UNIVERSIDADE FEDERAL DE ALAGOAS

INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DANIEL DE MELO PIMENTEL

Uma Proposta para Aprimorar o Anonimato em Transações *Bitcoin* com Suporte à Auditoria.

Maceió
2017

Daniel de Melo Pimentel

Uma Proposta para Aprimorar o Anonimato em Transações *Bitcoin* com Suporte à Auditoria.

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Informática da Universidade Federal de Alagoas, como requisito parcial para obtenção do grau de Mestre em Informática.
Orientador: Prof. Dr. Leandro Melo de Sales

Maceió
2017

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Bibliotecário Responsável: Helena Cristina Pimentel do Vale

P644u Pimentel, Daniel de Melo.
Uma proposta para aprimorar o anonimato em transações bitcoin com suporte à auditoria / Daniel de Melo Pimentel. – 2017.
94 f.: il.

Orientador: Leandro Melo de Sales.
Dissertação (Mestrado em Informática) – Universidade Federal de Alagoas. Instituto de Computação. Programa de Pós-Graduação em Informática, Maceió, 2017.

Bibliografia: f. 66-67.
Apêndice: f. 68-69.
Anexos: 70-94.

1. Bitcoin. 2. Moeda virtual. 3. Auditoria. 4. Assinatura de grupo – Anonimato. 5. Serviços online – Transações. I. Título.

CDU: 004.05:336.741.24

UNIVERSIDADE FEDERAL DE ALAGOAS

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Informática – PpgI
Instituto de Computação
Campus A. C. Simões BR 104-Norte Km 14 BL 12 Tabuleiro do Martins
Maceió/AL - Brasil CEP: 57.072-970 | Telefone: (082) 3214-1401

Membros da Comissão Julgadora da Dissertação de Mestrado de Daniel de Melo Pimentel, intitulada: *"Uma Proposta para Aprimorar o Anonimato em Transações Bitcoin com Suporte à Auditoria"*, apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas em 19 de maio de 2017, às 14h00min, na Sala de Reuniões do Instituto de Computação da UFAL.

COMISSÃO JULGADORA

Leandro Melo de Sales

Prof. Dr. Leandro Melo de Sales
Programa de Pós-graduação em Informática – UFAL
Orientador

Rafael de Amorim Silva

Prof. Dr. Rafael de Amorim Silva
Programa de Pós-graduação em Informática – UFAL
Examinador

Aydano Pamponet Machado

Prof. Dr. Aydano Pamponet Machado
Programa de Pós-graduação em Modelagem Computacional de Conhecimento – UFAL
Examinador

Agradecimentos

Agradeço primeiramente aos meus pais, Maria e Sebastião Pimentel, pelo amor incondicional, apoio e dedicação. Agradeço também aos meus irmãos, Paulo e Lilian Pimentel, por todo apoio.

Ao meu orientador, Prof. Dr. Leandro Sales por suas orientações. E ao Prof. Dr. Thiago Sales por compartilhar seus conhecimentos sobre criptografia. Aos Prof. Dr. Roberta Lopes e Francisco Barros por compartilhar seus conhecimentos matemáticos. E aos Prof. Dr. Rafael Amorim e Prof. Dr. Aydano Pamponet pelas suas contribuições em minha vida acadêmica.

Agradeço a todos os *hackers* das comunidades de software livre, em especial ao Richard Stallman, que faz do mundo um lugar melhor para todos através do software livre.

Por fim, agradeço a todos os meus amigos que me ajudaram, certamente este trabalho não teria sido concluído sem a ajuda e incentivo de vocês.

Resumo

Neste trabalho, apresenta-se uma avaliação de desempenho da aplicação da técnica de Assinatura de Grupo no sistema *Bitcoin* e um estudo sobre anonimato e auditoria. O *Bitcoin* tem como objetivo prover uma moeda virtual e um sistema de transação *online* anônima. Todavia, pesquisas recentes relatam que é possível quebrar o anonimato das transações *Bitcoin*, por meio de técnicas de rastreabilidade cronológica nas transações *Bitcoin* e da análise dos endereços de rede. Como consequência da quebra de anonimato das transações *Bitcoin*, a privacidade dos usuários e todo ecossistema *Bitcoin* são afetados negativamente. Por este motivo, neste trabalho, propõe-se a inclusão de técnicas de Assinaturas de Grupos no sistema *Bitcoin* para aumentar o anonimato nesse sistema, porém em tempo hábil, cerca de 10 minutos. A técnica de Assinatura de Grupo gera diversos grupos distintos para diversas transações *Bitcoin*, dificultando assim a rastreabilidade dessas transações. Após a implementação dessa técnica em uma versão modificada do sistema *Bitcoin*, avaliou-se tal proposta por meio de experimentos executados em simulações. Através dos resultados dos experimentos e análises estatísticas, constatou-se que a abordagem com a inclusão da Assinatura de Grupo é viável para implantação no sistema *Bitcoin* desde que os grupos sejam pequenos, cerca de 500 clientes. Para todos os casos avaliados, verificou-se que com o uso da Assinatura de Grupo as transações *Bitcoin* se tornam anônimas. Porém, avaliou-se que o desempenho das transações *Bitcoin* com a técnica de Assinatura de Grupo nesse sistema tem lentidão aproximada de 50% em comparação com as transações *Bitcoin* sem essa técnica. Além disso, em grupos pequenos com Assinatura de Grupo, obtêm-se um elevado nível de anonimato e permite-se auditoria, porém em grupos grandes com aproximadamente mais de 500 clientes essa técnica não se mostra tão eficiente, pois as transações começam a exceder o tempo hábil.

Palavras-chaves: *Bitcoin*. Assinatura de Grupo. Desempenho. Anonimato. Auditoria.

Abstract

This work show an avaliation about the performance of the application of Group Signature technique in Bitcoin system and a study about the anonymity and auditory. The Bitcoin' goal is provide a virtual currency and an anonymous online transaction system. However, recent researches show that can be to break the anonymity of transactions through the chronological traceability technique in the Bitcoin transactions and analysis of network addresses. As a result of breaking anonymity of Bitcoin transactions, users' privacy and all Bitcoin ecosystem are affected negatively. For this reason, in this work, it was proposed to include Group Signatures techniques in the Bitcoin system to increasing of anonymity in Bitcoin transactions but with audity possibility in accept time, more or less 10 minutes. The Group Signature generate a lot of dinstinct groups to dinstinct Bitcoin transactions. After to include the Group Signature technique in a modified version of Bitcoin system, we evaluated this technique in Bitcoin system through experiment in simulations. Through the experiments and statistic analysis, it was found that the approach of including the Group Signature is feasible for implementation in Bitcoin system with small groups, 500 clients. For the all cases analyzed, it was verified that the use of Group Signature in Bitcoin transactions increase the anonymity. Therefore, it was verified that the performance of Bitcoin transactions with Group Signature technique show a delay in nearly 50% less than current Bitcoin system whitout this technique. Nevertheless, in small groups with Group Signature get a better anonymity level and audity, but in big groups with more than 500 clients this technique not is good because the transactions time is over.

Keywords: Bitcoin. Group Signature. Performance. Anonymity. Audity.

Sumário

Lista de ilustrações

Lista de tabelas

Lista de códigos fonte

Lista de abreviaturas e siglas

ASIC	*Application Specific Integrated Circuits*
API	*Application Programming Interface*
AWK	*Aho, Weinberg e Kernighan*
BASH	*Born Again Shell*
BIP	*Bitcoin Improvement Proposal*
BTC	*Bitcoin*
CPU	*Central Processing Unit*
CSV	*Comma-Separated Values*
DoS	*Denied of Service*
ECDSA	*Elliptic Curve Digital Signature Algorithm*
GNU	*GNU is Not Unix*
GPL	*GNU Public License*
GuixSD	*Guix System Distribution*
HTML	*HyperText Markup Language*
IP	*Internet Protocol*
JSON	*JavaScript Object Notation*
MIT	*Massachusets Institute of Technology License*
0MQ	*Zero-eM-Queue*
ONU	Organização das Nações Unidas
P2P	*Peer-to-Peer*
POW	*Proof-Of-Work*
RCP	*Rate Control Protocol*
RPC	*Remote Procedure Call*

SHA	*Secure Hash Algorithm*
TCP	*Transport Control Protocol*
URL	*Uniform Resource Locators*
UDP	*User Datagram Protocol*

1 Introdução

A demanda dos usuários por métodos de compras e pagamentos seguros através da *Internet* tem crescido a cada ano. O crescimento das transações *online* na *Internet* é resultado do aumento da facilidade do acesso residencial à banda larga e do acesso sem fio, além das facilidades e comodidades que os métodos de compras e pagamentos virtuais oferecem. Essa popularização resulta em um aumento da demanda para os sistemas de transações *online* seguras, elevando a complexidade relacionada a segurança e a privacidade dos usuários nesse tipo de transação.

Nesse contexto, os desafios relacionados à segurança e privacidade dos usuários em transações virtuais de forma segura necessitam ser resolvidos (HEWETT; KIJSANAYOTHIN, 2009). Por isto, faz-se necessário pesquisar e propor novas soluções relacionadas ao anonimato em transações eletrônicas a fim de proporcionar aos usuários compras e pagamentos seguros através da *Internet*.

Atualmente, os sistemas de transações financeiras na *Internet* vêem sofrendo uma mudança radical em seu paradigma. A maior mudança que ocorreu nos últimos anos nas transações monetárias virtuais foi a criação de um novo sistema eletrônico de pagamento e moeda virtual que garante o anonimato denominada *Bitcoin* (BTC) (NAKAMOTO, 2017a).

A disseminação da moeda virtual *Bitcoin* tem levado à mudanças relacionadas aos métodos de como são realizadas as transações *online*. Como consequência, diversos pesquisadores buscaram validar o nível de anonimato do sistema *Bitcoin* e apontaram falhas relacionadas ao anonimato. As transações do sistema *Bitcoin* podem ser rastreadas e há a possibilidade de revelar a identidade de alguns usuários desse sistema.

1.1 Problemática

Apesar de usar alguns mecanismos de segurança, o sistema *Bitcoin* armazena publicamente todas as transações através de uma grande base de dados descentralizada chamada *blockchain*. Algumas informações sensíveis dos usuários do sistema *Bitcoin* como os identificadores de suas carteiras virtuais, endereços IP (*Internet Protocol*) e fusos horários são visíveis por todos os usuários desse sistema (BRADBURY, 2015). Alguns autores analisaram informações oferecidas pela *blockchain* e chegaram a conclusão que é possível extrair algumas informações dessa base de dados que podem ser utilizadas para quebrar o anonimato das transações *Bitcoin*.

Biryukov et al. em seu experimento afirmam que aproximadamente 11% dos IPs dos usuários *Bitcoin* são rastreáveis. Os autores constataram ainda que cerca de 60% dos usuários não usam nenhum tipo de serviço de *proxy* para ocultar o seu endereço IP. Sem *proxy*, facilita-se que usuários maliciosos possam obter os IPs de usuários autênticos do sistema *Bitcoin* (BIRYUKOV; KHOVRATOVICH; PUSTOGAROV, 2014). Dupont e Squicciarini conseguiram determinar a localização física de alguns usuários *Bitcoin* correlacionando 11 fusos horários desses usuários com perfis de um determinado *forum Bitcoin*, *Bitcointalk*. Os autores conseguiram 72% de ocorrência que revelam as possíveis localizações desses usuários (DUPONT; SQUICCIARINI, 2015).

Diante desse cenário, observa-se que o *Bitcoin* ainda possui fragilidades no que diz respeito ao anonimato dos seus usuários. A privacidade dos usuários do sistema *Bitcoin* está diretamente relacionada com o anonimato desse sistema.

A privacidade dos usuários *Bitcoin* é importante pois operações financeiras *online* devem ser realizadas em sigilo a fim de garantir a segurança dos usuários, diminuir crimes cibernéticos, roubo de identidade e furto das carteiras virtuais. Isto significa que há necessidade de realizar pesquisas para aprimorar o mecanismo de anonimato do sistema *Bitcoin*, mas sem afetar o desempenho das transações desse sistema.

Atualmente, uma transação *Bitcoin* deve ser finalizada em no máximo dez minutos, logo se uma transação exceder esse tempo esta se torna inviabilizada pelo sistema. Além disso ao resolver o problema de anonimato no *Bitcoin*, cria-se outro problema vinculado à auditoria. Sem auditoria é possível realizar fraudes e crimes como lavagem de dinheiro, portanto é fundamental que exista algum tipo de mecanismo que permita realizar transações anônimas em tempo máximo de dez minutos, e que ainda possibilite efetuar auditoria a fim de aumentar a confiabilidade e aceitação da moeda virtual *Bitcoin*.

1.2 Objetivos

O objetivo deste trabalho é aprimorar o sistema de anonimato com suporte à auditoria em sistema *Bitcoin*.

1.2.1 Objetivos específicos

- Implementar uma versão do sistema *Bitcoin* modificada com suporte à técnica de Assinatura de Grupo para melhorar o anonimato em transações *Bitcoin*.
- Estudar o desempenho do *Bitcoin* em comparação à versão atual, com foco nas questões de privacidade e auditoria.

1.3 Metodologia

O presente trabalho de pesquisa usou uma abordagem quantitativa. Portanto, avaliou-se o desempenho da técnica de Assinatura de Grupo no sistema *Bitcoin* por meio de simulações e quantificaram-se os resultados a fim de comparar com o atual sistema *Bitcoin*. Além disso, utilizou-se de formalizações para comprovar a veracidade do anonimato e auditoria através do uso da Assinatura de Grupo no sistema *Bitcoin*.

1.4 Relevância

Em 2011, havia aproximadamente 60 mil usuários *Bitcoin*, porém esse número cresceu para cerca de 33 milhões em meados de 2014 (GERVAIS et al., 2014). Com o crescimento dos usuários, consequentemente as transações também aumentaram em proporções semelhantes (SMITH, 2017). Quanto mais transações existir mais valorizado é a cripto-moeda *Bitcoin*, pois o seu valor é baseado na oferta e procura. Portanto, é de suma importância permitir que transações *Bitcoin* continuem sendo realizadas e que novos clientes *Bitcoin* transacionem a fim de valorizar ainda mais essa cripto-moeda.

Baseado no crescimento das transações *Bitcoin*, é vital garantir um elevado nível de anonimato para os seus usuários a fim de garantir segurança e privacidade em transações *online*. Todavia, as transações *Bitcoin* devem ser resolvidas em tempo hábil e serem totalmente anônimas. Sabendo-se que um dos pilares do sistema *Bitcoin* é garantir o anonimato em transações monetárias, qualquer fragilidade no sistema de anonimato dessa cripto-moeda pode inviabilizar a segurança dos seus usuários, impactando negativamente na viabilidade e crescimento do *Bitcoin*.

Elevando-se o anonimato também diminui-se as chances de realizar auditoria. Auditoria é outro fator relevante, pois muitas empresas, principalmente as empresas públicas, precisam permitir que se realize auditoria a fim conter fraudes e garantir que os recursos financeiros foram realmente direcionados de forma correta. Portanto, permitir auditoria pode aumentar a confiabilidade das empresas em relação ao *Bitcoin* e consequentemente aumentar o número de transações, que valoriza e fortalece o ecossistema *Bitcoin*.

Deste modo, estudar a aplicabilidade da técnica de Assinatura de Grupo no sistema *Bitcoin* com o foco em melhorar o seu sistema de anonimato e garantir auditoria com desempenho satisfatório, torna o trabalho relevante em termos práticos. O *Bitcoin* é a primeira cripto-moeda a surgir, a mais usada no mundo e a que possui o maior valor monetário agregado atualmente. Além disso, o *Bitcoin* é realidade na indústria, academia, mercado financeiro e tem sido adotada em alguns países (WALLACE, 2011). Portanto, resolver problemas relativos ao anonimato e permitir auditoria em tempo hábil pode trazer maior confiabilidade dos usuários e empresas no *Bitcoin*.

1.5 Estrutura do Documento

O restante deste documento está organizado da seguinte forma:

- no Capítulo 2, apresentam-se os principais conceitos relacionados ao sistema *Bitcoin*;
- no Capítulo 3, apresenta-se um estudo da aplicabilidade da Assinatura de Grupo no sistema *Bitcoin* para aumentar o anonimato e auditoria;
- no Capítulo 4, apresentam-se as análises dos experimentos, os detalhes estatísticos realizados e os resultados obtidos sobre desempenho da implementação *Anonycoin*;
- no Capítulo 5, apresentam-se os trabalhos relacionados e um comparativo com a abordagem *Anonycoin*;
- no Capítulo 6, apresentam-se as conclusões e trabalhos futuros;
- no Apêndice A, apresentam-se detalhes a respeitos das abstrações do sistema *Bitcoin* realizadas no *Anonycoin*;
- no Anexo A, apresentam-se o código fonte completo da simulação *Anonycoin*;
- no Anexo B, apresentam-se detalhes sobre o Algoritmo de Assinatura Digital de Curva Elíptica ou *Elliptic Curve Digital Signature Algorithm* (ECDSA) usado pelo *Bitcoin* e pela simulação *Anonycoin*;
- no Anexo C, apresentam-se detalhes do Algoritmo de *Hash* Seguro ou *Secure Hash Algorithm* (SHA) usado pelo *Bitcoin* e pela simulação *Anonycoin*.

2 Fundamentação Teórica

Este capítulo apresenta as principais estruturas do sistema *Bitcoin* e seus elementos fundamentais.

2.1 *Bitcoin*

Bitcoin é uma moeda virtual que foi criada para permitir transações *online* rápidas e seguras de forma anônima. O anonimato do *Bitcoin* é alcançado por meio de criptografia, portanto, é comum ouvir o termo cripto-moeda que faz referência a uma moeda virtual criptografada. Além do uso de criptografia, o *Bitcoin* usa uma arquitetura de rede descentralizada do tipo *Peer-to-Peer* (P2P). A rede P2P do *Bitcoin* permite realizar transações anônimas descentralizadas de forma segura, garantindo assim a privacidade dos usuários (HOBSON, 2013).

O *Bitcoin* foi desenvolvido por um programador misterioso que usa o pseudonônimo de Satoshi Nakamoto, atualmente (2017), esse programador continua anônimo. O *Bitcoin* é a primeira cripto-moeda a surgir, sendo a mais promissora dentre todas as cripto-moeda nos dias atuais e é fonte de inspiração para o desenvolvimento de novas cripto-moedas. O *Bitcoin* possui o maior número de usuários e transações entre todas as cripto-moedas, além de ser a moeda virtual mais valorizada nos dias atuais (SMITH, 2017).

Na Figura 1, ilustram-se a história do *Bitcoin* e os seus principais elementos.

Figura 1 – História e principais elementos do *Bitcoin* (Imagem adaptada de (EVANS-PUGHE; NOVIKOV; VITALIEV, 2014)).

2.2 *Wallet*

As carteiras virtuais ou *wallets* são *softwares* clientes que possuem um par de chaves criptográficas, chave privada e chave pública, que usam o algoritmo ECDSA (Anexo B). A partir da chave pública é gerado uma sequência de caracteres criptografados chamada de *hash* que representa essa chave. O *Bitcoin* usa uma função do tipo *Secure Hash Algorithm* (SHA-512) para gerar o *hash* da chave pública (Anexo C).

Na Figura 2, observa-se a sequência dos principais serviços criados pelas *wallets*.

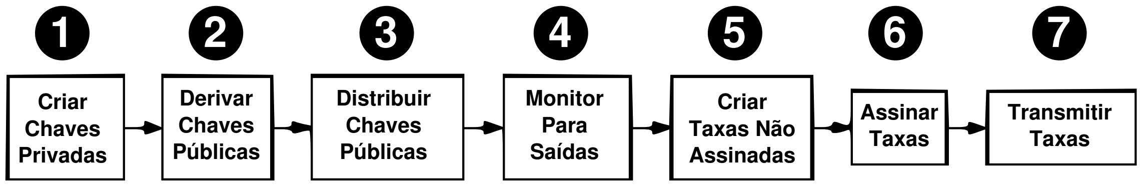

Figura 2 – Fluxos de serviços criados pelas *Wallets* (Imagem adaptada de (NAKAMOTO, 2017b)).

1. as *wallets* geram as chaves privadas usando o algoritmo ECDSA;
2. a partir das chaves privadas são derivadas as chaves públicas e os seus respectivos *hashes* do tipo SHA-512;
3. as chaves públicas são distribuídas entre os participantes de uma determinada transação *Bitcoin*;
4. a rede *Bitcoin* monitora as transações a fim de serem validadas pelos mineradores;
5. são gerados incentivos ou taxas para cada transação mas não assinadas pois não foram validadas pelos mineradores;
6. após as validações dos mineradores, taxas são assinadas e aplicadas as transações validadas com sucesso;
7. as taxas são distribuídas para o minerador da transação *Bitcoin* como forma de incentivo pelo esforço computacional.

As *Wallets* podem enviar ou receber *bitcoins*, além de armazenar todas as transações já realizadas pelo sistema *Bitcoin* por meio da rede P2P. Atualmente, existem *wallets offline* e *wallets online*, como pode ser visto na Tabela 1.

Tabela 1 – *Wallets online* e *offline* (Tabela adaptada de (HURLBURT; BOJANOVA, 2014)).

Ambiente	***Wallets***
Windows, Mac, Linux	*MultiBit, BitcoinQT, Armory, Electrum, Hive*
Android	*BitcoinWallet, coinbase.com*
iOS	*coinbase.com*
QR code scan	*BitcoinWallet*
SMS	*coinbase.com*
Web browsers	*coinbase.com, blockchain.info*

As transações entre *wallets* mantém um histórico cronológico, transação de entrada e transação de saída, onde a atual transação faz referência a transação anterior a fim de evitar fraudes virtuais, como pode ser observado na Figura 3.

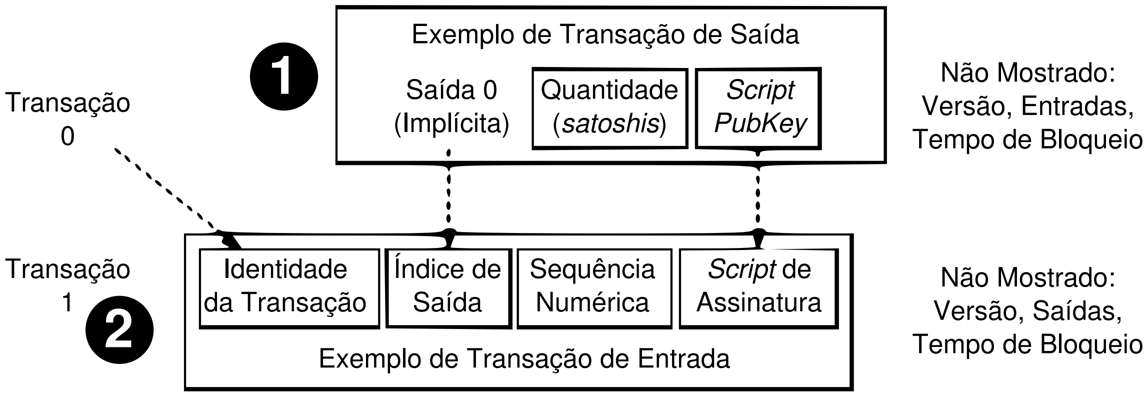

Figura 3 – Exemplo de transação *Bitcoin* de entrada e saída (Imagem adaptada de (NAKAMOTO, 2017b)).

