UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

PEDRO VINÍCIUS NOVAIS RIBEIRO

TOLERÂNCIA A FALHAS BIZANTINAS APLICADAS AO SISTEMA *BITCOIN*

Vitória da Conquista

2018

PEDRO VINÍCIUS NOVAIS RIBEIRO

TOLERÂNCIA A FALHAS BIZANTINAS APLICADAS AO SISTEMA *BITCOIN*

Trabalho de conclusão de curso submetido ao Curso de Ciência da Computação da Universidade Estadual do Sudoeste da Bahia, como requisito necessário para obtenção do título de bacharel

Vitória da Conquista

2018

PEDRO VINÍCIUS NOVAIS RIBEIRO

TOLERÂNCIA A FALHAS BIZANTINAS APLICADAS AO SISTEMA *BITCOIN*

BANCA EXAMINADORA**:**

_____

Orientador: Prof. Me. Dr. Marco Antônio Dantas Ramos

_____

Profa. Me. Cléia Santos Libarino

Convidado 1

_____

Prof. Me. Dr. Fábio Moura Pereira

Convidado 2

Vitória da Conquista

2018

Dedicatória

Dedico esse trabalho acadêmico a todas as pessoas que assim como eu vieram de uma origem simples, de pais batalhadores que venceram na vida e mostraram que é possível sim através da educação mudar a realidade a qual estamos inseridos e também repassar a todos ao seu redor que podemos dar sempre o melhor em todas as ocasiões, vencendo na vida e construindo um novo cenário.

Dedicando também a toda minha família que me inspira todos os dias e me motiva a superar todas as barreiras da vida através da união e do amor.

Agradecimentos

Momentos especiais trazem também consigo apoio durante uma longa jornada de luta e vitórias, mas são nos momentos difíceis que podemos reconhecer todas as pessoas que estão ao nosso lado, nos dando força e suporte para que continuemos em busca de todos os nossos objetivos durante a vida, assim gostaria de agradecer imensamente a toda a minha família, que me trouxe ensinamentos de vida e uma ótima educação para que pudesse encerrar mais um ciclo em minha vida, especialmente a minha Mãe Eliane que sempre esteve ao meu lado em todos momentos me motivando e me cobrando para que pudesse sempre evoluir, ao meu pai Arney pelo apoio incondicional em minhas escolhas e por sempre me mostrar o caminho correto ao longo da vida, a minha tia Arlene por ter me tratado como um filho e me ajudado a construir um caráter digno de homem, a minha avó Marinalva por todos os ensinamentos de vida e simplicidade inigualável, ao meu tio Márcio pelas conversas e exemplo de homem a ser seguido e a todos os meus outros tios(as) e primos(as) por sempre me mostrarem que a família é o bem mais precioso desse mundo independentemente de todas as outras coisas. Um agradecimento especial a minha digníssima namorada e companheira Talize, por ter me apoiado nos momentos de estudo e de construção da vida acadêmica e por todo amor.

Agradeço também ao corpo docente da UESB, professores iluminados que fizeram parte de toda construção educacional e profissional que pude receber, não esquecendo do anjo iluminado chamado Celina que sempre esteve disposta e com um enorme coração para me ajudar durante todo o curso.

Resumo

O surgimento de uma nova economia mundial trouxe à tona novas formas de se organizar e de se relacionar comercialmente, o dinheiro já não é tão visto como meio mais seguro de se negociar devido ao aumento vertiginoso da insegurança pública, o “dinheiro de plástico” roubou a cena e com ele trouxe um novo olhar para as formas de pagamento, o cartão de crédito fez com que os usuários se sentissem mais seguros e fez com que a sociedade se modernizasse financeiramente. O surgimento de novas formas de negociação criou o dinheiro digital, uma revolução que promete causar muita discussão na economia mundial, o meio digital cria novas faces e comodidades às pessoas, com a moeda não poderia ser diferente, o *BITCOIN* surge de forma descentralizada, ou seja, independente de um órgão que regularize as transações, assim os próprios usuários trocam moedas entre si sem nenhuma taxa ou tributação. Mas isso não seria um grande perigo? Seria se o sistema de código aberto do *BITCOIN*, desenvolvido em comunidade e idealizado pelo Satoshi Nakamoto, não fosse tão seguro. O sistema consiste de uma forte criptografia e verificações pelos próprios usuários que recebem recompensas por essas atividades de verificação, se a transação puder ser realizada e se a mesma for legal, tornando todo o sistema mais vantajoso se o usuário tentar ajudar a minerar do que se o usuário tentar fraudar. O presente trabalho visa analisar a questão da segurança e os seus principais fatores que asseguram usuários anônimos realizarem transações financeiras em uma rede *Peer-to-Peer.*

Palavras-Chave: BITCOIN, P2P, nova moeda digital, *Blockchain*, carteira virtual, consenso, criptomoeda, falhas bizantinas, gasto duplo.

ABSTRACT

The emergence of a new world economy has brought to the fore new ways of organizing and of relating commercially, money is no longer seen as safer way to negotiate due the increase of public insecurity, “plastic money” has stolen the scene and with it brought a new look at the ways of payment, the credit card made users feel more secure and made the companies modernize themselves. The outbreak of new forms of negotiation has created digital money, a revolution that can cause much discussion in the world economy, digital data create new faces and conveniences for people, which could not be different currencies, *Bitcoin* emerges as a decentralized currency, in other words, free from entities that regulates the transactions, that means users could exchange coins with each other without any taxes or duty. But would not that be a great danger? It would be if the open-source system of bitcoin, developed in community and idealized by satoshi nakamoto, was not so secure. The system consists of strong encryption and verifications by the users, who receive rewards for these verification activities, if the transaction can be performed and if the transaction is legal, making the whole system more advantageous if the user tries to help mining than if the user tries to cheat. This paper aims to analyze the security issue and its main factors that assure anonymous users to carry out financial transactions in a peer-to-peer network.

Key-words: BITCOIN, P2P, new digital currency, *Blockchain*, bitcoin wallet, consensus, crypto, Byzantine fault, double spend.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A economia mundial vive períodos de incertezas e inseguranças, as altas taxas de juros, inflação de uma moeda fazem as pessoas analisarem com muito zelo ao investir e até mesmo poupar o dinheiro, muitos querem apenas segurança, outros a maior taxa de lucratividade, mas se sentir seguro em meio a tantas crises globais vivenciadas, fazem os indivíduos verdadeiros reféns dos órgãos centralizadores de moedas de cada país. Não existe um controle fiscal rigoroso de todas transações e de como o dinheiro ali recolhido de milhões de pessoa se comporta e para onde vai, tornar tudo isso transparente aos olhos dos usuários seria uma solução ideal, ao mesmo tempo que não ter um órgão ditando regras e taxas de maneira impositiva aos usuários seria perfeito, porém o alto poderio dos detentores da economia global nunca foram ameaçados por nenhum método que pudesse revolucionar o setor. A computação, no entanto, possui ferramentas que permitem um simples projeto revolucionar a vida de milhões de pessoas, trazendo inúmeros benefícios e agilizando os processos cotidianos das vidas pessoas.

Em meados de 2008 um usuário denominado Satoshi Nakamoto – até então desconhecido - lançou em uma comunidade de desenvolvimento a maior revolução do sistema financeiro mundial, era um artigo "*BITCOIN: A peer-to-peer electronic cash system*", o *software* proposto realizava transações de uma criptomoeda totalmente descentralizada e sem taxações tributárias para os usuários, e tudo isso de uma forma totalmente segura(FILHO, 2017). A arquitetura *Peer-to-Peer*(P2P) utilizada no sistema BITCOIN proporciona que todos usuários se conectem diretamente para troca de informações através da rede computadores e verifiquem a confiabilidade daquele novo bloco de dados – Transação que irá ser realizada - antes do mesmo ser inserido em uma espécie de Registro com todas as transações, se o mesmo for válido, os nós – denominação para cada computador conectado ao sistema P2P – irão entrar em um consenso e inserir de forma segura os novos dados e repassá-los aos outros usuários, que terão que verificar o novo bloco e posteriormente adicionar o mesmo ao seu registro local informando que aquela transação é válida. A rede consiste de usuários ativos denominados mineradores, que funcionam como atuadores no controle e fiscalização de todas as transações, sem necessidade de uma autoridade superior por trás de tudo isso, de maneira transparente e com os próprios mineradores ativos

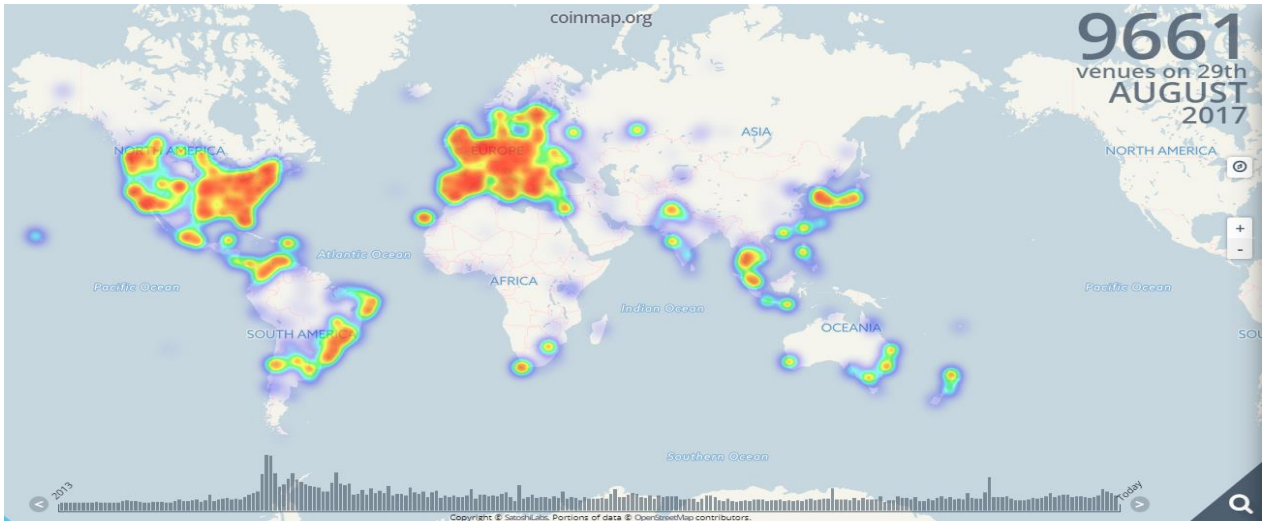
sendo recompensados por seu esforço computacional para manter o sistema seguro. Um dos pontos importantes é a segurança e integridade dos dados, então mesmo que alguns usuários da rede fossem desativados ou na pior das hipóteses um país inteiro sumisse do mapa na rede de computadores, o uso do sistema Peer-2-Peer aliado ao Registro de Transações que cada usuário possui já é suficiente, basta que apenas um nó esteja ativo para que todos os outros possam se recuperar e atingir o estado de consistência até a última transação segura.

1.1 Problemática

A facilidade de poder fazer uma simples transferência financeira totalmente segura em questão de segundos e sem qualquer método que te identifique socialmente fez que o *BITCOIN* tornasse a vida das pessoas mais ágeis, porém também trouxe à tona questões em relação ao uso da criptomoeda.

O que pode ser uma enorme vantagem para comerciantes e fornecedores que visam receber seus pagamentos com menores taxas de juros e de cartão de crédito, além de não ter que receber dinheiro “vivo” em seus estabelecimentos. O gráfico a seguir fornecido pelo site *CoinMap* revela através de um mapa de calor o número de estabelecimentos comerciais no globo terrestre que fazem uso ativo de *BITCOIN,* são restaurantes, pousadas, empresas de publicidade, supermercados e até mesmo locadoras de veículos.

Figura 1: Mapa de calor dos estabelecimentos que já aceitam BITCOIN



Fonte: CoinMap[1]

[1] https://coinmap.org/#/world/42.29356419/11.77734375/2

O número de usuários mal-intencionados que começaram a se utilizar das vantagens trazidas pela moeda para se aproveitar de todos os benefícios de uma maneira ilegal e receber pagamentos através da rede de computadores por atos ilegais e que causam prejuízos a terceiros, cresceu vertiginosamente. No ano de 2017 um grande número de usuários foi infectado pelo *"WannaCry"*, que consistia em um sistema de sequestro de dados que cobrava pela devolução desses dados ao próprio usuário. Segundo o portal Olhar digital[1], os responsáveis pelo ataque receberam em torno de 61,6 bitcoins em resgates, totalizando US$ 148 mil, ou R$ 461 mil (cotação da moeda na época)

O presente trabalho visa verificar a questão da segurança envolvendo esse novo sistema e como ele atua sobre a rede mundial de computações de forma distribuída e assíncrona, assim visamos relacionar a segurança como fator primordial envolvendo transações importantes que representem valores monetários e porque o sistema se tornou revolucionário ao garantir a integridade para ambas partes que visam realizar transações; Os prós e contras do uso no cenário atual de *BITCOINS*, visto que o ser humano é tecnológico e se desenvolve através da sua dinamicidade e capacidade de criação. Assim a análise feita por este trabalho irá diversificar a forma como essa tecnologia é vista, aceita e utilizada por seus usuários, abordando a questão da segurança e correlacionando com os ramos da Ciência da Computação responsáveis por estudos sobre segurança em redes e como ponto principal em relação a criptomoeda.

