Um mecanismo leve de consenso e confiança para controle de acesso em redes IoT baseadas em Blockchain

Afonso Barbosa de Souza Neto, Paulo A. L. Rego e Marcos D. Ortiz

[1]Universidade Federal do Ceará (UFC)
Avenida José de Freitas Queiroz, 5003 – Cedro – Quixadá – Ceará

afonsoneto121@gmail.com, {pauloalr,mdo}@ufc.br

Abstract. *Since its creation, the Internet has undergone several transformations, among them the Internet of Things (IoT), which is composed of a set of objects connected to the Internet in order to provide services to users. The idea is to connect the physical and digital world. To ensure the security of devices and users, security mechanisms need to meet the characteristics inherent to IoT. In this context, this paper proposes a security mechanism for IoT based on Blockchain technology, which is a model that seeks decentralization as a security measure. We use the same concept of Blockchain applied in crypto-coins, but with a consensus mechanism based on trust between nodes. A prototype of the solution was developed and evaluated to demonstrate that the proposed solution is capable of obtaining consistent results.*

Resumo. *Desde sua criação, a Internet vem sofrendo diversas transformações, dentre elas a Internet das Coisas (do inglês, Internet of Things - IoT), que é composta por um conjunto de objetos diversos conectados à Internet e que provêm serviços aos usuários. A ideia é, cada vez mais, conectar o mundo físico ao digital. Para assegurar a segurança desses dispositivos, e dos próprios usuário, é necessário que os mecanismos de segurança atendam às características próprias da IoT. Nesse contexto, este trabalho apresenta uma proposta de segurança para IoT baseada em Blockchain, onde é utilizado um mecanismo de consenso baseado na confiança entre os nós. Um protótipo da solução foi desenvolvido e avaliado para demonstrar que o mecanismo proposto é capaz de obter resultados consistentes.*

1. Introdução

A Internet das Coisas (IoT) vem atraindo a atenção de pesquisadores e entusiastas da área de tecnologia por ser considerada uma evolução da Internet. Além de permitir a interação entre os mundos físico e virtual, a IoT também colhe informações do ambiente e usa padrões existentes na Internet para prover serviços e análises [Gubbi et al. 2013].

A IoT tem impactos significativos em vários aspectos da vida cotidiana e comportamento de usuários em potencial. Nesse contexto, a automação residencial e a vida assistida são apenas alguns exemplos de possíveis cenários em que o novo paradigma tem um papel importante [Atzori et al. 2010]. Essa natureza pervasiva da IoT em diversas aplicações do cotidiano das pessoas impõe requisitos de segurança para suas transações e controle de acesso [Farooq et al. 2015]. Ao mesmo tempo, os modelos de segurança tradicionais tendem a centralizar todo o processamento em um único local, o que acaba

prejudicando a escalabilidade [Roman et al. 2013]. A *Blockchain* (BC) é naturalmente descentralizada e tem recursos de segurança para atender tais requisitos.

A *Blockchain* foi proposta junto com a criação do *Bitcoin*, a primeira criptomoeda totalmente descentralizada [Nakamoto 2008], cujo objetivo é deixar o processo de autenticação e criação de novas moedas com os membros da rede de maneira confiável e auditável. Usuários *Bitcoin* são conhecidos por uma chave pública modificável, gerando e transmitindo transações para a rede para transferir dinheiro. Essas transações são armazenadas em blocos, quando um bloco está cheio, ele é encadeado à BC por meio de um processo de mineração. Para minerar um bloco, alguns nós específicos na rede tentam resolver um quebra-cabeça criptográfico chamado *Proof of Work* (prova de trabalho). O primeiro nó que conseguir resolver o quebra-cabeça envia o bloco para a BC. Essa prova de trabalho tem a característica de ser muito difícil de solucionar e muito fácil de verificar se está correta [Dorri et al. 2017a]. Devido a esse mecanismo de consenso entre os nós, a BC possui algumas vantagens, tais como maior transparência, uma vez que todas as transações são publicas e auditáveis; e menos intermediários.

Os dispositivos IoT podem se beneficiar da natureza descentralizada da BC como medida de segurança, onde nós podem trocar informação de maneira confiável, assim como acontece no *Bitcoin* [Greve et al. 2018]. Os vários benefícios proporcionados pela BC a tornam uma solução atraente para abordar o problema de autenticação e controle de acesso em IoT. Contudo, a forma como ela está implementada no *Bitcoin* não pode ser diretamente adotada para IoT pelos seguintes motivos [Dorri et al. 2017b]:

- Complexidade do algoritmo de consenso: o *Proof of Work*, usado no *Bitcoin*, requerer um grande poder computacional e consumo de energia, que são recursos escassos em dispositivos IoT.
- Latência: existe um atraso associado à confirmação de um novo bloco na rede *Bitcoin*, uma vez que as transações podem levar algumas horas para serem confirmadas. No entanto, isso não é um problema para tais redes. Já os usuários e aplicações de redes IoT possuem requisitos de tempo de resposta mais rigorosos.

Dessa forma, este trabalho propõe um mecanismo de controle de acesso, baseado em *Blockchain* para cenários IoT, que implementa um modelo de consenso e confiança mais adequado à natureza dessas redes.

Este artigo está organizado da seguinte forma. A Seção 2 apresenta a solução proposta, a Seção 3 apresenta os experimentos realizados para avaliar nossa proposta, enquanto a Seção 4 apresenta os trabalhos relacionados e a Seção 5 conclui o trabalho.

2. Mecanismo Proposto

As soluções de segurança convencionais requerem a existência de uma entidade central e/ou grande capacidade de processamento para validar as transações dos usuários. Além dos problemas inerentes à existência de uma entidade central (e.g., ponto único de falha e não suporte à escalabilidade), é necessário que os participantes confiem nessa entidade central e entre si.