1. a Transação 0 representa a Transação de Saída, ou a transação anterior. Na *blockchain* todas as transações são ligadas entre si formando um histórico de transações sequenciais;

2. a Transação 1 representa a Transação de Entrada, ou a transação atual, onde o Identificador dessa transação é ligado ao da transação anterior, assim como o seu Índice de Saída e o *Script* de Assinatura.

Todas as transações são validadas por uma linguagem desenvolvida especificamente para o *Bitcoin*, denominada *Script* (EVANS-PUGHE; NOVIKOV; VITALIEV, 2014). Essa linguagem usa a criptografia ECDSA e não tem suporte a *loop* ou recursividade a fim de evitar que o sistema *Bitcoin* trave ou sofra ataques que possam gerar falhas de segurança.

2.3 *Mining*

Os mineradores *bitcoin* ou *miners* adicionam blocos para uma grande base de dados pública denominada *blockchain*, criando um histórico de transações.

Quando um determinado *miner* valida uma transação Bitcoin por meio da linguagem *Script*, em um tempo médio de 10 minutos, gera-se um novo bloco válido e doa-se 25 *bitcoins* para esse *miner* como forma de incentivo.

As validações de uma transação *Bitcoin* ocorre através da verificação na *blockchain*, verificando se o BTC a ser transacionado nunca foi usado e a autenticidade dos *hashes* dessa transação. Essas validações são chamadas de *Proof-of-Work* (PoW), e são necessárias para garantir a integridade das transações e evitar ataques como o o gasto duplo (*double-spend*). O *double-spend* segmenta a rede *Bitcoin*, e faz com que uma moeda já criada anteriormente possa ser criada novamente, criando um problema de integridade e de confiabilidade pois cada BTC é único.

Atualmente os recursos computacionais dos computadores pessoais e dispositivos móveis para realizar o PoW é insuficiente. Entretanto, além da mineração tradicional de *bitcoin*, denominada *solo miner*, há a mineração compartilhada chamada de *pool miner* (TAYLOR, 2014). Pode ser observado esses dois tipos de mineração nas Figuras 4 e 5, respectivamente.

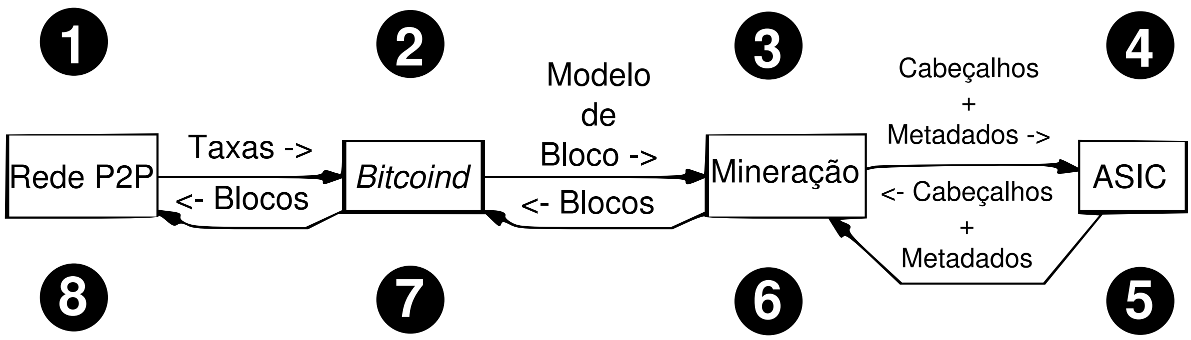

Figura 4 – Fluxo de trabalho de mineração *Bitcoin* do tipo *Solo Mining* (Imagem adaptada de (NAKAMOTO, 2017b)).

1. as *Wallets* realizam transações que são propagadas na rede P2P *Bitcoin* em conjunto com as taxas, que são incentivos para os mineradores, para o serviço *Bitcoind*;

2. o *Bitcoind*, um serviço que é executado em segundo plano (*background* ou *daemon*), define os modelos de blocos a serem minerados e os envia através do protocolo *Remote Procedure Call* (RPC) para o *Software* de Mineração;

3. o *Software* de Mineração recebe os Cabeçalhos dos blocos e os Metadados a serem processados em plataformas especificas denominadas *Application Specific Integrated Circuits* (ASIC);

4. a plataforma ASIC inicia o processamento do PoW e valida as transações;

5. a plataforma ASIC retorna para o *Software* de Mineração os Cabeçalhos e os Metadados;

6. o *Software* de Mineração recebe os Cabeçalhos e os Metadados e envia os Blocos para o *Bitcoind*;

7. o *Bitcoind* repassa os Blocos, incluindo a primeira transação conhecida como *coinbase*, as taxas e os incentivos (25 BTC), já validados para a rede P2P. Esses blocos serão propagados para todos os clientes *Bitcoin* formando a *blockchain*. Os blocos não válidos serão descartados.

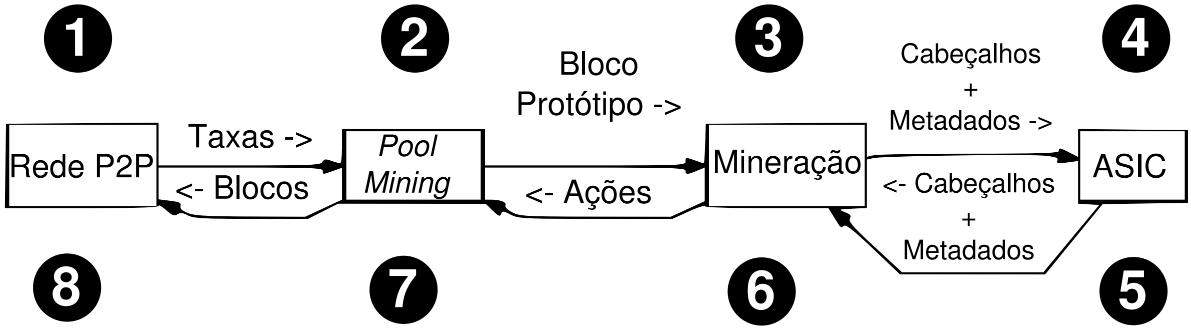

Figura 5 – Fluxo de trabalho de mineração *Bitcoin* do tipo *Pool Mining* (Imagem adaptada de (NAKAMOTO, 2017b)).

1. as *Wallets* realizam transações que são propagadas na rede P2P *Bitcoin* com as respectivas Taxas para o serviço *Pool Mining*;
2. o *Pool Mining*, serviço que é executado em forma de *daemon* usando o *Bitcoind*, definem os Blocos Protótipos a serem minerados. Esses blocos são enviados para os *Softwares* de Mineração compartilhadas;
3. os *Softwares* de Mineração recebem os Blocos Protótipos e enviam os Cabeçalhos dos blocos e Metadados para serem processados nas plataformas ASICs;
4. as plataformas ASICs iniciam o processamento do PoW e valida as transações;
5. as plataformas ASICs retornam os Cabeçalhos e os Metadados para os *Softwares* de Mineração;
6. os *Softwares* de Mineração recebem os Cabeçalhos e os Metadados e os repassam para o *Pool Mining* Compartilhados;
7. o *Pool Mining* repassa os Blocos, incluindo o *coinbase*, as Taxas e os incentivos (25 BTC) que serão compartilhados, já validados para a rede P2P. Esses blocos serão propagados para todos os clientes *Bitcoin* formando a *blockchain*. Os blocos não válidos serão descartados.

Foram desenvolvidos dispositivos específicos para mineração de *Bitcoin* usando a arquitetura ASIC, porém os custos são bastantes elevados e a compra é feita somente por empresas através de encomendas de grandes lotes diretamente aos fabricantes.

Estipulou-se que haverá em torno de 21 milhões de *bitcoins* a serem minerados, após o último *bitcoin* minerado a funcionalidade do minerador continua a validar as transações mas sem criar novos BTCs (HURLBURT; BOJANOVA, 2014). As taxas de recebimento ao minerar um bloco são reduzidas pela metade a cada 4 anos, logo a taxa de mineração que atualmente é de 25 BTC será reduzida para 12,5 BTC e assim sucessivamente.

2.4 *Blockchain*

A *blockchain* prover uma grande base de dados que comporta todas as transações *Bitcoin*, essas transações são públicas e acessíveis por qualquer pessoa (BRADBURY, 2015).

Essa grande base de dados armazena o histórico de todas as transações *Bitcoin* válidas que já foram concluídas com sucesso até os dias de hoje. Todas essas transações são ligadas entre si a fim de evitar modificações ou fraudes.

Na Figura 6, observa-se como acontecem as ligações entre as transações *Bitcoin* na *blockchain*.

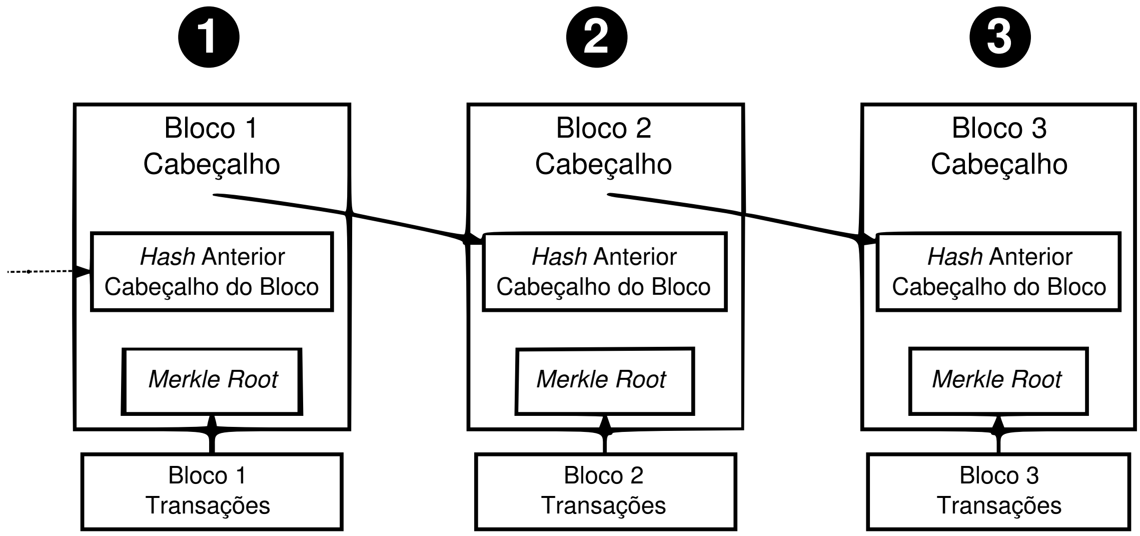

Figura 6 – *Blockchain Bitcoin* simplificado (Imagem adaptada de (NAKAMOTO, 2017b)).

1. o Bloco 1 contém o *hash* do bloco anterior, o *Merkle Root* que compacta todas as transações pertencentes a esse bloco;

2. o Bloco 2 contém uma ligação com o Bloco 1 através do *hash* do bloco anterior;

3. o Bloco 3 contém uma ligação com o Bloco 2 através do *hash* do bloco anterior. Esse esquema segue nos demais blocos que compõem a *blockchain*.

Cada bloco pode conter uma ou mais transações *Bitcoin*, sendo que a primeira transação de um novo bloco gerado chama-se de transação gênesis ou *coinbase*.

O sistema *Bitcoin* verifica todas as transações pertencentes a um determinado bloco da *blockchain* usando o esquema chamado de *merkle tree*. Nesse esquema, os *hashes* das transações compõem as ramificações de uma árvore binária onde a sua raiz, chamada de *Merkle Root*, é o *hash* que representa o bloco (ALAJEELY; AHMAD; DOSS, 2015).

3 Uma proposta para aprimorar o anonimato em transações *Bitcoin* com suporte à Auditoria

O presente trabalho de pesquisa foca no desempenho, anonimato e auditoria da moeda virtual *Bitcoin*. No primeiro caso, é de vital importância verificar se o desempenho da técnica de Assinatura de Grupo não irá apresentar custos relativos ao tempo das transações *Bitcoin*, impossibilitando a sua implementação em ambiente de produção. No segundo caso, é fundamental analisar o grau de anonimato oferecido pela solução proposta no sistema *Bitcoin*. O anonimato é a base para o sistema *Bitcoin* e é primordial para os usuários que lidam com dados financeiros. Por fim, no terceiro caso, faz-se necessário permitir auditoria no *Bitcoin*, mesmo com um elevado nível de anonimato a fim de permitir maior confiabilidade e adoção dessa moeda virtual.

Após pesquisas, observou-se que não havia nenhum simulador de transações *Bitcoin* disponíveis no estado da arte que permitisse incluir a técnica de Assinatura de Grupo e comparar o desempenho das transações *Bitcoin* com e sem essa técnica. Logo, desenvolveu-se o *Anonycoin* a fim de simular e medir o desempenho das transações *Bitcoin* com e sem a inclusão da técnica de Assinatura de Grupo, verificar o anonimato e permitir auditoria por meio dessa técnica.

3.1 Aplicabilidade da Assinatura de Grupo em Sistemas *Bitcoin*

A técnica de Assinatura de Grupo é composta por pares de chaves criptográficas, chave pública do grupo e chave privada de gerenciamento do grupo. A chave pública do grupo é compartilhada entre os seus participantes, já a chave privada de gerenciamento do grupo deve ser guardada em segurança por um membro confiável desse grupo (LIBERT; PETERS; YUNG, 2012).

Especificamente, em uma pesquisa recente realizada por Sales et al., desenvolveu-se um protocolo para prover autenticação e detecção de ataques *Sybil* em redes *Ad Hoc* veiculares com suporte ao controle de anonimato dos usuários usando a técnica de Assinaturas de Grupos (SALES, 2015). A técnica aplicada na referida pesquisa se mostrou bastante promissora, podendo-se aplicar em outros contextos, como no *Bitcoin*.

No contexto desse trabalho, o minerador definido na Seção 2.3 é quem gera o grupo. O minerador é responsável por validar uma transação e portanto uma parte confiável em uma transação *Bitcoin*. A fim de exemplificar a inclusão da técnica de Assinatura de

Grupo em uma transação *Bitcoin*, ilustra-se na Figura 7 essa implementação em partes do processo de uma transação *Bitcoin*.

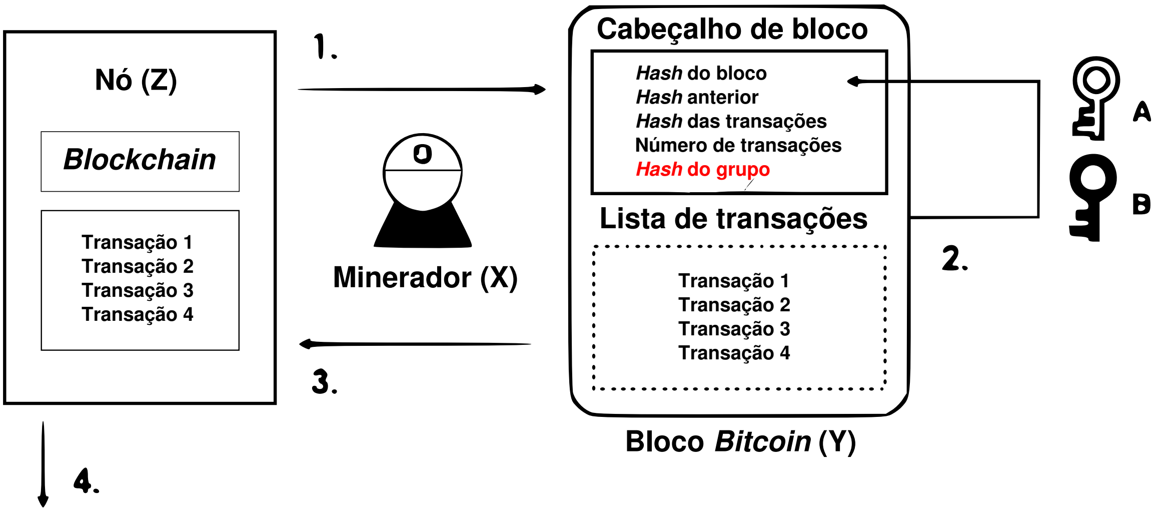

Figura 7 – Aplicação da Assinatura de Grupo em transações *Bitcoin* (Imagem adaptada de (BARRERA, 2017)).

1. duas *wallets* no nó Z iniciam uma determinada transação *Bitcoin* trocando suas chaves públicas criptográficas;
2. no bloco *Bitcoin* Y, o Minerador X cria um grupo gerando a chave pública do grupo (chave A) e a chave privada de gerenciamento do grupo (chave B);
3. cada transação é assinada por uma mensagem gerada a partir da combinação das chaves públicas das *wallets* participantes de uma transação. O Minerador X valida a transação e os BTCs transacionados através do PoW, a chave pública do grupo substitui a chave pública da *wallet*;
4. as transações realizadas com sucesso são adicionadas a *blockchain* e todas as *wallets* conectadas na rede *Bitcoin* P2P são atualizadas.

Com a inclusão da técnica de Assinatura de Grupo, as transações *Bitcoin* continuam públicas através da *blockchain* e da sua rede P2P. Porém, as informações sensíveis dos usuários, como as chaves públicas das *wallets* são criptografadas com a Assinatura de Grupo. Na versão atual do *Bitcoin*, a disponibilidade dessas informações pode comprometer o anonimato dos usuários.

Portanto, com a inclusão da técnica de Assinatura de Grupo, dificulta-se o rastreio das *wallets* e suas transações por meio de pesquisas cronológicas na *blockchain*. Com essa técnica, aumenta-se a dificuldade em realizar ligações entre as chaves públicas e os endereços IPs na rede *Bitcoin* P2P com *sites, foruns* e outras fontes de informações do usuário na Internet.

Diferente de outras abordagens com o mesmo intuito de elevar o anonimato no sistema *Bitcoin*, a inclusão da técnica de Assinatura de Grupo não modifica a arquitetura do sistema *Bitcoin*. Isto porque a técnica aqui proposta altera apenas estruturas que já existem na arquitetura do sistema *Bitcoin* como as chaves públicas das *wallets*, incluindo assim a chave pública do grupo. Nesse contexto, a seguir, apresenta-se alguns critérios relevantes em relação a aplicação da técnica de Assinatura de Grupo no sistema *Bitcoin* em comparação com outras abordagens no estado da arte:

- aumentar o anonimato do *Bitcoin*;
- manter o desempenho das transações *Bitcoin* dentro do aceitável;
- permitir auditoria no *Bitcoin*;
- reusar algoritmos do *Bitcoin*;
- reusar a infra-estrutura *Bitcoin*;
- não interferir nos processos de validação do *Bitcoin*;
- não alterar a arquitetura do *Bitcoin*.

Na Seção 3.2, apresentam-se as formalizações sobre a Assinatura de Grupo no sistema *Bitcoin* em relação ao anonimato e auditoria.

3.2 Anonimato e Auditoria

O processo de validação do anonimato e a verificação da auditoria foi realizado através do desenvolvimento de um modelo que define um grupo e todo o seu processo. Na Figura 8, desenvolveu-se uma representação gráfica do ciclo de vida completo de uma transação *Bitcoin* simplificada com a inclusão da Assinatura de Grupo para uma visão macro de todos os processos em relação ao anonimato e auditoria. A simplificação desse processo apresenta apenas quatro usuários no gráfico em um grupo, porém, na prática, esses grupos podem conter centenas de usuários. Isto é muito importante para criar aleatoriedade ao sistema e assim garantir o anonimato. Neste sentido, é importante avaliar o desempenho dessas combinações e, por isto, estudos acerca deste cenários serão apresentados e discutidos no Capítulo 4. Usou-se Teoria de Conjuntos para melhor definir um modelo que represente matematicamente os elementos gráficos (WEISS, 2017).

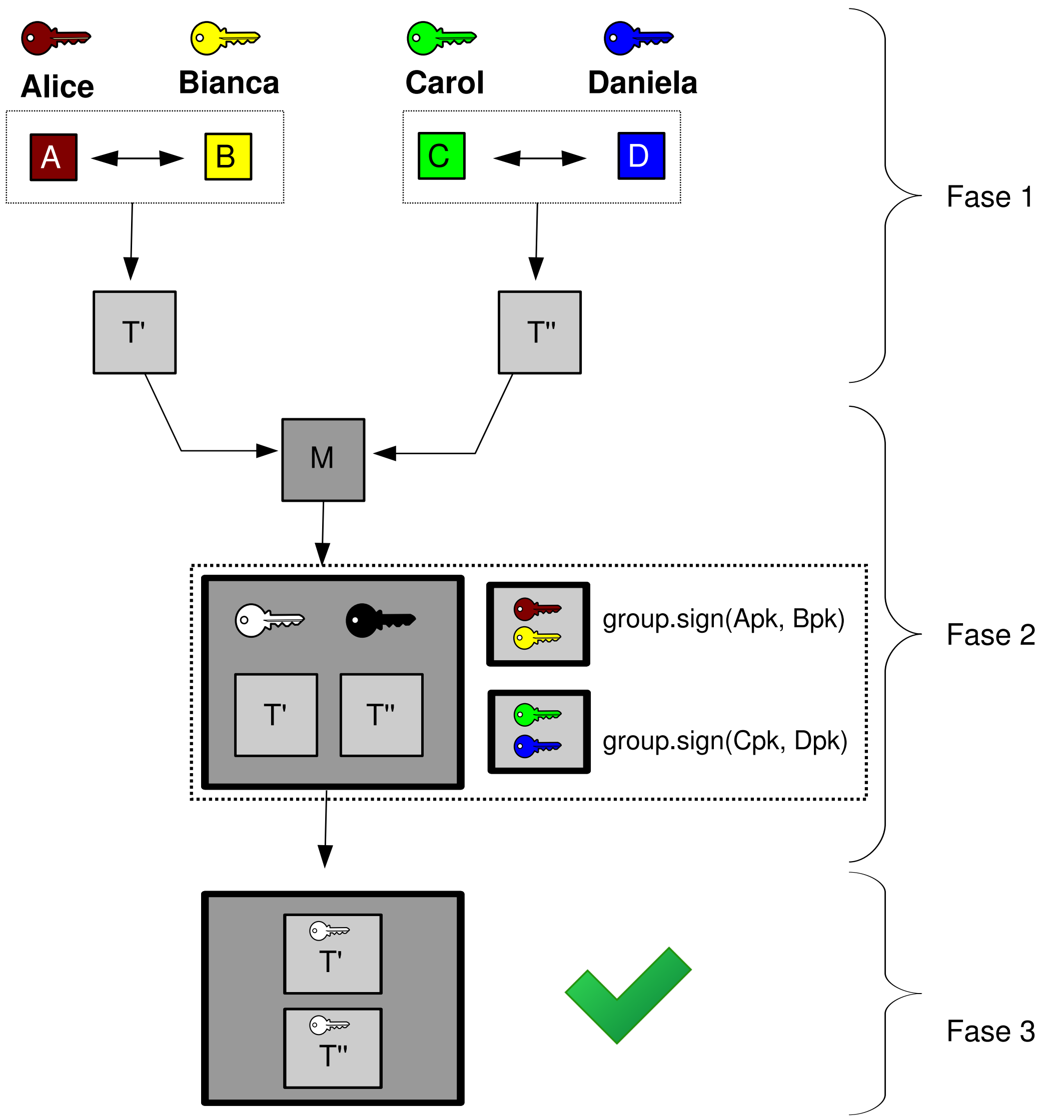

Figura 8 – Representação gráfica completa da inclusão da técnica de Assinatura de Grupo no sistema *Bitcoin*.

