Provendo Privacidade e Tolerância a Falhas em Cloud of Things

Luis Alberto B. Pacheco[1], **Eduardo A. P. Alchieri**[1], **Priscila Solis**[1]

[1]Departamento de Ciência da Computação
Universidade de Brasília (UnB)
Brasília – DF – Brasil

luisbelem@aluno.unb.br,{alchieri,pris}@unb.br

Abstract. *Through the Internet of Things (IoT) a large number of devices are connected to the internet, resulting in a huge amount of produced data. Cloud computing is currently adopted to store, process and access control to these data, this integration is called Cloud of Things - CoT. This approach is useful in personal networks, like residential automation and health care, since it facilitates the access to the information. Although this integration brings benefits to the users, it introduces a federation problem, the information leaves the user control and is housed at the cloud providers. In order for these technologies to be adopted, the users privacy must be ensured. This paper proposes a fault-tolerant architecture to privacy in Cloud of Things, which allows the users to control the data generated by the devices of their IoT networks.*

Resumo. *Através da Internet das Coisas (IoT), uma imensa quantidade de dispositivos são conectados à internet, gerando uma grande quantidade de dados. Atualmente propõe-se a utilização de computação em nuvem para o armazenamento, processamento, apresentação e controle de acesso a esses dados, a essa integração chama-se Cloud of Things - CoT. Essa abordagem é especialmente útil para redes domésticas e pessoais, tais como automação residencial e assistência médica, pois facilita o acesso a informação pelos indivíduos. Apesar de trazer benefícios aos usuários, a integração desses dois conceitos tecnológicos introduz um problema de acesso a informação, que sai da esfera de controle do usuário ao ser enviado para a nuvem. Para que haja uma adoção em massa dessa tecnologia é importante que a privacidade dos dados dos usuários seja mantida. Este trabalho propõe uma arquitetura tolerante a falhas e que provê privacidade em Cloud of Things, permitindo ao usuário o controle do acesso aos dados gerados pelos dispositivos de suas redes IoT.*

1. Introdução

Os avanços nas tecnologias de miniaturização de componentes eletrônicos e nas tecnologias de comunicação sem fio possibilitaram o advento da Internet das Coisas (IoT), onde os mais diversos itens do nosso cotidiano terão acesso à internet, trazendo uma enorme gama de benefícios à população [Chui et al. 2010]. São previstas bilhões de "coisas" conectando-se a internet para prover os mais diferentes tipos de informações aos usuários [Sundmaeker et al. 2010].

Os dispositivos de IoT podem estar presentes nos mais variados meios, mas comumente uma rede IoT é implementada através de sensores. Redes de sensores sem fio

(RSSF) são compostas por dispositivos que possuem uma unidade de processamento, um sensor que proporciona a interação com o mundo físico e uma antena para comunicação sem fio [Akyildiz and Vuran 2010]. A IoT será formada por milhares de RSSFs. Os dispositivos das RSSFs constantemente possuem tamanho e fonte de energia limitados, dessa forma a pilha de rede precisa possuir um baixo custo de processamento. Muitos protocolos em todas as camadas da pilha foram propostos com tal finalidade. Nas camadas Física e de Enlace o padrão IEEE 802.15.4 [Group 2006] foi lançado em 2006 e possui diversas atualizações, sendo atualmente utilizado por grande parte das redes de sensores. Já protocolos das demais camadas possuem uma grande fragmentação, com diferentes propostas para diferentes aplicações, gerando a necessidade da realização de uma tradução quando a comunicação deve ocorrer com a Internet, que é efetuada através de um *gateway*.

A comunicação direta da rede de sensores com a Internet, sem a necessidade de um *gateway*, traz grande benefício para a implementação da IoT, pois facilita o acesso direto à informação advinda do sensor pelo usuário [Granjal et al. 2015]. Existem vários trabalhos em andamento com o objetivo de criar adaptações de protocolos já utilizados na Internet para serem utilizados em redes de sensores. Na camada de rede, o 6LoWPAN [Kushalnagar et al. 2007] efetua compressão e encapsulamento de cabeçalhos IPv6 [Deering 1998] para que caibam em quadros do protocolo IEEE 802.15.4. O IPv6 possui grande capacidade de endereçamento, o que o faz um forte candidato à utilização na Internet das Coisas, que possuirá bilhões de dispositivos conectados. Na camada de transporte, é indicada a utilização do protocolo UDP [Postel 1980], que apesar de não oferecer funcionalidades de entrega confiável, ordenada e com checagem de erros como o TCP [Postel 1981], apresenta um *overhead* muito menor. Apesar dos sensores poderem se comunicar diretamente com outros dispositivos da internet, um *gateway* ainda pode ser utilizado para efetuar operações muito custosas para os pequenos dispositivos.

As limitações tecnológicas da IoT (armazenamento, processamento, comunicação) podem ser mitigadas com a utilização de Computação em Nuvem. Estes modelos são complementares uma vez que a IoT produz uma imensa quantidade de dados, enquanto que a Computação em Nuvem é capaz de fornecer mecanismos para armazenamento e processamento destes dados. De fato, aplicações podem utilizar os recursos virtualmente ilimitados de plataformas na nuvem para processar e apresentar as informações aos usuários. Outras características da computação em nuvem importantes para IoT são: escalabilidade inerente da tecnologia; aumento da segurança, pois o controle de acesso à informação é realizado na nuvem, a qual dispõe de poder computacional adequado para essa tarefa; maior possibilidade de alcance; dentre outros. Esta integração é chamada de *Cloud of Things* (CoT) e atualmente está sendo amplamente discutida [Botta et al. 2016].