1.1.1 Questões de pesquisa

Considerando o presente exposto em razão da problemática, obtemos situações que trazem dúvidas em relação ao uso do *BITCOIN* como meio seguro para transação de informações, portanto seu estudo mais aprofundado nos conduzirá a entender como essa moeda pode realmente fazer parte da vida cotidiana das pessoas e deixar de ser vista como uma real alternativa para aqueles usuários que buscam receber pagamentos ilegais através do anonimato, abordando a segurança das transações e estudos computacionais através de um meio inseguro, que é a rede

[1] https://olhardigital.com.br/fique_seguro/noticia/autores-do-wannacry-sacam-suas-bitcoins-ataque-rendeu-mais-de-r-500-mil/70177

mundial de computadores, e realizando estudos objetivando observar os pontos críticos e de falhas que essa tecnologia pode trazer consigo, e os pontos positivos que a mesma trouxe para solucionar problemas antigos em relação a segurança em transações financeiras envolvendo mecanismos computacionais. O mecanismo utilizado pelo BITCOIN para validar suas transações é uma estrutura conhecida como *BLOCKCHAIN*. Tal estrutura é fundamentalmente um banco de dados distribuído, no qual os nós são representados como processos ativos atuantes sobre a rede e os eventos são as transações no sistema.

1.2 Objetivos

1.2.1 Objetivos Gerais

Realizar uma análise especifica dos problemas de segurança do sistema *BITCOIN que* por ser distribuído e possuir dinamicidade de nós facilita a ocorrência de problemas como o gasto-duplo ou tentativas de falsificação de blocos que serão inseridos no *BLOCKCHAIN* e todo o comportamento do sistema durante essas tentativas de falsificação e invalidação de transações. Averiguando assim a tolerância a falhas do Sistema *BITCOIN* e demonstrando seus possíveis problemas.

1.2.2 Objetivos Específicos

O objetivo especifico deste trabalho é estudar a força do modelo de segurança adotado pelo BITCOIN, relacionando com os modelos de sistemas distribuídos, bem como os modelos de falhas, mais especificamente as falhas bizantinas. Avaliando, assim, as propriedades de segurança, as quais são possíveis de se garantir com tais mecanismos e suas eventuais falhas.

2 Fundamentação Teórica

2.1 Sistemas Distribuídos Tolerantes a Falhas

Sistemas distribuídos surgiram a partir do final dos anos 1970, quando as tecnologias de redes locais de comunicações começaram a amadurecer e ganhar impulso no mercado. Os primeiros sistemas eram limitados em relação à conectividade e aos recursos e serviços de rede disponíveis. A evolução das tecnologias de redes de comunicação locais, móveis, sem fio etc, permitiu a popularização de todo tipo de sistemas distribuídos, a tal ponto que não é exagero dizer que a quase totalidade das aplicações de computação com que uma pessoa comum tem contato hoje é composta de sistemas distribuídos de alguma maneira. Esta seção apresenta os sistemas distribuídos, características e propriedades, alguns modelos de sistemas mais relevantes e trata dos sistemas distribuídos tolerantes a falhas, objeto deste trabalho.

2.1.1 Sistemas Distribuídos

Couloris, Dollimore e Kindberg (2011) definem sistemas distribuídos como aqueles sistemas cujos componentes estão situados em computadores conectados por redes de comunicações e se coordenam e comunicam apenas através de trocas de mensagens. Para Tanembaum e Van Steen (2006) a experiência do usuário é definidora do sistema, ou seja, sistemas distribuídos são sistemas compostos por componentes independentes conectados por rede e que se apresentam para o usuário como um sistema único e coerente.

Essas três definições enfatizam aspectos importantes da área de sistemas distribuídos: são sistemas baseados em comunicação em rede, que apresentam para o usuário uma interface coerente e unificada, que esconde as complexidades relativas à comunicação e coordenação descentralizada, e que, uma vez que é sujeito a falhas de seus componentes, quando estas falhas não são toleradas adequadamente toda a ilusão de unidade e coerência é perdida.

Algumas propriedades axiomáticas decorrem dessa natureza descentralizada (COULORIS,2011):

- Concorrência - Uma vez que os componentes do sistema são distribuídos arbitrariamente e conectados apenas por alguma rede de comunicações, as tarefas executadas nessas componentes são naturalmente concorrentes. O compartilhamento de recursos e a coordenação dessas tarefas concorrentes são alguns dos grandes desafios decorrentes dessa concorrência.
- Ausência de relógio global - Ainda que existam protocolos de propagação do tempo a partir de servidores de tempo bastante precisos, a incerteza no tempo de entrega de mensagens entre componentes do sistema impõem um limite mínimo na acurária com que os relógios num sistema distribuído podem ser sincronizados (LUNDELIUS,1984). Isso torna o problema de sincronizar relógios dentro de uma acurácia determinada um grande desafio. Assim, determinar a ordem temporal de eventos só pode ser obtido de forma limitada.
- Independência de falhas - Um sistema distribuído está sujeito a falhas, e dada a natureza concorrente das tarefas no sistema, considera-se geralmente que as falhas que ocorrem naturalmente no sistema (sem a intervenção de algum personagem externo malicioso) são independentes umas das outras. O crescimento recente no número de ataques maliciosos contra sistemas de infraestrutura crítica, sistemas comerciais e financeiros, e mesmo ataques simultâneos a um grande número de usuários independentes tem feito com que pesquisadores considerem o caso da ocorrência de falhas coordenadas por agentes maliciosos.

Ainda que possamos enumerar uma vasta gama de exemplos de sistemas distribuídos, tais como o mensageiros instantâneos, as redes sociais, sistemas de *home banking* e o próprio *BLOCKCHAIN* utilizado pelo B*ITCOIN*, todos esses exemplos compartilham propriedades fundamentais como as enumeradas acima e são distribuídos geograficamente e conectados em rede, processam informação por trocas de mensagens e estão sujeitos a tipos equivalentes de falhas. O estudo dessas propriedades fundamentais e universais é realizado através da representação de sistemas distribuídos por modelos de sistemas. Nesses modelos, ditos modelos fundamentais, as componentes do sistema são chamadas genericamente de

“processos", sem maiores considerações sobre detalhes específicos. Tais modelos usualmente representam como os processos trocam mensagens, a natureza da comunicação entre eles, a existência ou não de limites temporais estritos e conhecidos para a troca de mensagens e o modo como se admite que os processos possam falhar. Coletivamente, esses modelos descrevem um sistema distribuído em detalhe suficiente para que se possa estudar propriedades fundamentais, descrever formalmente algoritmos distribuídos e determinar formalmente hipóteses acerca do comportamento desses algoritmos quando executados no sistema (COULORIS,2011).

Algumas das propriedades fundamentais são determinantes para o comportamento dos sistemas. A existência de limites temporais de processamento e entrega de mensagens permitem classificar os sistemas em síncronos, nos quais todos os limites são conhecidos, e assíncronos, em que não existem limites temporais (DWORK,1988). Em sistemas síncronos existem 2 limites temporais fixos e determinantes: o limite máximo para a diferença no tempo de computação entre os processos do sistema, chamado ϕ, e o limite máximo para o atraso na entrega de mensagens entre 2 processos quaisquer, chamado Δ. Num sistema síncrono, um envio de mensagem do processo pi para o processo pj e a consequente resposta leva um tempo teoricamente máximo $t = 2\Delta + \phi$. Esses tempos são garantidos por uma infraestrutura de redes e sistemas operacionais adequados. Em sistemas assíncronos não há tais limites ou garantias. Existem vários modelos de sistemas com características de sincronia intermediárias entre essas duas.

O modo como os processos do sistema podem falhar, se por falhas de parada, as mais simples, ou falhas arbitrárias, as chamadas falhas bizantinas é determinante para sabermos que tipo de resiliência, a capacidade de continuar operando corretamente na presença de falhas, podemos esperar do sistema. Lamport, Pease e Shostack (1982), Dwork, Lynch e Stockmeyer(1988), entre outros pesquisadores, determinaram limites máximos para a resiliência de processos em alguns modelos de sistemas e tipos de falhas.

Um importante resultado da computação distribuída, chamado **Impossibilidade FLP**, provado por Fisher, Lynch e Patterson (1985), diz que é impossível resolver o problema do consenso, um problema fundamental de

concordância de valores que será descrito na seção 4.2, na ocorrência de apenas uma falha de parada. Esse resultado correlaciona de forma determinante a sincronia com a resiliência do sistema, e foi motivador para a elaboração de diversos modelos de sistemas intermediários entre síncronos e assíncronos.

2.1.2 Sistemas Tolerantes a Falhas

Sistemas falham em algum momento. A área de sistemas tolerantes a falhas estuda técnicas, metodologias e estratégias que visam aumentar a confiabilidade de sistemas de informação e controle aplicados em todas as atividades humanas. O trabalho referencial de Avizienis *et al*, (2004), unifica a terminologia e conceituação para o campo, define conceitos fundamentais como erro, falha e defeito, até noções como confiança, dependência, confiabilidade e segurança em sistemas computacionais e estabelece uma taxonomia das abordagens para a tolerância a falhas, de modo abrangente e sistemático. Vamos rever estes conceitos fundamentais para construirmos a definição de sistema distribuído tolerante a falhas.

Um erro é o desvio do comportamento de um processo de sua especificação original. Um processo em estado de erro, ou um processo que apresenta erro é um processo cujo comportamento é anômalo. Essa anomalia de comportamento pode se revelar de um lado na parada total do processo, ele não apresenta comportamento algum; até o comportamento imprevisível, aleatório ou arbitrário, no qual o processo não para, mas não se comporta como especificado, assim causando grande perturbação ao sistema. Um processo em estado de erro é chamado também de processo faltoso.

Falha é a causa original do erro. Avizienis *et al* definem uma taxonomia para falhas em sistemas. Falhas podem ser internas ou externas. Quando a falha interna permite que uma falha externa prejudique o funcionamento do sistema, ela é dita uma vulnerabilidade. Ataques maliciosos exploram vulnerabilidades do sistema para obter controle sobre algum processo do mesmo, por exemplo. O erro é o comportamento provocado pela ocorrência ou ativação da falha.

Defeito ou falta ocorre quando o serviço oferecido pelo sistema diverge do serviço correto esperado. A falha é a causa última do erro, o erro se propaga

internamente pelos processos do sistema provocando desvios internos de funcionamento em cascata, e quando finalmente o serviço oferecido pelo sistema fica comprometido, temos um defeito. Defeitos são um desvio de comportamento externo. Esse desvio pode ser visível e mensurável ou não. Desvios não visíveis são, por exemplo, a entrega de algum conteúdo sobre o qual não existe um controle pré-definido. Um desvio visível está, por exemplo, na corrupção de algum protocolo conhecido pelos outros componentes do sistema. A mensuração desse desvio para fins de controle dos processos do sistema é chamada **detecção de defeitos.**

Tolerância a Falhas é o emprego de um conjunto de técnicas na construção do sistema com o objetivo de evitar a ocorrência de defeitos. Um sistema tolerante a falhas cumpre sua especificação mesmo na ocorrência de falhas. Segundo Avizienis et al o projetista dispõe de dois mecanismos básicos para isso: detecção de erros e recuperação. Detecção de erros consiste em alguma forma de codificação do comportamento do sistema que pode ser verificada. Anomalias na codificação representam a ocorrência de erros e as componentes comprometidas, usualmente as mensagens do sistema, são descartadas ou corrigidas.

A recuperação de erros ocorre quando o erro se manifesta internamente, mas sua propagação é contida por um de dois mecanismos básicos: isolamento ou compensação do erro. O isolamento do erro é associado com a detecção. Detecta-se o componente que está em erro (faltoso) e se discrimina esse componente, evitando a sua propagação no sistema. A compensação oferece um conjunto redundante de componentes corretos que compensam o comportamento anômalo do componente faltoso oferecendo alguma forma de resultado majoritário correto ao sistema.

Do ponto de vista deste trabalho, interessam as falhas chamadas falhas bizantinas. Falhas bizantinas são falhas nas quais a execução dos processos faltosos de um sistema desvia arbitrariamente de sua especificação original, ou seja, elas podem causar qualquer tipo de defeito no sistema, de maneira errática e imprevisível. Elas foram caracterizadas em (LAMPORT,1982). O artigo mostra que num sistema distribuído síncrono, na presença de processos bizantinos, o número de processos que falham *f* deve ser menor que um terço do número total de processos *n*. Dwork, Lynch e Stokemeyer (1988) mostraram que esta proporção vale para sistemas parcialmente síncronos. Isso torna a resiliência a processos bizantinos um processo de alto custo operacional.