1. na Fase 1, os usuários Alice, Bianca, Carol e Daniela transacionam entre si uma determinada quantia de BTC. Esses usuários trocam suas chaves públicas que posteriormente geram transações conforme definido formalmente a seguir:

   a) seja o conjunto finito das chaves privadas, definido por $S = \{s_1, s_2, ..., s_x\}$, tal que $x \in \mathbb{N}^*$;

   b) seja o conjunto finito das chaves públicas, definido por $P = \{p_1, p_2, ..., p_y\}$, tal que $y \in \mathbb{N}^*$;

   c) portanto, o conjunto finito das chaves criptográficas definido por $K = F_\pi(s_x, p_y)$, onde $s_x \in S$ e $p_y \in S$, tal que $F_\pi : P \times S \rightarrow K$. Logo, $F_\pi(x, y) = x + y$ é um polinômio de segunda ordem do tipo $\rho x^2 + \sigma y^2 + \tau$, onde $\rho \neq \sigma \neq \tau$, e $\rho, \sigma, \tau \in Primos$ a fim de garantir a autenticidade e unicidade das chaves. Sabendo que a função inversa F_π^{-1} define o segredo para resolver a criptografia definido na função anterior;

d) seja o conjunto finito dos usuários *Bitcoin* definidos por $U = \{u_1, u_2, ..., u_c\}$, tal que $c \in \mathbb{N}^*$;

e) seja o conjunto finito de transações, definido por $T = U \times U$, tal que $U \times U = \{(u_c, u_d)|c \neq d\}$ com $d \in \mathbb{N}^*$. Sabendo que os usuários u_c e η_d possuem chaves criptográficas, logo $(u_c, \eta_d) \leftrightarrow (k_\upsilon, k_\varphi) \in K$ define os pares de chaves criptográficas para cada usuário.

2. na Fase 2, o minerador valida as transações e cria um grupo, gerando assim um par de chaves (chaves privada de gerenciamento do grupo, na cor preta, e chave pública do grupo, na cor branca). As transações participam de um grupo que posteriormente são assinadas, conforme define-se a seguir:

   a) seja o conjunto finito dos mineradores, representado por $M = \{m_1, m_2, ..., m_e\}$ tal que $e \in \mathbb{N}^*$;

   b) seja o conjunto finito dos grupos, representado por $G = \{g_1, g_2, ..., g_f\}$ tal que $f \in \mathbb{N}^*$, de modo que g_f é gerado a partir de $m_h \in M$;

   c) a função que mapeia as transações para o minerador é definida pelo *powerset* conforme $F_\phi : M \to \mathfrak{p}(T)$;

   d) seja $F_h : T \to G$, uma função que denota as transações $t_i \in T$ conectados a um grupo $g_f \in G$, de modo que nenhuma transação t_i pode estar relacionado com dois ou mais grupos g_f;

   e) a função que mapeia as transações em um grupo é definida pelo *powerset* segundo $F_\chi : G \to \mathfrak{p}(T)$;

   f) seja o conjunto finito das chaves criptográficas da Assinatura de Grupo, definido por $A = \{a_1, a_2, ..., a_q\}$ tal que $q \in \mathbb{N}^*$;

   g) portanto, a função $F_t : G \to A$ denota que cada grupo $g_v \in G$ está relacionado a apenas uma assinatura de grupo $a_\tau \in A$;

   h) seja a função $F_\varrho : B \to A$ de modo que $(b_\alpha, b_\beta) \to b_\alpha + b_\beta = a_\varsigma$ tal que $a_\varsigma \in A$ e $b \in K$, garantindo assim a unicidade da assinatura.

3. na Fase 3, as transações são adicionadas a *blockchain*, porém as chaves públicas dos usuários não são mais disponibilizadas e sim a chave pública do grupo. Portanto, como todo grupo é único e não haverá repetição de chaves a serem rastreadas na *blockchain*, aumenta-se assim o nível de anonimato do *Bitcoin*.

Atualmente, o sistema *Bitcoin* permite que qualquer usuário de seu sistema consiga verificar se uma determinada transação foi completada com sucesso através da *blockchain*, permitindo também quebrar o anonimato.

A inclusão da técnica de Assinatura de Grupo no sistema *Bitcoin* define que as chaves públicas de todos os usuários que realizarem transações sejam substituídas pelas chaves pública dos grupos, impossibilitando consultas na *blockchain* e consequentemente aumentando o anonimato.

Com a inclusão da Assinatura de Grupo, como saber quem transacionou com quem? A resposta é simples: através da Assinatura do Grupo e a combinação das chaves públicas de quem transacionou, que corresponde a mensagem da assinatura. Ou seja, o minerador que detém a chave privada do grupo receberá a assinatura do grupo e as chaves públicas das *wallets* que participaram de uma determinada transação. O minerador é um ponto confiável no sistema *Bitcoin* devido o mesmo validar as transações e ser parte fundamental nesse sistema.

A validação da Assinatura de Grupo por meio do minerador ocorre através da verificação da veracidade da assinatura dessa transação e a combinação com ambas as chaves públicas das *wallets* participantes dessa transação, caso seja uma transação válida o minerador irá confirmar a transação, caso contrário não será confirmada.

Supondo que o governo brasileiro adote a cripto-moeda *Bitcoin*, e que exista uma determinada quantia de BTC reservada para ser investido em educação e que a Assinatura de Grupo esteja presente nessa cripto-moeda. Nesse cenário, imagina-se que as *wallets* destinadas a comprar materiais educacionais sejam pública através do Portal da Transparência do Governo Federal e todas as *wallets* dos fornecedores também são públicas, assim como as assinaturas dos grupos das transações já realizadas.

Nesse contexto, os usuários *Bitcoin* não poderão realizar pesquisas na *blokchain* a procura de outros usuários e nem realizar rastreamento cronológico nessa base de dados pois o que existe agora na *blockchain* são os BTCs transacionados e as chaves públicas dos grupos, portanto elevou-se o anonimato na perspectiva dos usuários.

Seguindo o mesmo pensamento e supondo que esse usuário queira validar as transações destinadas a educação do governo, qual seria o procedimento? O procedimento de validação ocorre por meio das chaves públicas das *wallets* que compõem a mensagem e a assinatura do grupo disponível no Portal da Transparência para verificar se realmente aquela quantia de BTCs foi realmente gasta com educação. Portanto, com esse caso descreve-se com um exemplo prático o processo de auditoria através da Assinatura de Grupo no sistema *Bitcoin*. A fim de exemplificar graficamente esse procedimento, criou-se a Figura 9.

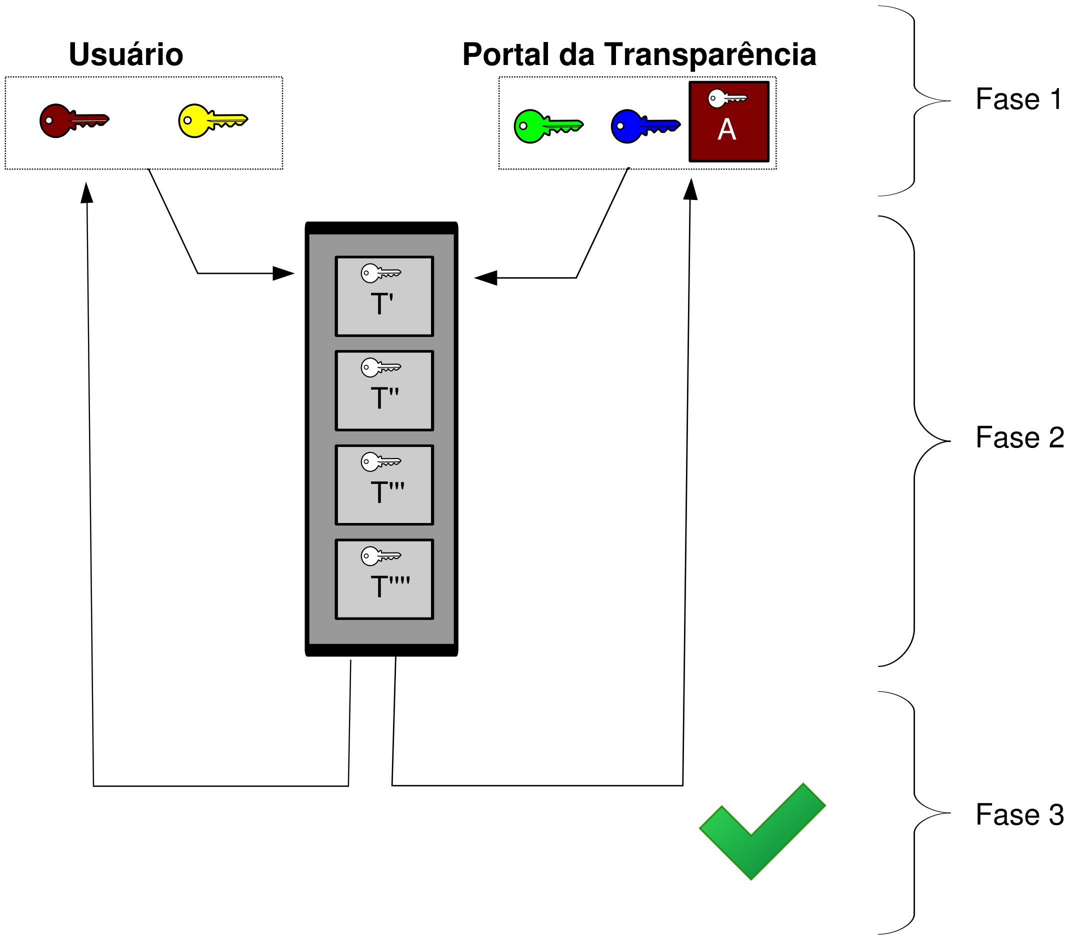

Figura 9 – Representação gráfica da auditoria por meio da inclusão da técnica de Assinatura de Grupo no sistema *Bitcoin*.

1. na Fase 1, um determinado usuário malicioso envia para o sistema *Bitcoin* duas chaves públicas aleatórias definidas por $p_\zeta, p_\kappa \in U$ e a possível mensagem da assinatura de grupo $b_\nu \in B$ a fim de quebrar a auditoria. O Governo Federal através do Portal da Transparência, disponibiliza as chaves públicas definido por $p_\lambda, p_\mu \in U$ e a assinatura de grupo $b_\xi \in B$ dessa transação a fim de permitir auditoria. Um usuário autêntico deseja auditar uma transação através do Portal da Transparência, logo esse usuário envia as informações p_λ, p_μ e b_ξ de uma transação disponibilizada por esse portal, a fim de validar essa transação e consequentemente verificar a veracidade da auditoria;

2. na Fase 2, a *blockchain* com Assinatura de Grupo disponibiliza as transações com as respectivas chaves públicas dos grupos e não mais as chaves dos usuários dessas

transações;

3. na Fase 3, verifica-se a auditoria por meio de um par de chaves públicas e a respectiva assinatura do grupo dessa transação definidas pela função inversa de F_{ϱ}^{-1}. Logo, se e somente se as chaves públicas e a assinatura forem autênticas. Logo, o sistema *Bitcoin* irá confirmar a transação. Nesse contexto, o usuário autêntico terá êxito na auditoria, pois ele possue as verdadeiras chaves que compõem a mensagem e a respectiva assinatura do grupo que permite validar essa transação por meio da função inversa. Por outro lado, o usuário malicioso não conseguirá quebrar a auditoria dessa transação, pois as chaves não correspondem a mensagem correta.

Portanto, só é possível realizar o procedimento da auditoria no sistema *Bitcoin* com a inclusão da técnica de Assinatura de Grupo através das chaves públicas das *wallets* que participaram de uma determinada transação e a sua respectiva assinatura. Ambas as *wallets* possuem a assinatura do grupo e conhecem as respectivas chaves públicas. Todavia, os outros usuários *Bitcoin* não possuem a assinatura do grupo e nem sabem as chaves públicas de quem transacionou. Desta forma, não é possível validar uma transação sem as chaves públicas e a assinatura do grupo, aumentando o anonimato e garantindo a auditoria com consenso.

Contudo, há a possibilidade de que um usuário malicioso faça combinações das chaves públicas do grupo e realize a captura das chaves públicas de todos os usuários *Bitcoin* ativos na rede a fim de realizar combinações e verificar quem transacionou com quem. Porém, além desse procedimento ser bastante custoso em termos computacionais, o atacante precisa da assinatura do grupo que somente os participantes das transações de um determinado grupo possui e o minerador que criou esse grupo.

Na Seção 3.3, discute-se como foi aplicado a Assinatura de Grupo no sistema *Bitcoin* através de um simulador desenvolvido no contexto deste trabalho, denominado *Anonycoin*. O *Anonycoin* foi desenvolvido porque não foram encontradas soluções simples para testar a aplicação da Assinatura de Grupo no sistema *Bitcoin*, a fim de verificar os resultados obtidos relacionados ao anonimato de forma objetiva e rápida, com suporte à auditoria.

3.3 *Anonycoin*

O *Anonycoin* foi desenvolvido especificamente para simular o sistema *Bitcoin* com suporte à Assinatura de Grupo. O simulador realiza testes de desempenho das transações *Bitcoin* com e sem a inclusão da técnica de Assinatura de Grupo. Após a realização de cada teste, geram-se *blockchains* com e sem Assinatura de Grupo que permite aplicar técnicas de quebra de anonimato. Desta forma, o *Anonycoin* foi desenvolvido a fim de verificar se os resultados obtidos relacionados ao desempenho, anonimato e auditoria são viáveis e eficientes quando se incluem a técnica de Assinatura de Grupo em relação ao modelo tradicional das transações *Bitcoin*.

Visando a aplicabilidade do *Anonycoin* no *Bitcoin* atual, faz-se necessário estudá-lo em uma instância de implementação mais próxima da realidade, ou seja, na rede de teste do *Bitcoin*, a *testnet* (NAKAMOTO, 2017b). Além disto, é igualmente importante destacar que uma das políticas dos desenvolvedores *Bitcoin* é a aprovação de uma *Bitcoin Improvement Proposal* (BIP), conforme Figura 10.

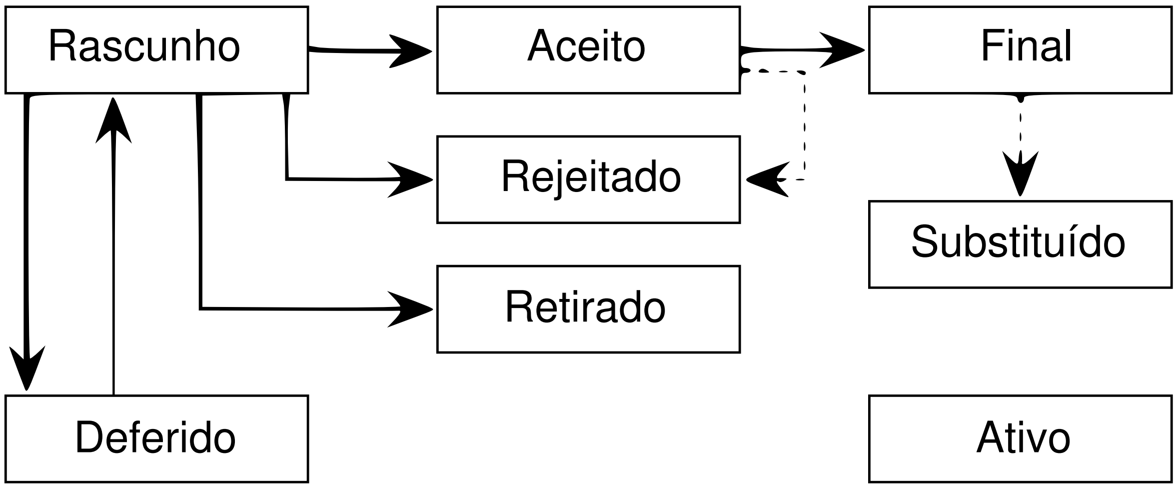

Figura 10 – Fluxograma de uma BIP (Imagem adaptada de (TAAKI, 2017)).

Logo, com o desenvolvimento do *Anonycoin*, pretende-se utilizá-lo como base para pesquisas sobre a eficácia da inclusão da técnica de Assinatura de Grupo em transações *Bitcoin* e validar o desempenho dessa técnica, o nível de anonimato e auditoria. Além disso, pretende-se disponibilizar o *Anonycoin* como *software* livre, o que permitirá o desenvolvimento de outras propostas para o sistema *Bitcoin*.

É importante salientar que na simulação do *Bitcoin*, utilizou-se diversos meios de abstração a fim de simplificar a pesquisa e focar no quesito desempenho, anonimato e auditoria do sistema *Bitcoin* com e sem Assinatura de Grupo. Todavia, pode-se simular o experimento em outras plataformas, sabendo que os resultados serão próximos dos resultados alcançados com variações apenas aplicado ao poder computacional do ambiente executado em relação ao desempenho, não atrapalhando nos resultados estatísticos.

Antes de entrar em detalhes de alguns aspectos técnicos importantes do desenvolvimento da Assinatura de Grupo no *Bitcoin*, a seguir, apresenta-se um resumo das abstrações e adaptações que foram realizadas na simulação do sistema *Bitcoin* no contexto deste trabalho:

1. simplificação das *wallets* (Seção 3.3.1);

2. simplificação das transações *Bitcoin* (Seção 3.3.2);

3. simplificação dos *miners* (Seção 3.3.3);

4. simplificação das *blockchains* (Seção 3.3.4);

5. adição da Assinatura de Grupo (Seção 3.1).

3.3.1 *Wallet*

Para obter *bitcoins* é preciso minerar. A fim de simplificar o processo de obtenção de *bitcoins* pelos usuários, gerou-se aleatoriamente uma determinada quantia de *bitcoins*, entre 10 a 100 BTC, para cada *wallet* gerada pelo *Anonycoin*.

O tipo de criptografia usado no *Anonycoin* é o mesmo usado pelo sistema *Bitcoin* para gerar as chaves públicas, chaves privadas e *hashes* das chaves públicas. O algoritmo usado por ambos sistemas se chama ECDSA (Anexo B).

Para mais detalhes sobre como foi implementado essas funcionalidades, pode-se verificar a Seção A.3.

3.3.2 Transação

No *Anonycoin*, gera-se uma determinada quantia de *wallets* e transações *Bitcoin* pré-definidas a nível de simulação. São necessários pelo menos 2 *wallets* em 1 transação, sendo que só é possível efetuar uma transação *Bitcoin* caso a *wallet* que irá enviar uma determinada quantia de *bitcoin* tenha saldo maior ou igual ao número estipulado de BTC a ser transacionado.

A linguagem *Script* do *Bitcoin*, discutida brevemente na Seção 2.2, foi simplificada a algumas funcionalidades:

- validação do saldo da *wallet* emissora, conforme a quantidade a ser transacionada;
- assinatura de uma transação;
- verificação da assinatura transacionada;
- adição da Assinatura de Grupo;
- verificação da Assinatura do Grupo.

A rede P2P foi desenvolvida usando o *software* distribuído *Zero-Em-Queue* (0MQ ou *ZeroMQ*), o mesmo *software* usado pelo *Bitcoin* para criar sua rede P2P a fim de que a simulação use o máximo da tecnologia já disponível no sistema *Bitcoin*.

Para mais detalhes sobre a implementação de uma transação no Anonycoin, verificar a Seção A.5.

3.3.3 *Miner*

O PoW na simulação resolve os *hashes* do tipo SHA-256 ao invés de SHA-512, essa abordagem permite diminuir consideravelmente o nível de dificuldade para minerar a fim de facilitar os experimentos realizados. A decisão de não resolver *hashes* do tipo SHA-512 foi embasada pelo fato de que é preciso um elevado poder computacional para resolver esse tipo de *hash*, sendo preciso plataformas específicas de mineração *Bitcoin*, conforme visto na Seção 2.3. Os *miners* não geram e nem propagam as taxas de mineração.

Para mais detalhes sobre a implementarão do *miner* no *Anonycoin*, verificar a Seção A.7.

3.3.4 *Blockchain*

Após todo o processo de mineração e validação, geram-se duas *blockchains* em formato de arquivo do tipo *Comma-Separated Values* (CSV) com as mesmas transações realizadas:

- ***bitcoin.csv*: ** *blockchain Bitcoin*;
- ***anonycoin.csv*: ** *blockchain Bitcoin* com a inclusão da Assinatura de Grupo.

As *blockchains* armazenam os seguintes dados: bloco, transação, BTC, *timestamp* e dificuldade. Todas as outras informações foram abstraídas a fim de simplificar o processo e focar no quesito desempenho das transações do sistema *Bitcoin*, anonimato e auditoria desse sistema.

Para mais detalhes sobre a implementação da *Blockchain* no *Anonycoin*, verificar a Seção A.8.

4 Análises e Resultados

Neste Capítulo, apresentam-se as análises e os resultados referentes aos experimentos realizados que estão distribuídos nas Seções 4.1 e 4.2, respectivamente.