Apesar dos imensos benefícios desta integração, é importante que as soluções assegurem as propriedades de segurança e, principalmente, a privacidade dos dados dos usuários gerados pelos dispositivos de IoT, uma vez que estes saem da esfera de controle do usuário ao serem enviados para a nuvem. Neste sentido, este trabalho propõe uma arquitetura tolerante a falhas que provê privacidade em *Cloud of Things*, permitindo que o usuário controle o acesso aos dados gerados pelos dispositivos que estão em seu controle. A arquitetura proposta permite um controle de acesso de granularidade fina sobre os dados, uma vez que os protocolos e controles são executados nos dispositivos de IoT

e não na borda da rede por um *gateway*, o qual é também um único ponto de falha que pode quebrar a disponibilidade ou as propriedades de segurança das aplicações caso seja comprometido.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 discute os principais trabalhos relacionados com a arquitetura proposta, que é detalhada na Seção 3. Uma ampla discussão relacionada com as soluções propostas é apresentada na Seção 4. Finalmente, as conclusões e trabalhos futuros são discutidos na Seção 5.

2. Trabalhos Relacionados

O modelo de serviços utilizado na computação em nuvem pode ser estendido para a integração com a IoT, de forma que as funcionalidades advindas da IoT possam ser oferecidas como serviço na nuvem. Por exemplo, sensores de temperatura espalhados pela cidade podem enviar seus dados para a nuvem, onde um serviço fornece, de maneira ubíqua e segura, acesso à essa informações para usuários em diversas plataformas.

Diversos trabalhos implementam a integração de IoT e computação em nuvem utilizando esta técnica. O IoTCloud [Fox et al. 2012] é uma plataforma de código aberto que tem por objetivo integrar os dispositivos IoT com a nuvem para o gerenciamento dos mesmos. Já o Nimbits [Sautner 2017] oferece uma solução a ser utilizada em plataformas de nuvem onde é possível coletar e processar dados de sensores.

O conceito de *Cloud of Things* envolve a integração de vários elementos, aumentando a complexidade das soluções de segurança que precisam ser adotadas [Roman et al. 2013]. Em redes IoT domésticas, o armazenamento dos dados do usuário na nuvem gera um problema relacionado a privacidade, pois a partir deste momento os dados estão sob controle de uma entidade diferente, e portanto podem ser utilizados para fins que o usuário não aprova. Para atender os requisitos de privacidade do usuário é preciso que apenas entidades previamente autorizadas manuseiem seus dados. Neste sentido, o SensorCloud [Eggert et al. 2014] propõe uma plataforma baseada na nuvem que integra redes de sensores e a Internet. Uma arquitetura baseada em camadas é implementada onde RSSFs conectam-se à nuvem por meio de pontos de confiança (dispositivos de borda) que são responsáveis pela *(i)* comunicação com a nuvem e *(ii)* aplicação de segurança. A arquitetura envolve apenas a comunicação a partir do ponto de confiança, o funcionamento da rede de sensores não é abordado.

O *User-driven Privacy Enforcement for Cloud-based Services in the IoT* (UPECSI) [Henze et al. 2016] estende o SensorCloud, implementando uma solução voltada a privacidade que abrange desde o processo de desenvolvimento de um serviço em nuvem até o usuário. Uma Linguagem de Desenvolvimento de Privacidade (do inglês *Privacy Development Language* (PDL)) foi desenvolvida para facilitar o desenvolvimento de serviços de nuvem com privacidade. A utilização da PDL permite ao desenvolvedor fornecer informações detalhadas sobre quais dados e como eles são utilizados pela aplicação. Esta funcionalidade é então utilizada pelo usuário para habilitar (ou não) certas funções do serviço de acordo com suas preferências. A segurança entre as redes IoT do usuário e a nuvem é realizada por um Ponto de Aplicação de Privacidade (do inglês *Privacy Enforcement Points* (PEP)). O PEP é um dispositivo na borda da rede de sensores que possui capacidade computacional e fonte de energia ilimitada, dessa forma é capaz de executar as tarefas necessárias para garantir a segurança e privacidade dos dados do usuário. Esta

arquitetura permite ao usuário alterar a Política de Privacidade de acordo com suas preferências, fornecendo um alto nível de controle sobre os dados armazenados na nuvem.

O Ponto de Aplicação de Privacidade é a ultima entidade sob o controle do usuário, como tal dispositivo não possui capacidades limitadas, é possível aplicar mecanismos de segurança robustos, como por exemplo a utilização do TLS na camada de transporte. De acordo com a Política de Privacidade configurada pelo usuário, o PEP determina quais dados podem sair da sua esfera de controle, e para quais serviços tais dados devem ser enviados. Os dados são armazenados na nuvem em forma cifrada, e a chave utilizada nesta operação é alterada periodicamente, possibilitando a remoção do acesso de um determinado serviço de nuvem quando desejado. Infraestrutura de Chave Pública (PKI) é utilizada para cifrar a chave simétrica que cifra os dados, a chave simétrica é cifrada uma vez para cada serviço que possui autorização sob os dados. Este mecanismo possibilita que apenas os serviços tenham acesso aos dados, nem mesmo o provedor de nuvem (que fornece o armazenamento) tem condições de acessar os dados do usuário.