O mecanismo aqui proposto utiliza BC para atender os requisitos de escalabilidade, enquanto implementa um algoritmo de consenso e confiança mais leve para reduzir os requisitos de processamento. Nas demais seções, vamos considerar um contexto de casas/prédios inteligentes, mas a solução pode ser aplicada a outros cenários de IoT.

2.1. Rede de Sobreposição

A rede consiste em dois níveis principais: a casa inteligente (do inglês, *Smart Home* - SH) e uma rede de sobreposição. Dispositivos IoT estão localizados no nível SH, enquanto a rede de sobreposição é formada por um conjunto de SH conectadas - semelhante a uma rede *peer-to-peer* que implementa uma arquitetura distribuída completamente descentralizada, em que todos os nós são equivalentes em termos de funcionalidade e tarefas que executam [Barcellos and Gaspary 2006].

Na rede de sobreposição, as SHs são agrupadas em *clusters* e cada *cluster* elege um líder (do inglês, *Cluster Head* - CH) que será responsável por armazenar a *Blockchain* e interagir com outros CHs. Espera-se que um nó selecionado como CH permaneça *online* por um longo período de tempo e tenha recursos suficientes, visto que processam transações de entrada e transações de saída que são geradas por outros CHs. Quando um dispositivo localizado na SH solicitar acesso a outro dispositivo localizado em outra SH, uma transação é criada por seu líder (CH) e enviada para os demais CHs para assim ser validada. A transação é armazenada em um local separado até a escolha de um novo nó minerador para juntar todas as transações em um bloco, validá-las e enviá-las para a rede. A Figura 1 ilustra a interação entre os diferentes membros da rede.

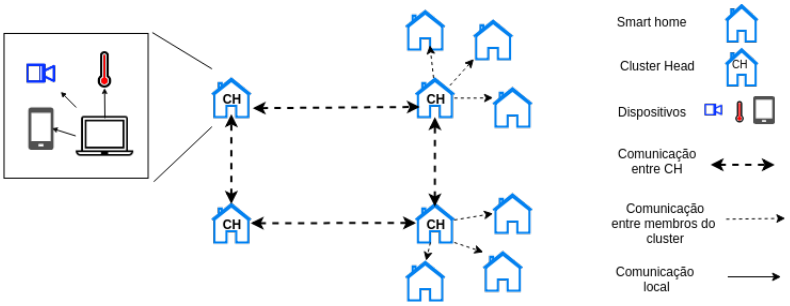


Figura 1. Cenário do trabalho proposto

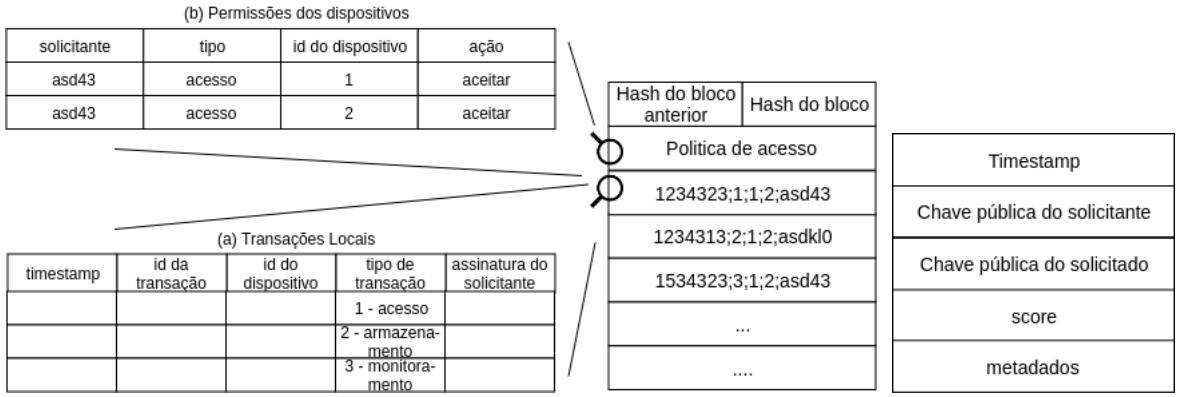
Cada SH possui uma *Blockchain* local para armazenar as permissões de cada dispositivo. Dessa forma, ela gerencia todas as transações locais de entrada e saída.

2.2. Estrutura dos blocos

Os blocos são usados para armazenar as transações geradas pelos CHs. Uma transação é definida como uma solicitação de acesso a um dispositivo IoT. A comunicação entre os dispositivos se dá através de transações que são armazenadas em blocos visíveis apenas para os CHs. Tais transações podem ser de três tipos: acesso, armazenamento ou monitoramento. Transações de armazenamento têm a função de armazenar alguma informação do dispositivo em locais como a nuvem ou em um disco local. A transação de acesso é usada por dispositivos que queiram acessar alguma informação em outros dispositivos ou nos locais de armazenamento. As transações de monitoramento concedem acesso ao dispositivo, e.g., acesso a uma câmera de monitoramento.

Transações podem ocorrer em uma rede local ou na rede de sobreposição. Transações locais são armazenadas na BC local e são compostas por cinco campos: *timestamp* (tempo de criação da transação), *ID* da transação (inteiro que identifica a transação como única), ID do dispositivo (chave pública do dispositivo solicitado), tipo da transação

como ilustrado na Figura 2(a)(a) e assinatura do solicitante (chave pública do dispositivo solicitante). As permissões dos dispositivos são armazenadas no cabeçalho - Figura 2(a)(b). A figura 2(a) ilustra um bloco local e seus campos.