4.1 Análise

No contexto de avaliação do desempenho das transações *Bitcoin* com e sem a inclusão da técnica de Assinatura de Grupo, realizou-se experimentações em um ambiente de simulação. Por meio da definição de um ambiente próximo do sistema *Bitcoin*, analisaram-se as principais métricas relativas ao desempenho da Assinatura de Grupo nesse sistema para garantir que as mesmas variáveis dentro do mesmo ambiente fossem medidas em um processo semelhante.

4.1.1 Objetivos e hipótese

O objetivo do experimento foi avaliar o desempenho da inclusão da técnica de Assinatura de Grupo no sistema *Bitcoin* em relação ao desempenho das transações desse sistema sem essa técnica.

4.1.2 Definição das variáveis

As variáveis foram definidas em 3 categorias: independentes, fatores e dependentes.

Na Tabela 2, apresentam-se as variáveis independentes usadas no experimento.

Tabela 2 – Variáveis independentes usadas no experimento.

Variável	**Valor**
Clientes	4, 8, 16, 32
Transações	120, 560, 2400, 9920
Grupos	10
Bloco	10

Na Tabela 3, apresentam-se os fatores considerados no experimento.

Tabela 3 – Fatores considerados no experimento.

Fator	**Valor**
Clientes por grupo	12, 56, 240, 992
Tempo (minutos)	10
Dificuldade	0 à 19920

As principais métricas para medir a técnica de Assinatura de grupo são apresentadas a seguir:

1. tempo máximo de uma transação: avalia-se o tempo relativo ao processo que uma transação precisa para ser concluída; tempo esse que não pode exceder 10 minutos;
2. tempo de mineração: o nível de dificuldade em validar uma transação é um processo que pode elevar o tempo de uma transação, principalmente se o minerador não dispor de poder computacional compatível para validar uma transação em tempo hábil.

Com base nessas métricas, determinou-se as variáveis dependentes, apresentadas na Tabela 4.

Tabela 4 – Variáveis dependentes usadas no experimento.

Variável	**Valor**
Transações com Assinatura de Grupo (*Anonycoin*)	A
Transações sem Assinatura de Grupo (*Bitcoin*)	B

4.1.3 População e amostras

A população total analisada foi de 13000 transações para o *Anonycoin*, que representa a inclusão da técnica de Assinatura de Grupo e 13000 transações para o *Bitcoin*. Porém, analisou-se estatisticamente apenas 9920 transações de ambas populações, pois nessa etapa se observou maior variância em relação ao desempenho das transações.

Da população de 13000, retirou-se uma amostra de 9920 transações de ambas populações. A amostra retirada indica uma maior variabilidade do tempo em minutos das transações. Essas amostras possuem as mesmas métricas e variáveis para ambos os experimentos, conforme visto nas Tabelas 2, 3 e 4, porém cada uma com Assinatura de Grupo e a outra sem Assinatura de Grupo.

4.1.4 Tratamentos

O processo de análise do desempenho das transações *Bitcoin* com e sem Assinatura de Grupo foi realizado em etapas. Em cada etapa, criou-se uma determinada quantidade de grupos com uma determinada quantidade de clientes que participaram de diversas transações. Portanto, analisou-se o impacto relativo ao desempenho das transações *Bitcoin* com a inclusão da técnica de Assinatura de Grupo em relação as transações *Bitcoin* tradicionais. Os grupos foram analisados baseado no tempo limite de no máximo 10 minutos, tempo hábil para realizar uma transação *Bitcoin*.

Na Tabela 5, apresentam-se os tratamentos considerados no experimento, definidos com base na combinação dos fatores apresentados na Tabela 3. Nesse contexto, executaram-se diversos ensaios distribuídos em 4 tratamentos que definem as fases.

Tabela 5 – Tratamentos executados no experimento.

Tratamento	**Clientes**	**Transações**	**Repetições**
1	4	120	10
2	8	560	10
3	16	2400	10
4	32	9920	10

Com relação a execução de cada tratamento, executaram-se 240 ensaios no tratamento 1, 1120 ensaios no tratamento 2, 4800 ensaios no tratamento 3 e 19840 ensaios no tratamento 4. Ou seja, repetiu-se dez vezes o mesmo tratamento para todos os sistemas avaliados a fim de verificar a quantidade de variações das populações. Repetiu-se diversas vezes os experimentos porém, com dez repetições observou-se variações significativas de desempenho das populações portanto, com cerca de dez repetições pode-se alcançar com 95% de certeza as anomalias relacionadas ao desempenho das abordagens.

Por fim, para a execução de cada ensaio para todos os tratamentos, independente do sistema a ser executado, determinou-se os seguintes critérios para a simulação com a inclusão da Assinatura de Grupo e sem Assinatura de Grupo:

1. a mesma quantidade de clientes, sendo acrescida proporcionalmente;
2. a mesma quantidade de transações, sendo incrementada proporcionalmente;
3. a mesma quantidade de *Bitcoin* transacionado;
4. o mesmo nível de dificuldade, que foram aumentados gradativamente a medida que as transações foram sendo realizadas;
5. a mesma quantidade de blocos;

6. a mesma quantidade de clientes em um grupo, sendo acrescida proporcionalmente.

4.2 Resultados

A fim de coletar informações relacionadas ao desempenho das transações *Bitcoin* com e sem a inclusão da técnica de Assinatura de Grupo, foi realizada uma série de experimentos que serão descritos nesta seção.

4.2.1 Fases

Com o objetivo de verificar o desempenho das transações *Bitcoin* e *Anonycoin*, realizou-se diversas transações através do simulador. Porém, essas transações foram divididas em quatro fases, com duas sub-fases cada, com as variáveis mencionadas na Seção 4.1.

4.2.1.1 Fase 1

Na Fase 1.1, criaram-se 4 *wallets* que realizaram 120 transações. No *Anonycoin*, essas transações foram distribuídas em 10 grupos, cada grupo com 12 *wallets*. O PoW inicia com zero e incrementado a medida que novas transações são realizadas. Conforme ilustra-se no gráfico da Figura 11, com eixo X intitulada "Número de Transações" e no eixo Y nomeada de "Tempo (minuto)". As transações *Bitcoin* e *Anonycoin* estão representadas no gráfico nas cores vermelha e azul, respectivamente.

Na Figura 11, observa-se que os tempos das transações *Anonycoin* foram aproximadamente o dobro do tempo das transações *Bitcoin*. Contudo, ambas as transações estão com os tempos abaixo do limite permitido, ou seja, tempo menor que dez minutos.

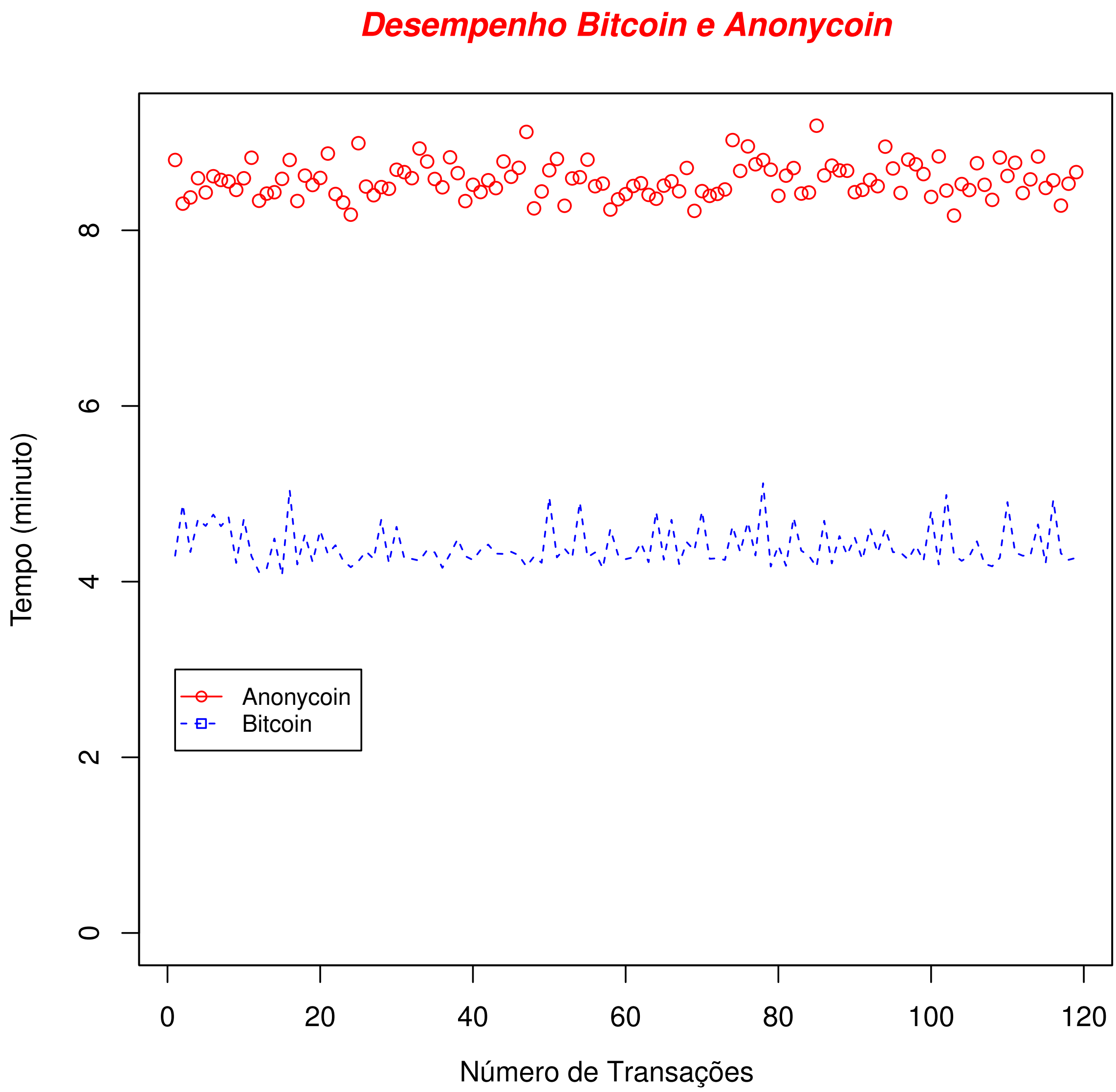

Figura 11 – Desempenho das transações *Bitcoin* e *Anonycoin* - Fase 1.1.

Na Fase 1.2, realizou-se o mesmo teste com as mesmas variáveis, com exceção do PoW que agora inicia-se com 10 mil e gradativamente incrementado, conforme pode ser observado na Figura 12.

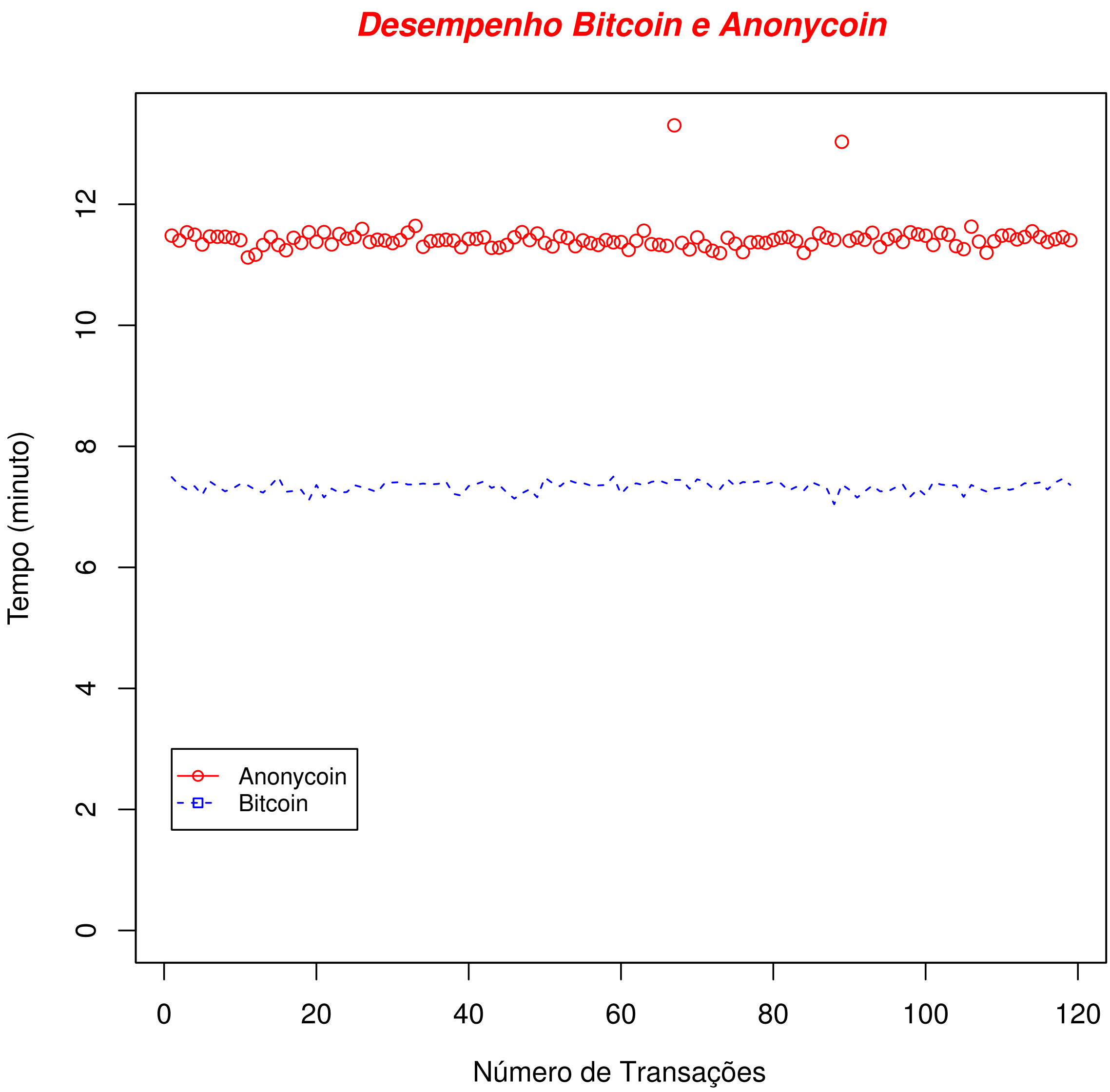

Figura 12 – Desempenho das transações *Bitcoin* e *Anonycoin* - Fase 1.2.

Na Figura 12, verifica-se que os tempos das transações *Anonycoin* excederam dez minutos. Contudo, as transações *Bitcoin* permanecem abaixo do tempo limite permitido pelo sistema. Ou seja, o tamanho do grupo não impactou negativamente no tempo das transações *Anonycoin*, mas apenas o PoW que elevou o tempo dessas transações.

4.2.1.2 Fase 2

Na Fase 2.1, criaram-se 8 *wallets* que realizaram 560 transações. No *Anonycoin*, essas transações foram distribuídas em 10 grupos, cada grupo com 56 *wallets*. O PoW inicia com zero e incrementado a medida que novas transações são realizadas. Conforme ilustra-se no gráfico da Figura 13, com eixo X intitulada “Número de Transações” e no eixo Y nomeada de “Tempo (minuto)”. As transações *Bitcoin* e *Anonycoin* estão representadas no gráfico nas cores vermelha e azul, respectivamente.

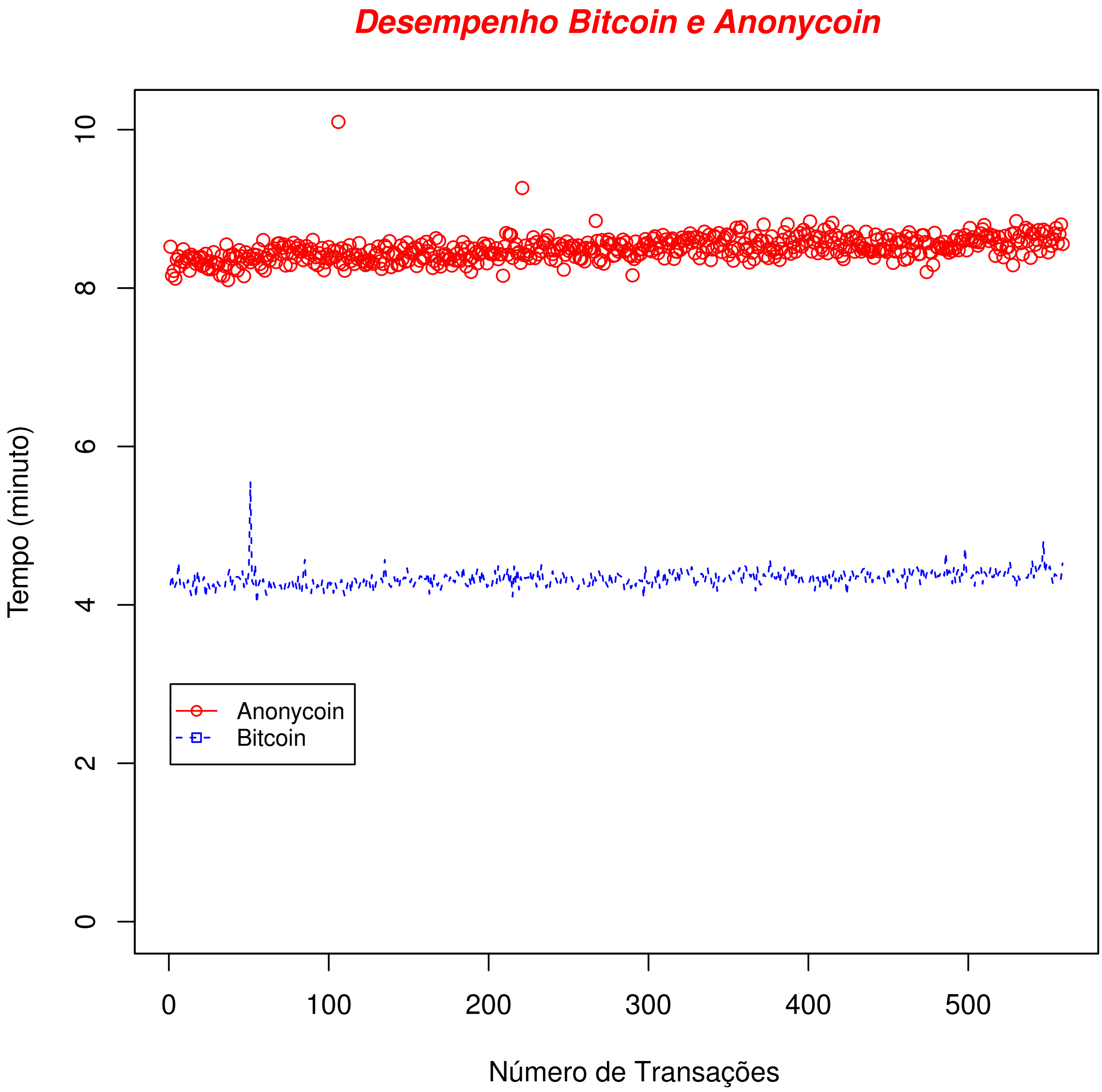

Figura 13 – Desempenho das transações *Bitcoin* e *Anonycoin* - Fase 2.1.

Na Figura 13, observa-se que os tempos das transações *Anonycoin* e *Bitcoin* continuam com as mesmas proporções da Fase 1, ou seja os tempos de ambas transações estão abaixo do limite permitido pelo sistema. Com exceção de uma transação *Anonycoin* que excedeu esse tempo limite.

Na Fase 2.2, as mesmas variáveis foram mantidas com exceção do PoW, conforme pode ser observado na Figura 14.

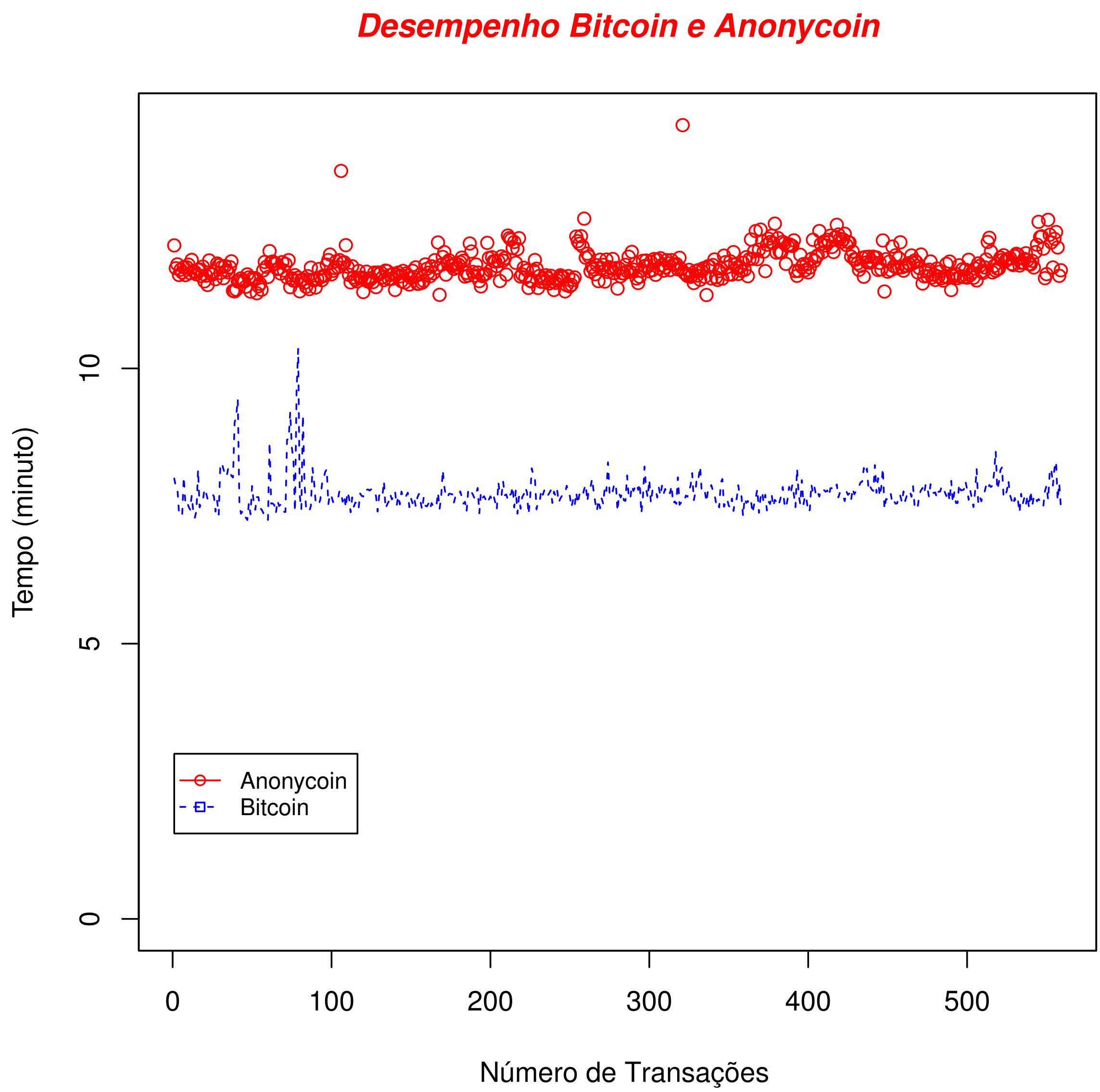

Figura 14 – Desempenho das transações *Bitcoin* e *Anonycoin* - Fase 2.2.