Não apenas o acesso aos dados é requisito de privacidade, a maneira como o provedor e os serviços de nuvem manuseiam os dados do usuário também devem ser levados em conta. Por exemplo, deve ser possível estipular um tempo de vida para o dado armazenado, ou até mesmo a localização de armazenamento do dado [Henze et al. 2013a]. A arquitetura UPECSI utiliza Anotações de Manuseio de Dados (do inglês *Data Handling Annotations*) [Henze et al. 2013b] para aplicar os requisitos de privacidade, tais anotações são enviadas com os dados para o provedor de nuvem, indicando como eles devem ser manuseados. Como mencionado anteriormente, a PDL também gera dados que possibilitam auditoria e monitoramento dos serviços de nuvem.

3. Arquitetura para Privacidade em *Cloud of Things*

Seguindo a tendência atual em que os dispositivos de uma rede IoT devem possuir capacidades de comunicação direta com a Internet, este trabalho propõe a remoção do Ponto de Aplicação de Privacidade. Os próprios dispositivos devem possuir funcionalidades de segurança e privacidade, e para tanto as funcionalidades do PEP foram categorizadas em protocolos com propósitos equivalentes, mas com implementação para dispositivos de baixo poder computacional. A arquitetura UPECSI envolve vários aspectos de um sistema de *Cloud of Things*, desde o desenvolvimento de serviços da nuvem até os protocolos de comunicação entre as redes IoT e o provedor de nuvem. O foco deste trabalho é a comunicação entre os dispositivos IoT e a nuvem, portanto os mecanismos definidos pela UPECSI que não são referentes a este aspecto não são mencionados na proposta, tais como a PDL citada na Seção 2.

A abordagem proposta apresenta pelo menos as seguintes vantagens: *(1)* melhora a tolerância a falhas da rede pois remove o dispositivo de borda, o qual propicia um ponto único de falha da arquitetura; *(2)* otimiza a segurança da rede pois remove um componente responsável por todas as tarefas relacionadas à segurança e, consequentemente, pode quebrar as propriedades de segurança do sistema uma vez que seja comprometido por um ataque bem sucedido; e *(3)* a execução da aplicação de mecanismos de segurança e privacidade nos dispositivos possibilita um controle fino sob os dados, pois cada dispositivo pode adotar uma política de segurança diferente.

A Figura 1 apresenta uma visão geral da arquitetura proposta, onde um usuário

possui várias redes IoT sob seu controle. O usuário vincula os dispositivos de suas redes ao provedor de nuvem, assim sendo os dados gerados por eles são armazenado na nuvem. Os serviços de nuvem com autorização podem então processar os dados e apresentá-los ao usuário. Uma Terceira Parte Confiável (TPC) é responsável por auditar e monitorar os serviços de nuvem, prover políticas de privacidade padrão (para usuários iniciantes) e participar de alguns mecanismos de segurança e privacidade envolvendo a comunicação dos dispositivos com a nuvem.

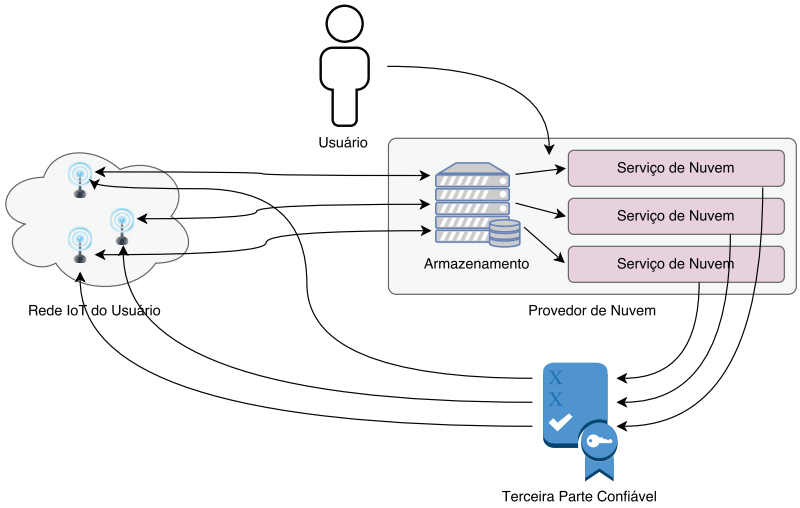

Figura 1. Arquitetura para Privacidade em *Cloud of Things*

Os dispositivos IoT armazenam os dados por eles gerados na nuvem de acordo com a Política de Privacidade. A aplicação da política possibilita que dados: *(1)* não sejam enviados para a nuvem, *(2)* sejam enviados cifrados, ou até mesmo *(3)* sejam enviados sem criptografia. Após o envio os requisitos de privacidade são de responsabilidade dos serviços de nuvem, pois os mesmos possuem as chaves necessárias para acessar os dados.

3.1. Vinculação dos Dispositivos ao Provedor de Nuvem

Assim como geralmente ocorre em provedores de nuvem privados, primeiramente o usuário precisa cadastrar uma conta para então ter acesso aos serviços. Após o processo de registro os usuários podem vincular seus dispositivos para então armazenar os dados gerados. O processo de vinculação é realizado com a utilização do protocolo OAuth 2.0 [Hardt 2012], um protocolo de código aberto para autorização segura. Após o processo de vinculação, os dispositivos podem enviar os dados gerados para a nuvem sem possuir as credenciais do usuário.

O protocolo OAuth 2.0 exige que a comunicação entre as partes seja segura para que o mesmo proporcione autorização segura. Na Internet essa exigência é cumprida com a utilização do procolo TLS, utilizado em conjunto com o TCP. Devido às características limitadas dos dispositivos IoT, ambos os protocolos (TCP e TLS) apresentam um custo muito alto para serem implementados nos dispositivos, dessa forma outra solução precisa ser utilizada. Este assunto atualmente está sendo estudado de forma ativa pela comunidade acadêmica, e já existem diversas propostas para prover canais seguros em redes IoT. Particularmente interessante, uma adaptação do Datagram Transport Layer

Security (DTLS)[McGrew and Rescorla 2010] está sendo estudada pelo grupo de trabalho *DTLS In Constrained Environments* (DICE) [Tschofenig and Fossati 2016]. O protocolo de vinculação é agnóstico ao protocolo da camada de transporte (que deve prover comunicação segura), portanto este trabalho não impõe a utilização de um protocolo específico.