Além da questão do custo, processos bizantinos podem não falhar de modo consistente. Kihlstrom et al (2003) classificam as falhas bizantinas em:

- Observáveis - As falhas observáveis podem ser por omissão, corrupção de mensagens ou mensagens mutantes.
    - Omissão - O processo faltoso omite mensagens que deveria enviar. Essa omissão pode ser generalizada ou discriminatória. O processo que omite mensagens de forma generalizada age como se tivesse sofrido uma parada.
    - Corrupção de mensagens - Neste caso o processo faltoso envia mensagens mal formadas ou injustificadas pelo protocolo original do sistema. Essa má formação pode estar associada a uma falha de identificação, quando o processo apresentou credenciais inválidas na mensagem, falha de conteúdo, quando o valor ou a estrutura da mensagem são ininteligíveis, ou ainda falha de justificativa, quando o processo faltoso não fornece uma base causal para o envio da mensagem errônea.
    - Mensagens mutantes - Ocorrem quando o processo faltoso envia mensagens válidas mas diferentes para diferentes processos.
- Não observáveis - As falhas bizantinas não observáveis não manifestam efeitos externos e podem apenas ser mascaradas.

2.1.3 O Problema do Consenso

A situação na qual os processos corretos que compõem um sistema distribuído precisam entrar em acordo a respeito de algum valor consistente com a especificação do sistema é recorrente em computação distribuída e está associada a uma ampla série de algoritmos fundamentais da área, tais como commit em bancos de dados distribuídos, replicação de dados, sincronização de relógios, ou a troca ordenada de mensagens. Paradigmas importantes da tolerância a falhas, como a eleição de líderes (GUERRAOUI, 1999), replicação de máquinas de estados (LAMPORT, 1978), sistemas de quórum (MALKHI, 1998) ou *broadcast* confiável com terminação (com sigla em inglês TRB) (CHANDRA,1996), dependem intrinsecamente de um acordo entre processos para serem realizáveis. Vários problemas representativos desse tipo de acordo foram estudados ao longo do tempo, para a determinação dos limites de

computabilidade desses problemas e a possibilidade da construção de protocolos que os resolvessem: o problema da consistência interativa estudado (PEASE,1980), o problema da concordância bizantina abordado por (LAMPORT, 1982) e o problema do consenso averiguado por (FISHER, 1983) são exemplos.

As soluções aqui apresentadas para todos esses problemas estão baseadas na construção de algum tipo de quórum adequado para tolerar processos faltosos em cada caso. Cada algoritmo varia na forma como o quórum é formado ou verificado, e no tipo de falha que o algoritmo pretende tolerar, o que determina o número de processos no quórum a formar. Resultados e métodos fundamentais em sistemas distribuídos foram fruto do estudo desses problemas e suas variantes.

Pease et al apresentam em (PEASE,1980) o Problema da Consistência Interativa, no qual os processos corretos de um sistema distribuído devem determinar um vetor de valores onde o i-ésimo elemento do vetor corresponde ao valor inicial do i-ésimo processo. No artigo, os autores introduzem noções importantes para a computação distribuída, como o de falha de valor arbitrário, que veio a integrar o que posteriormente se chamou de falha bizantina, ou a noção de quórum bizantino, quando determinam um número mínimo de processos necessário para mascarar os processos faltosos.

Como condições adicionais, o meio de comunicação é confiável e tem atraso desprezível. Além disso o emitente da mensagem pode sempre ser identificado pelo recipiente. No caso, os elementos do vetor correspondentes a processos faltosos podem ser arbitrários, desde que os mesmos valores sejam computados para eles. Os autores demonstram que algoritmos para a consistência interativa podem ser criados, desde que a proporção entre *n* e *f* seja n > 3f + 1, ou seja, o número de processos faltosos deve necessariamente ser inferior a um terço do total de processos. Os 2f + 1 processos restantes formam um quórum capaz de tolerar a falha de valor arbitrário de f processos e assim atingir consistência interativa. Lamport vai demonstrar o mesmo resultado para o problema da concordância bizantina.

O Problema da Concordância Bizantina foi inicialmente estudado por Lamport et al em (LAMPORT, 1982) e (LAMPORT,1983), com o nome de Problema dos Generais Bizantinos, e consiste em permitir que os processos corretos do sistema concordem num valor proposto por um deles ou num valor *default* comum na presença

de falhas de valor. De acordo com Lamport et al, um general comandante no exército bizantino teria de emitir uma ordem de ataque ou retirada para seus tenentes generais, mas todos são potenciais traidores, inclusive o general de emite a ordem. Lamport et al estuda o cenário mínimo para o problema, ou seja, o menor número de processos necessário para tolerar a presença de um general traidor, e enuncia o problema da seguinte forma (LAMPORT,1983):

Lamport demonstra em (LAMPORT, 1982) que o problema da concordância bizantina também não tem solução quando o número de processos faltosos for igual ou superior a um terço do total de processos do sistema. A partir da introdução da formulação com os generais bizantinos, os processos faltosos com falhas arbitrárias de valor passaram a ser chamados de **processos bizantinos**. O quórum formado a partir de n > 3f + 1 passou a ser chamado de **quórum bizantino**.

> A principal diferença entre o artigo de Lamport, Pease e Shostak (LAMPORT, 1982) e o artigo de Pease, Shostak e Lamport (PEASE, 1980), que é anterior àquele, reside no fato de o artigo de Pease se concentrar numa falha de valor arbitrário, ou seja, uma falha não observável, apesar de o autor mencionar a possibilidade para falhas detectáveis. Já (LAMPORT, 1982) generaliza o tipo de falha para abranger desde falhas de valor, como a de (PEASE, 1980), até o comportamento malicioso. Ainda assim, o resultado fundamental do artigo de Lamport, que é a demonstração do quorum bizantino para este modelo generalizado de falha, se mantém nas condições de (PEASE, 1980), mas pode ser melhorado consideravelmente com a introdução de alguma forma de identificação de processos que não seja forjável.

Problema do Consenso os processos podem propor valores dentre aqueles valores que julgam proponíveis, após o que um líder correto seleciona um valor proposto para consenso e os processos corretos devem concordar com este valor. Os processos não devem propor valores se estes não forem valores conhecidos por eles previamente. Os processos devem tomar uma decisão quando um valor adequado for proposto para eles. Assim posta, esta situação é uma instância do chamado problema do consenso, estudado em (PEASE,1980), (FISHER,1983), (DWORK,1988), (CHANDRA,1996), (MOSTEFAOUI,1999), (LAMPORT,1998), (LAMPORT,2001), (CASTRO,2002), (KIHLSTROM,2003), (FRIEDMAN,2005), (MACEDO,2008), (GORENDER,2007), (GORENDER,2011), (PASQUALETTI,2012), entre muitos outros. Este trabalho parte do enunciado de (CHANDRA,1996), a saber:

- Dado um sistema distribuído com N processos, os processos corretos propõem um valor e devem chegar a uma decisão unânime e irrevogável sobre algum valor que esteja relacionado entre os valores propostos, de acordo com as seguintes propriedades:
- Concordância - Nenhum par de processos corretos decide diferentemente;
- Validade Uniforme - Se um processo decide por um valor v, então v foi proposto por algum processo;
- Terminação - Todo processo correto deve em algum momento decidir sobre um valor;
- Integridade Uniforme - Todo processo decide no máximo uma vez.

A concordância diz respeito a que apenas um valor deve ser decidido consensualmente pelos processos corretos. A validade uniforme diz respeito a que os processos devem decidir coerentemente com os valores inicialmente disponíveis entre eles. Essas são propriedades relacionadas a *safety*: a decisão consensual deve ser conforme com a especificação do sistema. A propriedade de terminação indica que o sistema não deve ficar esperando indefinidamente por algum processo: se o valor proposto foi recebido, o processo deve então decidir. A integridade uniforme assegura que um processo não desista da decisão tomada e fique alternando de estado entre decidido e não decidido indefinidamente. Uma vez que tenha decidido, aquele estado é terminal. Assim sendo, terminação e integridade uniforme são propriedades de *liveness* (LYNCH,1991). A terminação pode não ser atingida enquanto a decisão consensual não acontecer, e a integridade uniforme garante que este processo seja monotônico, ou seja, uma vez que a decisão seja iniciada, todo processo correto que decidir permanece naquele estado. Alguns autores inclusive restringem a discussão de seus algoritmos de consenso à discussão sobre *safety*, deixando de lado a situação de como o algoritmo deve terminar enquanto não houver consenso. É o caso de Lamport com o algoritmo Paxos (LAMPORT,2001) e Paxos Bizantino (LAMPORT,2010).

2.2 Surgimento da moeda no mundo

A moeda surge na história da humanidade desde os seus primórdios como uma busca para organização das relações de negociação, a produção de mercadorias e a sua troca levou a humanidade a construir um símbolo que poderia representar um valor físico, este símbolo precisava naquele determinado momento ser representado, então pedras preciosas e moedas foram as bases rudimentares para a criação das transações monetárias.

Com a modernização das civilizações e a expansão territorial foram criadas então as moedas cunhadas em metais por grandes governantes como forma de poder, pois tinham um valor comercial e representativo dentro de um determinado território, além da facilidade trazida aos comerciantes ao terem um meio imutável para negociação dos seus produtos e organização da sociedade. Como segundo plano (WEATHERFORD, 1997) cita que a moeda facilitou a prática dos governantes na cobrança de tributos, o que antes realizado por meio de comódites com os camponeses que era pago com moedas que possuíam um valor especifico. Despertando também a consciência no indivíduo e nos governos que serviços poderiam ser pagos de forma mais simples, mais rápida e mais fácil, evitando ter que pagar por algo com uma saca de arroz, ou algo que necessitasse de um transporte maior. Por consequência, os benefícios da moeda fizeram com que a sociedade evoluísse e o homem se tornasse mais hábil em suas relações.

Ainda em sua Obra (WEATHERFORD, 1997) cita uma passagem de Gertrude Stein que diz: “o que diferencia o homem dos animais é dinheiro”, se tornando um marco divisor da história da humanidade nos períodos da antiga Grécia e Roma. A moeda é vista por (WEATHERFORD, 1997) em seu livro a história do dinheiro como um idioma compreendido por todas as nações, alertando também para os perigos que a importância do dinheiro ganhou no mundo ocidental e como forma de poder. Desde então a forma representativa da moeda ganha cada vez mais domínio, a sociedade se sofisticou dando um salto gigantesco em sua organização financeira, criando o dinheiro de papel impresso por cada país com variadas cotações e tributações, sendo a atual moeda controlada por entidades centrais que definem seus valores de mercado, altas e quedas.

Além do dinheiro impresso, a tecnologia obrigou os sistemas financeiros a se modernizarem, a internet realizou uma revolução no mundo e trouxe consigo as facilidades do mundo moderno e extremamente tecnológico, mostrando que homem não necessita de um contato físico para se relacionar com o outro, e como parte do processo de evolução o mesmo também percebeu que poderia realizar negociações a distância, realizando assim um processo de virtualização da moeda.

A moeda virtualizada pelos sistemas financeiros atuais é apenas uma representação de uma quantidade específica de moedas que um determinado indivíduo possui sob poder daquele determinado órgão centralizador. A moeda não deixou de ser o principal meio de troca, apenas assumiu novos papeis no cenário global, mas ainda sim necessitávamos de algo novo, totalmente independente e gerido por um grupo de usuários que realmente mantinham ativo aquele sistema.

O Sistema denominado *BITCOIN* é algo totalmente novo tanto para o mercado financeiro como para a ciência da computação, pois pela primeira vez um software poderia gerir toda uma rede interligada de usuários que trabalhavam em prol de uma única moeda, global e sem raízes atreladas a um determinado órgão monetário.

3 BITCOIN

Nesta seção iremos tratar sobre a criptomoeda *BITCOIN* e suas principais funcionalidades inerentes a ciência da computação, desde o seu surgimento em uma comunidade de tecnologia até o funcionamento intrínseco dos seus mecanismos internos de segurança, abordando como essa tecnologia abrange diversas áreas de estudo da área de segurança da Tecnologia da informação.

O sistema *BITCOIN* criou uma criptomoeda que surgiu a partir de uma divulgação em uma comunidade cientifica de um usuário chamado Satoshi Nakamoto, segundo o (MARCO CANUT, 2016) o BITCOIN pode ser definido como um programa de computador que cria uma rede global de notarização de transações que podem transferir valores entre partes distintas de forma descentralizada e de livre ingresso com auditória automática de todas as transações. O que torna todo o sistema um novo tipo de tecnologia no ramo da ciência da computação, a transação totalmente segura de uma criptomoeda através da gestão dos usuários. Portanto, imaginemos a revolução criada com o surgimento do *BITCOIN*, todos os sistemas financeiros atuais

possuem um órgão gestor e centralizado, pois lidar com o dinheiro de milhões de pessoas é algo que deve sim ser muito bem fiscalizado e os governos visam sempre estar no controle da moeda vigente em seu determinado território, contudo a tecnologia quebra barreiras até então inimagináveis até mesmo para os atuais sistemas. Surgiu um sistema financeiro que não necessita de um controle estatal ou centralizado, uma nova moeda totalmente acessível e global onde membros podem transacionar operações com taxas ínfimas e sem o controle rígido de um sistema ultrapassado.

