Capítulo 4

Infraestruturas de Autenticação e de Autorização para Internet das Coisas

Michelle S. Wangham[†], Marlon Cordeiro Domenech[†], Emerson Ribeiro de Mello[*]

[†] Universidade do Vale do Itajaí
wangham@univali.br, marloncdomenech@gmail.com

[*] Instituto Federal de Santa Catarina
mello@ifsc.edu.br

Abstract

The next step in growth of the Internet is the extensive integration of networked physical objects, known as things. The Internet of Things (IoT) paradigm is characterized by the diversity of devices that cooperate to achieve a common goal. In this environment, compose of constrained devices, the widespread adoption of this paradigm depends of security requirements like secure communication between devices, privacy and anonymity of its users. This chapter presents the main security challenges and solutions to provide authentication and authorization on the Internet of Things

Resumo

O próximo salto no crescimento da Internet está na ampla integração de objetos físicos do dia a dia, denominados coisas, conectados em rede. O paradigma da Internet das Coisas (Internet of Things – IoT) é caracterizado pela diversidade de dispositivos que cooperam entre si a fim de atingir um objetivo comum. Neste ambiente, composto por dispositivo com poucos recursos computacionais, garantir a comunicação segura entre estes dispositivos, a privacidade e o anonimato de seus usuários são requisitos de segurança fundamentais para a ampla adoção deste paradigma. Neste capítulo, são apresentados os principais desafios e soluções de segurança para prover autenticação e autorização na Internet das Coisas.

4.1. Introdução

O próximo salto no crescimento da Internet está fundamentado no paradigma da Internet das Coisas (*Internet of Things* – IoT) o qual abrange uma infraestrutura de *hardware*, *software* e serviços que conectam objetos físicos, denominados como coisas, à rede de computadores [ITU 2005, COMMUNITIES 2008]. Segundo [Atzori et al. 2010], a ideia básica de IoT consiste na presença de uma diversidade de objetos que interagem e cooperam entre si a fim de atingir um objetivo comum, para tal compartilham informações utilizando métodos de endereçamento único e protocolos de comunicação padronizados.

A integração entre sensores e atuadores sobre a Internet forma a base tecnológica para o conceito de ambientes inteligentes, nos quais a informação gerada por um objeto pode ser compartilhada entre diversas plataformas e aplicações [Gubbi et al. 2013]. O conceito de ambientes inteligentes engloba diferentes tecnologias, tais como redes de sensores sem fio (RSSF) e sistemas de identificação por rádio frequência (*Radio-Frequency IDentification* – RFID) integrados para rastrear estados das coisas, como sua localização, temperatura, movimentos, etc [Atzori et al. 2010].

Outro conceito importante no cenário de IoT é o de Web das Coisas (*Web of Things* – WoT). A principal característica na WoT é a adoção de protocolos usados amplamente em aplicações web, como por exemplo, o HTTP, cujo principal ganho está na facilidade de integração entre os serviços da WoT e outros serviços e sistemas disponíveis na Internet [Guinard e Trifa 2009]. Com o aumento da adoção de aplicações para IoT e WoT, a preocupação com a segurança das informações aumentará o sucesso do uso desta tecnologia emergente e assim estará fundamentado no nível de segurança que o ambiente poderá fornecer para os usuários, como por exemplo, a confidencialidade dos dados trafegados, bem como a privacidade dos usuários [ITU 2005].

A IoT apresenta requisitos singulares que demandam abordagens diferenciadas acerca da segurança. Segundo [Babar et al. 2011], acrescentar mecanismos de segurança em dispositivos embarcados com restrições computacionais pode ser um desafio. Diante da heterogeneidade dos dispositivos, desenvolver mecanismos de segurança que possam ser executados em diferentes plataformas é um requisito importante para a IoT. Por fim, os autores afirmam que o acesso físico aos dispositivos é facilitado em função do tipo de ambiente nos quais os objetos estão inseridos. Assim, são necessárias não só a proteção lógica mas também física destes dispositivos.

Dentre o conjunto de requisitos de segurança para IoT, cabe destacar: a gestão de identidade de usuários e dispositivos; a confidencialidade dos dados trocados na comunicação; a disponibilidade de recursos e sistemas; e o controle de acesso à rede para garantir somente dispositivos autorizados [Babar et al. 2011].

Pode-se atender a estes requisitos de segurança por meio de uma infraestrutura de autenticação e de autorização. Com esta infraestrutura, é possível implantar a gestão de identidades de forma a impedir que usuários ou dispositivos não autorizados tenham acesso aos recursos, impeça que usuários ou dispositivos legítimos acessem recursos para os quais não estejam autorizados e permita que usuários ou dispositivos legítimos tenham acesso aos recursos a estes autorizados [Liu et al. 2012]. Embora a autenticação e autorização de usuários seja bem abordada na literatura, a autenticação e autorização de dispositivos

não é bem caracterizada e, segundo [Miorandi et al. 2012], é um desafio de pesquisa neste cenário.

O objetivo deste capítulo é discutir os desafios de segurança e as infraestruturas de autenticação e de autorização que proveem gestão de identidades para Internet das Coisas. As seguintes questões chaves são analisadas neste capítulo: autenticação única (*Single Sign On -SSO*) de usuários e de dispositivos, gerenciamento de relações de confiança entre domínios administrativos diferentes e interoperabilidade entre mecanismos de autenticação e de autorização. Conceitos, requisitos e soluções tecnológicas encontradas na literatura são complementados com a apresentação de dois cenários de uso para IoT que demonstram a aplicabilidade das infraestruturas de autenticação e de autorização.

Este capítulo está dividido em sete seções. Nesta primeira seção, foi apresentada uma contextualização, destacando os objetivos e a motivação para a escolha do tema. Na Seção 4.2, são apresentados os principais conceitos, características, bem como as tecnologias envolvidos e os domínios de aplicação na IoT. A Seção 4.3 apresenta as principais características de IoT que fazem com que a segurança seja tratada de maneira distinta em relação aos demais sistemas computacionais, bem como os principais requisitos de segurança na IoT, as ameaças e os ataques descritos na literatura. Na Seção 4.4, são descritos os principais conceitos e técnicas de autenticação de usuários e de dispositivos e de mecanismos de autorização apropriados ao cenário para IoT. As principais infraestruturas de autenticação e de autorização adotadas na Internet e que são também empregadas na Internet das Coisas são analisadas na Seção 4.5. A Seção 4.6 apresenta uma análise dos principais projetos de pesquisa que tratam a gestão de identidades para IoT e os trabalhos acadêmicos mais relevantes que discutem infraestruturas de autenticação e de autorização. Por fim, a Seção 4.7 traz uma síntese dos principais aspectos da gestão de identidades na IoT analisados e as tendências de pesquisa nesta área.

4.2. Visão Geral sobre Internet das Coisas

O próximo passo para o crescimento da Internet é a integração de objetos físicos do dia a dia (coisas) às redes de comunicação [COMMUNITIES 2008]. Em 2010, havia aproximadamente 1,5 bilhão de computadores pessoais e mais de 1 bilhão de celulares com acesso à Internet. Para 2020, é esperado que algo entre 50 e 100 bilhões de dispositivos estejam conectados à Internet [CERP-IoT 2010]. [Babar et al. 2011] afirmam que na IoT estão coisas como roupas, mobília, carros, *smartcards*, dispositivos médicos, medidores de consumo e máquinas industriais. O paradigma de IoT integra uma grande variedade de conceitos e áreas, tais como: eletrônica, automação, redes de comunicação, biotecnologia, mecânica e tecnologia dos materiais [Xiang e Li 2012].