(a) Bloco Local (b) Estrutura da transação

Figura 2. *Estrutura dos blocos e transações*

Transações dos nós de sobreposição são realizadas entre os CHs e sua estrutura é formada pelos campos: *timestamp*, chave pública do solicitante, chave pública do solicitado, metadados e *score*. *Timestamp* é o tempo em que a transação foi criada, a chave pública do solicitante e solicitado é a identidade dos dispositivos, metadados contém a informação do tipo da transação (acesso, monitoramento e armazenamento) e o *score* é o resultado da verificação da transação (se foi bem sucedida, o *score* recebe verdadeiro, caso contrário recebe falso - Figura 2(b)).

Os blocos da camada de sobreposição (Figura 5) seguem uma estrutura semelhante a uma BC convencional, eliminando os campos relacionados à mineração(*Nonce* e Dificuldade). No cabeçalho, é adicionado um campo de reputação, que estabelece o grau de confiança entre os nós (ver Seção 2.5), que incrementa ao criador do bloco a quantia de 1; por último, as transações são armazenadas em forma de lista.

2.3. Comportamento do Nó

Neste trabalho, uma transação é definida como uma solicitação de acesso a um dispositivo e o acesso pode partir de um usuário na rede ou de outro dispositivo de uma *Smart Home*. A solicitação segue o modelo da Figura 3, sendo validada pelos próprios membros da rede. Os Chs, ou nós mineradores, são dispositivos com um maior poder computacional e disponibilidade (e.g., celulares, computadores, *Beaglebone*), e são os responsáveis por organizar e gerenciar toda a rede.

2.4. Consenso

Quando um CH é escolhido como minerador, ele verifica todas as transações do bloco e envia um *broadcast* para os demais CHs na rede e então espera por um período variável para que não faça parte da próxima eleição de minerador. Assim como no *Bitcoin* [Nakamoto 2008], uma transação é efetivada quando uma determinada quantidade de blocos seguintes são verificados e adicionados à BC. Os CHs podem manter 3 estados possíveis: seguidor, candidato e líder. Seguidores são os CHs que não irão participar da

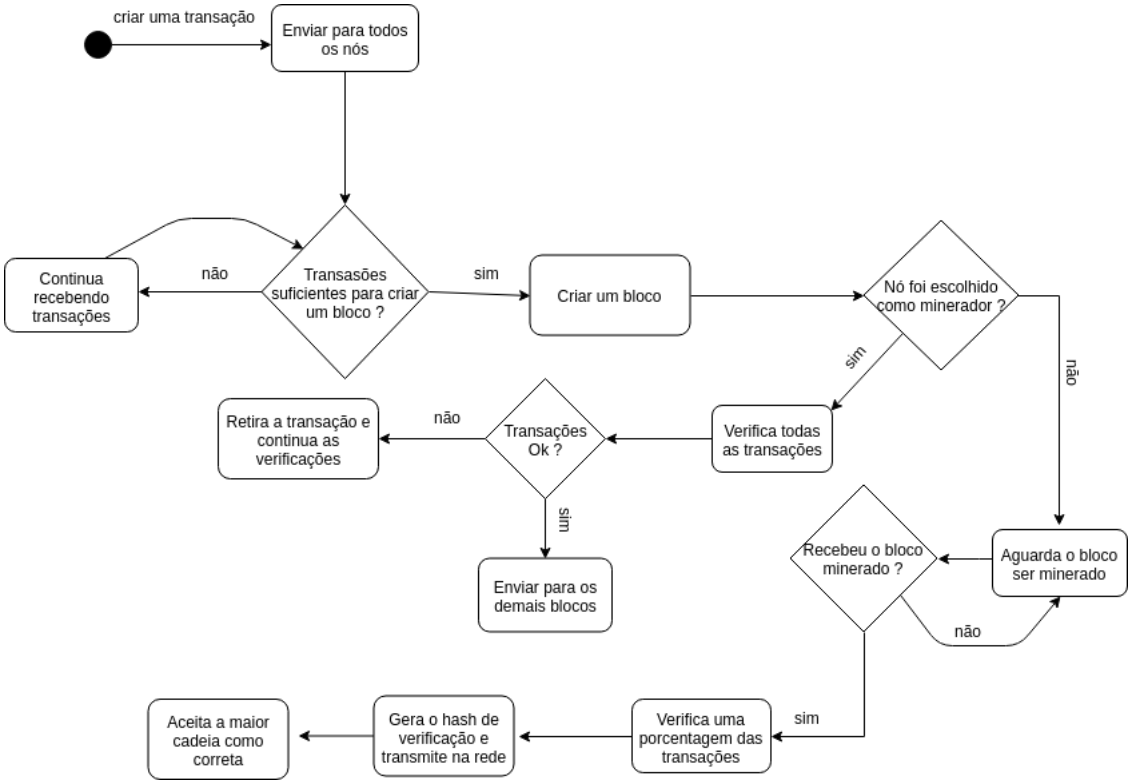
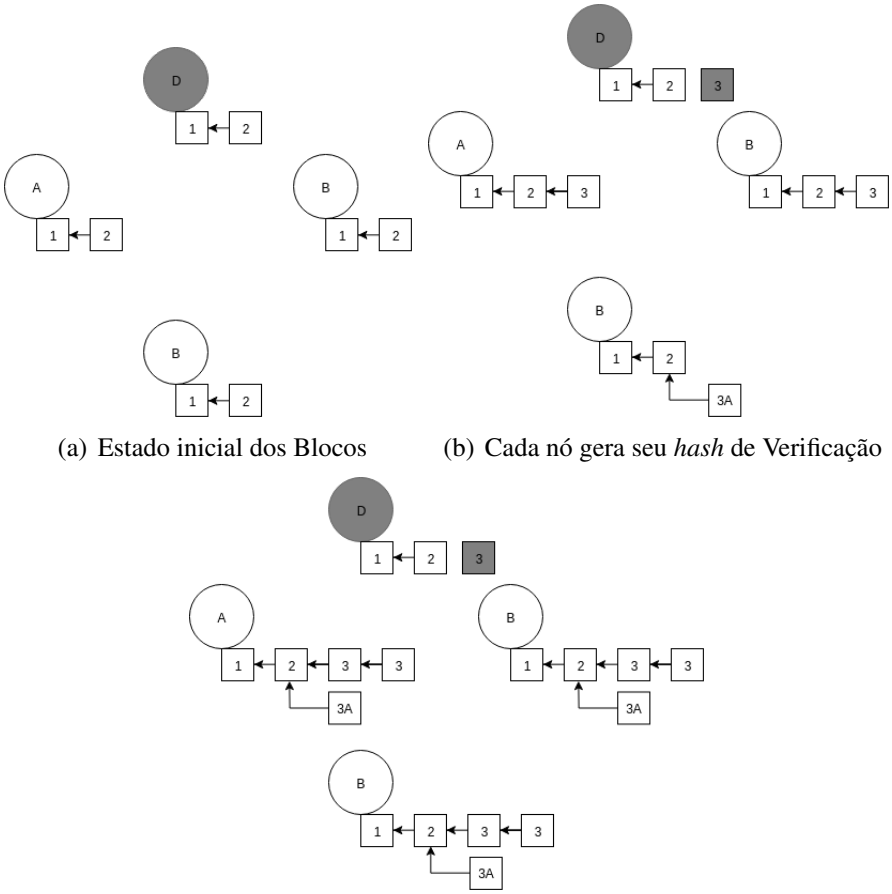