O *BITCOIN* se mostrou um sistema totalmente independente que torna possível usuários distintos se relacionarem através de uma moeda totalmente virtual, moeda a qual é também regulada pela compra, venda dos próprios usuários, e ao contrários dos sistemas atuais podem realizar uma auditoria a qualquer momento do que está acontecendo não só com as suas moedas, mas com todas as moedas de todos usuários. Esse tipo de sistema totalmente independente, como o criado pelo *BITCOIN* assusta os grandes órgãos que detém todo o poder financeiro econômico mundial, criando divergências em relação a sua comercialização e adesão de novos usuários, o que gera desconforto por parte dos órgãos centralizadores, pois pela primeira vez na história, o dinheiro da população pode não mais estar sob o seu poder.

Vale ainda salientar que o sistema é alimentado por melhorias e uma comunidade de usuários que buscam o tempo inteiro a correção imediata das falhas e resolução dos problemas que venham a ocorrer no sistema. O que torna seu código aberto e acessível a todos que desejem conhecer o sistema nos seus mínimos detalhes, o que consiste em uma evolução da transparência de um sistema financeiro que é aberto a toda população mundial.

3.1 Vantagens Do Bitcoin

Mas ainda assim, qual seria a vantagem de se utilizar BITCOINS ao invés de utilizar o dinheiro propriamente dito, como dólares ou reais? Em (Ulrich,2014) são mencionados aspectos vantajosos no uso do *BITCOIN,* como os menores custos de transação, nos sistemas atuais altas taxas são empregadas pelos órgãos centralizadores de moeda, assim pagamentos com cartões, transações diárias possuem um custo alto, diferentemente do *BITCOIN* que por ser descentralizado

oferece taxas ínfimas se comparadas aos modelos atuais – apenas taxas pagas ao minerador por desprender um poder computacional ao verificar os dados quer serão transacionados – permitindo uma maior liberdade até mesmo aos que não possuem limites altos de crédito ou acesso a um banco propriamente dito.

Segundo (Ulrich,2014) ainda existe um estimulo a inovação tecnológica e financeira, dando margem a programadores para desenvolverem em cima do protocolo *BITCOIN,* sistemas que podem ir de apostas a grandes ações de empresas diariamente. É uma porta aberta para novas oportunidades de mercado e inovação em todos os aspectos. Uma revolução digital no mercado financeiro, que acaba causando certo abalo a estruturas de poder que por décadas vem gerindo e regulando o poder de compra e venda de toda a população mundial.

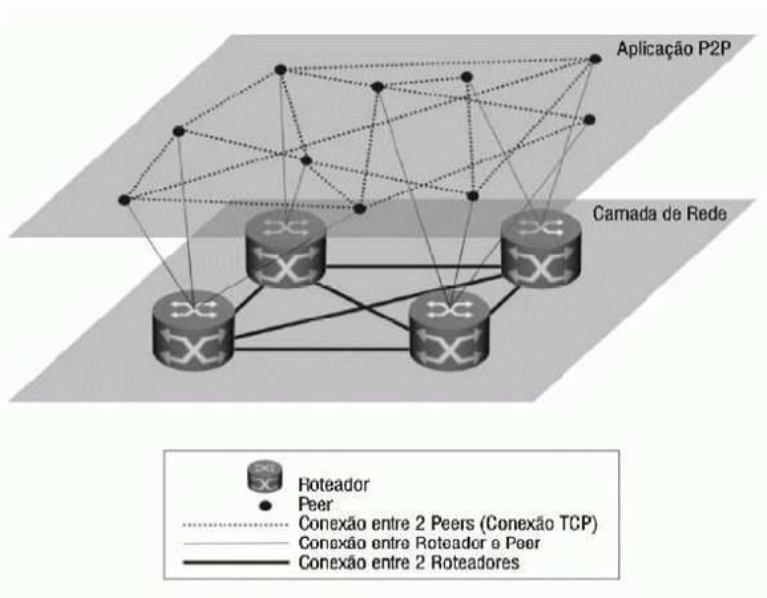
Além disso, o sistema *BITCOIN* traz uma inovação na questão da segurança financeira e digital, que sempre foi um elo fraco e que causava receio por parte dos usuários de qualquer sistema financeiro que atuasse por meios digitais, pois suponhamos que um malfeitor consiga ingressar nos sistemas financeiros atuais, os danos causados pelo mesmo seriam gigantescos visto que os sistemas ainda são antigos, muitos da década de 70, 80 e portanto apresentam um modelo de segurança antiquado, totalmente ultrapassado e que ainda não é visto com bons olhos pelos cidadãos, portanto sempre houve um pedido implícito ao ramo da computação por uma tecnologia que pudesse realmente suprir as necessidades de segurança e ao mesmo tempo ser totalmente aberto e claro a todos que quisessem auditar e verificar seus dados. No Sistema do *BITCOIN* os nós são partes atuantes através da rede Peer-2-Peer, realizando auditoria das informações, armazenando e repassando apenas informações válidas, essas informações serão verificadas por nós mineradores, os quais são responsáveis por desprender um poder de cálculo computacional afim de verificar se todos os dados daquela determinadas transação são corretos e podem ser inseridos no *BLOCKCHAIN* (Cadeia de Blocos) de maneira segura.

3.2 Redes P2P

A estrutura topológica de um sistema define suas formas de atuação sobre uma estrutura de comunicação, portanto, iremos definir nesta seção como é estruturada a forma de comunicação entre usuários do sistema *BITCOIN* na rede mundial de computadores e como os usuários ativos interagem sob essa rede estruturada fazendo com que o sistema possa funcionar de uma maneira distribuída e assíncrona, que é um dos objetos de estudo do presente trabalho.

Para entendermos melhor o funcionamento do *BITCOIN* precisamos ter em mente o funcionamento da arquitetura utilizada pela rede *BITCOIN*, portanto imagine que um computador conectado a rede mundial de computadores é denominado nó (peer), quando esses determinados nós(peers) trocam informações através da rede de computadores, a internet, os mesmos vão optar por uma topologia/arquitetura que é forma como esses usuários se organizam dentro de uma rede. Na rede *Peer-to-peer* – ou P2P, como iremos chamar essa rede daqui pra frente e os *peers* de nós – todos os nós trocam informações com todos os nós participando de um *Broadcast* (Difusão) de informações através da rede, ou seja, um derramamento de informações para que todos nós atuantes na rede possam receber um determinado dado, assim todos os nós realizam o papel de cliente (receptor dos dados) e servidor (emissor de dados) de forma descentralizada, o que facilita que usuários ao redor de todo o mundo se conectem sem uma central especifica. As redes P2P ganharam grande notoriedade devido aos compartilhadores de arquivos que quebravam o arquivo em partes pequenas e distribuíam para todos os nós. Aqueles usuários que já tivessem baixado determinada parte do arquivo poderiam também ajudar os outros nós enviando partes do arquivo a fim de ajuda-los a concluir seu processo de download.

Figura 2: Representação de uma aplicação P2P sob uma camada de Rede



Fonte:Coutinho(2006)

A grande vantagem da utilização das redes P2P é não ter uma base central que irá reter todas as informações, pois suponhamos que alguma catástrofe ou atentado ocorra justamente no servidor detentor de todos os dados, toda a rede seria prejudicada pela perda dos dados, pois os mesmos estavam concentrados em um único lugar e assim os outros clientes conectados a esse determinado servidor sofreriam uma desconexão, inutilizando totalmente o sistema ou tornando-o inacessível por um determinado momento. Portanto, mesmo que um nó na rede P2P perca informações ou venha a sofrer algum tipo de dano tanto de software como de hardware, posteriormente ele poderá recuperar os dados perdidos através da propagação pelos nós que estão em estados sólidos na rede.

No *BITCOIN* a rede P2P tem fundamental importância, pois todos os nós fazem parte do controle de transações do *BLOCKCHAIN.* Na literatura, o autor (DE LUCENA, HENRIQUES, 2016) o caracteriza como uma espécie de livro caixa onde todas as transações ficam armazenadas, assim que os nós recebem um determinado dado, realizam a verificação dos mesmos buscando a corretude das informações e espalham a resposta encontrada após a verificação para todos os nós ativos naquele determinado momento na rede, afirmando se os dados estão realmente corretos e precisos ou se foram modificados ou danificados durante o caminho propositalmente ou acidentalmente. O que reforça ainda mais o fator segurança tornando a sua

adulteração praticamente impossível (DE LUCENA, HENRIQUES, 2016), a integridade dos dados também é garantida devido ao fato de todos os nós possuírem uma cópia dos dados, portanto mesmo que um desastre ocorra e inutilize alguns nós eles poderão posteriormente se recuperar e baixar os dados do *BLOCKCHAIN*, pois todos os outros nós conectados a rede têm uma cópia dos dados salvos e irão atualizar os arquivos do nó que foi prejudicado até que o mesmo atinja o estado de consistência e possa participar ativamente da rede novamente.

Portanto é preponderante entender esse conceito de comunicação, pois será importante para o entendimento de todo o sistema nos próximos capítulos que serão abordados. Devido a comunicação entre os nós ser realizada através de Broadcast dos dados, os mesmos se tornam visíveis a todos os participantes ativos da rede, portanto um mecanismo de segurança que busque inibir o acesso de membros maliciosos deve ser fundamental no sistema *BITCOIN*, assim as técnicas que visem inibir a adulteração e retransmissão dos dados serão abordadas na seção 3.3.

3.3 Função Hash

Como visto na seção anterior, os dados irão circular através da rede devido ao tipo de arquitetura, portanto o adequado é que esses dados sejam encriptados/codificados de maneira segura, garantindo que os mesmos continuem íntegros e não sejam adulterados ou visualizados sem permissão ao longo do percurso.

A rede mundial de computadores não é o meio de comunicação mais seguro para trafego de dados, devido ao fato dos dados trafegarem por vários servidores até chegar ao destino final, principalmente dados que envolvem moedas e capitais financeiros o que exige um cuidado com a segurança mais rebuscado, porém existem técnicas que melhoram a segurança da informação a ponto de tornar uma mensagem que circula pela rede muito integra, concisa e segura fazendo com que qualquer modificação que foi realizada na mensagem seja identificada por outros usuários e os mesmos relatem aquela mensagem como uma mensagem inválida.

Métodos criptográficos através da função *HASH* garantem a integridade e evitam que os dados sejam modificados ao longo do meio de comunicação, abordaremos a função *HASH* utilizada no sistema *BITCOIN* nesta presente seção e sua contribuição na parte de segurança para o sistema.

No sistema *BITCOIN,* os nós irão enviar algumas informações pertinentes ao bloco como por exemplo: Chaves de assinatura digital – que devem ser guardadas com o máximo sigilo – valor a ser transferido, destinatário e outros campos que serão abordados na seção 3.5.1. Assim todas essas informações deverão passar por uma função *HASH*, a qual consiste em uma função matemática que manipula uma mensagem gerando uma nova sequência ou resumo e que possuirá o mesmo tamanho da mensagem, que é o *HASH* através de algoritmos do tipo MD5, SHA-1 e SHA-256 (RODRIGUES, 2017) que irá atuar sobre a mensagem com um digito verificador, mas para que os dados da mensagem sejam enviados da maneira correta, alia-se ao *HASH* as assinaturas digitais – que serão vistas na seção 3.6 – permitindo que toda a informação seja validada, além de realmente reconhecer que o emissor daquela mensagem está correto, através da sua chave privada.

Cada função *HASH* possui seu algoritmo, no caso do *BITCOIN* é utilizado um algoritmo duplo que cria uma mensagem randômica de 256 bits, denominada SHA-256. A importância dessa técnica, consiste na invalidação de blocos elaborados por usuários maliciosos, através de um mecanismo similar ao digito verificador, que ao passar a mensagem pela função *HASH* irá retornar um valor(digito verificador) através de um campo denominado "*Nonce*" do bloco, que garantirá que a mensagem seja integra pois será gerado um número para verificação e qualquer modificação em algum dado do bloco que foi gerado irá gerar um novo digito verificador o que leva a uma fácil verificação em qualquer tipo de adulteração.

Todavia para que essas mensagens sejam verificadas e o cálculo do seu *HASH* seja validado são necessários processos ativos, denominados mineradores, que serão vistos na próxima seção, os mesmos são responsáveis pela fiscalização dos blocos que circulam pela rede e recebem recompensas a cada calculo que exige poder de processamento desprendido.

3.4 Mineradores

Nesta seção trataremos dos usuários ativos do sistema, os quais realizam uma troca do seu poder computacional em busca de recompensas dentro do sistema através de moedas. Os Mineradores desempenham papel fundamental ao criar uma

força que protege o sistema de invasores e garante que todos os blocos sejam verificados antes de serem adicionados ao *BLOCKCHAIN.*

A criptomoeda necessita surgir de algum lugar e o método de inserção de moeda é a mineração, que recompensam usuários (nós mineradores) após os mesmos realizarem algum tipo de esforço computacional através da força bruta em prol da rede e na busca de novos blocos, são como taxas de recompensa pelo trabalho realizado. O processo funciona como uma alusão a busca do ouro – que seriam as criptomoedas – propriamente dita, assim o minerador irá desprender parte do seu tempo na busca pelo ouro e aquele que achar uma nova pedrinha de ouro(realizar os cálculos e confirmar que o bloco foi verificado) será recompensado com *bitcoin*, esse tipo de recompensa faz com que os usuários busquem ajudar todo o sistema através da mineração.