Segundo o relatório [ITU 2005], a IoT pode trazer mudanças à sociedade em geral na maneira como o indivíduo se relaciona com o ambiente, assim como na maneira como serão realizados os processos de negócio. Além da comunicação e informação a qualquer momento, em qualquer lugar, na IoT é possível também a conectividade para qualquer coisa, como pode ser visto na Figura 4.1.

Conforme [Gubbi et al. 2013], os avanços e a convergência das tecnologias de sistemas microeletromecânicos, comunicação sem fio e eletrônica digital resultaram no desenvolvimento de dispositivos em miniatura com a capacidade de sentir, computar e

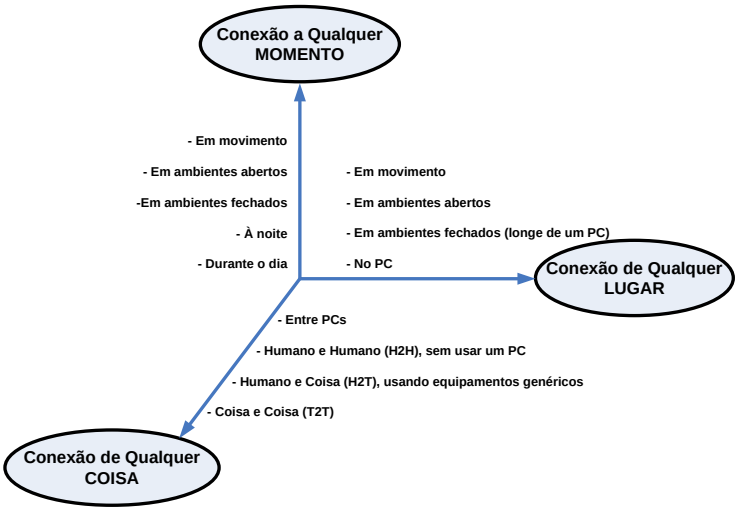

Figura 4.1: Nova dimensão para comunicação e compartilhamento de informação [ITU 2005]

se comunicar via rede sem fio a curtas distâncias. Deste cenário deriva o conceito de ambientes inteligentes (*smart environments*). Vários países estão desenvolvendo projetos de Cidades Inteligentes (*smart cities*), que oferecem experiências inovadoras em transporte, preservação ambiental, convivência e economia de energia. Mundialmente, se reconhece o potencial da tecnologia de IoT para criar ambientes inteligentes através dos *smart objects* [Schaffers et al. 2011].

As principais características da Internet das Coisas estão indicadas no mapa conceitual da Figura 4.2 e são:

- A IoT pode ser caracterizada como uma rede mundial de coisas/objetos/dispositivos interconectados que se comportam como entidades ativas [Roman et al. 2011b];
- As coisas (dispositivos) na IoT, muitas vezes, possuem restrições de recursos como memória RAM ou ROM, poder de processamento e energia [Hummen et al. 2013];
- Mecanismos de comunicação de alguns dispositivos, na maioria das vezes sem fio, possuem baixa potência de transmissão e baixa taxa de dados [Mahalle et al. 2010];
- Há uma grande quantidade de coisas (dispositivos) com ciclo curto de vida, o que exige uma alta capacidade de gerenciamento [Fongen 2012];
- Integra coisas (dispositivos) heterogêneos, o que demanda uma preocupação em relação a interoperabilidade entre estes [Atzori et al. 2010, Mahalle et al. 2012];
- A rede possui uma topologia dinâmica, pois muitos nós entram e saem da rede com frequência [Mahalle et al. 2012, Hanumanthappa e Singh 2012];
- Pode ser caracterizada como um ambiente contendo um grande número computadores ou dispositivos invisíveis que colaboram com o usuário, ou seja, um ambiente pervasivo e ubíquo [Hanumanthappa e Singh 2012];

- Na IoT, os usuários podem interagir com as coisas em seu ambiente físico e virtual de diversas maneiras [Mahalle et al. 2012].

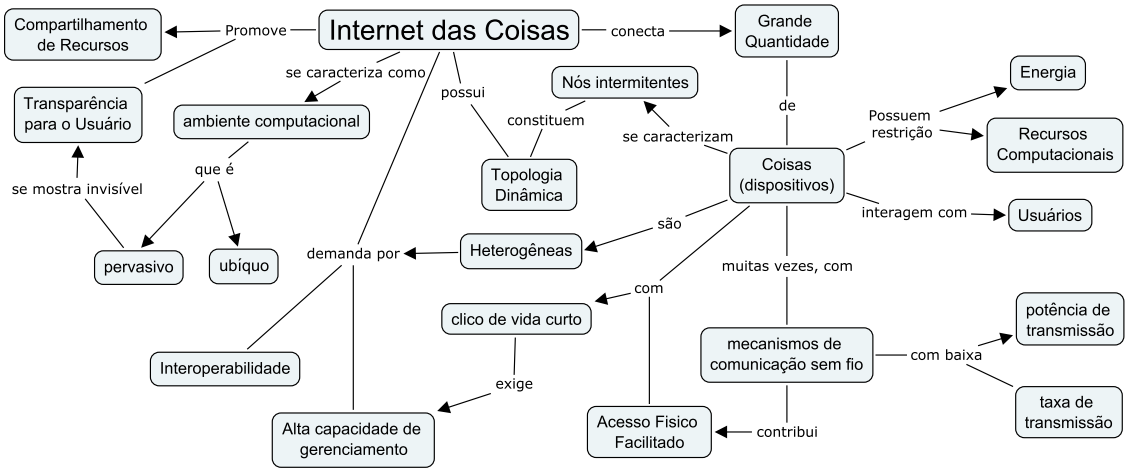

Figura 4.2: Mapa Conceitual sobre Internet das Coisas

De acordo com [Roman et al. 2011b], as coisas na IoT possuem cinco características principais e uma opcional. São estas:

- Existência: coisas que existem no mundo real podem também existir no mundo virtual (IoT), por meio de dispositivos de comunicação embarcada;
- Auto conhecimento (do inglês, *sense of self*): todas as coisas têm, explicitamente ou implicitamente, uma identidade que as descreve. As coisas podem processar informação, tomar decisões e se comportar de maneira autônoma;
- Conectividade: as coisas podem iniciar a comunicação com outras entidades. Dessa maneira, a comunicação com entidades nas suas proximidades ou em ambientes remotos é possível.
- Interatividade: as coisas podem interoperar e colaborar com uma variedade de entidades heterogêneas, seja humanos ou máquinas virtuais ou reais. Desse modo, estas produzem e consomem uma grande variedade de serviços;
- Dinamicidade: as coisas podem interagir entre si a qualquer momento, lugar ou maneira. Estas podem entrar e sair de uma rede conforme quiserem, não estando limitadas a um único local físico, podendo usar uma grande variedade de interfaces;
- (opcional) Ciência do ambiente: sensores podem permitir que as coisas percebam as características de seu ambiente, por exemplo, a sobrecarga da rede ou a radiação da água. Esta característica é opcional, pois nem todas as coisas possuem esta capacidade, como por exemplo, um objeto com uma etiqueta RFID.