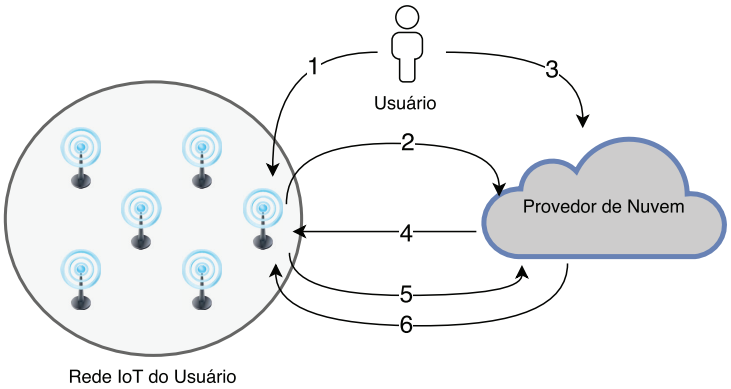

Figura 2. Processo de vinculação.

O usuário pode vincular um único dispositivo ou múltiplos dispositivos por vez. Neste trabalho mostramos como vincular um único dispositivo por vez, pois o processo de vinculação de múltiplos dispositivos pode ser realizado apenas com a utilização de um dispositivo de borda, de forma que todos os dispositivos IoT compartilham a mesma identificação e o usuário não será capaz de aplicar uma política de privacidade diferente a cada dispositivo. A vinculação é realizada utilizando o processo definido como Concessão de Código de Autorização (do inglês *Authorization Code Grant*) do protocolo OAuth, este tipo de permissão necessita da intervenção do usuário apenas na fase de configuração. Após receber a autorização, os dispositivos podem enviar dados para a nuvem sem utilizar as credenciais do usuário.

A Figura 2 apresenta o processo de vinculação para um dispositivo, o qual possui os seguintes passos:

1. O usuário inicia o processo como o *Resource Owner* acessando a página web de vinculação do dispositivo;
2. O dispositivo solicita o Código de Autorização para a nuvem;
3. O usuário é então redirecionado a uma página para inserir suas credenciais;
4. Após receber as credenciais do usuário a plataforma de nuvem envia o Código de Autorização ao dispositivo;
5. De posse do Código de Autorização, o dispositivo solicita um Token de Acesso;
6. A plataforma de nuvem envia o Token de Acesso.

Ao finalizar o processo o dispositivo possui um Token de Acesso que precisa ser anexado a todas as mensagens enviadas à nuvem. O provedor de nuvem autoriza o recebimento e armazenamento dos dados de acordo com o Token de Acesso. O processo de vinculação para múltiplos dispositivos é o mesmo, mas é executado pelo dispositivo de borda, que, ao receber o Token de Acesso, o envia a todos os dispositivos da rede. Como todos os dispositivos possuem o mesmo token, não há distinção de dispositivos pelo provedor de nuvem. Após vincular todos os dispositivos o usuário também pode criar redes lógicas para facilitar o gerenciamento.

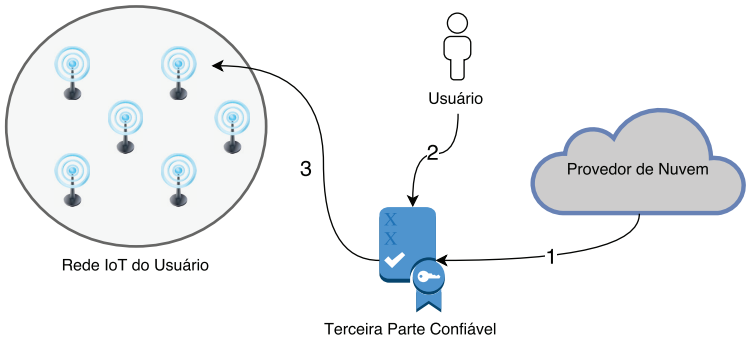

Figura 3. Atualização da Política de Privacidade.

3.2. Aplicação da Política de Privacidade

Uma Política de Privacidade (PP) define se um dado pode ser armazenado na nuvem, como ele será armazenado (cifrado ou não) e o que pode ser feito com este dado por um serviço de nuvem. Diferente de políticas de privacidade comuns, onde o usuário tem apenas a opção de aceitar ou rejeitar ela por completo, na arquitetura proposta o usuário tem o poder de alterar a sua política de privacidade com o tempo. Esta funcionalidade permite ao usuário assumir controle real sob os dados que envia para a nuvem. Serviços de nuvem fornecem ao usuário uma interface onde é possível analisar quais são os dados utilizados pelo serviço, e para quais propósitos, os usuários então podem habilitar ou desabilitar funções específicas do serviço de modo a restringir o acesso a determinados dados de seus dispositivos. Por exemplo, um usuário pode desabilitar a utilização de dados de localização por um determinado serviço se assim desejar.