Na Figura 14, pode-se observar que os tempos das transações *Anonycoin* excederam dez minutos. Contudo, a maioria das transações *Bitcoin* permanecem com tempo de suas transações abaixo do tempo limite permitido pelo sistema com exceção de uma transação. Logo, apenas o PoW influenciou no desempenho das transações diretamente.

4.2.1.3 Fase 3

Na Fase 3.1, criaram-se 16 *wallets* que realizaram 2400 transações. No *Anonycoin*, essas transações foram distribuídas em 10 grupos, cada grupo com 240 *wallets*. O PoW inicia com zero e incrementado a medida que novas transações são realizadas. Conforme ilustra-se no gráfico da Figura 15, com eixo X intitulada "Número de Transações" e no eixo Y nomeada de "Tempo (minuto)". As transações *Bitcoin* e *Anonycoin* estão representadas no gráfico nas cores vermelha e azul, respectivamente.

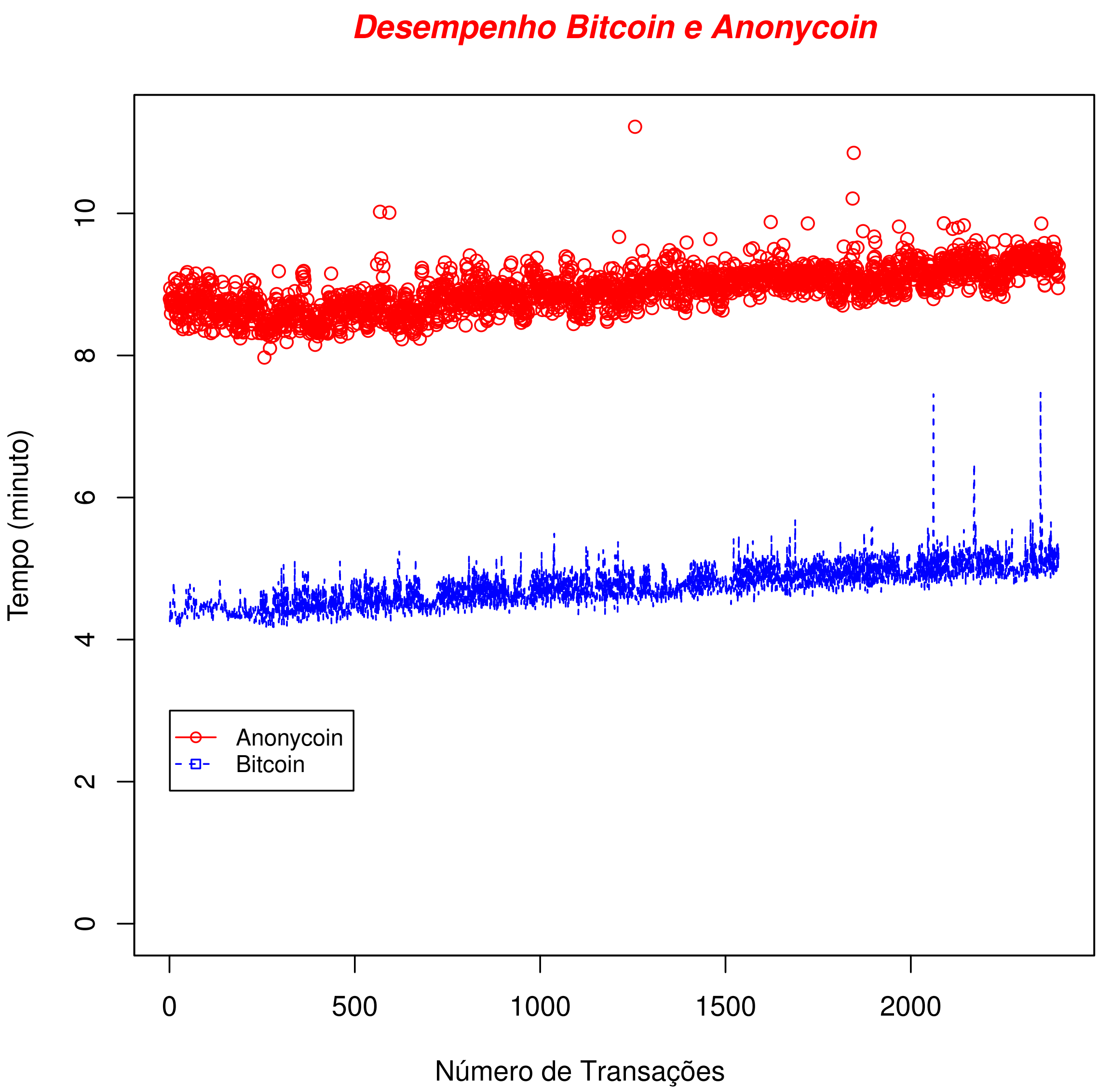

Figura 15 – Desempenho das transações *Bitcoin* e *Anonycoin* - Fase 3.1.

Na Figura 15, observa-se que os tempos das transações *Anonycoin* e *Bitcoin* continuam proporcionais. Contudo, ambas as transações estão com os tempos abaixo do limite permitido pelo sistema com exceção de algumas transações *Anonycoin*.

Na Fase 3.2, usou-se as mesmas variáveis da Fase 3.1 com exceção do PoW que inicia-se com 10 mil e vai sendo incrementado, conforme pode ser observado na Figura 16.

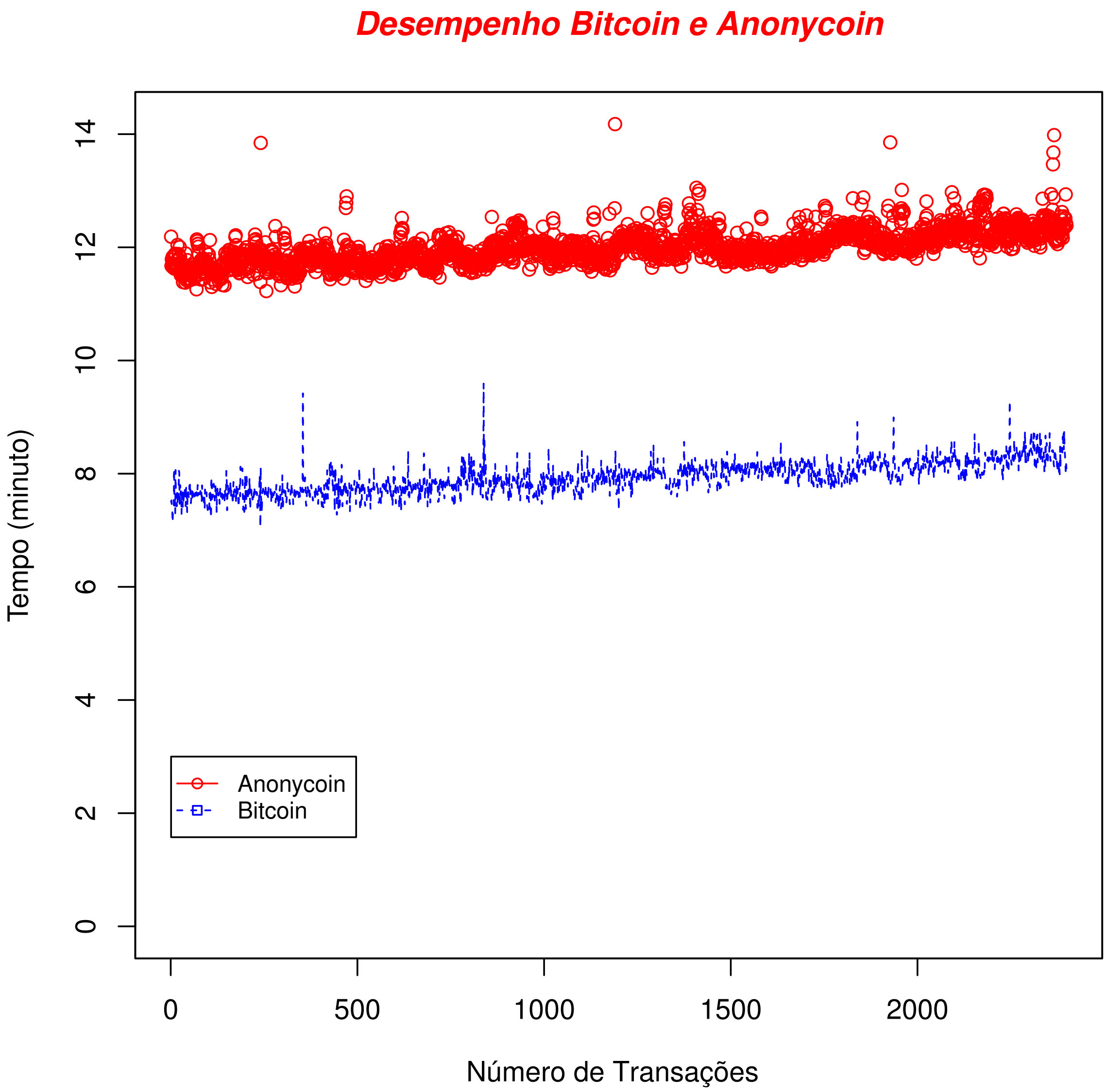

Figura 16 – Desempenho das transações *Bitcoin* e *Anonycoin* - Fase 3.2.

Na Figura 16, observa-se que os tempos de todas as transações *Anonycoin* continuam acima de dez minutos. Contudo, a maioria das transações *Bitcoin* permanecem com tempo de suas transações abaixo do tempo limite permitido pelo sistema. Ou seja, o número de transações e o tamanho do grupo continuam não afetando diretamente o desempenho do anonimato em grupo, apenas o PoW tem influenciado até a presente fase.

4.2.1.4 Fase 4

Na Fase 4.1, criaram-se 32 *wallets* que realizaram 9920 transações. No *Anonycoin*, essas transações foram distribuídas em 10 grupos, cada grupo com 992 *wallets*. O PoW inicia com zero e incrementado a medida que novas transações são realizadas. Conforme ilustra-se no gráfico da Figura 17, com eixo X intitulada "Número de Transações" e no eixo Y nomeada de "Tempo (minuto)". As transações *Bitcoin* e *Anonycoin* estão representadas no gráfico nas cores vermelha e azul, respectivamente.

Na Figura 17, observa-se que os tempos das transações *Anonycoin* continuam cerca do dobro do tempo das transações *Bitcoin* em proporcionalidade linear. Contudo, aproximadamente metade das transações *Anonycoin* começaram a exceder o limite permitido pelo sistema. Enquanto que a maioria das transações *Bitccoin* continuam com o tempo de suas transações dentro do tempo hábil. Portanto, observa-se que além do PoW, um número de clientes acima de 992 em um grupo pode começar a afetar negativamente os resultados relativos ao desempenho do *Anonycoin*.

Na Fase 4.2, as mesmas variáveis foram mantidas, apenas o PoW que inicia-se com 10 mil e incrementado a medida que novas transações são realizadas, conforme pode ser observado na Figura 18.

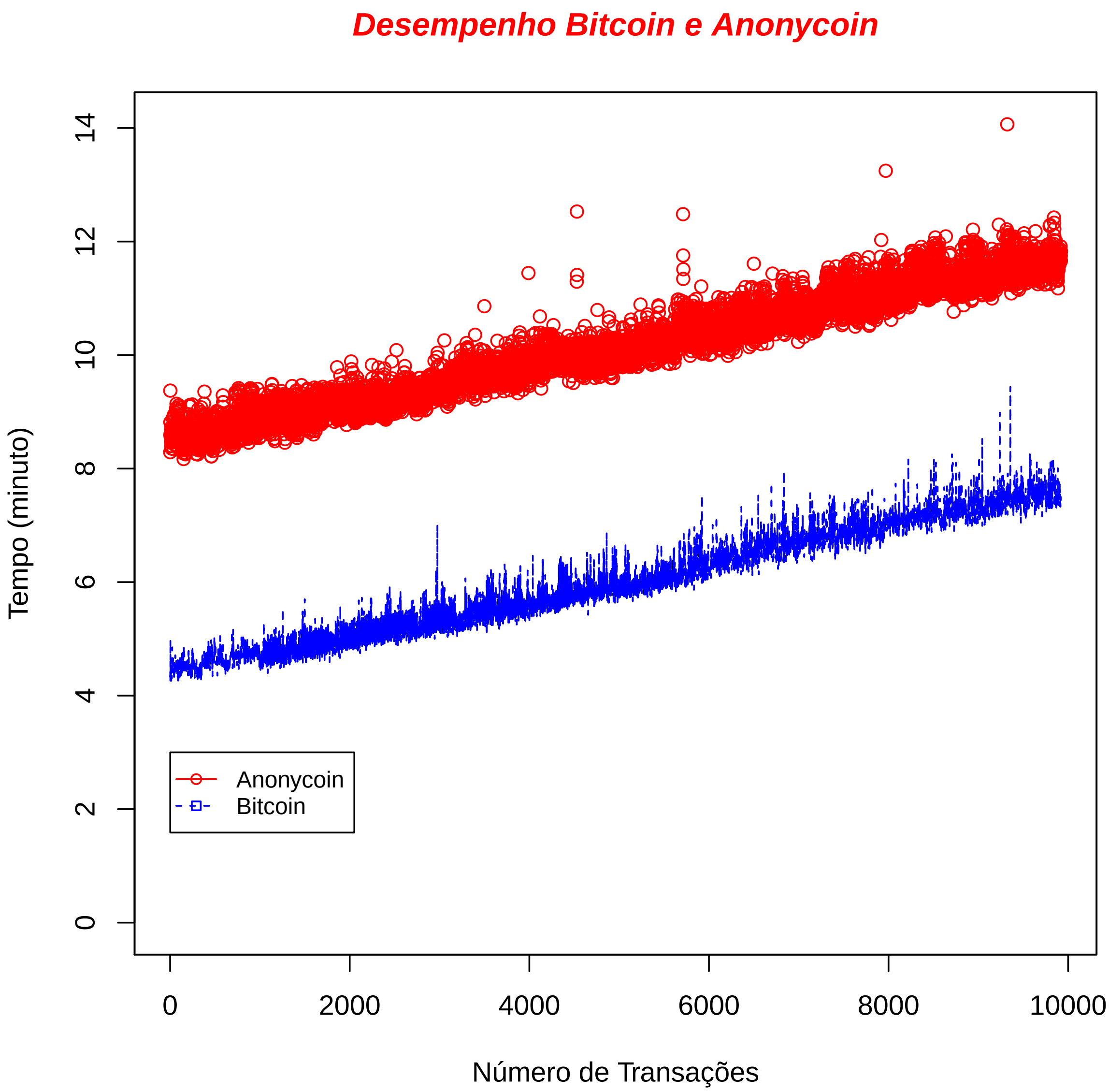

Figura 17 – Desempenho das transações *Bitcoin* e *Anonycoin* - Fase 4.1.

Na Figura 18, observa-se que os tempos de todas as transações *Anonycoin* excederam dez minutos. Contudo, apenas partes das transações *Bitcoin* começaram a exceder o tempo limite permitido pelo sistema devido o PoW e ao elevado número de transações. Porém, a proporcionalidade de tempo entre as soluções continuam semelhantes em razões lineares.

Observa-se que com um grupo inferior a 992 clientes o desempenho do *Anonycoin* apresenta resultados dentro do limite de tempo de dez minutos. Com essa quantia de clientes em um grupo verifica-se que o anonimato é alcançado conforme apresentado pela função $F_{\varrho} : B \rightarrow A$, onde define-se a unicidade da Assinatura de Grupo em cada transação pertencente a somente um grupo conforme visto na Seção 3.2. Portanto, a auditoria é alcançada pela assinatura do grupo e as respectivas chaves públicas de quem participou de cada transação, segundo definido pela função F_{ϱ}^{-1}, observada na mesma seção.

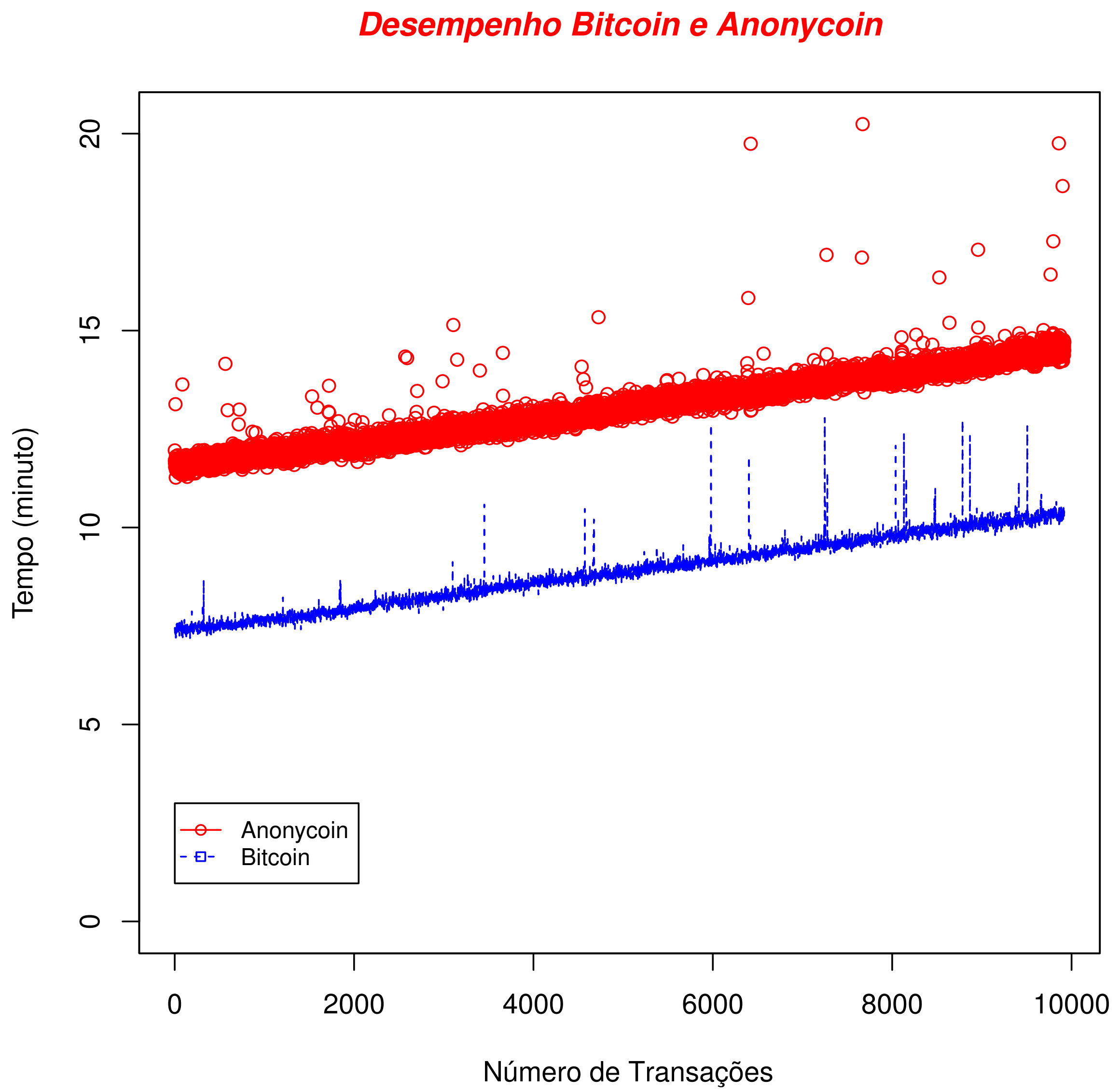

Figura 18 – Desempenho das transações *Bitcoin* e *Anonycoin* - Fase 4.2.

4.3 Estatísticas do experimento

A partir dos dados coletados, observados no gráfico da Figura 17, calcularam-se alguns valores relativos aos tempos das transações *Bitcoin* com e sem Assinatura de Grupo a fim de analisar o tempo das transações de ambas as soluções e verificar as causas da variância principalmente nas transações com Assinatura de Grupo. Calculou-se esses dados usando a função *summary* da linguagem de programação R e obteve-se:

- ***Anonycoin:***
  - Mínima: 8.169;
  - Primeiro quadrante: 9.283;

  - Mediana: 10.080;
  - Média amostral: 10.120;
  - Terceiro quadrante: 10.930;
  - Máxima: 14.070.

- ***Bitccoin:***
  - Mínima: 4.220;
  - Primeiro quadrante: 5.190;
  - Mediana: 5.945;
  - Média amostral: 6.019;
  - Terceiro quadrante: 8.866;
  - Máxima: 9.440.

Portanto, observa-se que o tempo mínimo de uma transação *Bitcoin* com a aplicação da técnica de Assinatura de Grupo foi de aproximadamente 8,16 minutos nessa simulação. Contudo, o tempo mínimo de uma transação *Bitcoin* sem essa técnica foi cerca de 4,22 minutos. Verifica-se também que, o tempo máximo de uma transação *Bitcoin* com a técnica de Assinatura de Grupo é de aproximadamente 14,07 minutos. Todavia, o tempo máximo de uma transação *Bitcoin* sem essa técnica foi de cerca de 9.44.

Em seguida, calculou-se a Média Amostral, que é resultado da soma de todos os dados dividido pela quantidade total da amostra. Ke e Xiang apresentam a Equação da Média Amostral, visualizada na Equação 4.1 (SEBER; LEE, 2003).

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.1}$$

Aplicou-se a Média Amostral usando a função *mean* da linguagem R:

- ***Anonycoin:*** 10.12012;
- ***Bitcoin:*** 6.018846.

Após o cálculo da Média Amostral, calculou-se a Variância da amostra. Yau define a Equação da Variância Amostral, conforme a Equação 4.2 (SEBER; LEE, 2003).