[Gubbi et al. 2013] afirmam ainda que o ambiente de IoT culmina com a geração de enormes quantidades de dados que precisam ser armazenados, processados e apresentados

de uma maneira eficiente e fácil de interpretar. Os autores afirmam que o conceito de Computação em Nuvem (*Cloud Computing*) completa o conceito de IoT no intuito de prover o sensoriamento ubíquo. Uma infraestrutura em nuvem, na qual seja possível armazenar uma grande quantidade de dados e fornecer esses dados para que aplicações possam ser construídas, com requisitos de disponibilidade, capacidade de processamento e alocação de recursos sob demanda, é necessária para que os ambientes inteligentes sejam de fato escaláveis e altamente disponíveis.

Pode-se citar como exemplo desta integração (Cloud e IoT) o projeto europeu OpenIoT cujo objetivo é prover um *middleware open-source* para o desenvolvimento de aplicações de IoT, usando o modelo baseado em computação em nuvem. Os objetos conectados à Internet podem ser acessados por serviços IoT na nuvem. Por exemplo, o sensoriamento deste objeto pode ser um serviço disponibilizado na nuvem (*Sensing as a Service*). Através do uso de serviços de IoT, usuários podem configurar e desenvolver aplicações de IoT. O projeto visa ainda fornecer infraestrutura e aplicações IoT na nuvem, formando uma nuvem de coisas (*Cloud of Things*) [OPENIoT 2012].

Para compreender o paradigma da IoT, [Atzori et al. 2010] apresentam os principais conceitos, tecnologias e padrões envolvidos, a partir de três perspectivas, a saber:

- **Visão orientada às coisas**. Considera as coisas como itens simples, por exemplo, etiquetas RFID (*Radio-Frequency IDentification*), porém, não somente estas coisas simples. Trata de aspectos como endereçamento único e global (para acesso direto às coisas por meio da Internet) e da identificação unívoca das coisas. Um dos pontos relevantes dessa visão é que, para a efetiva concretização da IoT, contata-se a necessidade de aumentar a inteligência das coisas (conceber *smarts things*);
- **Visão orientada à Internet**. Responsável pelos protocolos necessários e como esses devem ser adaptados para permitir a troca de informações entre as coisas na IoT. Nessa visão, estão as pesquisas e padrões que tratam da adaptação do protocolo IP para o ambiente de IoT;
- **Visão orientada à Semântica**. A quantidade de coisas conectadas à rede (Internet do Futuro) está destinada a ser muito alta. Esta visão incluem questões como: representação, armazenamento, busca e organização da grande quantidade de dados gerados na IoT. Neste contexto, as tecnologias semânticas deverão desempenhar um papel fundamental, pois serão usadas para modelagem das coisas, para extração do conhecimento dos dados, para o raciocínio sobre os dados gerados na IoT, para criação de ambientes de execução semânticos e para definição da arquitetura que irá acomodar os requisitos da IoT.

4.2.1. Tecnologias Envolvidas

[Atzori et al. 2010] afirmam que a realização do conceito de IoT no mundo real é possível por meio do uso e integração de várias tecnologias habilitadoras. Nesta seção, são apresentadas algumas tecnologias que tornam a IoT realizável, são estas: RFID (*Radio Frequency Identification*), WSN (*Wireless Sensor Network*) e WSAN (*Wireless Sensor and Actuator Network*), EnHANTs (*Energy-Harvesting Active Networked Tags*), NFC (*Near-Field Communications*) e RSN (*RFID Sensor Network*).

RFID (*Radio Frequency Identification*)

RFID é um método para transmissão de informações sem a necessidade de contato físico ou linha de visada entre as partes participantes da comunicação [Yan et al. 2008]. Um dispositivo RFID, muitas vezes chamado apenas de etiqueta (*tag*) RFID, é um microchip projetado para transmissão de dados sem fio, que transmite dados em resposta a interrogação feita por um leitor RFID [Juels 2006].

As etiquetas RFID podem ser: passivas, semi-passivas e ativas. A etiqueta passiva não tem fonte de energia acoplada e a energia utilizada para a transmissão dos dados é obtida através do sinal enviado pelo leitor RFID. As etiquetas semi-passivas possuem baterias que alimentam o circuito durante a fase de recebimento de dados, sendo que o envio é feito com a energia captada do sinal enviado pelo leitor. Já as etiquetas ativas têm a energia fornecida pela bateria nas fases de recepção e transmissão de sinal, além de poderem iniciar uma comunicação. Outro ponto é que as etiquetas ativas podem ser lidas a distâncias de 100m ou mais. Essas características das etiquetas ativas têm, em contrapartida, um custo final mais alto [Atzori et al. 2010].

Cada etiqueta RFID tem um código único, que faz parte do sistema chamado EPC (*Eletronic Product Code* - Código Eletrônico de Produtos), suportado pela EPCGlobal. Esse código por si só não identifica o objeto ao qual a etiqueta está associada. Contudo, com a associação do EPC com as informações de um banco de dados, é possível saber qual é o objeto referenciado pela etiqueta [GS1-EPCglobal 2009].

WSN (*Wireless Sensor Network*) e WSAN (*Wireless Sensor and Actuator Network*)

Conforme [Xiang e Li 2012], um sensor é um equipamento físico que serve para detectar ou sentir um sinal externo, normalmente, uma característica do ambiente, como por exemplo, luz, temperatura e umidade, e transmitir essa informação para outros dispositivos. Uma Rede de Sensores Sem Fio (RSSF), do inglês *Wireless Sensor Network* (WSN), é uma combinação de sensores, computação embarcada para comunicação sem fio e tecnologia de processamento distribuído.

Redes de sensores são compostas de grandes quantidades de nós de sensoriamento, instalados na área ou próximo a esta, na qual se deseja extrair informações. Em uma RSSF, a transmissão das informações captadas pelos sensores é feita para alguns, normalmente um, dos nós da rede, chamado sorvedouro ou *sink*. A infraestrutura de transmissão das informações pode ser feita por uma rede de múltiplos saltos (cada nó age como um roteador de mensagens) [Atzori et al. 2010].

A posição dos nós de sensoriamento não precisa ser pré-determinada, o que faz com que redes de sensores precisem de protocolos que tenham a capacidade de auto-organizar a rede. Os nós são providos de um pequeno processador, o que permite que estes possam processar dados antes de transmiti-los e não simplesmente transmiti-los [Akyildiz et al. 2002].

Nas redes de sensores e atuadores sem fio, do inglês *Wireless Sensor and Actuator Network* (WSAN), os sensores captam as informações e passam para os atuadores, para

que estes façam o processamento das informações e tomem a decisão de atuar no ambiente (o que pode ocorrer de diversas maneiras) [Martinez et al. 2008].

EnHANTs (*Energy-Harvesting Active Networked Tags*)

EnHANTs (*Energy-Harvesting Active Networked Tags*) são equipamentos que, por serem pequenos, flexíveis e terem independência energética, podem ser acoplados a objetos como roupas e livros, que normalmente não estão interconectados. O desenvolvimento dessa tecnologia é possível devido aos avanços na comunicação em Banda Ultra Larga (*Ultra Wide Band* - UWB), cujo consumo de energia é baixo, e de materiais de captação de energia baseados em semicondutores orgânicos. Considerando complexidade, banda passante, tamanho e requisitos de energia, as etiquetas EnHANTs ficam entre as tecnologias de Redes de Sensores e RFID.