A Política de Privacidade é aplicada toda vez que um dispositivo IoT envia dados para a nuvem, tal aplicação é realizada pelo próprio dispositivo. A Figura 3 apresenta o processo de atualização de uma PP:

1. O serviço de nuvem envia a Política de Privacidade à Terceira Parte Confiável (TPC) para auditoria. Ao receber a Política de Privacidade, a TPC audita a PP e gera uma Configuração de Privacidade (CP);
2. O usuário acessa a CP por meio de uma interface (navegador, *smartphone*, etc) e realiza as configurações desejadas;
3. A TPC envia aos dispositivos a CP revisada pelo usuário.

3.3. Controle de Acesso dos Dados

Após vincular os dispositivos ao provedor de nuvem e configurar as Políticas de Privacidade referentes aos serviços utilizados, é necessário garantir que os dados vindo dos dispositivos do usuário são armazenados e acessados apenas por entidades autorizadas.

O controle de acesso é implementado utilizando mecanismos baseados em criptografia. Assim como no processo de vinculação, a comunicação entre os dispositivos IoT e a nuvem deve ser segura, o que é garantido por um protocolo da camada de transporte. Os dados sensíveis vindo dos dispositivos são armazenados cifrados na nuvem (dados que não são sensíveis podem ser armazenados sem criptografia), dessa forma é prevenido o acesso ao dado até mesmo pelo provedor de nuvem. As chaves utilizadas para decifrar os dados são armazenadas em um repositório de chaves (também na nuvem).

Antes do envio do dado, o dispositivo IoT deve filtrá-lo de acordo com a Configuração de Privacidade, dessa forma decidindo se o dado deve deixar a esfera de controle do usuário. O dado é então cifrado com um algoritmo de criptografia simétrica. A chave utilizada por tal algoritmo é então cifrada com a chave pública dos serviços autorizado a acessar o respectivo dado. Caso haja mais de um serviço utilizando o mesmo dado a chave simétrica deve ser cifrada uma vez para cada serviço. O dado é armazenado apenas uma vez no provedor, independente do número de serviços que o acessarão.

A chave simétrica é atualizada periodicamente, prevenindo que novos serviços acessem dados armazenados previamente à sua autorização, ou que serviços com autorização cancelada continuem a acessar os dados do usuário. As chaves antigas continuam no repositório de chaves, possibilitando o acesso a dados já armazenados por novos serviços.

Como pode ser visto, os dispositivos IoT executam operações criptográficas. No caso de dispositivos limitados, com fonte de energia restrita, operações criptográficas podem diminuir significativamente o tempo de vida da rede. Duas abordagens são propostas para mitigar este problema, as chaves criptográficas são gerenciadas pelos dispositivos IoT (Seção 3.3.1) ou pela Terceira Parte Confiável (Seção 3.3.2). A seguir tais abordagens são apresentadas e suas vantagens e desvantagens discutidas.

3.3.1. Gerenciamento das Chaves pelos Dispositivos IoT

A Figura 4 apresenta a abordagem onde o próprio dispositivo é responsável por criar as chaves simétricas, cifrá-las com as chaves públicas dos serviços e enviá-las para o repositório de chaves no provedor de nuvem.

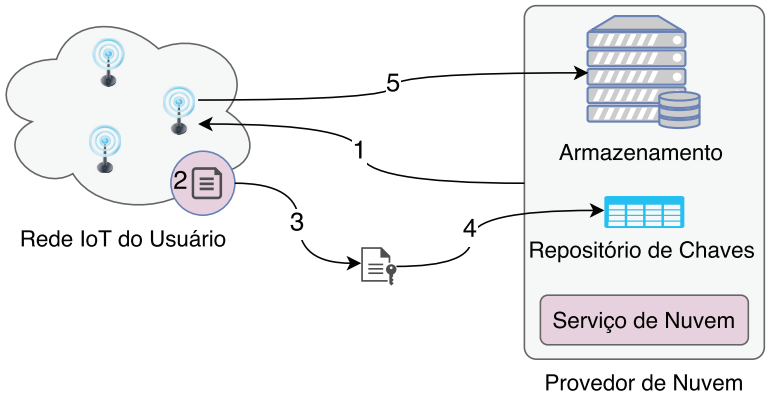

Figura 4. Gerenciamento das Chaves pelos Dispositivos IoT.

Esta abordagem é proposta para dispositivos que possuam capacidade computacional e energética suficientes para executar as seguintes tarefas (Figura 4) sem impactar significativamente seu tempo de vida:

1. Receber as chaves públicas dos serviços pelo provedor de nuvem;
2. Criar periodicamente chaves simétricas;
3. Cifrar as chaves simétricas com as chaves públicas dos serviços autorizados;
4. Enviar as chaves simétricas cifradas ao repositório de chaves;
5. Enviar dados cifrados ao provedor de nuvem.

3.3.2. Gerenciamento das Chaves por uma Terceira Parte Confiável

A Figura 5 apresenta a segunda abordagem, onde uma Terceira Parte Confiável é responsável pela maioria das tarefas relacionadas ao controle de acesso dos dados, atenuando a carga nos dispositivos IoT. A TPC cria as chaves simétricas, recebe as chaves públicas dos serviços, cifra as chaves simétricas e as envia ao repositório de chaves. As chaves simétricas também são enviadas aos dispositivos.

