].

A próxima geração de Blockchains apoiadas por transações sofisticadas e contratos inteligentes tem o potencial de viabilizar organizações autônomas no futuro [9]. Pesquisadores alegam que a lacuna entre teoria e prática faz com que a tecnologia não seja ainda totalmente entendida [9].

Além disso, a grande variedade de implementações disponíveis e de estudos empíricos realizados ou em andamento dificultam saber qual (plataforma de) Blockchain vai prevalecer [9].

Esta incerteza é refletida na imaturidade relativa das plataformas Blockchain. Plataformas mais maduras, como a Ethereum, tendem a ser mais estáveis, possuem documentação melhor e têm sido mais estudadas do ponto de vista da segurança.

Porém, por serem mais conhecidas e utilizadas, também estão mais expostas a ataques. Por outro lado, plataformas mais novas, como a Hyperledger, ainda apresentam uma instabilidade relativa, documentação dispersa e poucos exemplos de utilização com código fonte disponível.

Finalmente, de modo geral, o desenvolvedor de software interessando na tecnologia precisa investir bastante tempo no aprendizado das plataformas Blockchain e suas ferramentas para desenvolvimento de aplicações. Além disso, mesmo adotando métodos ágeis para o desenvolvimento rápido de aplicações, as questões de segurança precisam ser tratadas com atenção.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] S. Underwood, “Blockchain beyond bitcoin,” Communications of the ACM, vol. 59, no. 11, pp. 15–17, 2016.

[2] P. Franco, Understanding Bitcoin: Cryptography, engineering and economics. John Wiley & Sons, 2014.

[3] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, “A Fistful of Bitcoins: Characterizing Payments Among Men with No Names,” Commun. ACM, vol. 59, no. 4, pp. 86–93, 2016.

[4] A. Braga and R. Dahab, “Introdução à Criptografia para Programadores: Evitando Maus Usos da Criptografia em Sistemas de Software,” in Caderno de minicursos do XV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais — SBSeg 2015, 2015, pp. 1–50.

[5] M. Swan, Blockchain - Blueprint for a New Economy. 2015.

[6] “Hyperledger - Blockchain Technologies for Business.” [Online]. Available: https://www.hyperledger.org.

[7] “Ethereum Blockchain App Platform.” [Online]. Available: https://www.ethereum.org.

[8] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran, and S. Chen, “The blockchain as a software connector,” Proceedings - 2016 13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016, pp. 182–191, 2016.

[9] F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: A technical survey on decentralized digital currencies,” IEEE Communications Surveys and Tutorials, vol. 18, no. 3, pp. 2084–2123, 2016.

[10] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, “Research Perspectives and Challenges for Bitcoin and Cryptocurrencies,” IEEE Symposium on Security and Privacy, pp. 104–121, 2015.

[11] C. Barski and C. Wilmer, Bitcoin for the Befuddled. No Starch Press, 2014.

[12] C. Dannen, Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners. Apress, 2017.

[13] K. Krombholz, A. Judmayer, M. Gusenbauer, and E. Weippl, “The Other Side of the Coin: User Experiences with Bitcoin Security and Privacy,” Financial Cryptography and Data Security 2016, 2016.

[14] S. Eskandari, D. Barrera, E. Stobert, and J. Clark, “A First Look at the Usability of Bitcoin Key Management,” USEC 2015 San Diego, CA, USA, no. February, 2015.

[15] Z. Wan, D. Lo, X. Xia, and L. Cai, “Bug characteristics in blockchain systems: a large-scale empirical study,” in Proceedings of the 14th International Conference on Mining Software Repositories, 2017, pp. 413–424.

[16] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, “Elliptic curve cryptography in practice,” in Financial Cryptography and Data Security, Springer, 2014, pp. 157–175.

[17] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making Smart Contracts Smarter,” Ccs, pp. 254–269, 2016.

[18] N. Atzei, M. Bartoletti, and T. Cimoli, “A survey of attacks on Ethereum smart contracts (SoK),” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10204 LNCS, no. July, pp. 164–186, 2017.