Comparada às etiquetas RFID, as EnHANTs terão uma fonte de energia, poderão se comunicar numa rede de múltiplos saltos e não precisarão depender da energia do sinal de um leitor. Em comparação com sensores, as etiquetas EnHANTs irão operar com taxas de dados menores e consumirão menos energia, transmitindo na maior parte das vezes a sua ID. Essa tecnologia permitirá aplicações além das permitidas pelo RFID. Ao invés de apenas identificar, será possível buscar por um objeto, devido à sua capacidade de operar continuamente e em rede [Gorlatova et al. 2010].

NFC (*Near-Field Communications*)

NFC (*Near-Field Communications*) é uma tecnologia sem fio, de curto alcance, que permite a comunicação entre equipamentos com etiquetas NFC, como por exemplo, placas/pôsteres, celulares, mercadorias, tickets, dentre outros e que pode ser usado para troca de dados entre equipamentos a uma distância máxima de 10cm [Ahson 2012]. NFC opera na frequência de 13,56MHz e consegue transmissões de dados de até 424Kbps. Traz os benefícios do RFID, já que os leitores NFC são compatíveis com etiquetas RFID (podem ler e escrever nas etiquetas). A tecnologia NFC permite a comunicação entre entidades do dia a dia, o que facilita a criação do cenário de IoT. Interações via NFC podem ser habilitadas somente mediante iniciativa e permissão do usuário. Devido à curta distância, a outra ponta de comunicação via NFC é fisicamente conhecida [Cavoukian 2012].

Deve ser ressaltado que há ainda um grande esforço a ser realizado em termos de padronização desta tecnologia, tanto de interfaces como de protocolos. Conforme [Cavoukian 2012], há ainda desafios em termos de segurança e privacidade para que a adoção da tecnologia seja feita de forma mais ampla.

RSN (*RFID Sensor Network*)

Uma rede de sensores de RFID (RSN) consiste de leitores RFID e sensores RFID, que estendem a funcionalidade do RFID para prover sensoriamento. As RSNs combinam as vantagens das etiquetas RFID, tais como: capacidade de identificação, baixo custo, vida

longa, tamanho reduzido, capacidade de serem ativados, com as vantagens das redes de sensores, a saber: capacidade de sensoriamento, comunicação em rede e maior capacidade de processamento [Buettner et al. 2008].

O representante mais forte das RSNs é a tecnologia WISP (*Wireless Identification and Sensing Platforms*[1]) do Intel Labs, que pode ser empregada em cenários que não são adequados para as etiquetas RFID e para as redes de sensores, como em locais nos quais as baterias não podem ser trocadas facilmente ou que precisam de identificação e sensoriamento com custo baixo e possibilitam o uso de leitor móvel. Contudo, algumas limitações desta tecnologia ainda precisam de investigações, tais como, a integração de unidades de sensoriamento em uma rede *mesh* (constituindo uma verdadeira rede de sensores) e a eficiência energética, permitindo que os nós da rede consigam armazenar energia e processar dados de forma eficiente [Buettner et al. 2008].

Smart Gateway

Conforme Mahalle et al. (2010), um dos fatores que torna desafiadora a integração dos objetos (coisas) com a Internet são as restrições de conectividade. Alguns dispositivos não suportam diretamente a conectividade com a Internet, por meio do protocolo IP. Para estes casos, é possível utilizar um dispositivo intermediário entre a Internet e o objeto, chamado de *Smart Gateway*. Esse dispositivo fornece uma interface de comunicação dos objetos com sistemas finais na Internet e vice-versa. Isso permite que sistemas finais na Internet, que podem ser outros objetos, se comuniquem com os objetos através desse *Smart Gateway*, sendo que este recebe as mensagens vindas da Internet e repassa ao objeto por meio de sua API (*Application Programming Interface*) de comunicação específica, e vice-versa.

Em outras palavras, o *Smart Gateway* atua como uma ponte entre a Internet e os dispositivos inteligentes. Este tem um endereço IP e compreende os diferentes protocolos dos dispositivos conectados a este através do uso de controladores (*drivers*) dedicados. Assim, requisições vindas da Internet que visam acessar algum dispositivo irão passar pelo *Smart Gateway*, que irá redirecionar a requisição através do protocolo específico.

Padronização

A padronização é um aspecto importante para a concretização da IoT. Observa-se uma intensa colaboração entre as entidades de padronização, Grupos de Interesse e Alianças de fabricantes das tecnologias envolvidas, o que demonstra um grande interesse da indústria [Atzori et al. 2010].

Dentre os padrões existentes, destaca-se o padrão IEEE 802.15.4 que define as LR-WPAN (*Low-Rate Wireless Personal Area Networks*), que são redes sem fio projetadas para ter um baixo custo, baixo consumo de energia e pequeno alcance. Esse padrão define apenas as camadas física e de enlace do modelo OSI, sendo que as camadas superiores não são especificadas [Atzori et al. 2010] [Baronti et al. 2007].

[1] https://wisp.wikispaces.com/

A *Zigbee Alliance* é constituída por um grupo de empresas que padroniza e mantém as especificações do **Zigbee** que define as camadas superiores do modelo OSI para serem usadas sobre a especificação IEEE 802.15.4. A camada de rede do Zigbee é encarregada de fazer a organização e o roteamento sobre uma rede de múltiplos saltos, já a camada de aplicação provê um *framework* para o desenvolvimento de aplicações distribuídas [Baronti et al. 2007].

Com a intenção de integrar os dispositivos que estão em conformidade com o padrão IEEE 802.15.4 em uma rede IP e, assim, na Internet, foi criado pela IETF o grupo de trabalho de integração do IPv6 às redes IEEE 802.15.4, o **6LoWPAN** (*IPv6 over Low power Wireless Personal Area Networks*) [IETF 2007].

Com a intenção de difundir o protocolo IP como padrão de comunicação entre objetos, foi criada a *IPSO Alliance*, uma organização sem fins lucrativos com mais de sessenta companhias membros, dentre estas líderes mundiais dos mercados de TI, comunicação e energia. Dentre os objetivos da organização, estão: apoiar as entidades de padronização, como a IETF (Internet Engineering Task Force), no desenvolvimento de padrões que envolvem o protocolo IP e objetos inteligentes, assim como promover testes de interoperabilidade e auxiliar as indústrias na descoberta de novos mercados que envolvem a integração do IP com objetos inteligentes [Alliance 2013].