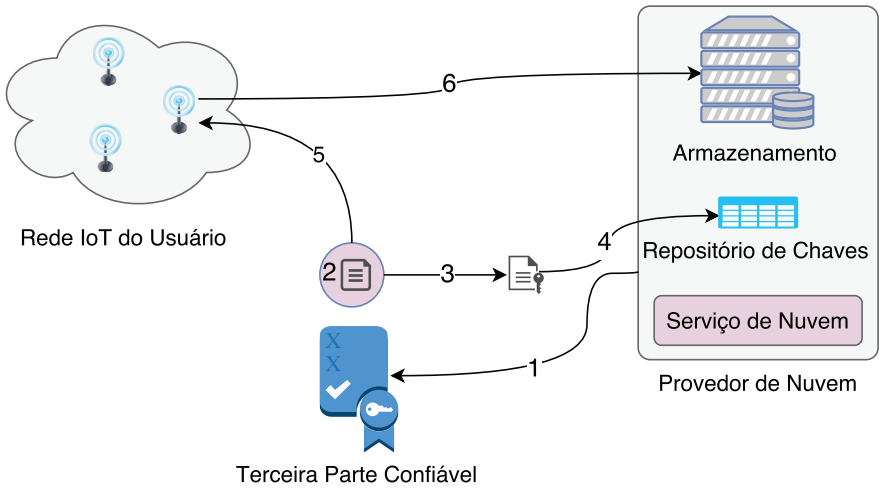

Figura 5. Gerenciamento das Chaves por uma TPC.

Esta abordagem é vantajosa para dispositivos limitados, onde as tarefas migradas à TPC diminuiriam o atraso na comunicação e aumentariam a expectativa de vida do aparelho. Nesta abordagem são executados os seguintes passos:

1. **TPC**: Recebe as chaves públicas dos serviços pelo provedor de nuvem;
2. **TPC**: Cria periodicamente chaves simétricas;
3. **TPC**: Cifra as chaves simétricas com as chaves públicas dos serviços autorizados;
4. **TPC**: Envia as chaves simétricas cifradas ao repositório de chaves;
5. **TPC/Dispositivo IoT**: Envia (de forma segura) as chaves simétricas aos dispositivos IoT;
6. **Dispositivo IoT**: Envia dados cifrados ao provedor de nuvem.

Em ambas as abordagens os dados são cifrados duas vezes, uma vez na camada de transporte, pelo protocolo que garante um canal seguro, e outra vez na camada de aplicação, pelo controle de acesso aos dados. A criptografia efetuada na camada de aplicação é sempre necessária, pois o dado é armazenado em forma cifrada. Consequentemente, para prevenir que o dado seja cifrado duas vezes é necessário remover a cifragem realizada na camada de transporte. A mensagem enviada pelos dispositivos contem os dados cifrados e o Token de Acesso, que então precisa ser protegido na camada de aplicação. Isso pode ser alcançado simplesmente compartilhando uma chave entre o provedor de nuvem e o dispositivo de forma segura. Este processo pode ser realizado na etapa de vinculação, onde é utilizado um canal seguro, desta forma os dispositivos podem cifrar o Token de Acesso com a chave compartilhada e enviar todas as informações de forma segura.

3.4. Políticas de Privacidade Flexíveis

O controle de acesso aos dados fornecido pela arquitetura proposta permite apenas o acesso de agentes previamente autorizados. Entretanto, dependendo de certas situações, seria conveniente diminuir a privacidade do usuário, como por exemplo em emergências médicas. Em situação normal os dados do usuário iriam apenas para o médico vinculado, mas em uma emergência qualquer médico disponível teria acesso aos dados do usuário para poder trata-lo.

Para tratar situações como a descrita o usuário deve criar Políticas de Privacidade alternativas. Tais políticas são acionadas por eventos internos, definidos por valores coletados que ultrapassam um limiar, ou externos, comumente gerados pelos serviços. O processo de criação e atualização da PP Flexível é idêntico ao da PP comum (ver Seção 3.2).

Dados que serão liberados quando políticas flexíveis forem acionadas passam por um processo de criptografia semelhante aos dados comuns, no entanto a chave simétrica utilizada não é criptografada com a chave pública dos serviços e sim com uma chave pública criada pela TPC. Quando eventos indicam o acionamento da PP Flexível, o dispositivo aciona a TPC, que por sua vez envia a chave privada (referente a chave pública criada) aos serviços que devem ter acesso aos dados.

Para prevenir acesso a dados gerados quando PP Flexíveis deixam de estar em vigor, o par de chaves pública/privada deve ser reconfigurado, i.e., a chave privada antiga deve ser revogada e uma nova chave privada deve ser obtida. Esse mecanismo, diferente da aplicação de privacidade em situações comuns, requer a utilização da TPC, pois a mesma apresenta maior confiabilidade que os dispositivos IoT.

4. Discussões

Esta seção apresenta uma ampla discussão a respeito de aspectos relacionados com a arquitetura apresentada na seção anterior. Primeiramente é discutida a abordagem de transferência das funcionalidades do *gateway* para os dispositivos, principalmente em relação a segurança da comunicação fim-a-fim. Em seguida é discutida a proposta desse trabalho, indicando suas limitações, vantagens e desvantagens.

4.1. Comunicação Direta dos Dispositivos com a Internet

Um aspecto importante da arquitetura proposta é a utilização de protocolos equivalentes aos da Internet nos dispositivos IoT, proporcionando a habilidade de comunicação direta entre os dispositivos e a nuvem. Baseado no UPECSI [Henze et al. 2016], este trabalho transfere as funcionalidades do Ponto de Aplicação de Privacidade para os dispositivos IoT, adaptando tais funcionalidades de acordo com a natureza limitada dos dispositivos. Apesar dos dispositivos se comunicarem diretamente com a nuvem, *gateways* podem ser utilizados para tradução de protocolos. Por exemplo, o IEEE 802.15.4 define um pacote de no máximo 127 bytes, mas muitos protocolos da Internet possuem cabeçalhos e mensagens que utilizam a maior parte deste pacote, deixando pouco espaço para dados da aplicação (por exemplo, o cabeçalho do IPv6 possui no mínimo 40 bytes).