Figura 3. Visão geral do comportamento de um nó

eleição do próximo nó, candidatos são os CHs aptos a participar e o líder é o CH escolhido. Dessa forma, quando o bloco é minerado, o CH espera por um tempo variável e passa para o estado de seguidor.

Como não é possível garantir que todos os CHs que irão verificar o bloco sejam confiáveis, o algoritmo de consenso trabalha com a ideia de maior cadeia. Alguns CHs que verificam o bloco podem atribuir falsos acessos a dispositivos e comprometer a integridade do bloco e da rede, ou pode corrigir uma transação falsa gerada por algum minerador mal intencionado. A cadeia mais longa é marcada como válida e cadeias distintas são descartadas. Para isso, o algoritmo usa um *hash* no cabeçalho do bloco, chamado *hash* de verificação, cuja função é assinar o bloco após a verificação das transações. Se o *hash* da verificação for igual ao *hash* do bloco minerado, então nenhuma transação foi modificada. Se o *hash* for diferente, então o CH alterou alguma transação do bloco e, portanto, cria-se um *fork* na BC. Assim, tem-se dois ramos da cadeia, um principal e outro secundário, que irão coexistir até que um possua a maior quantidade de blocos.

A Figura 4 exemplifica o conceito de maior cadeia. Na Figura 4(a), é dado o estado inicial dos blocos, com o nó **D** como minerador e o restante sendo nós de verificação. Na Figura 4(b), o nó **D** minera o bloco 3 e envia para os demais nós. Cada nó verifica as transações e gera seu próprio *hash* de verificação. O nó **B** tentou criar uma transação falsa alterando assim seu *hash* de verificação. Na Figura 4(c), o bloco verificado é transmitido para a rede e é anexado aos outros blocos. Percebe-se que o ramo com mais blocos (1,2,3,3) passa a ser o ramo correto, resolvendo o *fork* da rede. Por fim, os blocos duplicados podem ser descartados, restando os blocos (1,2,3).



(a) Estado inicial dos Blocos (b) Cada nó gera seu *hash* de Verificação

(c) Cada nó envia sua verificação

Figura 4. Exemplo de cadeira mais longa

O Algoritmo 1 representa a rotina que é executada sempre que um novo bloco é gerado para a rede. Após a escolha do minerador, o CH escolhido executa a verificação das transações pelo campo metadados da transação, que contém informações sobre o CH que gerou a transação e informações do dispositivo solicitado. Os demais CHs executam o algoritmo de confiança, que analisa a tabela de confiança.

Algoritmo 1: Algoritmo de consenso

```
begin
    inicializa o evento com time zero;
    minerador = escolhaMinerador();
    acesso = FALSE;
    if minerador == VERDADEIRO then
        bloco = criarBloco();
        repeat acesso = verificarPermissao(bloco.trasacao[i].metadados);
        transacao[i].status = acesso;
        i++;
        until acabar as transações;
        geraHashDeVerificacao(bloco);
    else
        confiança();
    end
    escolhaDaMaiorCadeia();
end
```

A última rotina a ser executada é a escolha da maior cadeia, apresentada no Algoritmo 2, onde o CH aguarda a chegada dos demais blocos e, em seguida, armazena o *hash* de verificação em uma lista. No fim do processo, a maior cadeia de *hash* é anexada à BC.

Algoritmo 2: Maior cadeia

```
begin
    listaHash = vazio;
    repeat bloco = recevBloco();
    id = estaNaLista(bloco.hashverificacao,listaHash);
    if id >= 0 then
        listaHash[id].quantidade = +1;
    else
        addNaLista(bloco.hashverificacao);
    end
    until bloco == NULL;
    anexar na Blockchain a maior cadeia;
end
```

2.5. Confiança

Os nós criam confiança um no outro à medida que sua confiança aumenta. Dois critérios são utilizados para estabelecer a confiança entre os nós: quantidade de transações verificadas e sua reputação. A quantidade de transações verificadas representa o número total de transações que um CH minerou. A reputação representa a quantidade de **blocos** minerados anteriormente por um CH. A Tabela 1 mostra a quantidade de transações que precisarão ser verificadas (em porcentagem). Quanto maior a confiança no minerador, menor será a quantidade de transações que precisarão ser verificadas, reduzindo a demanda por processamento.