4.2.2. Web das Coisas - *Web of Things* (WoT)

Há uma tendência nas pesquisas atuais em tratar a Internet das Coisas como Web das Coisas, nos quais os padrões abertos da Web são empregados para prover o compartilhamento de informação e a interoperabilidade entre dispositivos [Zeng et al. 2011]. Segundo estes autores, alguns motivos favorecem a WoT, dentre estes, destacam-se:

- A Web se tornou o principal meio de comunicação na Internet;
- Diversos pequenos servidores web embarcados estão disponíveis. Estes podem ser construídos em apenas alguns KBytes;
- Navegadores Web estão disponíveis para quase todos as plataformas, de computadores a smart phones e tablets, o que os tornam a interface de usuário padrão de fato para uma gama de aplicações;
- A tecnologia integradora dos serviços web tem se mostrado indispensável na criação de aplicações distribuídas interoperáveis para Internet;
- Coisas inteligentes (*smart things*), com servidores web incorporados, podem ser abstraídas como serviços web e perfeitamente integradas na web existente;
- É natural a reutilização de tecnologias e padrões web existentes para unificar o mundo cibernético ao mundo das coisas físicas
- Como as tecnologias web existentes podem ser reutilizadas e adaptadas, é possível construir novas aplicações e serviços com a participação das coisas inteligentes.

Em suma, diferente do ponto de vista tradicional da IoT, que associa ao dispositivo um endereço IP e os torna interconectados na Internet, a WoT habilita os dispositivos a "conversarem"na mesma língua, de modo que estes possam se comunicar e interagir livremente na Internet. A interoperabilidade é particularmente essencial para concepção de sistemas com dispositivos heterogêneos produzidos por diferentes fabricantes. Na WoT, a integração dos dispositivos ocorre no nível de aplicação, acima da conectividade de rede [Zeng et al. 2011, Guinard et al. 2011].

Há dois métodos para integrar coisas à Web: integração direta e integração indireta. Na direta, via *smart things*, é requerido que todas as coisas tenham um endereço IP ou que tenham um IP habilitado quando conectados à Internet. Servidores web devem ser embarcados nas coisas/dispositivos para que estas possam se entender por meio da linguagem da Web. Já na integração indireta, nem todos os dispositivos podem ter recursos computacionais suficientes para embarcar um servidor web, como por exemplo uma etiqueta RFID. Além disso, algumas vezes não é necessário integrar diretamente todas as coisas inteligentes dentro da Web, quando se considera custo, energia e a segurança. Como solução para integração indireta, pode-se usar de um proxy, chamado de *smart gateway*, localizado entre os dispositivos inteligentes e a Web (ver Figura 4.3).

Os serviços Web são definidos pela W3C com um sistema de software projeto para suportar a comunicação máquina para máquina (do inglês, *machine-to-machine* (M2M) interoperável sobre uma rede. Como a W3C atesta, existem dois grandes paradigmas de serviços web: serviços Web em conformidade com o REST [Fielding e Taylor 2002] (chamados de serviços web *RESTful*) e os serviços web arbitrários (conhecidos como WS*) [Group 2004]. O objetivo principal dos serviços web *RESTful* é manipular os recursos da web usando um conjunto uniforme de operações sem estado. No segundo, usa-se um conjunto arbitrário de operações. Ambos os paradigmas podem ser adotados por *smart things* ou *smart gateways*.

As tecnologias chaves dos WS* são: o protocolo SOAP, a linguagem de descrição de serviços web (*Web Service Description Language*- WSDL), o *Universal Description Discovery and Integration*(UDDI) e a *Business Process Execution Language* (BPEL).

Conforme definido em [Fielding e Taylor 2002], o REST (*Representational State Transfer*) é um estilo de arquitetura de software, desenvolvido como um modelo abstrato da arquitetura web, que que pode ser aplicado no desenvolvimento de sistemas distribuídos fracamente acoplados, denominados *RESTful*. O conceito básico do REST é que qualquer coisa é modelada como recurso, ou particularmente como recursos HTTP, com uma URI (*Uniform Resource Identifier*). Os sistemas *RESTful* são menos acoplados, mais leves, eficientes (menos complexos) e flexíveis do que os sistemas baseados em Serviços Web arbitrários (WS*) que utilizam o protocolo SOAP. Essas características fazem do REST a opção mais adequada para ser embarcada em dispositivos com restrição de recursos e para permitir a fácil composição de serviços web (i.e., *mashup*) [Guinard e Trifa 2009, Zeng et al. 2011]. A WoT facilita a criação de *mashups* físicos (objetos físicos). Um *mashup* Web é uma aplicação que utiliza diversos recursos web e os usa para criar outra aplicação

Conforme ilustrado na parte direita da Figura 4.3, um dispositivo inteligente com um servidor web embarcado é capaz de tornar diretamente acessível na web a sua API RESTful para que outros dispositivos ou usuários tenham acesso aos seus recursos. Quando

um servidor web não é possível ou desejado, pode-se utilizar um *smart gateway* para interconectar um dispositivo não acessível diretamente como um recurso Web. Um Smart Gateway é um servidor Web que esconde a comunicação entre os dispositivos de rede (por exemplo, Bluetooth e Zigbee) e os clientes, através de controladores dedicados atrás de um serviço *RESTful*.

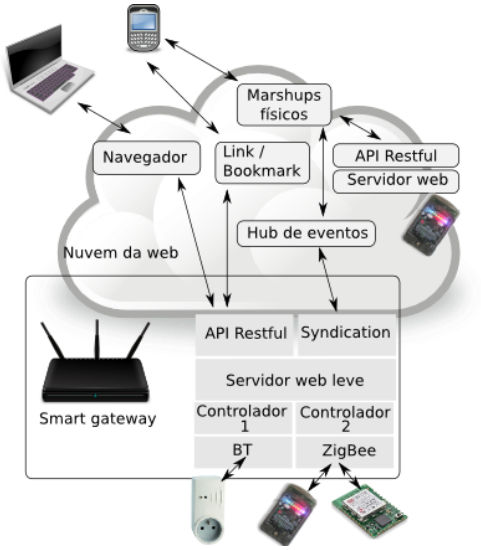

Figura 4.3: Integração Direta e Indireta na Web das Coisas

4.2.3. Domínios de Aplicações na Internet das Coisas

As aplicações de IoT podem ser divididas em três grandes áreas [Xiang e Li 2012]:

- Social: aplicações que envolvem o desenvolvimento e cuidados social, urbano e humano, como por exemplo, serviços públicos do Governo e saúde eletrônica (*e-health*) para cidadãos;
- Ambiental: aplicações que envolvem a proteção, monitoramento e desenvolvimento de recursos ambientais, por exemplo, aplicações na pecuária, agricultura, reciclagem de recursos, gestão de energia, dentre outros.; e
- Industrial: aplicações financeiras e comerciais entre companhias, organizações e outras entidades, por exemplo, aplicações para pecuária, agricultura, reciclagem de recursos, gestão de energia, dentre outros.

[Atzori et al. 2010] agrupam os diferentes tipos de aplicações possíveis para IoT em quatro domínios distintos dos anteriores. São estes:

- Transporte e logística: aplicações que envolvem meios de transporte de pessoas e de mercadorias, assim como aplicações para rodovias;
- Cuidados com a saúde: aplicações para auxílio de cuidados com a saúde das pessoas;

- Ambientes inteligentes: aplicações que tornam escritórios, casas, plantas industriais e ambientes de lazer inteligentes; e
- Pessoal e social: aplicações que habilitam o usuário a interagir com outras pessoas para manter e criar relacionamentos.

4.3. Requisitos e Ameaças de Segurança na Internet das Coisas

De acordo com [Roman et al. 2011b], a segurança é identificada como um dos obstáculos a serem transpostos para o efetivo uso da Internet das Coisas. Ao prover segurança às aplicações de IoT, por meio de uma infraestrutura de autenticação e de autorização, é preciso garantir o comportamento autônomo dos objetos e a interoperabilidade entre estes.