O UPECSI define mecanismos de comunicação entre o *gateway* e a nuvem, este trabalho propõe uma arquitetura segura entre os dispositivos IoT e a nuvem, relativos apenas à camada de aplicação. Contudo, uma discussão a respeito da segurança na camada

de transporte é pertinente, pois é um tópico sendo amplamente estudado. Segurança nas camadas de Enlace e de Rede na Internet das Coisas, apesar de apresentar muitos desafios, pode ser considerada mais madura, já que o IEEE 802.15.4, o 6LoWPAN e o RPL [Winter 2012] são protocolos consolidados.

O grande dificultador da implementação de mecanismos de segurança na Internet das Coisas é o custo adicional causado pela quantidade de processamento e pelas mensagens extras necessárias para realização de operações de criptografia. Os protocolos da Internet utilizam a Criptografia de Chave Pública [Salomaa 2013], ou criptografia assimétrica, onde uma chave pública é utilizada para cifrar e uma chave privada para decifrar a mensagem. A criação e troca de chaves entre os pares são especialmente custosas, e por isso muitos trabalhos assumem o carregamento prévio das chaves nos dispositivos. Neste trabalho é assumida a utilização de um protocolo de camada de transporte que assegure um canal seguro e confiável. Apesar de várias propostas atuais com este fim [Sethi et al. 2012, Hummen et al. 2014, Tschofenig and Fossati 2016], o IETF atualmente trabalha para definir um protocolo padrão para utilização em redes de dispositivos limitados.

4.2. Arquitetura Proposta

Esta seção traz uma discussão acerca de aspectos relacionados com os protocolos implementados nos dispositivos de IoT.

4.2.1. Vinculação

O processo de vinculação é uma aplicação direta do protocolo OAuth 2.0. Visto que esse processo acontece apenas uma vez, a carga adicionada aos dispositivos pode ser negligenciada. Esse processo também pode ser realizado por um *gateway*, como definido no UPECSI, dessa forma todos os dispositivos partilham de um mesmo *Access Token*. Esse recurso agiliza a etapa de vinculação da rede, mas diminui a granularidade de controle do usuário, pois todos os dispositivos serão obrigatoriamente vinculados ao mesmo provedor.

O protocolo OAuth 2.0 presume a utilização de um canal seguro entre o dispositivo e o provedor, pois não dispõe de recursos para manter a confidencialidade das mensagens trocadas. Caso o processo de vinculação seja realizado por cada dispositivo, é necessário que eles disponham de uma interface web. Levando em conta sua natureza limitada, é importante que a implementação dessa etapa seja realizada de forma otimizada.

4.2.2. Aplicação da Política de Privacidade

A aplicação da Política de Privacidade é realizada da mesma forma que no UPECSI, transferindo o processo do *gateway* ao dispositivo. A alteração da PP é realizada através do serviço de nuvem, por meio de uma interface *Web*. O dispositivo IoT recebe a nova informação assim que gerada, dessa forma o processo de atualização da PP não ocasiona uma carga considerável aos dispositivos, já que não deve acontecer com frequência. O procedimento de aplicação da PP é realizado por cada dispositivo, o qual de posse da PP deve filtrar todas as mensagens a serem enviadas ao provedor (previamente ao envio).

Como esta filtragem não exige grande processamento, sua implementação também não deve ocasionar aumento significativo da carga no dispositivo.

4.2.3. Controle de Acesso aos Dados

A privacidade do usuário é garantida por meio de criptografia. O dado é armazenado no provedor de forma cifrada, e apenas os serviços autorizados possuem a chave para decriptá-lo. Este mecanismo permite que apenas entidades previamente autorizadas tenham acesso ao dado. Como já discutido, operações criptográficas podem ser custosas para dispositivos IoT. Embora os dispositivos IoT precisem cifrar os dados, algumas operações podem ser transferidas para a Terceira Parte Confiável, alivando a carga nos dispositivos. Nesta abordagem o gerenciamento das chaves é realizado pela TPC, e os dispositivos precisam apenas receber as chaves e cifrar os dados. Esta abordagem é recomendada a redes IoT onde os dispositivos possuam recursos limitados.

Os próprios dispositivos podem ser responsáveis por criar as chaves simétricas periodicamente e enviá-las aos serviços autorizados, sem o intermédio da Terceira Parte Confiável. Este método introduz maior carga aos dispositivos, que ficam responsáveis por tarefas de alto custo computacional: criação de chaves simétricas, encriptação das chaves simétricas com as chaves públicas dos serviços e envio das chaves encriptadas aos serviços. Vale notar que a encriptação e o envio é realizada uma vez para cada serviço sendo utilizado, e esse processo repete-se periodicamente, deve-se ter cautela ao utilizar tal método, pois caso o período de renovação das chaves seja muito pequeno os dispositivos podem ter uma sobrecarga. A vantagem desta abordagem é a remoção da Terceira Parte Confiável do processo, tornando o protocolo mais seguro pois diminui a superfície de ataque. Este método é indicado para redes IoT de moderada capacidade computacional e de grande disponibilidade de energia.

Em ambas as abordagens o dado será cifrado duas vezes, uma na camada de transporte, por um protocolo que garanta um canal seguro, e outra na camada de aplicação, pela arquitetura proposta neste trabalho. A cifra realizada pela camada de aplicação será sempre necessária, pois é nesta forma que o dado é armazenado na nuvem. Para remover a criptografia da camada de transporte é preciso assegurar o envio confidencial da mensagem na camada de aplicação. A mensagem enviada pelos dispositivos possui os dados cifrados e o *Access Token*, que agora precisa ser protegido, o que pode ser alcançado compartilhando uma chave entre a plataforma de nuvem e o dispositivo anteriormente ao envio dos dados. Este compartilhamento pode ser realizado ao final do processo de vinculação, o qual utiliza um canal seguro. Desta forma, os dispositivos IoT podem cifrar o *Access Token* e os dados gerados.