Para exemplificar a confiança entre CHs, dado um estado de BC como o da Figura 5, é possível estabelecer a confiança no nó percorrendo todos os blocos. Assumindo que a BC está em um estado válido, o CH **D** minerou um novo bloco e enviou na rede. Os

Tabela 1. Tabela de confiança

Quantidade de transações verificados anteriormente						
Reputação		≥ 10 e < 20	≥ 20 e < 30	≥ 30 e < 40	≥ 40 e < 50	≥ 50
	$\geq$5 e <10	80%	70%	60%	50%	40%
	$\geq$10 e <15	70%	60%	50%	40%	30%
	$\geq$15 e <20	60%	50%	40%	30%	30%
	$\geq$20 e <25	50%	40%	30%	30%	20%
	$\geq$25	40%	30%	30%	20%	20%

demais CHs executam o algoritmo para estabelecer a confiança no CH **D**. Eles calculam a reputação do CH comparando o *hash* do bloco com o *hash* de verificação. Se forem diferentes, a contagem no bloco é descartada e passa ao bloco anterior; se forem iguais, a reputação e quantidade de transações são contabilizas. Ao fim do processo, é retornada uma tupla contendo a reputação e a quantidade de transações. Neste exemplo, a confiança no CH **D** seria: Reputação +2 e Transações +16. A Tabela 1 mostra que este CH ainda não tem confiança suficiente, então todas as transações devem ser verificadas.

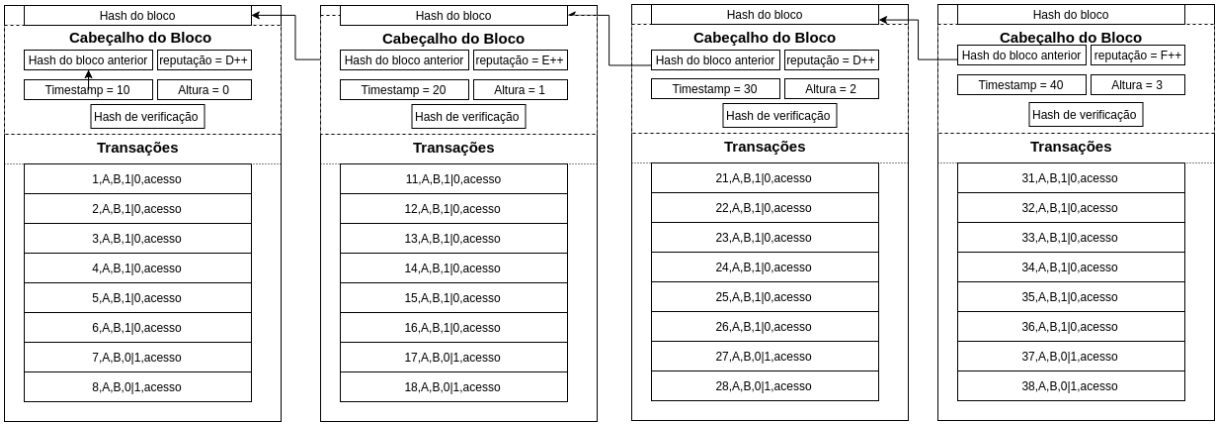


Figura 5. Estrutura da Blockchain do trabalho proposto

3. Avaliação do Mecanismo Proposto

Nesta seção, são apresentados o cenário de avaliação e os resultados das simulações realizadas. A solução proposta foi avaliada através da técnica de simulação com o *Network Simulator* (NS-3) baseado no trabalho de [Gervais et al. 2016], que implementa uma versão do *Bitcoin* usando o NS3. Para envio de mensagem do bloco, foi usada a implementação Json para C++ desenvolvida por [Tencent 2016]. Por fim, os resultados obtidos foram comparados aos de [Dorri et al. 2017a].

Esta seção está dividida da seguinte forma: A Subseção 3.1 apresenta o cenário de testes com a rede de sobreposição, as métricas de validação e a configuração da simulação; enquanto a Subseção 3.2 apresenta a análise comparativa dos resultados através de gráficos.

3.1. Cenário de Testes

O cenário simulado corresponde à estrutura dos nós de sobreposição, nós mineradores ou CH, abstraindo toda a estrutura interna da *smart home*. Ou seja, a simulação contempla

apenas o funcionamento da estrutura de sobreposição apresentada na Seção 2. Isso se deve ao fato do NS3 não fornecer ferramentas adequadas para avaliar/executar os dispositivos IoT com suas características específicas de comportamentos/recursos.

A fim de simplificar a comunicação entre CH e a *smart home*, as permissões dos dispositivos são armazenadas no CH, assim como a estrutura da BC. As permissões seguem o modelo da política de acesso apresentada na Figura 2(a). A estrutura da BC é representada por uma variável inteiro e uma lista de blocos. As transações, que representam solicitações de acesso a dispositivos IoT, são criadas de forma aleatória, seguindo o modelo da Figura 2(b), sempre antes da mineração do bloco. O fluxo de eventos da simulação segue conforme a Figura 3.

As métricas adotadas na validação do trabalho foram: i) tempo médio de validação de uma transação, ii) tempo de mineração do bloco, iii) quantidade de transações falsas que alcançaram sucesso. O tempo de validação de uma transação representa o tempo que uma transação levou para ser efetivada (i.e., o acesso do usuário foi aceito ou recusado). Assim como no *Bitcoin*, ou qualquer outra criptomoeda, uma transação é considerada efetivada quando ela for minerada em um bloco e quando for sucedida por uma quantidade mínima de blocos. Neste trabalho, a quantidade adotada como padrão foi cinco blocos, já que um valor menor implica em maiores chances da transação ser falsa ou desfeita, pela regra da maior cadeia. O tempo de mineração do bloco mede o algoritmo de confiança, pois, a quantidade de transações que precisará ser verificada diminui à medida que um CH aumenta sua reputação. A Tabela 2, que mede a confiança de um CH, foi adaptada a fim de acelerar o teste e a análise dos resultados. A mudança ocorreu na reputação que foi diminuída, dessa forma o critério adotado para consenso torna-se menos rígido e possibilita o aumento contínuo da confiança entre os CHs. Por fim, a quantidade de transações falsas que alcançaram sucesso representa a taxa de falha do modelo na presença de usuários mal intencionados.