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{4.2}$$

Executou-se a função *var* da linguagem R, e obteve-se os seguintes resultados:

- ***Anonycoin:*** 0.9017807;
- ***Bitcoin:*** 0.9173767.

Verificou-se o desvio padrão usando a função *sd* do R e obteve o resultado a seguir:

- ***Anonycoin:*** 0.9496214;
- ***Bitcoin:*** 0.9577978.

Calculou-se também a mediana das amostras usando a função *median* do R, obtendo-se:

- ***Anonycoin:*** 10.07601;
- ***Bitcoin:*** 5.94546.

Usou-se a Equação de Distribuição Normal, também conhecida como Distribuição de Gauss, que é definida pela Função de Densidade Probabilística que resulta na Equação 4.3. Nessa equação, μ é a Média Amostral (Figura 4.1) e σ é a Variância (Figura 4.2) (SEBER; LEE, 2003).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\epsilon^{-(x-\mu)^2/2\sigma^2} \tag{4.3}$$

Yu afirma que se a variável X segue a Distribuição Normal, então pode-se calcular a Equação de Distribuição Normal Padrão (SEBER; LEE, 2003), visto na Equação 4.4.

$$X = N(\mu, \sigma^2) \tag{4.4}$$

A fim de conhecer o tamanho máximo de clientes *Bitcoin* conforme variável dependente grupo, usando a técnica de Assinatura de Grupo, e baseado na variável independente de tempo (10 minutos) aplicou-se o método estatístico da Regressão Linear no *Anonycoin* e *Bitcoin*.

Peternelli apresenta o Modelo Linear de Primeiro Grau, conhecido também por Regressão Linear Simples (SEBER; LEE, 2003; PETERNELLI, 2017), visto na Equação 4.5.

$$Y_i = \beta_o + \beta_1 X_i + e_i \tag{4.5}$$

No qual:

- Y_i: valor observado para a varíavel dependente Y;
- β_0: constante de regressão;
- β_1: coeficiente de regressão;
- X_i: i-ésimo nível da variável independente X;
- e_i: erro entre Y_i e o ponto da curva.

Logo, a linearidade descreve corretamente a relação funcional entre dois pontos X e Y. A normalidade é esperada para que não exista tendências e que a estatística funcione de forma correta entre as amostras (YAU, 2017).

Usou-se a função para Regressão Linear Simples "lm"da linguagem R aplicada a uma amostra de 5 mil transações *Bitcoin* com e sem Assinatura de Grupo, obtendo-se os resultados desse modelo a seguir:

Resultado do modelo residual

- Mínima: 1.31762;
- Primeiro quadrante: 0.19106;
- Mediana: 0.00052;
- Terceiro quadrante: 0.18348;
- Máxima: 2.71269.

Coeficientes

- Estimado: 4.909518 e 0.847132;
- Erro padrão: 0.042577 e 0.008141;
- Valor de t: 115.3 e 104.1;
- Valor de p ($> |t|$): 2.2e-16.

Os erros padrão do modelo residual foi de 0.2736 sobre 4997 graus de liberdade. O R^2 padrão foi de 0.6843, com o R^2 ajustado de 0.6842. O valor de F foi de 1.083e+04 aplicado entre 1 e 4997 DF com p-value $<$ 2.2e-16.

Logo, se a Distribuição Normal tem como resultado $\mu = 0$ e $\sigma = 1$, então pode-se aplicar a Distribuição Normal Padrão. A Equação de Shapiro-Wilk verifica se uma determinada amostra segue a Distribuição Normal, como observa-se na Equação 4.6 (SEBER; LEE, 2003):

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{4.6}$$

A fim de verificar a normalidade das amostras, aplicaram-se o teste de Shapiro-Wilk usando a linguagem R e obtêve-se o resultado para o valor de p $<$ 2.2e-16. O valor de p indica se as amostras estão normalizadas e seguem a Distribuição Normal se o valor de p for maior que 0,05.

A fim de comparar a homogeneidade das amostras, gerou-se uma análise visual dos dados usados que devem ser distribuído igualmente. Conforme ilustra-se no gráfico da Figura 19, com eixo X intitulada “Filtrada” e no eixo Y nomeada de “Residual”.

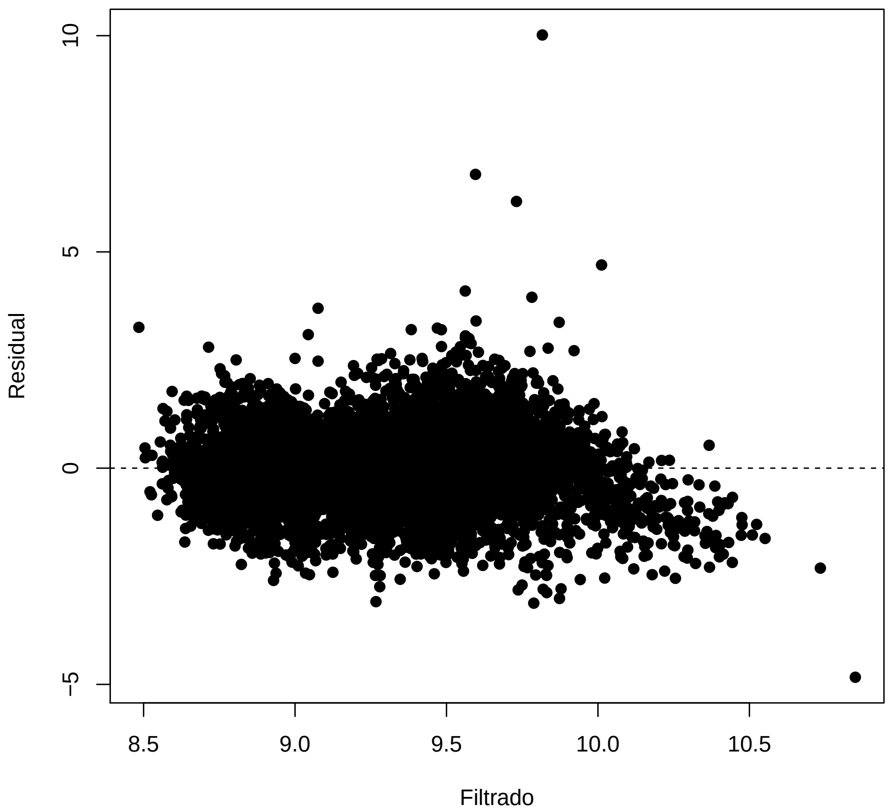

Figura 19 – Análise visual para homogeneidade dos resíduos.

Portanto, com o intuito de verificar se as amostras são distribuídas de forma homogênea gerou-se uma representação gráfica conforme pode ser observado na Figura 20, com uma linha descontínua para separar as amostras *Bitcoin* com e sem Assinatura de Grupo, onde o eixo X representa o *Bitcoin* e o Y representa o *Anonycoin* respectivamente.

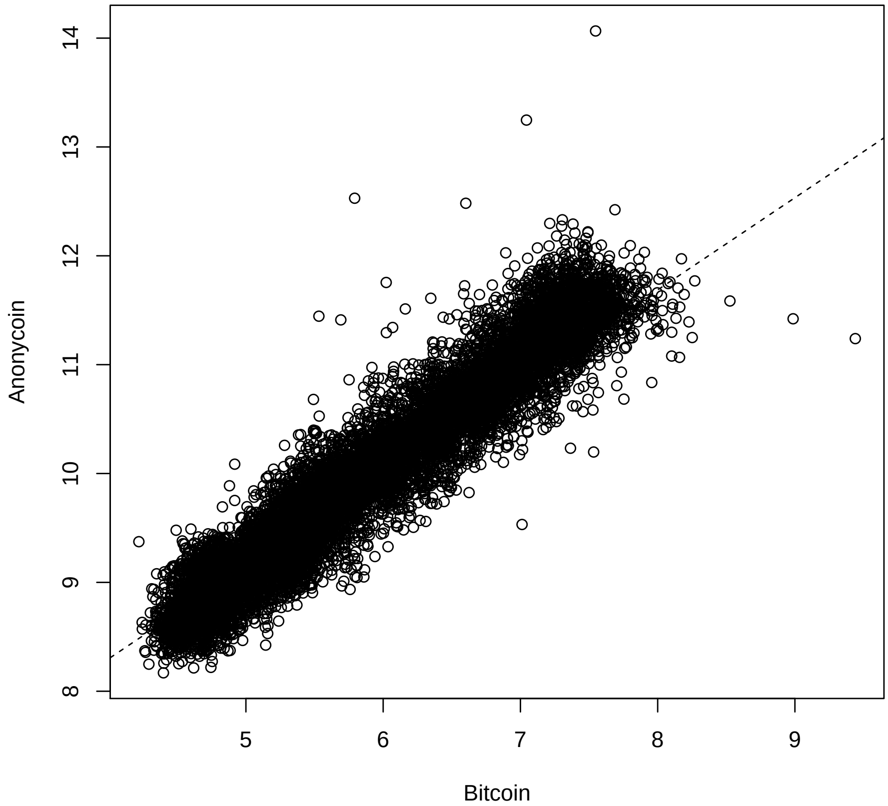

Figura 20 – Distribuição homogênea usando Regressão Linear.

Logo, pode-se comparar a média amostral de ambas as amostras com base no tempo de dez minutos pois elas são homogêneas.

4.3.1 Conclusões

Neste trabalho, apresentou-se o a inclusão da técnica Assinatura de Grupo nas transações *Bitcoin* por meio da simulação *Anonycoin*. Nesse contexto, através do *Anonycoin* realizou-se um estudo comparativo sobre o desempenho das transações do sistema *Bitcoin* com e sem Assinatura de Grupo. Foi definido, através de formalizações, o anonimato e a auditoria da Assinatura de Grupo.

Na perspectiva de implementação, efetuou-se um estudo da API de desenvolvimento do sistema *Bitcoin*. Estudou-se detalhadamente a especificação original do *Bitcoin* e suas funções, registrando-se quais funções poderiam ser implementadas tal como especificadas e quais deveriam ser adaptadas. Nesse contexto, definiram-se funções e mecanismos que foram apenas citados, sem definição detalhada, ou parcialmente definidos na especificação original do *Bitcoin*.

Logo, implementou-se e configurou-se a Assinatura de Grupo no *Bitcoin* através de simulação. A implementação provê uma abstração para diversos partes do sistema *Bitcoin*, de modo que os processos em execução fossem simplificados mas próximos dos reais. Esse simulador é de suma importância para novas pesquisas acadêmicas pois através deste pode-se medir o desempenho de transações *Bitcoin* e comparar com outras técnicas além do anonimato em grupo a fim de validar uma outra abordagem.

A simulação foi de suma importância para verificar os limites da técnica de Assinatura de Grupo no sistema *Bitcoin* no que tange a quantidade de clientes em um grupo, a fim de realizar transações em tempo hábil em comparação com o sistema tradicional. Apesar de que foi exemplificado o anonimato e auditoria com apenas quatro clientes a fim de facilitar a explicação por meio de formalizações, mas vale resaltar que cada grupo pode conter cerca de 500 clientes por grupo com resultados positivos em relação ao anonimato e auditoria.

Essa pesquisa é importante para a indústria pois resolve problemas relacionados ao anonimato no sistema *Bitcoin*, aumentando assim a confiabilidade dessas empresas nesse sistema já que há um crescimento exponencial do uso dessa criptomoeda no mundo corporativo em todo mundo. Outro fator relevante mencionado por essa pesquisa é auditoria com consenso, único dentre os trabalhos relacionado, oferecendo maior segurança e transparência para prestação de contas para o setor público e privado.

Neste sentido, este trabalho contribui diretamente com o avanço no desenvolvimento do anonimato e auditoria por meio da técnica de Assinatura de Grupo no sistema *Bitcoin*, viabilizando o seu desempenho.

4.3.2 Aplicações

A proposta de uma solução para elevar o nível de anonimato em transações *Bitcoin* e permitir auditoria em tempo hábil através da técnica de Assinatura de Grupo é viável e aplicável. Portanto, sob o ponto de vista de implementação da técnica, observa-se resultados positivos em relação ao anonimato, auditoria e ao desempenho, desde que sejam criados grupos de até 500 clientes que usam Assinatura de Grupo no sistema *Bitcoin*, pois é possível obter resultados satisfatórios em relação ao anonimato em tempo hábil e permite-se auditoria nesse sistema. Porém, com grupos acima de 500 clientes a técnica de Assinatura de Grupo se torna ineficiente, pois eleva o tempo da transação acima de 10 minutos.

Como resultado final deste trabalho, mostra-se que é possível implementar um elevado nível de anonimato para os usuários *Bitcoin* usando uma técnica simples como a Assinatura de Grupo. A Assinatura de Grupo possibilita que se efetue o processo de auditoria em transações *Bitcoin* sem alterar a sua infra-estrutura e reusando tecnologias existentes.

A auditoria por meio da Assinatura do Grupo é eficiente em termos teóricos conforme as formalizações apresentadas na Seção 3.2, desde que as chaves públicas de quem transacionou e a assinatura propriamente dita não sejam disponibilizadas, a não ser que esse seja com o objetivo de permitir que todos possam acompanhar a transação. Portanto, a única forma de realizar auditoria é através das chaves públicas das *wallets* participantes das transações e da assinatura do respectivo grupo. De outra forma, não há como realizar o procedimento de auditoria e todas as transações e usuários *Bitcoin* serão anônimos.

4.3.3 Trabalhos Futuros

A seguir, apresentam-se as propostas de trabalhos futuros relacionados a Assinatura de Grupo no sistema *Bitcoin*:

1. é necessário implementar a Assinatura de Grupo no sistema *Bitcoin* atualmente em operação, a fim de verificar se os resultados obtidos em simulação são os mesmos em produção com milhões de transações sendo realizadas;

2. é importante verificar se em transações *Bitcoin* múltiplas e paralelas a Assinatura de Grupo é eficaz em relação ao seu desempenho;

3. pesquisar formas de otimizar o tempo das transações com grupos acima de 500 clientes;

4. aplicar técnicas de segmentação da rede *Bitcoin*, a fim de verificar como os grupos se comportam em relação a auditoria;

5. efetuar testes reais usando a rede de teste *testnet* do *Bitcoin* para homologar a Assinatura de Grupo e verificar se os resultados são próximos do apresentado aqui.

5 Trabalhos Relacionados

Neste capítulo, apresentam-se os trabalhos relacionados com foco em anonimato *Bitcoin* na Seção 5.1 e uma comparação desses trabalhos em relação ao anonimato, desempenho e auditoria com o *Anonycoin* nas Seções 5.2, 5.3 e 5.4, respectivamente.

5.1 Trabalhos relacionados com foco em anonimato *Bitccoin*

Corrigan-Gibbs et al. desenvolveram um método que garante um elevado nível de anonimato com a integridade da comunicação e prestação de contas usando grupos anônimos (CORIGAN-GIBBS; FORD, 2010) chamado de *Dissent*. Essa técnica foi testada somente em pequenos grupos com cerca de 44 nós descentralizados e mensagens de até 16 Megabytes. Não foi implementado no sistema *Bitcoin* e nem tem foco nesse sistema. É importante salientar que utilizou-se um método com limitações relacionados ao tamanho do grupo e das mensagens compartilhadas. Essas limitações podem impactar diretamente no desempenho do sistema *Bitcoin* devido a grande quantidade de nós e das transações desse tipo de sistema.

Ziegeldorf et al. propõem uma novo tipo de serviço de combinações descentralizadas intitulado *CoinParty* (ZIEGELDORF et al., 2015). O serviço mescla informações de diversas *wallets* e transações de um sistema *Bitcoin* usando embaralhamento de conexões de forma descentralizada para aumentar e garantir o anonimato. Contudo, a alta latência nas transações causada pelo embaralhamento das conexões em um serviço intermediário é um ponto fraco na implementação desse método no sistema *Bitcoin*. Portanto, uma lentidão nas conexões do sistema *Bitcoin* poderia inviabilizar as transações e consequentemente a sua implementação nesse sistema.

Miers et al. apresentam o *Zerocoin*, um novo sistema *e-cash* baseado no *Bitcoin*, conhecido também de *Altcoin*. Esse sistema é distribuído e usa técnicas criptográficas para quebrar o elo entre as transações sem adicionar pares confiáveis, sendo uma extensão criptográfica compatível com o *Bitcoin* (MIERS et al., 2013). Essa abordagem permite um elevado nível de anonimato para o *Bitcoin* porém a complexidade da criptografia baseada em logaritmo duplo-discreto aumenta o tamanho dos blocos a serem minerados. Portanto, a complexidade, o tempo e a mudança de arquitetura proposta pelo *Zerocoin* elevam a dificuldade de implementação no *Bitcoin*.

Danezis et al. criaram o *Pinocchio Coin*, uma variante do *Zerocoin* que usa curvas elípticas e emparelhamentos bilineares em vez de um logaritmo duplo-discreto de verificação (DANEZIS et al., 2013). Como resultado de tal abordagem, observou-se uma verificação mais eficiente em relação a complexidade e ao desempenho, já que o *Pinocchio Coin* usa aritmética quadrática. Contudo, além de herdar alguns problemas do *ZeroCoin*, como mudança de arquitetura, não há uma implementação completa do *Pinocchio Coin* ou uma evidências contundentes com base em análise científica para o sistema *Bitcoin*, nem mesmo simulações, sendo somente um caso de estudo bastante preliminar.

Ruffing et al. desenvolveram o *CoinShuffle*, um protocolo que realiza embaralhamento das transações a fim de garantir um elevado nível de anonimato (RUFFING P. MORENO-SANCHEZ, 2017). Observa-se que tal abordagem obteve um elevado nível de anonimato. Contudo, o desempenho do protocolo foi medido em simulação com grupos de até 50 participantes obtendo-se 40 segundos. Todavia, os autores não medem o tempo de cada transação e sim o desempenho do protocolo em si.

Na Tabela 6, disponibiliza-se todas as características a serem avaliadas no trabalhos relacionados e no *Anonycoin*.

Tabela 6 – Características a serem avaliadas nos trabalhos relacionados.

Identificador	**Característica**
1	Compatível com o sistema *Bitcoin*?
2	Analisa anonimato?
3	Analisa desempenho?
4	Analisa auditoria?
5	Usa somente as tecnologias *Bitcoin*?
6	É *Altcoin*?
7	Há um protótipo funcional?
8	O código fonte está disponível?

Na Tabela 7 há uma comparação entre as principais características dos trabalhos relacionados, identificado na Tabela 6, e a abordagem *Anonycoin*.

Tabela 7 – Comparação entre os trabalhos relacionados.

Projeto	**Característica**							
-	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Anonycoin	X	X	X	X	X		X	X
Dissent		X	X				X	X
CoinParty	X	X	X				X	
ZeroCoin	X	X	X			X	X	
Pinocchio Coin	X	X	X			X		
CoinShuffle	X	X	X		X		X	X

5.2 Comparação de anonimato entre o *Anonycoin* e outras abordagens

Nesta Seção, apresenta-se um comparativo relacionado ao anonimato da simulação *Anonycoin* e seu desempenho com outras abordagens *Bitcoin*.

O protocolo *Dissent* garante um elevado nível de anonimato com proteção contra ataque de negação de serviço (DoS (*Denial of Service*) (CORIGAN-GIBBS; FORD, 2010). Esse protocolo usa o conceito da técnica de Assinatura de Grupo, servindo de inspiração para a inclusão dessa técnica no *Bitcoin* através do *Anonycoin*. Porém, os autores não fazem nenhuma referência ao sistema *Bitcoin* e só desenvolveram um protótipo do protocolo *Dissent*. Diferente desse protocolo, o *Anonycoin* é uma simulação da inclusão da técnica de Assinatura de Grupo no sistema *Bitcoin* que garante o anonimato.

O protocolo *CoinParty* é um serviço de embaralhamento de transações *Bitcoin* externo ao sistema *Bitcoin*, para aumentar e garantir o anonimato (ZIEGELDORF et al., 2015). Apesar de elevar o anonimato, esse serviço externo desenvolvido pelo *CoinParty* pode ser um ponto fraco em relação a sua possível implementação no sistema *Bitcoin*. Diferente do *CoinParty*, o *Anonycoin* eleva o anonimato mas usa as próprias tecnologias do sistema *Bitcoin* sem adicionar serviços de terceiros, diminuindo assim a latência das transações desse sistema.

O sistema *Zerocoin* é baseado no *Bitcoin*, mas usa *blockchains* descentralizadas e algoritmos criptográficos baseados em logaritmo duplo-discreto para garantir o anonimato (MIERS et al., 2013). Essa abordagem permite um elevado nível de anonimato, assim como o *Anonycoin*, porém a complexidade da criptografia usada e a mudança de arquitetura como a descentralização da *blockchain* dificulta a sua inclusão no sistema *Bitcoin*, além de aumentar a latência das transações desse sistema. O *Anonycoin* é focado no sistema *Bitcoin*, além de usar a própria infra-estrutura do sistema *Bitcoin* e usar os mesmos algoritmos criptográficos.

O sistema *Pinocchio Coin* é uma variante do *Zerocoin*, que usa um algoritmo baseado em curvas elípticas e emparelhamentos bilineares em vez de um algoritmo baseado em logaritmo duplo-discreto para aumentar o anonimato (DANEZIS et al., 2013). Apesar de diminuir a complexidade relacionado ao anonimato através do uso de um algoritmo mais simples porém eficiente, o *Pinocchio Coin* sofre dos mesmos problemas que o *Zerocoin* como mudança de arquitetura do sistema *Bitcoin* e não há uma implementação do *Pinocchio Coin*. Diferente do *Pinocchio Coin*, o *Anonycoin* usa as mesmas infra-estrutura do sistema *Bitcoin* e é um protótipo real e implementável em produção no sistema *Bitcoin*.

O sistema *CoinShuffle* é semelhante ao *CoinParty*, contudo usa as tecnologias *Bitcoin* para realizar o embaralhamento das transações (RUFFING P. MORENO-SANCHEZ, 2017). Essa abordagem atinge um alto nível de anonimato através de sua técnica, inclusive em transação única. Apesar do *Anonycoin* não usar embaralhamento de transações, as abordagens são semelhantes em relação ao grupo e ambas propostas atingem o elevado nível de anonimato por meio de formalizações. Contudo, o *CoinShuffle* usa embaralhamento das transações *Bitcoin* em grupos, enquanto o *Anonycoin* usa grupos e altera as chaves públicas dos clientes.

5.3 Comparação de desempenho entre o *Anonycoin* e outras abordagens

Nesta Seção, apresenta-se um comparativo relacionado ao desempenho da simulação *Anonycoin* com outras abordagens que tem o foco em aumentar o anonimato em transações *Bitcoin*.