4.2.4. Políticas de Privacidade Flexíveis

Políticas de Privacidade Flexíveis foram propostas para aumentar a flexibilidade da arquitetura. Este protocolo possibilita ao usuário definir limites, para que serviços de nuvem sejam habilitados ou desabilitados. Este mecanismo é especialmente útil em situações excepcionais, onde pode ser mais vantajoso abdicar da privacidade.

Uma vez que o principal propósito das Políticas de Privacidade Flexíveis são situações excepcionais, pode ser o caso em que o próprio dispositivo IoT deixe de funcionar, desta forma o acionamento da PPF é possível apenas por um agente externo. Para assegurar o correto funcionamento em uma situação como esta a TPC é responsável pela aplicação da PPF, gerenciando as chaves e autorizando os serviços quando necessário. Este método torna o protocolo mais confiável e introduz pouca carga aos dispositivos, pois apenas uma mensagem é enviada do dispositivo à TPC para ativação da PPF.

5. Conclusões

A integração de Internet das Coisas com Computação em Nuvem traz muitos benefícios para as aplicações, pois enquanto a primeira produz uma grande quantidade de dados, a segunda possui poder suficiente para armazenar e processar estas informações, incluindo soluções para *big data*. Um aspecto extremamente relevante nesta integração é a segurança, principalmente relacionada com a privacidade dos dados dos usuários, visto que os mesmos saem da esfera de controle do usuário ao serem enviados para a nuvem.

Visando contornar ou pelo menos mitigar este problema, este trabalho propõe uma arquitetura para que os usuários possam controlar o acesso aos dados gerados pelos dispositivos de IoT em seu controle. A solução proposta possibilita uma granularidade fina neste controle, i.e., o controle pode ser realizado por dispositivo. Como trabalhos futuros, pretende-se definir cenários e analisar o desempenho das soluções propostas.

Referências

Akyildiz, I. F. and Vuran, M. C. (2010). *Wireless sensor networks*, volume 4. John Wiley & Sons.

Botta, A., de Donato, W., Persico, V., and Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, 56:684–700.

Chui, M., Löffler, M., and Roberts, R. (2010). The internet of things. *McKinsey Quarterly*, 2(2010):1–9.

Deering, S. E. (1998). Internet protocol, version 6 (ipv6) specification. RFC 2460.

Eggert, M., Häußling, R., Henze, M., Hermerschmidt, L., Hummen, R., Kerpen, D., Pérez, A. N., Rumpe, B., Thißen, D., and Wehrle, K. (2014). Sensorcloud: Towards the interdisciplinary development of a trustworthy platform for globally interconnected sensors and actuators. In *Trusted Cloud Computing*, pages 203–218. Springer.

Fox, G. C., Kamburugamuve, S., and Hartman, R. D. (2012). Architecture and measured characteristics of a cloud based internet of things. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pages 6–12. IEEE.

Granjal, J., Monteiro, E., and Silva, J. S. (2015). Security in the integration of low-power wireless sensor networks with the internet: A survey. *Ad Hoc Networks*, 24:264–287.

Group, W. W. P. A. N. W. W. (2006). *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE Standard for Information Technology.

Hardt, D. (2012). The oauth 2.0 authorization framework. RFC 6749.

Henze, M., Großfengels, M., Koprowski, M., and Wehrle, K. (2013a). Towards data handling requirements-aware cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 2, pages 266–269. IEEE.

Henze, M., Hermerschmidt, L., Kerpen, D., Häußling, R., Rumpe, B., and Wehrle, K. (2016). A comprehensive approach to privacy in the cloud-based internet of things. *Future Generation Computer Systems*, 56:701–718.

Henze, M., Hummen, R., and Wehrle, K. (2013b). The cloud needs cross-layer data handling annotations. In *Security and Privacy Workshops (SPW), 2013 IEEE*, pages 18–22. IEEE.

Hummen, R., Shafagh, H., Raza, S., Voig, T., and Wehrle, K. (2014). Delegation-based authentication and authorization for the ip-based internet of things. In *Annual IEEE International Conference on Sensing, Communication, and Networking*.

Kushalnagar, N., Montenegro, G., and Schumacher, C. (2007). Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals. Technical report.

McGrew, D. and Rescorla, E. (2010). Datagram transport layer security (dtls) extension to establish keys for secure real-time transport protocol (srtp).

Postel, J. (1980). User datagram protocol (udp). RFC 768.

Postel, J. (1981). Transmission control protocol (tcp). RFC 793.

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.

Salomaa, A. (2013). *Public-key cryptography*. Springer Science & Business Media.

Sautner, B. (2017). Nimbits. `http://bsautner.github.io/com.nimbits/`, Acessado em Março de 2017.

Sethi, M., Arkko, J., and Keränen, A. (2012). End-to-end security for sleepy smart object networks. In *IEEE Conference on Local Computer Networks Workshops*.

Sundmaeker, H., Guillemin, P., Friess, P., and Woelfflé, S. (2010). Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision*.

Tschofenig, H. and Fossati, T. (2016). Transport layer security (tls)/datagram transport layer security (dtls) profiles for the internet of things. RFC 7925.

Winter, T. (2012). Rpl: Ipv6 routing protocol for low-power and lossy networks. RFC 6550.