Tabela 2. Tabela de confiança adaptada

Quantidade de transações verificados anteriormente						
Reputação		$\geq$10 e <20	$\geq$ 20 e <30	$\geq$30 e <40	$\geq$ 40 e <50	$\geq$ 50
	$\geq$ e <5	80%	70%	60%	50%	40%
	$\geq$5 e <10	70%	60%	50%	40%	30%
	$\geq$10 e <15	60%	50%	40%	30%	30%
	$\geq$15 e <20	50%	40%	30%	30%	20%
	$\geq$20	40%	30%	30%	20%	20%

Os gráficos gerados possuem barras de erro, em torno das médias das vinte rodadas, utilizando um intervalo de confiança de 95%. Todas as métricas foram coletadas a partir da BC gerada ao final de cada simulação. Para isso foi criado um nó de controle que exerce apenas a função de gerar a BC ao final da execução (esse nó não participa de nenhuma etapa de mineração). Foram feitas simulações utilizando cinco, dez, quinze e vinte CHs, cada um gerando até cinco transações por bloco, como no trabalho de [Dorri et al. 2017a]. A Tabela 3 resume os parâmetros de configuração da simulação.

Tabela 3. Configuração da simulação

Parâmetro	Valor
Cenário	Sobreposição
Geração das transações	Aleatório
Quantidade de CH	5,10,15,20
Taxa (transação/CH)	Aleatório (1 a 4)
Número de rodadas	20
Intervalo de Confiança	95%

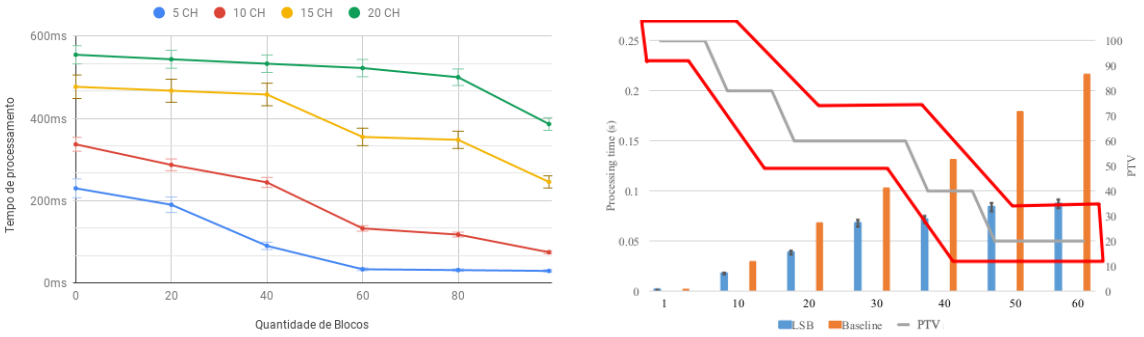
Os experimentos foram realizados em um computador com processador Intel Core Intel® Core™ i5-5200U CPU@2.20GHz x4, 7.7 GB de memória RAM e sistema operacional Ubuntu 18.04.1 LTS 64 bits e a versão 3.28 do NS3. O código fonte do mecanismo proposto está disponível no GitHub[1].

3.2. Análise dos Resultados

Os resultados são apresentados de forma comparativa ao trabalho de [Dorri et al. 2017a] usando as métricas tempo de mineração do bloco, tempo médio de validação de uma transação e quantidade de transações falsas descobertas.

3.2.1. Tempo de mineração do bloco

O mecanismo proposto usa um algoritmo de confiança distribuído que diminui o número de transações que devem ser verificadas à medida que os CHs adquirem confiança um nos outros (como explicado na Subseção 2.5). Foi usado uma quantidade variada de CHs (cinco, dez, quinze e vinte) em cada rodada da simulação com uma instância de BC com 100 blocos. Além disso, foi utilizada uma versão reduzida da tabela de confiança (Tabela 2) a fim de facilitar a observação da diminuição no tempo de mineração do bloco.



(a) Este trabalho (b) Resultados de [Dorri et al. 2017a]

Figura 6. *Tempo de mineração do bloco*

Como pode ser observado na Figura 6(a), o experimento mostra como o algoritmo de confiança reduz o tempo de mineração do bloco. Devido à concorrência entre os CHs, sua reputação cresce de forma mais lenta, o que impacta diretamente no tempo de

[1]Código-fonte disponível em: `https://github.com/afonsoneto121/tcc`

mineração. Com uma quantidade menor de CHs, o tempo de mineração do bloco tende a diminuir mais rápido (gráfico azul da Figura 6(a)), enquanto com uma quantidade maior de CHs percebe-se que o tempo de mineração reduz, porém de forma mais lenta (gráfico verde da Figura 6(a)).

Comparando os resultados obtidos com o trabalho de [Dorri et al. 2017a] (Figura 6(b)), onde os autores usaram duas implementações de BC, uma chamada de *Baseline*, que corresponde a uma implementação sem o algoritmo de confiança, e outra chamada de LSB, que corresponde à implementação dos autores com o algoritmo de confiança. PTV representa a quantidade de transações verificadas em porcentagem. Vale destacar que não são consideradas, no tempo de mineração, outras tarefas, tais como verificação de listas de chaves, geração de novos blocos, etc. Os resultados encontrados neste trabalho ficaram um pouco acima dos encontrados por [Dorri et al. 2017a]. Ao analisar a área destacada, nota-se que o algoritmo de confiança proposto por [Dorri et al. 2017a] chega à quantidade mínima de verificações (20%). Porém, eles utilizam uma tabela de confiança que se resume à quantidade de transação verificadas anteriormente.