4.3.1. Requisitos de Segurança na IoT

Tendo em vista as características da IoT, [Alam et al. 2011, Roman et al. 2011b] apontam diversos requisitos de segurança para Internet das Coisas e indicam quais **propriedades de segurança** devem ser garantidas, sendo estas:

- Confidencialidade: dados sensíveis de usuários ou organizações podem estar contidos nas transações na Internet das Coisas e, portanto, a confidencialidade de tais dados deve ser assegurada;
- Integridade: dados armazenados e transmitidos não devem ser alterados, removidos ou incluídos por usuários ou dispositivos não autorizados;
- Disponibilidade: manter os serviços/recursos da Internet das Coisas disponíveis para acesso por usuários e dispositivos autorizados em qualquer momento e a partir de qualquer lugar, provendo assim o acesso a dados de forma contínua;
- Autenticidade: necessidade de autenticação mútua, pois os dados de IoT são usados para diferentes processos de tomada de decisão e atuação, sendo que é necessário que tanto o consumidor de recursos/serviços como o provedor sejam autenticados; e
- Privacidade: se refere a necessidade de prover aos usuários meios para que estes controlem a exposição e a disponibilidade dos seus próprios dados e informações e tenham maior transparência sobre como e por quem seus dados são usados.

[Babar et al. 2010] apontam alguns outros requisitos de segurança que precisam ser garantidos em IoT, dentre estes:

- Gestão de Identidades: lida com a identificação e autenticação dos usuários e dos dispositivos/coisas em um sistema. Também controla acessos aos recursos deste sistema associando direitos e restrições de acesso, de acordo com a identidade estabelecida (autenticação e autorização);
- Comunicação segura de dados: inclui a autenticação dos pares da comunicação, assegurando a confidencialidade e integridade dos dados transmitidos, impedindo o repúdio de uma transação e protegendo a identidade das entidades;

- Acesso seguro à rede: garante a possibilidade de conexão de rede ou o acesso a um serviço apenas para dispositivos autorizados; e

- Resistência à violação: mantem os aspectos de segurança, mesmo quando o dispositivo for acessado fisicamente por um atacante.

Segundo [Mahalle et al. 2013b], a grande escala e escopo da IoT aumentam as opções de interação dos usuários com os sistemas, levando a necessidade de estender os modelos atuais de privacidade, segurança e gestão de identidades para incluir a forma como os usuários interagem com os objetos. Neste sentido, também são levantados os requisitos de que deve ser possível identificar os objetos de maneira única, ou seja, diferenciar um objeto do outro, além de permitir a autenticação única de objetos na IoT [Mahalle et al. 2013a].

Por fim, [Xiaohui 2012] e [Roman et al. 2011b] destacam o requisito da tolerância a faltas, que nos cenários em geral, se refere ao sistema não falhar e funcionar normalmente, mesmo diante da presença de uma falta. Na Internet das Coisas, a tolerância a faltas consiste no sistema recuperar a transmissão de dados e reparar a estrutura da rede (p. ex. a sua topologia) de forma autônoma, mesmo diante de faltas em nós ou nos enlaces da rede.

Na Figura 4.4 são ilustrados os principais requisitos de segurança para a Internet das Coisas.

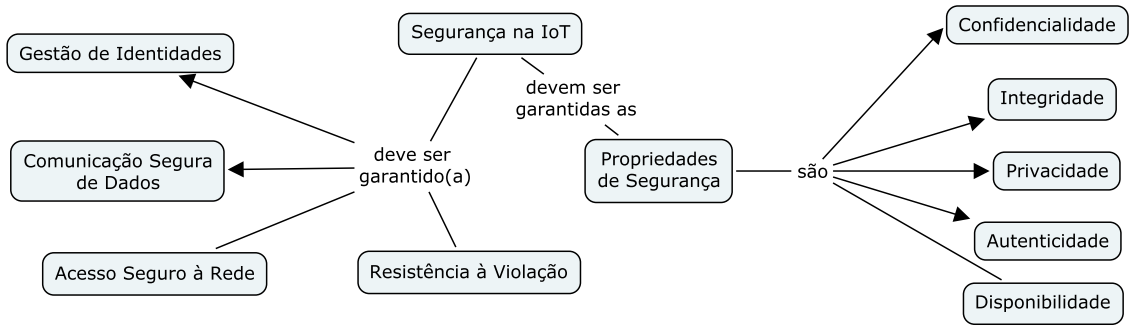

Figura 4.4: Mapa Conceitual com os Principais Requisitos para a Internet das Coisas

4.3.2. Ameaças e Ataques na IoT

Em [Akram e Hoffmann 2008a], os autores afirmam que a IoT possibilita que sistemas computacionais se tornem ubíquos e transparentes para os usuários. Essa transparência, juntamente com a onipresença, são potenciais ameaças para a privacidade dos usuários, bem como impõe dificuldades para garantir a confidencialidade e a integridade dos dados que ali trafegam. O compartilhamento de dispositivos com outras pessoas é uma das principais ameaças de segurança contra a privacidade dos usuários, pois os dados podem ser facilmente obtidos por pessoas não autorizados, uma vez que esta pessoa bastaria ter acesso físico ao dispositivo [Jindou et al. 2012].

Em [Liu et al. 2012], afirma-se que antes da existência da IoT, sistemas digitais corrompidos eram em sua maioria incapazes de atuar no mundo físico, porém no cenário

da IoT, dispositivos corrompidos podem atuar e influenciar o mundo físico diretamente. Por exemplo, um dispositivo que possua sensor de fumaça deve avisar uma central de controle sempre que detectar fumaça no ambiente. Se este dispositivo for corrompido, poderá emitir alertas falsos ou mesmo pode deixar de emitir alertas diante de uma real situação de perigo com fumaça.

No cenário da Internet das Coisas, quando um nó envia dados para um outro nó da rede ou mesmo para um nó acessível através da Internet, esses dados podem ser armazenados temporariamente nos nós intermediários que atuam como roteadores. Assim, entre a origem e o destino de uma determinada informação, podem existir diversos nós intermediários que, se forem maliciosos, poderão alterar a informação em trânsito ou ainda não encaminhar a informação para o destino final [Conzon et al. 2012].

[Babar et al. 2011] apresentam uma divisão dos tipos de ataques na Internet das Coisas em cinco categorias, relacionadas a seguir:

- **Ataques Físicos**: são ataques que violam o hardware do dispositivo e são difíceis de executar, pois o material necessário para executar os ataques é caro. *De-packaging* de um chip, *micro-probing* e *layout reconstruction* são técnicas usadas para esse tipo de ataque;
- **Ataques no canal de comunicação**: ataques baseados em dados recuperados dos dispositivos responsáveis por operações criptográficas. Esses dados são obtidos através de análise de temporização, radiação emitida, potência consumida, dentre outras fontes, que permitem que a chave de criptografia usada seja inferida;
- **Ataques de análise de criptografia**: ataques com foco no texto cifrado, buscando encontrar a chave de criptografia para assim obter o texto em claro. Um dos ataques dessa categoria é o ataque do Homem do Meio (*Man in the Middle* - MITM);
- **Ataques de software**: exploram vulnerabilidades dos softwares presentes no dispositivo. Inclui ataques de exploração de estouro do buffer (*buffer overflow*) e uso de programas cavalos de tróia, *worms* e vírus para injetar código malicioso no sistema;
- **Ataques de rede**: no meio sem fio a transmissão é por difusão (*broadcast*) e assim há vulnerabilidades inerentes ao próprio meio. Nessa categoria entram ataques como captura e análise de tráfego (*eavesdropping*), negação de serviço (*Denial of Service* – DoS), corrupção de mensagens, ataques de roteamento, dentre outros.