O processo de geração de novas moedas um dia irá atingir o seu limite máximo, que foi estabelecido em 21 milhões de BITCOINS pelo próprio algoritmo, assim o controle da inflação é estabelecido evitando que os preços da criptomoeda se elevem indiscriminadamente, evitando a sua desvalorização. Isso significa que um dia o processo de mineração não será mais recompensado com novos BITCOINS criados pela rede, como é feito atualmente, passando então os mineradores a serem recompensados com uma taxa de serviço, mantendo a busca ativa por novos blocos algo ainda vantajoso. Em um estudo realizado por (Ulrich,2014), foi estimado que o ultimo satoshi – nomenclatura dada a menor unidade de BITCOIN que equivale a 0,00000001 BTC – será produzido no ano de 2140. O que garante essa previsão, segundo Ulrich, é que mesmo que o poder computacional dos mineradores aumente a progressão de dificuldade para mineração desses BITCOINS também serão aumentadas.

Portanto para entendermos como a segurança da rede funciona basta entender que para os nós mineradores, torna-se mais vantajoso ser um braço atuante do sistema do que se tornar parte contrária ao sistema e tentar causar algum tipo de dano ao mesmo, no *BITCOIN* os usuários podem assumir esse papel de minerador e receber pequenas recompensas, chamadas provas de trabalho(Proof of work), durante essas provas de trabalho, o minerador irá realizar operações matemáticas complexas em busca de um a solução, somente após todos os cálculos realizados pelo minerador e sua confirmação de que tudo realmente é seguro e consistente que

um bloco poderá ser considerado válido e integro para só então ser adicionado ao *BLOCKCHAIN*, porém o número de mineradores no sistema cresceu devido ao seu conhecimento, o que acaba causando uma verdadeira corrida em busca da validação dos blocos, somente irá receber a recompensa aquele minerador que resolver seus cálculos de validação em primeiro lugar e propagar essas informações na rede, onde todos os nós ativos deverão entrar em consenso e reconhecer que aqueles *BITCOI*NS da recompensa pertencem realmente ao minerador vencedor e adicionar o bloco ao *BLOCKCHAIN,* assim posteriormente o minerador receberá sua recompensa em *BITCOINS*.

Todo o segredo por parte da mineração das criptomoeda, consiste nos cálculos matemáticos que são adaptáveis ao poder computacional do minerador, pois suponhamos que 2 usuários ingressem no sistema de mineração, porém com sistemas computacionais com poderes distintos, um com grande poder computacional enquanto que outro com pequeno poder computacional concorrem pela validação de um mesmo bloco, assim lhe serão dados problemas computacionais distintos, os quais através de probabilidade serão resolvidos em uma média de 10 minutos, essa regulação na dificuldade de mineração é feita a cada 2016 blocos. Todo esse mecanismo computacional torna a mineração difícil devido ao uso do *HASH* SHA-256 que possui 256 bits, assim seu *HASH* começa por uma quantidades de dígitos 0 e encontrar um bloco que contenha uma quantidade de zeros na parte inicial do seu *HASH* é extremamente baixa, o que obriga o minerador a realizar diversas buscas até encontrar o valor correto. Todavia, esse mecanismo citado torna o sistema ainda mais forte a invasão e falsificação de blocos.

Portanto se uma maioria de mineradores honestos realizar o controle do sistema, a cadeia de blocos corretos adicionados ao *BLOCKCHAIN* crescerá de maneira concisa e integra, evitando que nós desonestos tentem modificar algo no sistema, pois para isso eles deveriam refazer a prova de trabalho de todos os blocos e ainda continuar a minerar na mesma velocidade dos nós honestos (NAKAMOTO, 2008).

Todo esse trabalho de mineração dos blocos tem que ser adicionado a algum tipo de estrutura de registro e armazenamento desses dados, no caso do sistema *BITCOIN,* surge um novo modelo de estrutura, onde os blocos serão armazenados e organizados de maneira que garantam um ordem cronológica sem o uso de um relógio

Global, essa ferramenta será importante para evitar falhas bizantinas em sistemas assíncronos, como no caso no *BITCOIN*, portanto na próxima seção iremos entender como ficam organizados a cadeia de blocos com todas as transações do sistema.

3.5 Blockchain

A palavra *BLOCKCHAIN* é de origem da língua inglesa, portanto significa cadeia de blocos, que pela própria nomenclatura já descreve a ideia inicial, temos o conceito de blockchain como uma estrutura de dados – dizemos estrutura de dados, pois podemos modificar o seu funcionamento a depender na necessidade de utilização – que atua no controle financeiro do BITCOIN, assegurando a sua autenticidade, de fato o blockchain atua como se fosse um caderno de anotações manual de um pequeno comércio registrando todas as transações que foram realizadas até aquele determinado momento, criando um elo entre cada transação estabelecida de entrada e saída, o que tem um papel crucial para assegurar que todos os dados serão corretos e íntegros no sistema *BITCOIN*. Assim cada transação realizada será assinada de forma digital no registro de transações e todos os usuários da rede irão realizar o download do *BLOCKCHAIN* em suas máquinas antes de iniciar suas atividades na rede.

Figura 3: Representação da cadeia de blocos do *BLOCKCHAIN*

Fonte: (LEWIS, 2015)

A principal característica do sistema de armazenamento de blocos é possuir um rígido padrão que evite alterações, acusando qualquer modificação no livro caixa do sistema *BITCOIN* e que fosse facilmente detectada e quebrasse esse elo malicioso da corrente de ligação dos blocos e o isolasse. Para tanto foi criado uma forte ligação através de ponteiros *HASH* os quais ligam sempre novos blocos a blocos anteriores criando uma gigantesca cadeia de dados, onde foi dada a nomenclatura de base distribuída de dados que mantem uma lista encadeada (DE LUCENA,HENRIQUES, 2016).

De acordo ainda com (DE LUCENA,HENRIQUES, 2016) o *BLOCKCHAIN* obedece a alguns princípios fundamentais que irão garantir a integridade dos seus dados e tornar o sistema ainda mais seguro. Assim, são consideradas as funções de mão única – onde operações são realizadas em um único sentido, afim de não tornar conhecido os valores de entrada da função, os quais são geralmente realizados por *HASH* – também é fundamental o registro de tempo – assim qualquer alteração será demarcada em seu tempo e instante – e as assinaturas digitais – que inibem qualquer tipo de alteração nos blocos que serão gerados através de chaves públicas e privadas – assim (DE LUCENA,HENRIQUES, 2016) define o modus operandi do *BLOCKCHAIN* que se distribui por toda rede descentralizada do *BITCOIN* por seus usuários ativos.

Para que o *BLOCKCHAIN* seja atuante o sistema precisa ser preenchido com dados, estes dados serão organizados em forma de blocos e consequentemente uma estrutura de dados denominada arvore de Merkle irá resumir esses dados, afim de esses dados possam ser verificáveis através de um método de busca a partir da raiz da árvore, essa parte do sistema será analisada na seção 3.5.2 deste trabalho. Portanto o *BLOCKCHAIN* pode armazenar uma ou mais operações envolvendo a criptomoeda, na próxima seção veremos como funciona a estrutura de um bloco, que é a mínima parte do sistema, eles serão responsáveis por carregar as informações relativas as transações e assinaturas do criador do bloco, entre outros campos necessários para funcionamento do sistema.

3.5.1 Blocos

Abordaremos nesta seção a peça que irá compor e preencher com dados o *BLOCKCHAIN*, assim o bloco será a unidade responsável pelo registro unitário de

transações e armazenará informações sensíveis responsáveis por valor, carteira de destino, estampa de tempo e etc.

No Blockchain transações serão representadas por um bloco, que serão enviados para todos os outros nós na rede após a verificação da sua corretude, e em cada bloco vão existir informações que serão fundamentais para que os mineradores averiguem os dígitos verificadores do bloco ou *HASH* – O hash é uma função que irá verificar uma sequência/cadeia de caracteres e ao submeter uma determinada sequência a essa função hash ela irá retornar um código que irá conter um digito verificador como visto na seção 3.3, assegurando se a cadeia está ou não correta, portanto quanto mais dígitos verificadores menor será a probabilidade de falha ou erro – no BITCOIN são utilizados dígitos verificadores de 77 dígitos que podem ser números ou letras, o que praticamente anula a possibilidade de duplicatas. A seguir apresentaremos os campos de um bloco a partir de uma tabela.

Tabela 1: Representação dos campos de um bloco

Campo	**Descrição**
Versão	A versão das regras de validação que o bloco segue
ID	O *hash* do bloco, calculado com a função *double* SHA256 (a função SHA256 composta com ela própria).
Quantidade de transações	A quantidade de transações no bloco.
Lista de transações	As transações propriamente ditas
Bloco Anterior	O id do bloco imediatamente anterior (ponteiro hash). Esse campo cria uma estrutura acíclica chamada blockchain.
Raiz da arvore de Transações	O valor *hash* da raiz da árvore de Merkle que contém as transações do bloco (ponteiro *hash*).
Estampa de tempo	Data e hora da criação do bloco. O tempo é especificado em segundos desde a meia noite de 1/1/1970 UTC (*Coordinated Universal Time*).
Valor Alvo	Valor alvo que regula a dificuldade da prova de trabalho no momento da criação do bloco.
Nonce	Campo numérico relacionado ao cálculo da prova de trabalho

Fonte: FILHO(2016).

Cada bloco deve referenciar uma transação prévia que disponibilizou aquele valor, ou seja o bloco gerador do valor que será gasto no bloco atual, essa referência prévia será apontada para o valor do digito verificador do bloco anterior que será o ID de cada bloco e se algum usuário tentar maliciosamente forjar algum desses blocos o digito verificador será invalidado e portanto nada será inserido no *BLOCKCHAIN*(FILHO,2016). O tempo médio que um minerador leva para realizar o trabalho de força bruta na verificação dos dados de um bloco é regulado para durar em torno de 10 minutos através do campo Valor alvo, assim usuários que possuem maquinas distintas podem competir em determinado nível de igualdade dentro do sistema.

Um campo que merece destaque é o *NONCE,* responsável por determinar o valor que será encontrado ao realizar a verificação do bloco, se algo foi modificado o valor do campo *NONCE* nunca será atingido e portanto sua corretude será anulada e o bloco será invalidado. Vale destacar que a estampa de tempo(*TIMESTAMP)* serve para dificultar ainda mais as tentativas de modificação, pois o usuário malicioso deverá criar um bloco no mesmo instante de tempo que o bloco original foi criado, caso contrário o valor calculado pelo *HASH* nunca será idêntico ao original, tornando, portanto o bloco inválido, ele será identificado pela rede e posteriormente se tornará inutilizável.

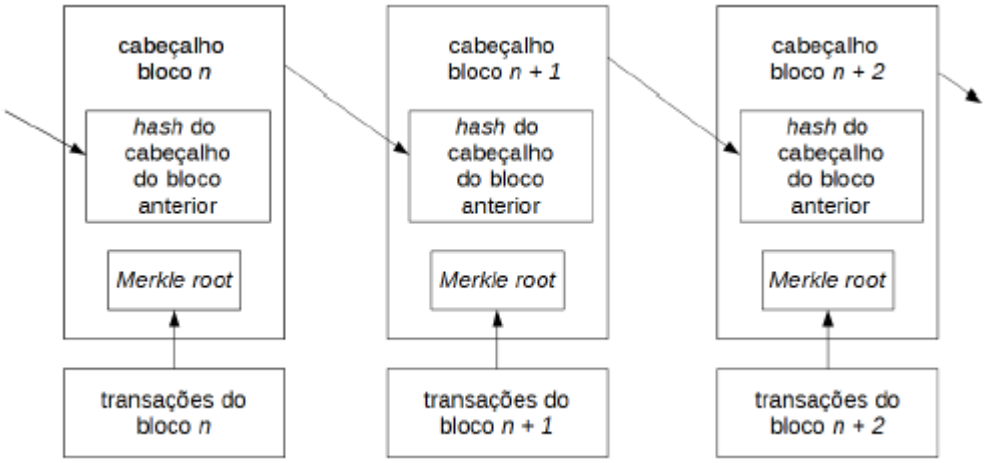
O sistema necessita de uma estrutura de dados para organizar e resumir todos as transações contidas na *BLOCKCHAIN*, para isso (NAKAMOTO, 2008) utilizou uma árvore que pudesse atuar através de ponteiros *HASH*, assim a árvore de Merkle realizará o papel de organizador através de ***BRANCH's*** *(*ramificações) que será visto na próxima seção.