3.2.2. Tempo médio de validação de uma transação

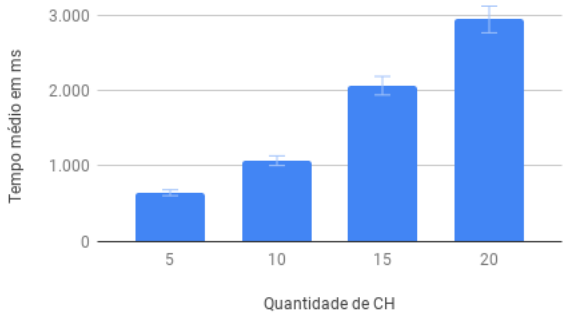


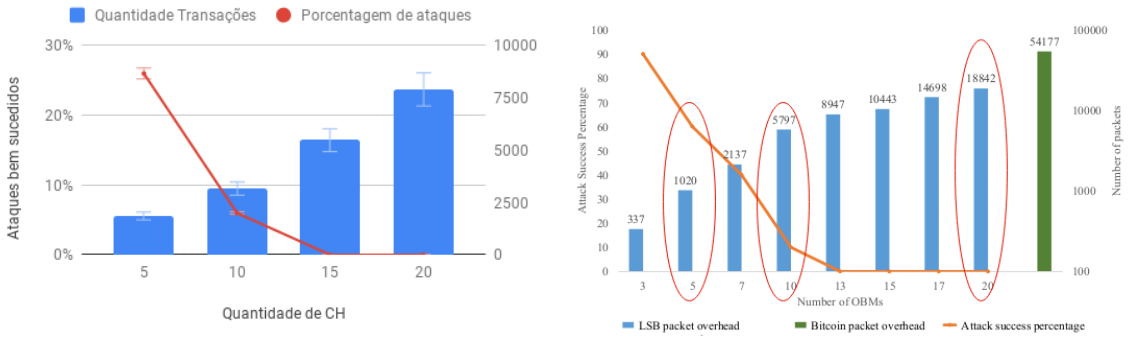
Figura 7. Tempo médio de validação de uma transação

A Figura 7 mostra o tempo médio que uma transação leva para ser efetivada. A contagem do tempo inicia-se quando um bloco é minerado. O estado atual do projeto ainda não é capaz de computar o tempo total de uma transação, diferença de tempo desde a solicitação de acesso por parte do usuário até a efetiva resposta da rede cedendo o acesso ou negando-o. A efetivação da transação segue conforme funcionam nas criptomoedas, onde uma transação é efetivada quando uma quantidade mínima de blocos é minerado. Essa quantidade mínima é igual a cinco, o que significa que as transações do bloco 1 só vão ser efetivadas quando o bloco 6 for minerado, o mesmo segue para os demais blocos. O resultado encontrado mostra que, devido à concorrência entre os CHs, o algoritmo de consenso não consegue ser totalmente eficiente reduzindo o tempo das transações, o que significa que a reputação de um CH aumenta de forma mais lenta fazendo com que mais blocos sejam verificados por completo. Além desse fator, um número maior de CH faz com que se aumente o número de mensagens trocadas, influenciando nos resultados encontrados. Assim como o tempo de mineração do bloco o tempo de validação da transação

tende a diminuir, à medida que mais blocos são minerados. O limite dessa redução é o valor mínimo da tabela de confiança.

3.2.3. Quantidade de transações falsas descobertas

Diferente das simulações anteriores, em que as transações eram geradas de modo aleatório, nesta a rede segue um padrão previamente estabelecido, para que no final da simulação a quantidade de transações falsas seja conhecida e avaliada. Para cada rodada da simulação, aproximadamente um terço dos CHs é marcado como usuário mal intencionado, criando uma transação falsa por bloco. Ao final da simulação, foi contabilizada a quantidade de transações que alcançaram sucesso.Uma transação alcança sucesso quando o campo score da transação é igual a verdadeiro. A Figura 8(a) demostra os resultados encontrados, indicando que à medida que mais CHs participam do processo de mineração, menor é a chance de uma transação falsa alcançar sucesso. A partir de quinze CHs, nenhuma transação falsa conseguiu ser bem sucedida.



(a) Este trabalho (b) Resultados de Dorri

Figura 8. *Quantidade de transações falsas descobertas*

Observando as áreas destacadas na Figura 8(b), nota-se uma pequena melhora nos resultados encontrados, que se deve ao fato do algoritmo de confiança usado neste trabalho usar critérios mais rígidos para aumentar a reputação de um CH, visto que o mecanismo de consenso adotado neste trabalho trabalha com dois requisitos para confiança(reputação e transações anteriores) enquanto [Dorri et al. 2017b] trabalha apenas com as transações anteriores.

4. Trabalhos Relacionados

Em [Sahraoui and Bilami 2014], os autores propõem uma nova forma de autenticação e controle de acesso para tornar dispositivos IoT seguros contra acesso não autorizado. O método proposto baseia-se em duas autoridades de autenticação: Autoridade de Registo (do inglês, *Registration Authority* - RA) e Autoridade de Registro Residencial (do inglês, *Home Registration Authority* - HRA,). Todos os dispositivos são registrados no RA, que é projetado para facilitar o processo de autenticação dos dispositivos. Da mesma forma, o HRA facilita o processo de autenticação para os usuários. Quando um usuário deseja acessar dados de um determinado dispositivo, a requisição é primeiro enviado para o RA, que verifica a autenticidade do usuário com o HRA. Assumindo que o usuário é

autenticado, o RA gera uma chave compartilhada para comunicação entre o usuário e o dispositivo.

A análise de segurança mostra que o modelo é seguro, porém a necessidade de cada dispositivo ter um RA e cada usuário um HRA pode ocasionar problemas de escalabilidade. No nosso projeto, a *Blockchain* é administrada por um *Custer Head*, Seção 2.1, e cada *smart home* gerecia seus dispositivos internos.