Assim, como o *Anonycoin*, o *Dissent* analisa o desempenho de seu protocolo porém com métricas diferentes. As métricas adotadas pelo *Dissent* são os tamanhos das mensagens que são cerca de 16 MB (*Megabytes*) e nós de até 16 clientes. Portanto, os autores do *Dissent* não implementaram no sistema *Bitcoin*. Sendo assim, não há como comparar o desempenho do *Anonycoin* com esse protocolo apesar de que os autores realizaram testes de desempenho com resultados satisfatório.

Os autores do *CoinParty* realizaram uma análise de desempenho em simulação com cerca de 300 transações e grupos de embaralhamento de transações que variam entre 3, 5, 7, 11, 13 e 15. Seu pior resultado foi de 150 minutos e seu melhor resultado foi de 3 minutos. O pior resultado do *Anonycoin* com grupo de 992 clientes em relação a desempenho foi de cerca de 14 minutos e o melhor resultado foi de aproximadamente 8 minutos com um elevado nível de dificuldade. Logo, o desempenho do *Anonycoin* em relação ao *CoinParty* é mais aceitável baseado no tempo hábil de uma transação *Bitcoin*. Porém, é importante salientar que ambas as soluções foram desenvolvidas em simulações distintas em ambientes diferentes, portanto os resultados reais só poderão ser mensurados quando ambas as soluções forem implementadas no sistema *Bitcoin* em produção e realizados testes de desempenho semelhantes com as mesmas variáveis. Contudo, os autores desenvolveram um simulador e analisaram o desempenho de sua abordagem usando máquinas virtuais, e analisaram o desempenho de seu protocolo isoladamente e não das transações *Bitcoin* com e sem sua abordagem a fim de permitir uma análise mais próxima da realidade como foi realizado pelo *Anonycoin*.

O sistema *Zerocoin* analisa o desempenho relacionado à computação necessária para calcular o PoW de sua abordagem e não do desempenho de uma transação *Bitcoin* como realizado pelo *Anonycoin*. Os autores do sistema *Zerocoin* fazem testes de desempenho apenas entre sua abordagem e o sistema *Bitcoin*, pois o *Zerocoin* é outro tipo de criptomoeda denominada *Altcoin*. Logo, é inviável comparar os resultados de desempenho de soluções que não consideram as mesmas métricas.

Os autores do sistema *Pinocchio Coin* disponibilizaram uma equação matemática para calcular o desempenho de sua abordagem, porém não há resultados em simulação de seus experimento para análise e comparação com o *Anonycoin*. Os testes de desempenho dessa abordagem são apresentados pelos autores somente através de formalidades. O *Anonycoin* realiza testes em simulações para determinar o desempenho de sua abordagem e faz comparação com o sistema *Bitcoin* tradicional, além de formalizar o anonimato. Logo, a falta de uma não implementação em simulação do *Pinocchio Coin* inviabiliza a comparação dos resultados relacionados ao desempenho com o *Anonycoin*.

Os autores do *CoinShuffle* realizam testes de desempenho que são apresentados em relação apenas ao seu protocolo isoladamente. O *Anonycoin* realiza testes de desempenho de cada transação *Bitcoin* com e sem Assinatura de Grupo a fim de verificar a veracidade de sua abordagem em tempo hábil. Portanto, para uma análise comparativa eficiente é preciso que ambas as soluções usem as mesmas métricas a fim de que seja comparado com a maior veracidade possível o desempenho de ambas as soluções.

5.4 Comparação de auditoria entre o *Anonycoin* e outras abordagens

Todos os trabalhos relacionados analisam apenas o nível de anonimato e desempenho de suas abordagens. Todavia, nenhum desses trabalhos abordam o assunto relativo a auditoria.

Auditoria é um ponto interessante a ser abordado principalmente no sistema *Bitcoin*, pois para algumas empresas é de suma importância realizar esse procedimento. Um elevado nível de anonimato impossibilita realizar o processo de auditoria.

O *Anonycoin* é o único projeto de pesquisa dentre os trabalhos mencionados que permite a realização de auditoria como visto na Seção 3.2. Porém, é possível aplicar os conceitos de auditoria apresentado pelo *Anonycoin* nos trabalhos relacionados desde que seja realizados adaptações conforme cada abordagem.

6 Conclusões

Neste trabalho, apresentou-se o a inclusão da técnica Assinatura de Grupo nas transações *Bitcoin* por meio da simulação *Anonycoin*. Nesse contexto, através do *Anonycoin* realizou-se um estudo comparativo sobre o desempenho das transações do sistema *Bitcoin* com e sem Assinatura de Grupo. Foi definido, através de formalizações, o anonimato e a auditoria da Assinatura de Grupo.

Na perspectiva de implementação, efetuou-se um estudo da API de desenvolvimento do sistema *Bitcoin*. Estudou-se detalhadamente a especificação original do *Bitcoin* e suas funções, registrando-se quais funções poderiam ser implementadas tal como especificadas e quais deveriam ser adaptadas. Nesse contexto, definiram-se funções e mecanismos que foram apenas citados, sem definição detalhada, ou parcialmente definidos na especificação original do *Bitcoin*.

Logo, implementou-se e configurou-se a Assinatura de Grupo no *Bitcoin* através de simulação. A implementação provê uma abstração para diversos partes do sistema *Bitcoin*, de modo que os processos em execução fossem simplificados mas próximos dos reais. Esse simulador é de suma importância para novas pesquisas acadêmicas pois através deste pode-se medir o desempenho de transações *Bitcoin* e comparar com outras técnicas além do anonimato em grupo a fim de validar uma outra abordagem.

A simulação foi de suma importância para verificar os limites da técnica de Assinatura de Grupo no sistema *Bitcoin* no que tange a quantidade de clientes em um grupo, a fim de realizar transações em tempo hábil em comparação com o sistema tradicional. Apesar de que foi exemplificado o anonimato e auditoria com apenas quatro clientes a fim de facilitar a explicação por meio de formalizações, mas vale resaltar que cada grupo pode conter cerca de 500 clientes por grupo com resultados positivos em relação ao anonimato e auditoria.

Essa pesquisa é importante para a indústria pois resolve problemas relacionados ao anonimato no sistema *Bitcoin*, aumentando assim a confiabilidade dessas empresas nesse sistema já que há um crescimento exponencial do uso dessa criptomoeda no mundo corporativo em todo mundo. Outro fator relevante mencionado por essa pesquisa é auditoria com consenso, único dentre os trabalhos relacionado, oferecendo maior segurança e transparência para prestação de contas para o setor público e privado.

Neste sentido, este trabalho contribui diretamente com o avanço no desenvolvimento do anonimato e auditoria por meio da técnica de Assinatura de Grupo no sistema *Bitcoin*, viabilizando o seu desempenho.

6.1 Aplicações

A proposta de uma solução para elevar o nível de anonimato em transações *Bitcoin* e permitir auditoria em tempo hábil através da técnica de Assinatura de Grupo é viável e aplicável. Portanto, sob o ponto de vista de implementação da técnica, observa-se resultados positivos em relação ao anonimato, auditoria e ao desempenho, desde que sejam criados grupos de até 500 clientes que usam Assinatura de Grupo no sistema *Bitcoin*, pois é possível obter resultados satisfatórios em relação ao anonimato em tempo hábil e permite-se auditoria nesse sistema. Porém, com grupos acima de 500 clientes a técnica de Assinatura de Grupo se torna ineficiente, pois eleva o tempo da transação acima de 10 minutos.

Como resultado final deste trabalho, mostra-se que é possível implementar um elevado nível de anonimato para os usuários *Bitcoin* usando uma técnica simples como a Assinatura de Grupo. A Assinatura de Grupo possibilita que se efetue o processo de auditoria em transações *Bitcoin* sem alterar a sua infra-estrutura e reusando tecnologias existentes.

A auditoria por meio da Assinatura do Grupo é eficiente em termos teóricos conforme as formalizações apresentadas na Seção 3.2, desde que as chaves públicas de quem transacionou e a assinatura propriamente dita não sejam disponibilizadas, a não ser que esse seja com o objetivo de permitir que todos possam acompanhar a transação. Portanto, a única forma de realizar auditoria é através das chaves públicas das *wallets* participantes das transações e da assinatura do respectivo grupo. De outra forma, não há como realizar o procedimento de auditoria e todas as transações e usuários *Bitcoin* serão anônimos.

6.2 Trabalhos Futuros

A seguir, apresentam-se as propostas de trabalhos futuros relacionados a Assinatura de Grupo no sistema *Bitcoin*:

1. é necessário implementar a Assinatura de Grupo no sistema *Bitcoin* atualmente em operação, a fim de verificar se os resultados obtidos em simulação são os mesmos em produção com milhões de transações sendo realizadas;

2. é importante verificar se em transações *Bitcoin* múltiplas e paralelas a Assinatura de Grupo é eficaz em relação ao seu desempenho;

3. pesquisar formas de otimizar o tempo das transações com grupos acima de 500 clientes;

4. aplicar técnicas de segmentação da rede *Bitcoin*, a fim de verificar como os grupos se comportam em relação a auditoria;

5. efetuar testes reais usando a rede de teste *testnet* do *Bitcoin* para homologar a Assinatura de Grupo e verificar se os resultados são próximos do apresentado aqui.

Referências

ALAJEELY, M.; AHMAD, A.; DOSS, R. Malicious node traceback in opportunistic networks using merkle trees. *IEEE International Conference on Data Science and Data Intensive Systems*, IEEE, p. 1–6, 2015.

BARRERA, A. *A Guide to Bitcoin (Part I): A look under the hood*. 2017. Acesso em Abril. Disponível em: <http://tech.eu/features/808/bitcoin-part-one/>.

BIRYUKOV, A.; KHOVRATOVICH, D.; PUSTOGAROV, I. Deanonymisation of clients in bitcoin p2p network. *CCS*, ACM, p. 1–15, 2014.

BRADBURY, D. Security bitcoin. *Engineering & Technology*, ACM, p. 68–71, 2015.

CORIGAN-GIBBS, H.; FORD, B. Dissent: Accountable anonymous group messaging. *CCS*, ACM, p. 1–11, 2010.

DANEZIS, G. et al. Pinocchio coin: Building zerocoin from a sunccinct pairing-based proof system. *PETShop*, ACM, p. 1–3, 2013.

DUPONT, J.; SQUICCIARINI, A. C. Toward deanonymizing bitcoin by mapping users location. *ODASPY*, ACM, p. 1–3, 2015.

EVANS-PUGHE, C.; NOVIKOV, A.; VITALIEV, V. To bit or not to bit? *Engineering & Technology*, ACM, p. 82–85, 2014.

GERVAIS, A. et al. On the privacy provisions of boom filters in lightweight bitcoin clients. *ACSAC*, ACM, p. 1–10, 2014.

GUERON, S.; JOHNSON, S.; WALKER j. Sha-512/256. *Eighth International Conference on Information Technology: New Generations*, IEEE, p. 1–5, 2011.

HEWETT, R.; KIJSANAYOTHIN, P. On securing privacy in composite web service transactions. *ICITST*, IEEE, p. 1–6, 2009.

HOBSON, D. What is bitcoin? *FALL*, ACM, p. 40–44, 2013.

HURLBURT, G. F.; BOJANOVA, I. Bitcoin: Benefit or curse? *IT Pro May/June*, IEEE, p. 1–6, 2014.

LAMBA, S.; SHARMA, M. An efficient elliptic curve digital signature algorithm (ecdsa). *Machine Intelligence and Research Advancement (ICMIRA)*, IEEE, p. 1–5, 2013.

LIBERT, B.; PETERS, T.; YUNG, M. Group signatures with almost-for-free revocation. *In Advances in Cryptology-CRYPTO*, Springer, p. 171–298, 2012.

MIERS, I. et al. Zerocoin: Anonymous distributed e-cash from bitcoin. *IEEE Symposium on Security and Privacy*, IEEE, p. 1–16, 2013.

NAKAMOTO, S. *Bitcoin: A Peer-to-Peer Eletronic Cash System*. 2017. Acesso em Abril. Disponível em: <https://bitcoin.org/bitcoin.pdf>.

NAKAMOTO, S. *Bitcoin Developer Guide*. 2017. Acesso em Abril. Disponível em: <https://bitcoin.org/en/developer-guide>.

PETERNELLI, L. A. *Capítulo 9 - Regressão linear e correlação*. 2017. Acesso em Maio. Disponível em: <http://www.dpi.ufv.br/~peternelli/inf162.www.16032004/index.html>.

ROTE N. VIJENDRAN, D. S. M. D. High performance sha-2 core using the round pipelined technique. *CONECCT*, IEEE, p. 1–6, 2015.

RUFFING P. MORENO-SANCHEZ, A. K. T. *CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin*. 2017. Acesso em Abril. Disponível em: <http://crypsys.mmci.uni-saarland.de/projects/CoinShuffle/coinshuffle.pdf>.

SALES, T. B. M. *Um Protocolo de Autenticação e Detecção de Ataques Sybil em Redes Ad Hoc Veiculares com Suporte ao Controle de Anonimato*. Tese (Doutorado) — UFCG, 2015.

SEBER, G. A. F.; LEE, A. j. *Linear Regression Analysis - 2 edition*. [S.l.]: Wiley, 2003. 1-582 p.

SMITH, P. *Total Number of Transactions Bitcoin*. 2017. Acesso em Abril. Disponível em: <https://blockchain.info/charts/n-transactions-total>.

TAAKI, A. *BIP Purpose and Guidelines*. 2017. Acesso em Abril. Disponível em: <https://github.com/bitcoin/bips/blob/master/bip-0001.mediawiki>.

TAYLOR, M. B. Bitcoin and the age bespoke silicon. *CASES*, IEEE, p. 1–10, 2014.

WALLACE, B. *The Rise and Fall of Bitcoin*. 2011. Acesso em Abril. Disponível em: <https://www.wired.com/2011/11/mf_bitcoin/>.

WEISS, W. A. R. *An Introduction to Set Theory*. 2017. Acesso em Abril. Disponível em: <http://math.toronto.edu/weiss/set_theory.pdf>.

YAU, C. *R Tutorial with Bayesian Statistics Using OpenBUGS*. 2017. Acesso em Maio. Disponível em: <http://www.r-tutor.com/elementary-statistics/simple-linear-regression/standardized-residual>.

ZIEGELDORF, J. H. et al. Coinparty: Secure multi-party mixing of bitcoins. *CODASPY*, ACM, p. 1–12, 2015.

Apêndices

APÊNDICE A – Abstrações Anonycoin

Alguns recursos definidos na especificação original do *Bitcoin*, não foram implementados na simulação. Esses recursos poderão ser desenvolvidos futuramente, caso necessário. Segue abaixo a lista dos recursos não implementados:

1. **Transação Gênesis**: na especificação original do *Bitcoin*, descrevem-se a transação gênesis ou *coinbase*. No contexto deste trabalho, não foi implementadas pois não diz respeito ao anonimato das transações *Bitcoin*. A complexidade poderia extrapolar o tempo definido para a pesquisa e desenvolvimento da simulação;

2. **Múltiplas transações:** o sistema *Bitcoin* permite múltiplas transações em um bloco. No contexto deste trabalho, esse procedimento não foi implementado a fim de simplificar as transações simuladas que ocorrem sequencialmente, uma por vez;

3. ***Mining***: nesse trabalho, o PoW foi simplificado a resolver *hashes* de SHA-256 e não SHA-512, pois o tempo e o poder computacional para resolver *hashes* do tipo SHA-512 poderia elevar o tempo para obter dados essenciais para essa pesquisa;

4. **Blockchain**: na especificação original do *Bitcoin*, descrevem-se *blockchain* como pequenos blocos interligados entre si que formam um grande histórico das transações *Bitcoin* através de *merkle tree*. No contexto deste trabalho, esse processo foi simplificado gerando sequencialmente os blocos pelos mineradores a fim de diminuir a complexidade. Além de que, a sua implementação inicial não interfere no quesito anonimato;

5. **Ataques de negação de serviço**: na especificação original do *Bitcoin*, há abordagens sobre ataques de negação de serviços, como o ataque do 51%, que pode inviabilizar o sistema *Bitcoin*. No contexto deste trabalho, não foi aplicado ataques na simulação. Partindo do ponto que as simulações foram executadas em um ambiente controlado por meio de simulação;

6. **Linguagem de Script**: na especificação original do *Bitcoin*, descreve-se um tipo de linguagem de programação, que não disponibiliza *loops* ou recursividade. A função dessa linguagem é validar as transações, portanto não aborda o tema anonimato em transações *Bitcoin*. Logo foi simplificada a validar a quantia de *bitcoin* transacionadas e a assinar essas transações.

Anexos

ANEXO A – Código Fonte *Anonycoin*

Neste Apêndice, apresentam-se o código fonte da simulação simplificada do atual sistema *Bitcoin* e a inclusão da Assinatura de Grupo nesse sistema. Essa simulação foi nomeada de *Anonycoin (Anonymity + Bitcoin)*.

Alguns arquivos como *README*, *COPYING*, *requeriments* e arquivos de documentação no formato HTML não foram incluídos neste Apêndice a fim de simplificar essa seção e apresentar somente os códigos fontes mais relevantes para esse documento.

O *Anonycoin* é *software* livre sob a licença *GNU GENERAL PUBLIC LICENSE* (GPL) versão 3 como pode ser conferido no arquivo *COPYING* do projeto.

O código fonte completo do *Anonycoin* está disponível no repositório na seguinte URL <https://gitlab.com/d4n1/anonycoin>.

O *Anonycoin* foi desenvolvido na língua inglesa com o intuito de ser universal, permitindo assim o seu uso por qualquer pessoa que saiba um pouco de inglês técnico.

A.1 Instalando o *Anonycoin*

O *Anonycoin* foi desenvolvido na linguagem de programação Python na versão 3, e todos os software usados são *softwares* desenvolvidos em Python. Portanto, para realizar a instalação é preciso instalar as seguintes dependências: ecdsa==0.13, pep8==1.7.0, pyfiglet==0.7.4, pyzmq==15.2.0 e requests==2.9.1. Todos esses pacotes Python estão no arquivo *requeriments*, logo deve-se executar o comando *pip install -r requeriments*.

A.2 *run.py*

O arquivo *Python run.py*, contém a lógica principal da simulação *Anonycoin*. Ao executar esse arquivo, há 4 opções em forma de menu. Observa-se a tela principal do *Anonycoin*, executando o arquivo *run.py* como mostrado na Figura 21.

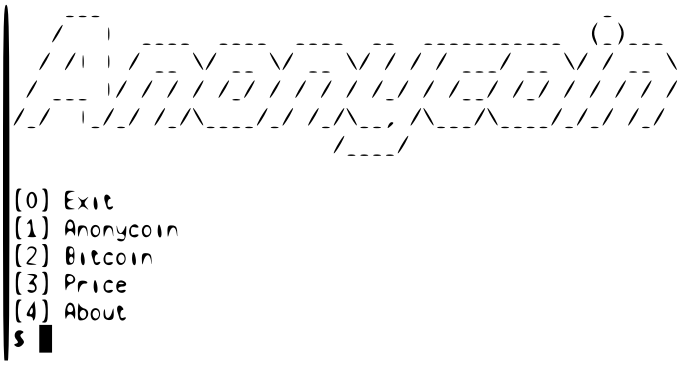

Figura 21 – *Anonycoin*: Simulador *Bitcoin*.

As opções desse menu são descritas abaixo:

- *Exit*: Finaliza a execução do sistema caso a opção escolhida não seja os números de 1 à 4;
- *Anonycoin*: Inicializa a execução da simulação do sistema *Bitcoin* com inclusão da Assinatura de Grupo;
- *Bitcoin*: Inicializa a execução da simulação do sistema *Bitcoin* sem a Assinatura de Grupo;
- *Price*: Retorna o valor atual da cripto-moeda *Bitcoin* em dólar;
- *About*: Exibe uma breve descrição sobre o *Anonycoin* com o nome do autor e seus *copyrights*.

O código visto em *run.py*, inicializa a rede, instância algumas classes e executa algumas funções. Esse código garante que pelos menos 2 *wallets* distintas serão criadas e ao menos 1 transação.

Executando-se o *run.py*, são gerados 2 arquivos do tipo CSV: *bitcoin-time.csv* e *anonycoin-time.csv*. O primeiro arquivo contém o tempo de execução de cada transação *Bitcoin* sem Assinatura de Grupo, e o segundo arquivo contém o tempo de execução de cada transação com Assinatura de Grupo.

O código fonte do arquivo *run.py* é apresentado a seguir:

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
    signature

Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.

Anonycoin is free software; you can redistribute it and/or modify it
```

```
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""

from random import randrange, uniform
from itertools import permutations
from shutil import move
from time import clock
from pyfiglet import Figlet
from wallet import Wallet
from network import Publisher, Subscriber
from transaction import Transaction
from miner import Miner
from btc import btc_price
from group import Group


BTC = 1
WALLETS = 32
choice = 0

def menu():
    """Menu screen."""

    ascii = Figlet(font="slant")
    print(ascii.renderText("Anonycoin"))
    print("[0] Exit")
    print("[1] Anonycoin")
    print("[2] Bitcoin")
    print("[3] Price")
    print("[4] About")
    global choice
    choice = input("$ ")

def run():
    """Run Bitcoin simulator."""
    menu()
```

```
    while (choice != "0"):
        block = 10000

        if choice == "1":
            for b in range(10):
                wallets = []

                for w in range(WALLETS):
                    wallet = Wallet()
                    wallets += [wallet]

                publisher = Publisher()
                publisher.run()

                p = permutations(wallets, 2)
                g = Group()

                for t in p:
                    start_clock = clock()

                    transaction = Transaction(t[0], t[1], BTC)
                    transaction.make()
                    t[0].send(t[1], BTC)

                    subscriber = Subscriber()
                    subscriber.run()
                    publisher.broadcast("btc")

                    miner = Miner(transaction)

                    miner.pow(block)

                    t[0].sign = g.sign(t[0].hash_key)

                    miner.transaction.inputs = g.hash_key
                    miner.transaction.outputs = g.hash_key

                    miner.mining(b, "anonycoin.csv")

                    stop_clock = (clock() - start_clock) * 60

                    with open("anonycoin-time.csv", "a", encoding="utf-8
                        ") as f:
                        f.write(str(stop_clock) + "\n")

                    block += 1
```

```
elif (choice == "2"):
    for b in range(10):
        wallets = []

        for w in range(WALLETS):
            wallet = Wallet()
            wallets += [wallet]

        publisher = Publisher()
        publisher.run()

        p = permutations(wallets, 2)
        miners = []

        for t in p:
            start_clock = clock()

            transaction = Transaction(t[0], t[1], BTC)
            transaction.make()

            t[0].send(t[1], BTC)

            subscriber = Subscriber()
            subscriber.run()
            publisher.broadcast("btc")

            miner = Miner(transaction)

            miner.pow(block)
            miner.mining(b, "bitcoin.csv")

            stop_clock = (clock() - start_clock) * 60

            with open("bitcoin-time.csv", "a", encoding="utf-8")
                as f:
                f.write(str(stop_clock) + "\n")

            block += 1

elif (choice == "3"):
    btc_price()

elif (choice == "4"):
    print("""
    Anonycoin --- Bitcoin simulator with anonymity throught
        group signature
```

```
            Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

            This file is part Anonycoin.