3.5.2 Arvore de Merkle

Em uma análise realizada por (FILHO, 2016) os blocos devem sempre ser referenciados a blocos anteriores, portanto é necessário um mecanismo de ligação que seja seguro entre os blocos, portanto uma estrutura de dados que seja inerente a ciência da computação, para isso são utilizados ponteiros *HASH* que realizam o enlace entre os blocos adicionados ao blockchain assegurando que a transação anterior irá apontar para uma próxima que seja válida. Para isso o ponteiro *HASH* irá

se estruturar de um ptr – que será o ponteiro – e um D – que será o *HASH* do receptor daquele ponteiro, seu valor apontado – constituindo uma tupla (ptr, D)

Figura 4: como funcionam os ponteiros no calculo do *HASH*



Fonte: (DE LUCENA, HENRIQUES, 2016)

O grande diferencial de um ponteiro HASH é que o mesmo garante que seja mantida a integridade na estrutura que será apontada, o que significa que para descobrir qual será o próximo bloco apontado, basta apenas calcular o *HASH*, caso sejam iguais, o ponteiro está apontando para a próxima estrutura de maneira correta. De uma maneira geral, esse tipo de abordagem de ponteiros possibilita um espaço de armazenamento menor, facilitando assim o seu uso pois basta armazenar os ponteiros *HASH's.*

Mas boa parte dessa introdução por ponteiros *HASH* analisada em (FILHO, 2016) se deve ao fato de que será aplicado em Arvores de Merkle, que são basicamente estruturas de dados baseadas em Arvores binárias onde seus ponteiros utilizarão *HASH*, sua grande vantagem é que todos os ramos serão verificados a partir do nó raiz, o que assegura de uma forma mais sólida que os dados se mantenham íntegros, pois todos os nós que ligam da raiz até o nó buscado são feitos por ponteiros *HASH* .

As transações de um bloco devem ser armazenados sob a forma de uma estrutura de dados, no modelo do sistema *BITCOIN* os blocos serão armazenados como uma árvore de Merkle. O valor do *hash* de uma transação é calculado, em seguida o próximo bloco também terá seu valor *hash* calculado e o valor do *hash* que

será gerado ao fim de todo o processo será chamado de Raiz de Merkle(Merkle Root), assim realizar modificações que quebrem todos os elos de ligação entre as cadeias se torna inviável.

A técnica de aliar *hash's* com a arvore de Merkle ainda receberá um aliado que tornada o sistema ainda mais forte, são as chaves para assinatura digital, que serão abordadas no próximo capitulo.

3.6 Chaves Públicas e Privadas

Os membros participantes da rede deverem armazenar o Blockchain em sua máquina de forma integra e atualizada, pois para acontecer uma transação, os blocos devem entrelaçar de onde veio o dinheiro e para onde o mesmo vai, assim sendo ele deverá possuir endereços que chamaremos de chaves criptografadas, que servem para garantir a sua segurança, uma de domínio público (Public Key) – que será repassada ao membro que irá realizar um depósito em sua conta e onde todos os usuários podem verificar suas transações – outra de domínio privado(Private Key) – que deve ser pessoal, intransferível e armazenada em um local seguro, pois ela vai dar acesso a sua conta e permitir o usuário gaste a sua criptomoeda e assine transações – essas chaves serão incluídas junto ao valor que será transferido no cabeçalho do bloco antes de sua inserção no blockchain.

O sistema funciona basicamente por meio de assinaturas digitais das mensagens, assim suponhamos que João deseja enviar uma mensagem para maria e essa mensagem seja assinada digitalmente, ambos irão possuir um par de chaves como as citadas anteriormente (pública e privada), para que João envie algo para maria ele deverá submeter a sua mensagem a uma função *HASH* que irá gerar uma sequência criptografada da mensagem e em seguida a chave privada do João é aplicada sobre a sequência criptografada e então é gerada uma nova sequência criptografada que irá garantir a integridade dos dados que agora poderão circular pela rede. Quando Maria receber a mensagem criptografada, ela deverá verificar a assinatura de João, para realmente se certificar que aquela mensagem é integra e autêntica, para isso ela utiliza a chave pública do João para realizar a verificação. A realização desse processo de verificação das assinaturas dá ao sistema *BITCOIN* mais robustez ao garantir a integridade dos dados, pois aninha técnicas inerentes a

computação para garantir que seja quase impossível uma cópia perfeitamente idêntica a mensagem original, provando que sistemas distribuídos e assíncronos podem se consolidar como seguros e íntegros. O método utilizado pelo *BITCOIN* para criar um esquema de assinaturas digitais é o *Elliptic Curve Digital Signature Algorithm* (ECDSA), como definido em (FILHO, 2016) esse esquema cria uma criptografia através de curvas elípticas e consegue alcançar bons níveis de segurança com chaves que possuam tamanhos relativamente pequenos.

As técnicas vistas anteriormente de assinatura digita e *HASH* são criptográficas, ainda assim, iremos abordar na próxima seção a criptografia, que é um fator preponderante em quase todas ações do sistema *BITCOIN.*

3.6.1 Criptografia

A comunicação dos dados em uma rede P2P deve ser sigilosa, visto que todos os usuários irão receber todas as informações e somente aqueles que possuírem permissão devem realmente acessar as informações com segurança, assim os usuários podem trocar informações através de um meio inseguro – que é uma rede de trafego de dados qualquer – e ainda assim seus dados serão mantidos concisos. Para isso é utilizada a criptografia, que é uma tecnologia que se baseia em cifrar e decifrar uma mensagem, tornando-a totalmente ilegível para qualquer usuário que receba aquele conteúdo e não possua um código para decifrar a mensagem. Para isso são utilizadas chaves e elas podem ter vários tamanhos, são as chaves que tornam o sistema criptográfico forte ou fraco, quanto maior for o tamanho dessa chave (quantidade de bits), maior será a codificação da mensagem e, portanto, ela estará mais segura. No BITCOIN as chaves Pública e Privada serão geradas para embaralhar toda a mensagem e qualquer um que intercepte a mensagem pelo caminho deverá conhecer as chaves de acesso para desembaralhar a mensagem e assim acessar todo o seu conteúdo, sendo apenas a chave pública conhecida o interceptador apenas irá saber o endereço a quem aquela mensagem é destinada e poderá visualizar a mensagem sem alterá-la, pois se ocorrer alguma modificação a mensagem será invalidada. Somente o detentor da chave privada poderá modifica-la e assiná-la afirmando a sua autenticidade.

O tipo de criptografia utilizada pelo BITCOIN é o assimétrico, definição utilizada por (FILHO, 2016), visto que são utilizadas duas chaves no embaralhamento (Privada - emissor) e desembaralhamento (Pública - receptor),. Portanto a segurança da chave privada é de fundamental importância na seguridade dos dados e transações, pois somente com as mesmas pode-se assinar uma mensagem e garantir a autenticidade daquele autor.

3.7 Transações

Todas as transações já realizadas são armazenadas no *BLOCKCHAIN* e podem ser vistas por todos, isso garante que o BITCOIN não trabalhe de forma totalmente anônima na rede, pois temos as chaves públicas sempre divulgadas nos blocos, ou seja, podemos verificar quando o usuário gastou ou recebeu, apenas não sabemos a identidade propriamente dita daquele determinado usuário. Contudo ainda assim temos um nível altíssimo de privacidade entre os usuários da criptomoeda, que por sua vez também garante uma transparência de todas as transações e uma possível auditoria por parte de qualquer usuário

4 Desenvolvimento

A base de funcionamento do sistema *BITCOIN* foi analisada através da fundamentação de cada parte do sistema, afim de demonstrar em uma primeira leitura como se realiza o processo de transações entres os mineradores (processos) ativos do sistema. Nesta seção do estudo, será realizada a relação de funcionalidade do sistema com objetos inerentes a ciência da computação, mais especificamente a parte de segurança de sistemas distribuídos. Portanto será estabelecida uma análise cientifica para tratamento das falhas apontadas na seção 2.1 e suas possíveis perturbações causadas ao sistema. Para isso serão averiguadas as perturbações em sistemas distribuídos detalhadas na seção 2.1 e estudos fundamentados ao longo da seção 3 correlacionados ao sistema *BITCOIN*.

4.1 Modelo de sistema

Para compreensão das seguintes seções, será definido o modelo especifico de sistema em estudo. O sistema *BITCOIN* funciona como um sistema distribuído concorrente assíncrono, assim sendo não possui um relógio global que sincroniza a prioridade e atuação dos processos no sistema e portanto não garante restrições temporais de entrega de mensagens como observado em 2.1.1, a ausência de um relógio global e o uso de processos distribuídos pela rede demanda necessidades técnicas especificas como: Integridade da causalidade dos eventos do sistema, consistência no consenso dos usuários. O fato de saber que por ser uma rede TCP/IP as mensagens irão chegar aos seus destinatários, porém não se sabe quando e se estão intactas devido a um atraso do meio de comunicação ou adulteração durante o trajeto pela rede. Consequentemente o estudo em questão examinará se essas necessidades técnicas de um sistema assíncrono e distribuídos são sanadas de maneira satisfatória com o comportamento anômalo de processos internos e externos ao sistema.

4.2 Falhas Bizantinas

Os mecanismos de funcionamento do *BITCOIN* vistos servem como base para que possamos estabelecer uma relação entre segurança e as falhas bizantinas mais conhecidas e exploradas na seção 2.1, assim será examinado o processo com que o

sistema *BITCOIN* consegue solucionar esses problemas de maneira segura e clara para os usuários demonstrando ser um sistema tolerante a falhas. Como definido em Avizienis *et al*, (2004) alguns conceitos são fundamentais para que a segurança seja mantida em sistemas computacionais. Portanto os nós ativos no sistema, que são os mineradores, possuem o comportamento de processos, isso irá facilitar a análise de um sistema tolerante a falhas.

No estudo realizado por (DE LIMA, F. GREVE, 2015) sobre sistemas distribuídos dinâmicos são estabelecidos formatos de mensagens para as falhas possam ser detectadas por todos processos ativos no sistema, um deles é o uso de um certificado que permita a verificação da mensagem através de um algoritmo, esse tipo de verificação é realiza no sistema *BITCOIN* através do uso da função *HASH,* que permite que todos os processos identifiquem caso uma mensagem chegue adulterada, tanto por fatores de escopo do algoritmo ou por manipulação.

Se um determinado processo passar a atuar no sistema com um estado de erro, ou seja, um processo omisso, apenas não participando ou deixando de enviar um resultado encontrado por conta de atraso no canal de comunicação, ou mensagens chegando atrasadas e portanto sem validade sendo assim um problema de omissão, em determinadas situações o processo não falha, ocorrem apenas perdas ou atrasos no meio de comunicação, assim o processo simplesmente irá perder a corrida pela mineração para um outro minerador concorrente e portanto não causará nenhum dano ao sistema, se tornando um processo faltoso e nunca conseguiria portanto finalizar a mineração de algum bloco e vencer a corrida, porém posteriormente ele deverá atualizar sua *BlockChain* até o último bloco para novamente voltar a condição de corrida e sair da condição de omissão temporária. Em um sistema síncrono, essa restrição poderia ser resolvida facilmente através de um uso de um temporizador (*TimeOut)* para detectar falhas de omissão onde o processo que deixou de responder ou enviou uma mensagem que não chegou ao seu destinatário, seria descartado do presente ciclo de eventos pois os processos iriam aguardar por um tempo indeterminado uma resposta que não chegaria acarretando em uma falha por *LIVENESS* – que é a propriedade que garante que o sistema se mantenha ativo, cumprindo a sua vivacidade, finalizando o processamento de dados ativos e iniciando novos processos na condição de corrida. O *BITCOIN,* no momento em que um

minerador inicia o processo de mineração ele deverá finalizar e dar a vez para uma nova condição de corrida. Porém em um sistema assíncrono como o do *BITCOIN* o *LIVENESS* será garantido quando ao menos um processo encontrar o resultado esperado e comunicar a todos os outros que aquela verificação possui relevância e está correta, essa é uma propriedade que deve ser garantida para sistemas distribuídos concorrentes e assíncronos, assim todos os processos que perderam a corrida realizarão a verificação dos dados e entraram em consenso que o processo foi honesto e vencedor da corrida. Logo o problema da omissão momentânea ou total é atendida pelo sistema de maneira factual e simples. O reconhecimento por parte dos outros membros se dará através da verificação da autenticidade do bloco (mineração), assim o mesmo será adicionado a um *BRANCH* (ramificação) e aquela que for reconhecida como a maior ramificação por todos os membros em consenso, será a cadeira honesta no *BLOCKCHAIN*.

O processo de detecção de falhas evidenciado em (DE LIMA, F. GREVE, 2015) supõe que para que processos bizantinos possam vir a atuar sobre um determinado sistema e para que o sistema seja tolerante a falhas bizantinas, ela deverá atender dois princípios fundamentais: Os processos ativos devem receber as mensagens, verificar sua corretude através de redundância e assinaturas digitais e também verificar a consistência das mensagens de acordo com o algoritmo do sistema. Para que esses dois requisitos definidos acima sejam atendidos, (DE LIMA, F. GREVE, 2015) define técnicas, como: Redundância, uso de assinaturas digitais não forjáveis e a validação de conteúdo através de certificados. Técnicas abordadas no presente estudo nas seções 3.6 e 3.3 respectivamente, que tornam o sistema analisado neste trabalho tolerante a falhas bizantinas, pois em seu algoritmo de funcionamento atende aos requisitos de falhas propostos em sistemas assíncronos distribuídos. Isso não garante que o sistema nunca possa vir a falhar, mas sim de que o sistema foi estruturado para que possa suportar tais falhas criando um forte elo de segurança através dessas técnicas.