O trabalho de [Steger et al. 2018] propõe uma arquitetura baseada em *Blockchain* para atualização segura de *software* automotivo. Os autores avaliam a implementação de uma prova de conceito em um sistema de atualização de software sem fio, fornecendo uma comunicação segura e eficiente entre todas as partes envolvidas. Os veículos inteligentes consistem em uma unidade simples que controla tarefas. Cada unidade comunica-se entre si e com o fabricante, que por sua vez verifica e distribuiu a nova versão do *software* que será instalada nas unidades. Os nós são agrupados em *clusters*, onde é eleito um líder para gerenciar a *Blockchain*, semelhante ao proposto em nosso projeto, mas os autores usam *Proof of Concept* como algoritmo de concesso, o que demanda muito processamento, enquanto nós utilizamos consenso em duas etapas: confiança entre os nós (validação) e a escolha do nó que irá minerar o bloco.

O trabalho de [Dorri et al. 2017b] propõe um modelo de segurança baseado em *Blockchain*, leve e escalável para IoT. O método proposto baseia-se na criação de uma arquitetura em dois níveis: (i) rede local, onde os dispositivos são gerenciados por uma *Blockchain* local, e uma rede de sobreposição, onde os gerentes de bloco (do inglês, *Overlay Block Mangers* - OBM) gerenciam uma *Blockchain* pública, confirmando e verificando novos blocos. O algoritmo de consenso garante que um bloco gerado é selecionado aleatoriamente entre os nós e é limitado no número de blocos que pode gerar. Para introduzir a aleatoriedade entre os blocos, cada OBM deve esperar por um tempo antes de gerar um novo bloco.

Nosso trabalho é inspirado em [Dorri et al. 2017b], diferindo-se no uso de algoritmo de consenso e confiança diferentes, voltados para a redução do processamento e, consequentemente, consumo de energia. A Tabela 4 sumariza as principais semelhanças e diferenças dos trabalhos relacionados e o presente trabalho.

Tabela 4. Comparação com os trabalhos relacionados

	[Sahraoui and Bilami 2014]	**[Steger et al. 2018]**	**[Dorri et al. 2017b]**	**Presente Trabalho**
Consenso	-	*Proof of Concept*	Transações	Confiança e maior cadeia
Autenticação e Controle de acesso	RA e HRA	-	Estrutura de blocos locais	Estrutura de blocos locais
Verificação das transações	-	Assinatura do fabricante	Verifica as permissões do solicitante	Verifica as permissões do solicitante
Blockchain	-	Cada nó tem a mesma versão da *Blockchain*	Nós podem ter diferentes versões da *Blockchain*	Cada nó tem a mesma versão da *Blockchain*
Encadeamento dos blocos	-	Blocos encadeados	Transações encadeadas	Blocos encadeados

5. Considerações Finais

Este trabalho apresenta uma abordagem descentralizada para segurança de dispositivos inteligentes em *smart homes*, utilizando uma abordagem baseada em BC. Desenvolvemos um algoritmo de consenso adequado para IoT em que há limitações de recursos e grande distribuição de dispositivos.

O trabalho proposto incorpora um método de confiança distribuída em que o tempo de processamento para validar novos blocos diminuem gradualmente à medida que os nós criam confiança um nos outros. Os experimentos realizados mostraram que com o aumento de número de CHs o mecanismo impede que usuários maliciosos criem transações falsas ao mesmo tempo que mantém o tempo de mineração e validação em valores aceitáveis para IoT.

Como trabalho futuro, pode-se melhorar os testes usando o próprio escalonador de eventos no NS3 para coordenar as transações com o intuito de verificar o funcionamento da estrutura de maior cadeia e seus impactos para os resultados já encontrados e desenvolver um método para gerar a tabela de confiança de modo dinâmico, que se adapte às necessidades da rede. Pretende-se também utilizar o Cooja para implementar a interação com os dispositivos IoT.

Referências

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

Barcellos, A. M. P. and Gaspary, L. P. (2006). Segurança em redes p2p: princípios, tecnologias e desafios. In *Simposio Brasileiro de Redes de Computadores (24.: 2006 maio: Curitiba, PR). Anais dos minicursos. Curitiba:[sn], 2006.*

Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2017a). Blockchain for iot security and privacy: The case study of a smart home. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*, pages 618–623. IEEE.

Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2017b). Lsb: A lightweight scalable blockchain for iot security and privacy. *arXiv preprint arXiv:1712.02969*.

Farooq, M. U., Waseem, M., Khairi, A., and Mazhar, S. (2015). A critical analysis on the security concerns of internet of things (iot). *International Journal of Computer Applications*, 111(7).

Gervais, A., Karame, G., Wüst, K., Glykantzis, V., Ritzdorf, H., and Capkun, S. (2016). On the security and performance of proof of work blockchains. In *Proceedings of the 23nd ACM SIGSAC Conference on Computer and Communication Security (CCS)*. ACM.

Greve, F., Sampaio, L., Abijaude, J., Coutinho, A., Valcy, Í., and Queiroz, S. (2018). Blockchain e a revolução do consenso sob demanda. *Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (Minicursos_SBRC)*, 36.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Roman, R., Zhou, J., and Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279.

Sahraoui, S. and Bilami, A. (2014). Compressed and distributed host identity protocol for end-to-end security in the iot. In *Next Generation Networks and Services (NGNS), 2014 Fifth International Conference on*, pages 295–301. IEEE.

Steger, M., Dorri, A., Kanhere, S. S., Römer, K., Jurdak, R., and Karner, M. (2018). Secure wireless automotive software updates using blockchains: A proof of concept. In *Advanced Microsystems for Automotive Applications 2017*, pages 137–149. Springer.

Tencent (2016). Repositório github, https://github.com/tencent/rapidjson.git. Acessado em Novembro de 2018.