[Bonetto et al. 2012] destacam que as redes sem fio, como as utilizadas na IoT, são propensas a diversos tipos de ataques, tais como: **captura de informações** (*eavesdroping*), que viola a propriedade da confidencialidade; **mascaramento**, no qual um nó se faz passar por outro, ferindo assim a propriedade da autenticidade; e ainda a **negação de serviço**, que viola a propriedade de disponibilidade. Sobre a negação de serviço, [Mahalle et al. 2012] citam a topologia dinâmica da rede, menor largura de banda e restrições de energia como vulnerabilidades que propiciam este tipo de ataque.

Em [Mahalle et al. 2013a], são descritas preocupações de segurança relacionadas com o ingresso dos dispositivos na rede. No momento do ingresso na rede, informações

sobre chaves criptográficas, parâmetros de domínio e outras configurações podem ser capturadas por entidades maliciosas e estas poderiam fazer uso dessas informações para interceptar e reencaminhar dados de forma a não ser percebido, caracterizando um ataque de ***man in the middle***. Além disso, caso o protocolo de estabelecimento de chaves seja comprometido, não só a confidencialidade da comunicação será comprometida, mas também a autenticidade dos nós participantes pode estar em risco, já que muitas vezes os nós comunicantes não tem conhecimento prévio um do outro. Segundo os autores, é possível realizar o ataque de esgotamento de recursos (DoS), uma vez que que neste ambiente os recursos computacionais e de energia são limitados.

[Mahalle et al. 2012] apontam que o ataque de *man in the middle*, pode levar ao ataque de mensagem antiga, no qual o atacante busca utilizar de mensagens antigas (interceptadas) para se comunicar com outros dispositivos, a fim de obter respostas desses dispositivos que inicialmente não seriam para ele, mas sim para o remetente da mensagem original.

[Jara et al. 2011] indicam a possibilidade de corromper mensagens de identificação ou de localização em arquiteturas de IoT que fazem uso do protocolo 6LoWPAN [Montenegro et al. 2007]. Corromper tais mensagens levaria a falhas na segurança da rede, uma vez que um intruso poderia enviar mensagens falsas de atualização sobre a localização de um nó, fazendo com que mensagens não chegassem ao seu destino ou fossem enviadas para o nó malicioso. Isso ainda permitiria a ocorrência de ataques de negação de serviço através de envio em massa de mensagens (*flood*).

[Nguyen et al. 2010] abordam dois outros tipos de ataques: chave compartilhada e *sybil*. No **ataque de chave compartilhada** (*shared-key attack*), o atacante conhece o mecanismo de distribuição de chaves do ambiente e, sabendo que dois nós estão próximos, supõe que estes compartilhem um mesmo espaço de chaves. O ataque ocorre quando a chave compartilhada pelos dispositivos também pode ser inferida pelo atacante, comprometendo a segurança do sistema. O **ataque *sybil*** é caracterizado quando um nó malicioso assume múltiplas identidades falsas com o objetivo de roubar ou forjar a identidade de um nó legítimo.

Por fim, [Liu et al. 2012] ressaltam também a existência do **ataque de controle de chaves** (*key control attack*), no qual uma dos participantes da comunicação força os demais participantes a escolherem chaves criptográficas dentro de um conjunto restrito de valores ou mesmo um valor pré-determinado. Desta forma, o atacante influencia o processo de escolha de chaves criptográficas de modo a facilitar a obtenção do controle sobre os dados trafegados.

4.4. Autenticação e Autorização na Internet das Coisas

Autenticação e Autorização (controle de acesso) são conhecidos como elementos centrais para tratar a segurança em sistemas distribuídos. Uma forma para prover estes controles é através de uma infraestrutura de autenticação e de autorização (IAA) que provê a gestão de identidades (*Identity Management - IdM*). IdM pode ser entendida como o conjunto de processos e tecnologias usados para garantir a identidade de uma entidade (usuário ou um dispositivo), garantir a qualidade das informações de uma identidade (identificadores, credenciais e atributos) e para prover procedimentos de autenticação e de autorização [ITU

2009]. As entidades envolvidas em um sistema de IdM são: (i) usuário ou dispositivo, entidade que utiliza um serviço fornecido por um provedor de serviços; (ii) provedor de identidades (*Identity Provider - IdP*), responsável por manter a base de dados de usuários do domínio e validar suas credenciais (autenticar usuários); e (iii) provedor de serviços (*Service Provider - SP*), que oferece recursos ou serviços aos usuários [Wangham et al. 2010].

Na IoT, os dispositivos podem pertencer a mais de uma rede ou domínio administrativo, cenário que [Horrow e Sardana 2012] chamam de *Internetwork of Things*. Esta situação pode afetar o funcionamento dos procedimentos de autenticação e de autorização, em função da mobilidade destes dispositivos entre redes diferentes.

Esta seção aborda os conceitos e modelos de autenticação e de autorização para Internet das Coisas, considerando as particularidades deste cenário. Na literatura, a autenticação na IoT é tratada de forma diferente para usuários e para dispositivos. Nas seções a seguir, os conceitos e técnicas apresentados também seguem essa distinção.

4.4.1. Autenticação de Usuários

Na Internet das Coisas, usuários interagem com muitos dispositivos inteligentes ou provedores de serviços (*Service Providers - SPs*) para obter algum serviço útil para eles. Para que um usuário acesse um objeto/dispositivo na IoT, muitas vezes, é necessário que este passe por um processo de autenticação.

Alguns trabalhos na literatura seguem o modelo de autenticação centralizada, baseada em uma terceira parte confiável. [Li et al. 2010] propõem o uso do LDAP (*Lightweight Directory Access Protocol*) em conjunto com o mecanismo de autenticação Kerberos, para prover autenticação única (Single Sign On) de usuários na IoT.

Em [Konidala et al. 2005], para que um usuário acesse um provedor de serviço, este precisa apresentar um *token* de acesso assinado (emitido) por uma terceira parte confiável (*Authorized Server - AS*) tanto para o usuário, quanto para o SP. Cada usuário precisa fazer um cadastro inicial neste servidor central (AS), o qual deve fornecer um identificador único e senha. Para obter o *token* de acesso (*capability*), o usuário precisa se autenticar no AS, fazendo uso de sua senha. O SP verifica a assinatura do token e analisa o conteúdo do mesmo para concluir o processo de autenticação do usuário.

[Rotondi et al. 2011] fazem uso de criptografia de chave pública como técnica de autenticação de usuários na IoT. Neste trabalho, as requisições de acesso são assinadas digitalmente com as chaves privadas correspondentes às chaves públicas presentes nos certificados. Dessa forma, é possível garantir a autenticidade dos usuários diante dos dispositivos. Este trabalho encontra-se descrito na Seção 4.6.1.5.