Contudo (DE LIMA, F. GREVE, 2015) mostra que não é possível diferenciar uma falha por omissão de um retardo ao receber mensagens devido ao fato do sistema ser assíncrono realizando assim uma conjectura e um corolário como definido a seguir:

> *“*Conjectura 1 *Num sistema assíncrono, é impossível detectar falhas bizantinas por omissão, de forma assíncrona, caso o padrão de troca de mensagens do algoritmo A seja do tipo* 1 → *n; isto é, se em determinado momento, o algoritmo A determina que apenas um processo envia mensagens aos demais (n).*
> Corolário 1 *A forma assíncrona da detecção de falhas bizantinas só pode ser adotada por protocolos simétricos, nos quais todos os nós executam o mesmo papel.” (DE LIMA, F. GREVE, 2015,* p.12*)*

A análise do presente corolário, justifica o fato de que se um bloco não envia sua mineração aos demais mineradores, ele nunca vencerá a corrida, porém nunca se saberá se foi por omissão ou algum tipo de retardado sofrido pelo meio de comunicação.

O processo pode não atuar da maneira como especificada pelo sistema fugindo ao escopo determinado pelo algoritmo, no caso uma falha por conta de uma vulnerabilidade ou algum outro processo malicioso, ele poderá minerar um bloco com erro no cálculo do *HASH,* com dados adulterados no bloco ou até mesmo um bloco que possua um *HASH* valido, entretanto ao tentar propaga-lo na rede para possível inserção no *BlockChain* seria detectada qualquer alteração por todos os usuários que recebessem aquele bloco modificado devido ao cálculo do *HASH* realizado por força bruta não bater com as assinaturas digitais, valores, data de criação ou qualquer outro campo que tenha sido modificado do bloco original, o que acabaria por invalidar o bloco malicioso devido ao pressuposto que a modificação causaria um valor diferente no campo do digito verificador(Nonce) e o mesmo passaria a ser tratado como um bloco órfão por todos os outros processos, levantando uma suspeita sobre aquele processo, o que é consequência do processo de redundância e uso de assinaturas não forjáveis com verificação posterior dos processos ativos da veracidade dos dados.

Retornando a identificação de mensagens anômalas, que podem ser mensagens que possuam duas versões diferentes e sejam propagadas pela rede afim de causar confusão na identificação de processos ativos, para que uma mensagem anômala seja identificada segundo KIHLSTROM (2003) um registro de transações deve ser gerado afim de coletar todas as trocas de mensagens e todos processos honestos devem reenviar as mensagens a todos os processos do sistema, porém isso funciona em uma rede ponto-a-ponto, em um sistema distribuído assíncrono as

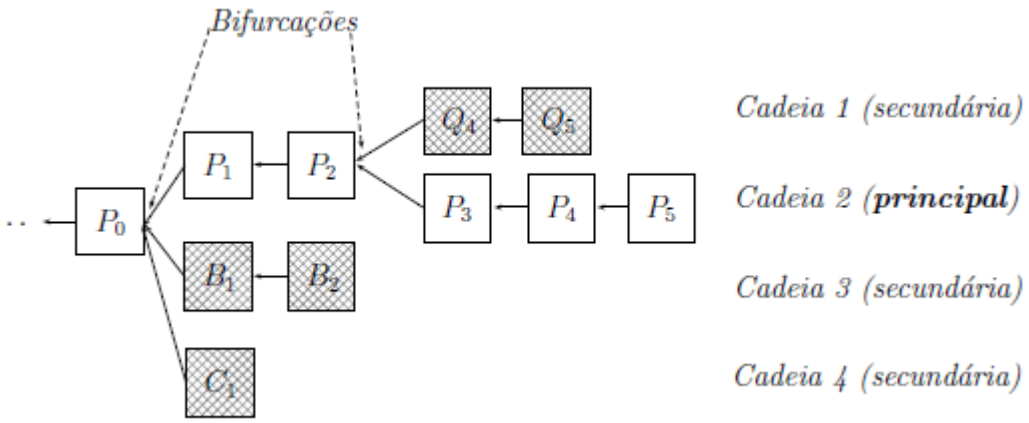
mensagens serão recebidas com o mesmo conteúdo por todos os processos honestos, evitando assim a ocorrência de mensagens anômalas, como observado em (DE LIMA, F. GREVE, 2015). O que garante assim um tratamento das falhas de segurança por conta do sistema ser distribuído e a troca de mensagens acontecer por meio de difusão, tornando a identificação dessas anomalias mais fácil.

4.3 Consenso

O consenso nada mais é do que a aceitação por parte dos processos de um resultado que seja legitimo e tido como verdade pela maioria do quórum de processos ativos naquele determinado momento de atividade do sistema. Em (GUERRAOUI, 1999) é estabelecido um quórum para que um evento passe a ser aceito pelo sistema distribuído e todos realizem um commit do resultado final afim de que todos os processos estejam atualizados em relação aos dados, para isso os processos devem entrar em acordo em relação a escolha dos dados a serem realizados o commit. No sistema *BITCOIN* cada vez que um processo finaliza a mineração de um bloco, o mesmo recebera o poder de um voto SIM para escolher qual a sequência(FORK) que o mesmo irá seguir, e ao escolher uma determinada sequência ele votará SIM para que aquela sequência seja a principal, portanto a cadeia será escolhida como principal, caso haja uma sequência concorrendo com a mesma, depois de 6 votos SIM dos processos ativos ela será confirmada como a cadeia principal.

Quando novos blocos são gerados na rede, eles serão adicionados a uma cadeia de blocos que já estão consistentes até aquele presente momento no *BlockChain*, porém se um bloco não for recebido por um determinado nó, ou caso esse bloco seja perdido, pode ser gerada uma bifurcação (*FORK)* entre a sequência de novos blocos, ou seja vão existir duas sequências e aquela que se tornar maior (receber os próximos 6 blocos) será considerada a cadeia principal através de consenso(FILHO,2016), pois recebeu 6 votos SIM em um período de 60 minutos ou 1 Hora, o que torna aquela cadeira confiável e integra.

Figura 5: Representação de uma bifurcação(FORK)



Fonte: figura retirada de (FILHO,2016)

O grande impasse do *BITCOIN* definido em FILHO (2016) é portanto definir através de consenso entre os nós ativos da rede qual será a cadeia considerada como principal, que conterá somente os blocos válidos de todas as transações. Para descobrir qual é a cadeia principal, basta avaliar e encontrar a que possuir maior extensão. Para isso foi definido em (RODRIGUES,2017) que na rede BITCOINS toda vez que um bloco é adicionado é computado um SIM para aceitação daquela cadeia como principal e a partir de 6 votos, ou seja 6 novos blocos adicionados, é praticamente impossível invalidar aquela cadeia – assim sendo uma nova cadeia de blocos leva em torno de 1 hora para que seja aceita como a principal – analisando do ponto de vista da segurança, enquanto toda a rede ainda continua a complementar a cadeia principal com novos blocos, um suposto atacante que queira prejudicar o sistema, deverá realizar toda a prova de trabalho que já foi realizada de todos esses blocos da cadeia principal e sabemos que ao realizar uma prova de trabalho o minerador tem que receber um algoritmo aleatório o qual irá solucionar e que qualquer modificação por mais ínfima que seja irá alterar o *HASH* do último bloco e consequentemente irá invalidar a prova de trabalho, além de gastar um poder de processamento para minerar todos os blocos enquanto rede ainda continua a minerar.

Esse número de 6 blocos adicionados a *BLOCKCHAIN* não é um valor arbitrário, o mesmo consiste em um tempo, pois como examinado, um processo leva em torno de 10 minutos para finalizar a mineração de um bloco e portanto 6 votos acarretariam em uma hora, esse é um fator de segurança para o sistema pois para que nós invasores tentem tomar o controle do sistema, eles precisarão de um poder

computacional superior a todos os processos ativos que estão minerando naquele momento e ainda terão que processar todos as provas de trabalho feitas em uma hora, o que se torna praticamente impossível do ponto de vista computacional.

Consequentemente podemos demonstrar que o poder computacional gasto para realizar todas essas modificações na rede seria muito alto e todas as tentativas seriam invalidadas devido a sua prova de trabalho que será adulterada e identificada através das assinaturas digitais, por isso podemos também afirmar que seria muito mais vantajoso para o atacante minerar blocos de forma correta e autentica, pois, assim receberia suas recompensas que seriam muito mais valorosas e contribuiriam para uma maior segurança da rede agindo pelo lado honesto do *BITCOIN*. Tudo isso torna o sistema ainda mais seguro e bem visto aos olhos dos usuários, pois por pior que sejam as intenções de um atacante, ele ainda prefere se render ao poder de fazer parte da corrente solidária da rede.

O minerador tem como principal função minerar novos blocos e adicioná-los ao BlockChain, pois somente assim irá receber a recompensa por toda a prova de trabalho executada para isso, mas de nada vale o minerador adicionar novos blocos a uma cadeia que será considerada invalida, pois o minerador não irá receber sua recompensa por todo o processamento o qual dispensou até validar aquele bloco e tentar adicioná-lo. Portanto ao escolher a cadeia mais confiável, ou seja, aquela possui uma cadeia maior e buscar consenso é vantajoso, o processo busca a melhor maneira de receber suas recompensas, pois se o mesmo tentar forjar que uma cadeia é a correta e essa cadeia não for, quando os outros processos verificarem através do *HASH* e das assinaturas digitais irão votar NÃO para recompensa daquele processo e todo o seu gasto computacional será perdido, pois aquele processo receberá uma FLAG de suspeito para aquele evento em questão e essa FLAG também será propagada pela rede.

Outro exame é a situação a qual uma cadeia secundária será formada e pode passar a ser a principal, para isso a cadeia secundária deve ultrapassar a cadeia principal até aquele dado momento em número de blocos, a partir de então ela passará a ocupar o lugar de cadeia principal e todos os blocos até aquele momento serão válidos. Analisando essa parte do *BITCOIN* o leitor pode inferir que algo malicioso seria algo fácil de ser introduzido na rede *BITCOIN*, ao inferir que um determinado atacante queira criar uma sequência secundária forjada afim de transferir

BITCOINS válidos de uma determinada carteira para outra – pois ao analisar superficialmente, o atacante só precisaria ultrapassar o número de blocos da cadeia principal – assim o atacante forja a criação de 4 novos blocos que retiram BITCOINS válidos de transação já realizada e presente na cadeia principal e os transfere para outra carteira de maneira maliciosa. Esse tipo de problema é conhecido como Gasto-Duplo e será analisado na seção 4.6.

4.4 Problema Dos Generais Bizantinos

Como visto na seção 2.1.2 foi abordado o problema dos generais bizantinos, que devem entrar em um consenso em qual a melhor decisão e a hora certa de atacar ou recuar, porém sem poder se comunicar com os outros generais. Esse problema causa uma inconsistência devido a integridade causal dos blocos que devem ser adicionados a *BLOCKCHAIN.* Assim iremos demonstrar como será resolvido o problema do consenso quando não existe comunicação direta com os outros nós.

O problema abordado nesta seção pode permitir que nós se tornem falsificadores ou nós que enviem mensagens duplicadas para outros nós afim de causar ambiguidade de informações que estão incorretas. Portanto, ao se tornar um nó ativo no sistema, dele deverá possuir uma *BLOCKCHAIN* consistente, ou seja, atualizada, e através do uso do *HASH* o nó poderá verificar a integridade dos dados através da recomposição do *HASH.*

A questão consiste em adicionar novos blocos e esperar que a rede os aceite adicionando-o ao *BLOCKCHAIN* ou não aceita-los, similar ao problema dos generais, que devem decidir quando atacar ou recuar sem se comunicar.

A resolução do problema se dá através da prova de trabalho(*proof-of-work*), pois os blocos sempre serão encadeados aos blocos anteriores que contem também o *HASH* do bloco anterior, o que é simples e rápido de se verificar a garantir a autenticidade do sistema através do incremento do campo *nounce*, e assim que o mesmo for identificado, o bloco poderá ser adicionado ao *BLOCKCHAIN* e o nó poderá propagar a informação de que o bloco foi minerado. Portanto o ato de minerar um bloco garante que o problema dos generais bizantinos seja elucidado e garante uma segurança para que o sistema continue a atuar de maneira integra.

Na próxima seção iremos abordar um problema recorrente na computação relacionado a sistemas financeiros, que é o gasto duplo, onde saques podem ser efetivados duas vezes mesmo sem a conta possuir saldo, veremos como o sistema se comporta frente a essa falha.

4.6 Gasto Duplo

A proposição de um usuário gastar uma quantia a qual não possui é um problema recorrente no uso de softwares de sistemas bancários, este problema é derivado de uma sequência de eventos que chegaram foram de ordem e permitiram que este tipo de gasto duplicado ocorresse, portanto em sistemas síncronos é importante manter a ordem dos eventos para que situações como o gasto duplo não seja recorrente. Portanto um relógio global e servidor central que receba essas mensagens e verifique se o gasto é possível para aquele determinado processo, funcionam de maneira satisfatória em sistemas síncronos, porém em sistemas assíncronos como o do *BITCOIN* foi necessário uma inovação através do aninhamento de técnicas que serão vistas nesta seção.