            Anonycoin is free software; you can redistribute it and/or
                modify it
            under the terms of the GNU General Public License as
                published by
            the Free Software Foundation; either version 3 of the
                License, or (at
            your option) any later version.

            Anonycoin is distributed in the hope that it will be useful,
                 but
            WITHOUT ANY WARRANTY; without even the implied warranty of
            MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See
                the
            GNU General Public License for more details.

            You should have received a copy of the GNU General Public
                License
            along with Anonycoin.  If not, see <http://www.gnu.org/
                licenses/>.
            """)

        menu()
    exit(0)


if __name__ == '__main__':
    run()
```

A.3 *wallet.py*

O arquivo *wallet.py*, define a classe *Wallet* que implementa uma carteira virtual *Bitcoin*. Usa-se o mesmo tipo de criptografia (ECDSA) do *Bitcoin*.

Essa classe possui 4 atributos a saber:

- *secret_key*: Chave privada gerada pela função *SigningKey* do pacote *Python ecdsa*;
- *public_key*: Chave pública gerada a partir da chave privada;
- *hash_key*: *Hash* da chave pública;

- *btc*: Quantidade de *bitcoins* fracionadas de 10 a 100. A quantidade é gerada randomicamente pela função *Python uniform*;
- *IP*: O endereço IP desse cliente.

E implementa 3 métodos:

- *__init__*: Método construtor da classe;
- *send*: Envia uma determinada quantia de *bitcoin*, decrementando o saldo atual de *bitcoins*;
- *receive*: Recebe uma determinada quantia de *bitcoin*, incrementando o saldo atual de *bitcoins*.

O código fonte de *wallet.py*, pode ser observado a seguir:

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
   signature

Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.

Anonycoin is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""

from random import uniform, randrange
from ecdsa import SigningKey
from ipaddress import ip_address


class Wallet:
    """Wallet Bitcoin simulator thread ECDSA (SHA1)."""

    def __init__(self):
```

```
        """Constructor.
        secret_key - wallet secret key
        public_key - wallet public key
        hash_key - wallet public key hash
        btc - btc amount
        ip - ip address
        """

        self.secret_key = SigningKey.generate()
        self.public_key = self.secret_key.get_verifying_key()
        self.hash_key = self.public_key.to_string()
        self.btc = round(uniform(10, 100), 2)
        self.ip = ip_address(randrange(100))
        self.sign = ""

    def send(self, wallet, btc):
        """Send Bitcoin.
        wallet - wallet address
        btc - btc amount
        """

        wallet.receive(btc)
        self.btc - btc

    def receive(self, btc):
        """Receive Bitcoin.
        btc - btc amount
        """

        self.btc + btc
```

A.4 *network.py*

O arquivo *network.py*, define a classe *Network* que implementa uma rede do tipo P2P. Foi usado o *software* distribuído *ZeroMQ*, que é usado pelo *Bitcoin*. Nesse arquivo há a implementação de 2 classes: *Publisher* e *Subscriber*.

A classe *Publisher* exerce a função de servidor, possuindo 2 atributos:

- *context*: Contexto do pacote ZMQ;
- *socket*: *Socket* do contexto do tipo *publisher* ZMQ.

E implementa 3 métodos:

- *__init__*: Método construtor da classe;
- *run*: Inicializa o servidor via protocolo TCP em qualquer endereço de rede e na porta 4444;
- *broadcast*: Envia uma mensagem para todos os nós que estão conectados na rede.

A classe *Subscriber* exerce a função de cliente, possuindo 2 atributos:

- *context*: Contexto do pacote ZMQ;
- *socket*: *Socket* do contexto do tipo *subscriber* ZMQ.

E implementa 2 métodos:

- *__init__*: Método construtor da classe;
- *run*: Inicializa o cliente via protocolo TCP no endereço local, *localhost* ou 127.0.0.1, e na porta 4444.

O código fonte de *network.py*, está disponível a seguir:

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
   signature

Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.

Anonycoin is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""

from random import randrange
import zmq
```

```
class Publisher:
    """Network Bitcoin simulator server (ZMQ)."""

    def __init__(self):
        """Constructor.
        context - context zmq
        socket - socket p2p
        """

        self.context = zmq.Context()
        self.socket = self.context.socket(zmq.PUB)

    def run(self):
        """Run thread."""

        self.socket.bind("tcp://*:4444")

    def broadcast(self, message):
        """Broadcast message.
        message - text message
        """

        message = bytes(message, 'utf8')
        self.socket.send(message)


class Subscriber:
    """Network Bitcoin simulator client."""

    def __init__(self):
        """Constructor.
        context - context zmq
        socket - socket p2p
        """

        self.context = zmq.Context()
        self.socket = self.context.socket(zmq.SUB)

    def run(self):
        """Run thread,"""
        self.socket.connect("tcp://localhost:4444")
```

A.5 *transaction.py*

O arquivo *transaction.py*, define a classe *Transaction* que implementa uma transação *Bitcoin* entre 2 *wallets* distintas.

Uma transação *Bitcoin* possui vários campos, na simulação foi implementado : *Inputs, Outputs, BTC, Sign.* As *inputs* referência os *outputs* da transação anterior, formando uma cadeia de transações.

A classe *Transaction* possui 4 atributos:

- *inputs*: A *wallet* que envia *bitcoins* na transação;
- *outputs*: A *wallet* que recebe *bitcoins* na transação;
- *btc*: Quantidade de *bitcoins* transacionados;
- *Sign*: A assinatura da transação.

E implementa 2 métodos:

- __init__: Método construtor da classe;
- *make*: Realiza uma transação *bitcoin*, assinando as *wallets* participantes. Esse método também reatribue às *inputs* os valores dos *hashes* da transação e a sua assinatura. Os *outputs* atribui a quantidade de *bitcoins* transacionados e o *hash* da *wallet* do receptor.

O código fonte de *transaction.py* está disponível logo abaixo:

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
   signature

Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.

Anonycoin is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""

from script import Script


class Transaction:
    """Transaction Bitcoin simulator thread."""

    def __init__(self, inputs, outputs, btc):
        """Constructor.
        inputs - wallet input
        outputs - wallet output
        btc - btc amount
        sign - sign transaction
        """

        self.inputs = inputs
        self.outputs = outputs
        self.btc = btc
        self.sign = ""

    def make(self):
        """Make transacton."""

        script = Script(self.inputs.secret_key)
        self.inputs = self.inputs.hash_key
        self.outputs = self.outputs.hash_key
        self.sign = script.sign(self.inputs)
```

A.6 *script.py*

O arquivo *script.py*, define a classe *Script* que implementa a assinatura de uma determinada transação usando o algoritmo ECDSA.

Para o sistema *Bitcoin*, foi desenvolvido uma linguagem de programação especifica chamada de *Script*. Essa linguagem realiza algumas validações como a assinatura da transação. O *Script*, não tem suporte a *loop* ou chamadas recursivas a fim de não ocorrer a possibilidade da rede parar por meio de ataques de negação de serviço.

Essa classe possui 3 atributos:

- *secret_key*: Chave privada gerada pela função *SigningKey* do pacote *Python ecdsa*;
- *public_key*: Chave pública gerada a partir da chave privada;

- *hash_key*: *Hash* da chave pública.

E implementa 3 métodos:

- *__init__*: Método construtor da classe;
- *sign*: Assina uma mensagem com a chave privada;
- *verify*: Verifica se a assinatura e a mensagem são válidas através da chave pública.

O código fonte de *script.py* está disponível a seguir:

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
   signature

Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.

Anonycoin is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""

from ecdsa import SigningKey


class Script:
    """Script language Bitcoin simulator"""

    def __init__(self, secret_key):
        """Constructor.
        secret_key - script secret key
        public_key - script public key
        hash_key - script public key hash
        """

        self.secret_key = secret_key
```

```
        self.public_key = self.secret_key.get_verifying_key()
        self.hash_key = self.public_key.to_string()

    def sign(self, message):
        """Signature.
        message - signature message
        """

        try:
            return self.secret_key.sign(bytes(message))
        except:
            return False

    def verify(self, sign, message):
        """Verify signature.
        sign - signature to be validated
        message - signature message
        """

        try:
            assert self.public_key.verify(sign, bytes(message))
            return True
        except:
            return False
```

A.7 *miner.py*

O arquivo *miner.py* define a classe *Miner* que implementa a mineração de *Bitcoin*. O PoW é gerado através da resolução de *hashes* SHA-256. Não é gerado o bloco gênesis, ou *coinbase*.

Essa classe possui 3 atributos:

- *transaction*: A transação;
- *timestamp*: Um conjunto de data, hora e fuso horário;
- *difficulty*: A complexidade para resolver o *hash* da transação.

E implementa 3 métodos:

- *___init___*: Método construtor da classe;
- *pow*: Calcula a dificuldade para resolver a transação;

- *mining*: Gera o bloco que será enviado para a *blockchain*.

O código fonte de *miner.py* está disponível logo abaixo:

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
   signature

Copyright(c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.

Anonycoin is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""

from datetime import datetime
import threading
import hashlib
from random import randrange
from blockchain import Blockchain


class Miner:
    """Miner Bitcoin simulator thread."""

    def __init__(self, transaction):
        """Constructor.
        transaction - transactor to be validated
        timestamp - mining timestamp
        difficulty - difficulty level to validate the transaction
        """

        self.transaction = transaction
        self.timestamp = datetime.now()
        self.difficulty = 0

    def pow(self, proof):
        """Proof-of-Work Bitcoin simulator using SHA-256."""
```

```
        secret = hashlib.sha256(str.encode(str(proof))).hexdigest()

        for i in range(1000000000):
            work = hashlib.sha256(str.encode(str(i))).hexdigest()
            if work == secret:
                self.difficulty = secret
                break

    def mining(self, block, simulation):
        """Mining Bitcoin simulator."""

        blockchain = Blockchain(block, self.transaction,
                                self.timestamp, self.difficulty)
        blockchain.blocking(simulation)
```

A.8 *blockchain.py*

O arquivo *blockchain.py*, define a classe *Blockchain* que implementa a base de dados que armazena todas as transações realizadas pelo sistema *Bitcoin*. A classe *Blockchain*, não gera o *merkle root*.

Essa classe possui 6 atributos:

- *block*: O bloco da *blockchain*;
- *transaction_input*: A transação de entrada;
- *transaction_output*: A transação de saída;
- *btc*: A quantidade BTC transacionado;
- *timestamp*: Um conjunto de data, hora e fuso horário;
- *difficulty*: A complexidade para resolver o *hash* SHA-256 da transação gerado pela classe *Miner*.

E implementa 2 métodos:

- ___*init*___: Método construtor da classe;
- *blocking*: Gera duas *blockchain* (*bitcoin* e *anonycoin*) com todas as transações realizadas com êxito. Essas *blockchains* são disponibilizadas em arquivos do tipo CVS.

O código fonte de *blockchain.py*, é apresentado a seguir:

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
    signature

Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.

Anonycoin is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""


class Blockchain:
    """Blockchain Bitcoin simulator."""

    def __init__(self, block, transaction, timestamp, difficulty):
        """Constructor.
        block - current block header
        transaction - transaction in the current block
        timestamp - timestamp generated by mining
        difficulty - difficulty level generated by mining
        """

        self.block = block
        self.transaction_inputs = transaction.inputs
        self.transaction_outputs = transaction.outputs
        self.btc = transaction.btc
        self.timestamp = timestamp
        self.difficulty = difficulty

    def blocking(self, simulation):
        """Make block Bitcoin writing Blockchain in CVS file.
        simulation - 0 (default Bitcoin) or 1 (bitcoin with group
            signature)
        """
```

```
        with open(simulation, "a", encoding="utf-8") as f:
            f.write(str(self.block) +
                    "\t" +
                    str(self.transaction_inputs) +
                    "\t" +
                    str(self.transaction_outputs) +
                    "\t" +
                    str(self.btc) +
                    "\t" +
                    str(self.timestamp) +
                    "\t" +
                    str(self.difficulty) +
                    "\n")
```

A.9 *group.py*

O arquivo *group.py*, define a classe *Group* que implementa a Assinatura de Grupo usando o algoritmo ECDSA.

Essa classe possui 3 atributos:

- *secret_key*: A chave privada de gerenciamento do grupo;
- *public_key*: A chave pública de do grupo;
- *hash_key*: O *hash* gerado a partir da da chave pública.

E implementa 3 métodos:

- *__init__*: Método construtor da classe;
- *sign*: Método que assina o grupo baseado em uma mensagem passado como parâmetro;
- *verify*: Método que verifica a assinatura do grupo, recebendo a assinatura do grupo e a mensagem passado como parâmetro;

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
   signature

Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.
```

```
Anonycoin is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""

import requests


def btc_price():
    """Return current BTC price."""

    try:
        url = "https://blockchain.info/stats?format=json"
        json = requests.get(url).json()
        btc = json["market_price_usd"]

        print("1 BTC = US$ " + str(btc))
    except:
        print("Without network!")
```

A.10 *btc.py*

O arquivo *btc.py*, define a função *btc_price* que retorna o atual valor da criptomoeda *Bitcoin* em dólar.

Essa função, usa o método *requests* para obter um arquivo do tipo JSON do site <http://blockchain.info>.

O código fonte de *btc.py*, está disponível abaixo:

```
"""Anonycoin --- Bitcoin simulator with anonymity throught group
    signature

Copyright (c) 2016 Daniel Pimentel <d4n1@d4n1.org>

This file is part Anonycoin.

Anonycoin is free software; you can redistribute it and/or modify it
```

```
under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 3 of the License, or (at
your option) any later version.

Anonycoin is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with Anonycoin.  If not, see <http://www.gnu.org/licenses/>.
"""

import requests


def btc_price():
    """Return current BTC price."""

    try:
        url = "https://blockchain.info/stats?format=json"
        json = requests.get(url).json()
        btc = json["market_price_usd"]

        print("1 BTC = US$ " + str(btc))
    except:
        print("Without network!")
```

A.11 *plots.r*

O arquivo *plots.r*, define as funções usadas pela linguagem R para plotar os gráficos e realizar as análises estatísticas do *Bitcoin* com e sem Assinatura de Grupo baseado nas respectivas *blockchains* geradas.

O código fonte de *plot.r*, está avaliada abaixo:

```
## time

anonycoin <- read.csv(file="anonycoin-time.csv")
bitcoin <- read.csv(file="bitcoin-time.csv")
max_y <- max(anonycoin[,1])
plot_colors <- c("red", "blue")
pdf("analysis-time.pdf")
plot(anonycoin[,1], col=plot_colors[1], ylim=c(0, max_y), xlab="Numero
   de Transacoes", ylab="Tempo (minuto)")
box()
```

```
lines(bitcoin[,1], pch=22, lty=8, col=plot_colors[2])
title(main="Desempenho Bitcoin e Anonycoin", col.main="red", font.main
    =4)
legend(1, 3, c("Anonycoin", "Bitcoin"), cex=0.8, col=plot_colors, pch
    =21:23,  lty=1:3)

## statistic
anonycoin <- read.csv(file="anonycoin-time.csv")
bitcoin <- read.csv(file="bitcoin-time.csv")

lmfunction<-lm(anonycoin$time[0:4999]~bitcoin$time[0:4999])

summary(lmfunction)

#Call:
#lm(formula = anonycoin$time[0:4999] ~ bitcoin$time[0:4999])

#Residuals:
#      Min        1Q    Median        3Q       Max
#-1.31762 -0.19106  0.00052  0.18348  2.71269

#Coefficients:
#                      Estimate Std. Error t value Pr(>|t|)
#(Intercept)          4.909518   0.042577   115.3   <2e-16 ***
#bitcoin$time[0:4999] 0.847132   0.008141   104.1   <2e-16 ***
#---
#Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

#Residual standard error: 0.2736 on 4997 degrees of freedom
#Multiple R-squared:  0.6843,   Adjusted R-squared:  0.6842
#F-statistic: 1.083e+04 on 1 and 4997 DF,  p-value: < 2.2e-16

shapiro.test(rstudent(lmfunction))

#Shapiro-Wilk normality test

#data:  rstudent(lmfunction)
#W = 0.98593, p-value < 2.2e-16

pdf("lm-analysis.pdf")
plot(rstudent(lmfunction) ~ fitted(lmfunction), pch=19, xlab="Filtrado",
     ylab="Residual")
abline(h=0, lty=2)

pdf("lm-graph.pdf")
plot(anonycoin$time~bitcoin$time, xlab="Bitcoin", ylab="Anonycoin")
abline(lmfunction, lty=2)
```

ANEXO B – Algoritmo de Assinatura Digital de Curva Elíptica

Neste anexo, apresentam-se detalhes sobre o Algoritmo de Assinatura Digital de Curva Elíptica (Elliptic Curve Digital Signature Algorithm - ECDSA) usado no sistema *Bitcoin* e também no simulador *Anonycoin*.

B.1 Curva Elíptica

As curvas elípticas tem ligação com integrais elípticas da matemática, origem do seu nome. As curvas elípticas podem ser utilizadas para determinar o comprimento do arco de uma elipse. Estes podem ser definidos como um conjunto de pontos discretos no plano de coordenadas, satisfazendo a Equação B.1.

$$y^2[+xy] = x^3 + ax^2 + b \pmod p \tag{B.1}$$

O ECDSA são formações de pares de chaves, chave privada e chave pública. Essas chaves são usadas na geração e verificação da assinatura. Lamba ((LAMBA; SHARMA, 2013)) disponibiliza os algoritmos que descrevem os passos para geração das chaves pública e privada, bem como a geração e verificação de assinatura.

B.1.1 Geração de Chaves

Dado um ponto "G", a chave privada "d" e a chave pública pode ser gerada através dos seguintes passos:

1. Selecionar um inteiro randômico "d" no intervalo de $[0, n-1]$;
2. Computar $Q = d \times G$, obtendo um ponto de multiplicação. "Q" e "G" são pontos da curva elíptica;
3. O par de chaves é (d, Q) onde "d" é a chave privada e "Q" a chave pública.

B.1.2 Geração de Assinatura

Usando a chave privada, a assinatura pode ser gerada através de uma mensagem "M" conforme a seguir:

1. Escolher um inteiro randômico "k" com $1 \leq k \leq n-1$;
2. Computar $k \times G = (x_1, y_1)$ e $r = x_1 \bmod n$. Se $r = 0$ então retorna para o passo 1;
3. Computar $k^{-1} \bmod n$;
4. Computar $z = h^{-1}(M)^2$;
5. Calcular $s = k^{-1}(z + d \times r) \bmod n$. Se $s = 0$ então retorna para passo 1;
6. A assinatura do *hash* da mensagem "z" é (r, s).

B.1.3 Verificação de Assinatura

A autenticidade da mensagem recebida pode ser verificada pelos seguintes passos:

1. Verificar se "r" e "s" são inteiros no intervalo $[1, n-1]$;
2. Computar $z = h^{-1}(M)$;
3. Computar $w = s^{-1} \mod n$;
4. Computar $u_1 = z \times w \pmod n$ e $u_2 = r \times w \pmod n$;
5. Computar $X = u_1 G + u_2 Q$. Se $X = O_\infty$ então será rejeitado a assinatura;
6. Caso contrário, computar $v = x_1 \bmod n$ onde $X = (x_1, y_1)$;
7. Aceitar a assinatura se e somente se $v = r$.

ANEXO C – Algoritmo de *Hash* Seguro

Neste anexo, apresentam-se detalhes sobre o Algoritmo de *Hash* Seguro (*Secure Hash Algorithm* - SHA) usado no sistema *Bitcoin* e também no simulador *Anonycoin*.

Algoritmos de *hashing* são funcionalidades rápidas, seguras e robustas para verificar a autenticidade de uma transação computacional. O SHA-512 opera sobre 80 repetições em 128 blocos de bytes (GUERON; JOHNSON; WALKER, 2011).

A partir de uma dada *string*, é possível gerar um *hash* único. Esse *hash* serve para verificar a integridade de um arquivo, *software* ou o que se possa imaginar. O *Bitcoin* usa o *hash* do tipo SHA-512 para validar as *wallets* e suas transações.

Na Figura 22, há a arquitetura simplificada de um algoritmo de *hashing* do tipo SHA-512.

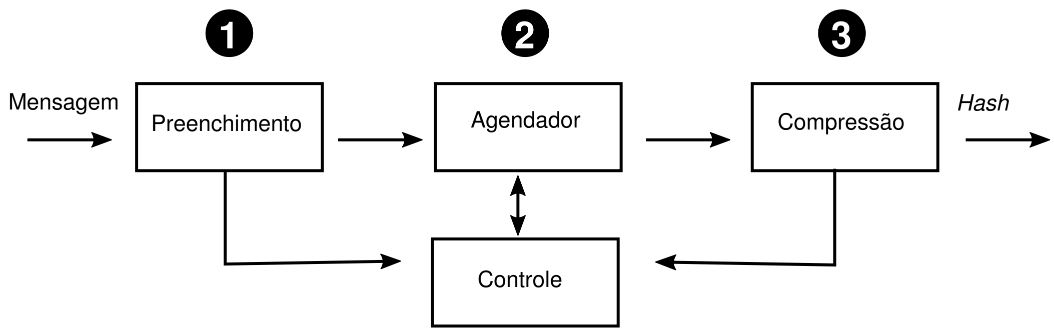

Figura 22 – Arquitetura simplificada de um algoritmo SHA-512 (Imagem adaptada de (ROTE N. VIJENDRAN, 2015))

1. Após receber uma dada mensagem, executa-se algoritmos matemáticos que cifram o texto na unidade de Preenchimento. Essa unidade garante que a cifra será única baseada no texto da mensagem;

2. A mensagem cifrada pode ser encaminhada para a unidade de Agendamento ou para a unidade de Controle. A unidade de Agendamento define quantas compressões serão realizadas. A unidade de Controle irá gerenciar as mensagens;

3. A unidade de Compressão irá finalizar o *hash*. Caso preciso, o *hash* pode voltar para a unidade de Controle até que o processo seja finalizado com êxito.