O problema denominado na área da computação de Gasto Duplo, foi definido em (DE LIMA, F. GREVE, 2015) como a tentativa de um usuário transferir um determinado valor financeiro duas vezes sem possuir aquele saldo em sua conta, portanto corresponde ao pagamento a dois usuários de um mesmo saldo insuficiente. (DE LIMA, F. GREVE, 2015) Ainda constata que pela primeira vez o problema do gasto duplo foi solucionado sem a utilização de um mediador ou órgão centralizador que pudesse realizar essa verificação. Assim os *BITCOINS* que deveriam ser gastos em uma transação são gastos em outros de maneira inválida.

Como todos os processos possuem uma cópia de todas as transações até aquele determinado momento, se torna fácil verificar através do atrelamento de todas as transações, como visto na seção 3.5, se o usuário realmente possui aquele crédito a ser gasto através da sua chave pública, que mostra a quantidade de moedas que aquele determinado usuário possui e se a mensagem foi assinada com a chave privada pelo processo que deseja transferir aquele crédito em moedas para outro

usuário (DA SILVA, 2014). Ainda assim todas as transações são interligadas no *BLOCKCHAIN* através de ponteiros *HASH* ,vistos na seção 3.5.2, que criam uma relação de causalidade dos eventos, determinando que nenhuma moeda emergiu ou foi gasta de forma inválida no sistema. Essa relação de causalidade das transações que ocorre nos sistemas síncronos é realizada também no sistema *BITCOIN* porém sem o uso de relógio global que ordene a sequência de mensagens quando elas chegam ao ponto central. Essa inovação na parte de segurança propiciou autonomia ao sistema devido ao fato de transações serem realizadas sem o acompanhamento e fiscalização de terceiros.

4.7 Ataque Denial Of Service

O sistema BITCOIN já possui alguns mecanismos incorporados para prevenção de alguns ataques de segurança em redes distribuídas (MARTINS, CARVALHO,2014) , dentre eles e dos mais comuns está o ataque DOS(Denial of Service ou ataque por negação de serviço), onde maquinas que são agrupadas, através de infecção ou não, realizando requisições a um determinado servidor ou host de dados, afim de sobrecarregar o sistema e torna-lo instável, inseguro e em determinados casos inacessível, o que seria um grande problema para o sistema BITCOIN.

A resolução desse problema se dá pela escalabilidade no tempo de mineração de um bloco, que é regulado e percorre um tempo em torno de 10 minutos, assim, para que um ataque aconteça, seria necessário sempre derrubar o nó que irá minerar o bloco através de muitas solicitações, entretanto mesmo que esse bloco seja atacado não tem como prever qual será o próximo bloco a minerar vencer a corrida. Outro fator importante também é a ausência de um servidor, devido ao fato da rede ser distribuída. Assim as requisições para sobrecarregar o sistema não tem um direção única devido a arquitetura da rede.

4.8 Fator Humano

Os sistemas podem falhar, de fato é algo que acontece caso ocorram problemas de engenharia de software ou erros não previstos, mas ainda assim existe um fator determinante ao se trabalhar com sistemas, os usuários. É importante saber lidar com qualquer tipo de sistema, ainda mais aqueles que geram ônus financeiros caso sejam mal utilizados, portanto ser um usuário consciente e bem informado faz com que os riscos inerentes ao fator humano sejam reduzidos.

A criptomoeda também corre os seus riscos e pode vir a ser roubado e perdido como qualquer outra forma de moeda no mundo atual, porém a atenção do usuário no manuseio de suas chaves criptografadas é de suma importância. A chave privada uma vez perdida levará o usuário a perda dos seus BITCOINS pois ela é pessoal e intransferível, como dito anteriormente e responsável pelo acesso do usuário a sua carteira – onde ficam guardadas as suas moedas – que é um software que realiza o gerenciamento dos *BITCOINS*. O fator humano é ponto crucial na segurança dos dados dos usuários, visto que o sistema funciona em uma rede e não existe contato pessoal e direto entre os nós que irão realizar transações, portanto negociações podem ser realizadas com pessoas do outro lado do mundo, que o usuário talvez mal conheça, assim sendo a troca de informações ao realizar uma transação deve ser cuidadosa devido a quantidade de golpes que podem ser aplicados a usuários iniciantes e que ainda não possuam um determinado discernimento necessário para negociar suas moedas na rede *P2P*.

Outro ponto crucial é a manutenção da segurança da máquina do usuário, visto que malwares podem infectá-la e subtrair chaves através de *KEYLOGGERS*, assim podendo realizar qualquer tipo de transação financeira dentro da rede e até mesmo furtar a carteira realizando o saque de todos os *BITCOINS* para uma outra carteira. Assim é estritamente recomendado que o sistema operacional esteja atualizado e totalmente verificado em questões de segurança.

Obviamente é recomendado que se utilize um bom antivírus e guarde suas chaves em locais seguros e se ainda assim forem em algum dispositivo eletrônico, que seja desconectado da internet, pois as chaves permitem acesso total ao dados da carteira do usuários, onde qualquer outro usuário pode manipulá-lo e transferir

fundos para outras contas. Evitar o uso do sistema de carteiras em computadores ou celulares que sejam utilizados para fins diversos é uma outra recomendação essencial na seguridade dos seus dados. Com a recente valorização das criptomoedas, cresceram também o número de ataques virtuais em busca de *BITCOINS*, portanto os usuários devem estar atentos a saúde dos dispositivos utilizados.

5 Conclusão

Ao longo do presente trabalho, tratamos das questões de segurança envolvendo o *BITCOIN*, desde os seus pontos positivos e também as suas possíveis falhas e pontos negativos. Sabemos que muito ainda deve ser melhorado para que o sistema se torne realmente aceito por todos e demonstre total segurança, porém já observamos que muitos problemas já foram solucionados pelo sistema.

Problemas recorrentes como o gasto-duplo e a necessidade de um central regulatória para definir se as transações são válidas e possíveis, mostram como o sistema pode solucionar problemas antigos. Como objetivo principal desta analise foram as falhas bizantinas e a maneira como o sistema se comporta perante as mesmas, concluímos que o sistema age de forma satisfatória diante requisitos básicos e até mesmo avançados para garantir que a integridade dos dados e das transações sejam mantidas, assegurando que o *LIVENESS* seja mantido e sistema continue operando mesmo em situações adversas.

A relação estabelecida entre o teórico e a pratica de um sistema que abrange diversas áreas do estudo de sistemas distribuídos, mostra mais uma vez que novos métodos podem

5.1 Trabalhos futuros

Como colaboração para possíveis trabalhos, podemos citar o problema da escalabilidade do *BITCOIN* que vem sendo bastante abordado devido ao seu crescimento e conhecimento acelerado da plataforma.

O sistema *BITCOIN* funciona como um grande banco de dados distribuído entre todos os usuários, que troca informações através de uma rede distribuído e assíncrono, assim a agilidade de suas trocas de dados deve se dar de forma efetiva e que consiga suprir toda a demanda atual, que por sinal, se tornou maior devido a visibilidade das criptomoedas. Portanto a necessidade de melhorar a forma como as transações são processadas e o seu tempo, fizeram com que duvidas inerentes ao tempo gasto – em torno de 10 minutos – para processar uma transação surgissem. Essa escalabilidade do sistema carece de melhorias e, portanto, é uma área vasta para análise de estudos de caso.

Portanto podemos definir essa análise devido ao número de transações adicionadas ao *Blockchain* que cresceu exponencialmente devido a quantidade de novos usuários ativos no sistema e as grandes movimentações da moeda, criando assim um gargalo entre o gerenciamento do tempo das transações e o seu volume.

Surge assim alguns projetos inovadores e que pretendem resolver o problema da escalabilidade, um deles é o projeto do *LIGHTNING NETWORK* que é uma inovação no próprio sistema do *BITCOIN* pois idealiza que o sistema suporte um número indefinido de transações entre os usuários com custos ainda mais baixos, proporcionando um salto ainda maior em relação a tecnologia do protocolo.

Portanto essa inovação traria mudanças de escalabilidade em relação ao sistema do *BITCOIN* condicionando o mesmo ao estudo da segurança e escalabilidade de suas transações e surgindo um novo leque de estudos para os próximos trabalhos.

6 **Referências**

AVIZIENIS, Algirdas et al. Basic concepts and taxonomy of dependable and secure computing. **IEEE transactions on dependable and secure computing**, v. 1, n. 1, p. 11-33, 2004.

BAKKER, Arno; STEEN, Maarten Van; TANENBAUM, Andrew S. A wide-area Distribution Network for free software. **ACM Transactions on Internet Technology (TOIT)**, v. 6, n. 3, p. 259-281, 2006.

CHANDRA, Tushar Deepak; TOUEG, Sam. Unreliable failure detectors for reliable distributed systems. **Journal of the ACM (JACM)**, v. 43, n. 2, p. 225-267, 1996.

Coulouris, George, Jean Dollimore and Tim Kindberg; Gordon Blair. Distributed Systems: Concepts and Design (5th Edition), 2011.

DA SILVA, Douglas Emanuel. Aspectos de segurança na rede Bitcoin. Disponível em: < http://www.cdn.ueg.br/source/observatorio_inhumas/conteudoN/3322/CAP_8__CR_ASPECTOS_DE_SEGURANCA_NA_REDE_BITCOIN.pdf.pdf> Acesso em: 19 agosto.2017.

DE LIMA, Murilo Santos; GREVE, Fabíola Gonçalves Pereira. Detectando Falhas Bizantinas em Sistemas Distribuídos Dinâmicos. Revista Brasileira de Redes de Computadores e Sistemas Distribudos, p. 9-21, 2009.

DE LUCENA, Antônio Unias; HENRIQUES, Marco Aurélio Amaral. Estudo de arquiteturas dos blockchains de Bitcoin e Ethereum. <Disponível em: https://pdfs.semanticscholar.org/eb41/c8ea5c5d191c909d3e107ec84d5e441794c0.pdf> Acesso em: 10 outubro. 2017.

DWORK, Cynthia et al. Fault tolerance in networks of bounded degree. **SIAM Journal on Computing**, v. 17, n. 5, p. 975-988, 1988.

FILHO, Márcio Barbosa de Oliveira. UTILIZANDO O PROTOCOLO BITCOIN PARA CONDUÇÃO DE COMPUTAÇÕES MULTILATERAIS SEGURAS E JUSTAS" Disponível em: <http://repositorio.ufpe.br/handle/123456789/17143> Acessado em: 15 setembro. 2017.

KIHLSTROM, Kim Potter; MOSER, Louise E.; MELLIAR-SMITH, P. Michael. Byzantine fault detectors for solving consensus. **The Computer Journal**, v. 46, n. 1, p. 16-35, 2003.

LAMPORT, Leslie. The weak Byzantine generals problem. Journal of the ACM (JACM), v. 30, n. 3, p. 668-676, 1983.

LAMPORT, Leslie; SHOSTAK, Robert; PEASE, Marshall. The Byzantine generals problem. **ACM Transactions on Programming Languages and Systems (TOPLAS)**, v. 4, n. 3, p. 382-401, 1982.

LEWIS, Antony. A gentle introduction to blockchain technology. **BraveNewCoin, http://bit. ly/2jdE8iz**, 2015. Disponível em: https://bravenewcoin.com/assets/Reference-Papers/A-Gentle-Introduction/A-Gentle-Introduction-To-Blockchain-Technology-WEB.pdf Acesso em: 22 de dezembro de 2017.

LUNDELIUS, Jennifer; LYNCH, Nancy. An upper and lower bound for clock synchronization. **Information and control**, v. 62, n. 2-3, p. 190-204, 1984.

MALKHI, Dahlia; REITER, Michael. Byzantine quorum systems. **Distributed Computing**, v. 11, n. 4, p. 203-213, 1998.

NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Disponível em : <http://bitcoin.org/bitcoin.pdf> .Acesso em: 20 julho. 2017.

PEASE, Marshall; SHOSTAK, Robert; LAMPORT, Leslie. Reaching agreement in the presence of faults. **Journal of the ACM (JACM)**, v. 27, n. 2, p. 228-234, 1980.

RODRIGUES, Carlo Kleber da Silva. Sistema Bitcoin: uma análise da segurança das transações. **iSys - Revista Brasileira de Sistemas de Informação**, [S.l.], v. 10, n. 3, p. 5-23, sep. 2017. ISSN 1984-2902. Disponível em: <http://www.seer.unirio.br/index.php/isys/article/view/5949>. Acesso em: 21 feb. 2018.

TEMPEST SECURITY INTELLIGENCE. **MARCO CARNUT** - Introdução ao Blockchain e à Rede Bitcoin. Fevereiro 2016. Disponível em: <https://www.youtube.com/watch?v=kQIQHGUEDv8 > Acesso em: 5 Agosto. 2017.

ULRICH, Fernando. **Bitcoin**: a moeda na era digital. São Paulo: Instituto Ludwig von Mises Brasil, 2014. 122 p. Disponível em: <http://www.elivros-gratis.net/scripts/download.asp?SEC=14&FL=Fernando-Ulrich-Bitcoin.zip&NOME=Bitcoin%20-%20A%20Moeda%20na%20Era%20Digital&AUTOR=Fernando%20Ulrich>. Acesso em: 20 julho. 2017.