Devido à característica da IoT, na qual nem sempre usuários e dispositivos estão em um mesmo domínio de rede, uma abordagem de autenticação centralizada, por exemplo, que faz uso de um centro de distribuição de chaves (*Key Distribution Center* – KDC), pode não ser indicada para realizar a autenticação do usuário [Liu et al. 2012]. Uma abordagem centralizada pode ser empregada apenas se um único e amplamente aceito provedor de identidades ou KDC estiver disponível.

Um modelo de gestão de identidades mais adequado ao cenário de IoT é o modelo

baseado em identidades federadas [Akram e Hoffmann 2008c, Liu et al. 2012]. Neste modelo, o usuário se autentica no provedor de identidades do seu domínio e este provedor fornece as afirmações necessárias para que os provedores de serviços de outros domínios da IoT confiem na autenticação realizada e recebam o atributos do usuário. Em [Liu et al. 2012], os autores propõem um mecanismo para autenticação de usuários que segue o modelo de identidades federadas, conforme apresentado na seção 4.6.1.7. [Akram e Hoffmann 2008c] destacam o uso do OpenID como uma das soluções para gestão de identidades federadas do *Hydra Middleware* (maiores detalhes na Seção 4.6.2.1).

4.4.2. Autenticação de Dispositivos

[Mahalle et al. 2012] propõem um método de autenticação mútua para IoT focado em dispositivos que estejam em um único domínio. Um dispositivo, ao ingressar na rede, recebe um par de chaves assimétricas e um parâmetro de domínio de um centro de distribuição de chaves confiável (*Key Distribution Center* – KDC). Esse parâmetro de domínio foi utilizado pelo KDC no processo de geração do par de chaves entregue ao dispositivo, com base em um protocolo de criptografia de curvas elípticas (*Elliptic Curve Cryptography* – ECC). Assim, quando dois dispositivos desejam se comunicar dentro do domínio de um mesmo KDC, eles usam o protocolo ECCDH para o estabelecimento de uma chave privada, que será utilizada para a comunicação entre eles. A base para o estabelecimento dessa chave é o parâmetro de domínio e a chave pública de cada um dos dispositivos.

Após o estabelecimento dessa chave privada, ocorre o processo de autenticação entre os dispositivos. Esse processo se dá através de protocolo de desafio resposta, que utiliza como base a chave privada estabelecida, um *timestamp* e um número aleatório gerado por uma das partes da comunicação. Também são utilizados no processo de autenticação a habilidade apresentada por cada um dos dispositivos. A habilidade é um *token* que contém a identidade do dispositivo, um conjunto de direitos de acesso e um *hash* dos dois campos anteriores. Esse *hash* é aplicado com o método CBC MAC, visando garantir a integridade das mensagens. Por fim, o dispositivo que será acessado verifica se o *token* de habilidade enviado pelo outro dispositivo é igual ao que este tem armazenado. Se sim, e se o resultado do desafio-resposta for correto, o processo de autenticação mútua está finalizado e foi bem-sucedido.

[Kothmayr et al. 2012] apresentam uma arquitetura de segurança para IoT baseada no *Datagram Transport Layer Security* (DTLS) [Rescorla e Modadugu 2012] e fazem uso de certificados digitais no processo de autenticação de dispositivos. Nesta arquitetura, são propostos três atores: *publisher* – dispositivo produtor; *subscriber* – dispositivo consumidor de recursos; e servidor de controle de acesso – equipamento com maior poder computacional e responsável por aplicar o controle de acesso aos recursos dos dispositivos produtores.

São apresentados dois cenários, um no qual os dispositivos produtores possuem *Trusted Platform Modules* (TPMs) e outro no qual os dispositivos não possuem TPMs. Para o primeiro cenário, o produtor está apto a fazer *handshake* completo do DTLS com o consumidor do recurso, sendo a autenticação mútua realizada através de certificados X.509, emitidos por uma Autoridade Certificadora (AC), reconhecida por ambos.

Para o segundo cenário (dispositivos com restrições), a autenticação do produtor é feita através do uso de uma chave compartilhada (*Pre-Shared Key* – PSK) da suíte de cifragem do TLS [Eronen e Tschofenig 2005]. Neste cenário, o dispositivo possui um conjunto de dados aleatórios pré-instalados, chamados *protokeys*, que são usados para gerar a PSK de uma sessão. No processo de autenticação, o produtor gera uma identidade de sessão, composta por sua identidade[2] e alguns dados gerados aleatoriamente no momento da autenticação. Em seguida, uma PSK é gerada por meio da aplicação de uma função HMAC sobre essa identidade de sessão, tendo como chave neste processo a *protokey*.

O consumidor do recurso, por sua vez, deve se autenticar no servidor de controle de acesso, o qual tem conhecimento das *protokeys* e da identidade de sessão do produtor. Assim, o servidor de controle de acesso é capaz de gerar a PSK do dispositivo produtor para essa sessão e de repassá-la ao consumidor. Dessa maneira, o servidor de controle de acesso valida a identidade do consumidor para o produtor, além de validar a identidade (de sessão) do produtor para o consumidor. Sendo assim, o servidor de controle de acesso precisa ser uma terceira parte confiável nesta arquitetura.

De acordo com [Hummen et al. 2013], as restrições dos objetos inteligentes da IoT demanda por mecanismos de segurança mais leves. O uso de certificados digitais para autenticação de dispositivos é, em muitos casos, considerado impraticável. Neste trabalho, os autores tiveram como objetivo comprovar que, com algumas modificações no processo de *handshake* do protocolo do DTLS, o uso de certificados digitais torna-se um método viável de autenticação em muitos cenários da IoT.

Os autores descrevem três modificações no processo de *handshake* do protocolo do DTLS, visando sua otimização em função das restrições computacionais dos dispositivos da IoT. A primeira modificação é voltada para ambientes em que o dispositivo se comunica com um objeto ou serviço fora de seu domínio, passando por um *gateway*. Neste caso, durante o *handshake* do DTLS, o *gateway* verifica se o certificado não foi expirado ou revogado, bem como valida a cadeia de certificação em nome do dispositivo, passando para este somente as mensagens de *handshakes* DTLS que utilizarem certificados válidos.

Outra forma de otimização em função das restrições computacionais é a retomada de sessão DTLS, na qual as operações criptográficas mais custosas e as verificações de certificados são realizadas apenas uma vez em um *handshake* inicial. Para que isso seja possível, é necessário que o cliente, o servidor ou ambos mantenham informações sobre a sessão DTLS que foi estabelecida mesmo depois que esta seja desfeita. Isso permite que a sessão DTLS seja reestabelecida futuramente em um tempo menor e com menor custo computacional [Hummen et al. 2013].

A terceira modificação sugerida por [Hummen et al. 2013] consiste na delegação do primeiro *handshake* para o dono do dispositivo. Desta forma, quando um cliente desejar acessar o dispositivo, o primeiro *handshake* DTLS, que inclui operações criptográficas custosas e verificações de certificados, é feito entre o dono do dispositivo e o cliente. Em seguida, o dono do dispositivo finaliza a sessão DTLS com o cliente e passa as informações de retomada de sessão para o dispositivo que, possuindo esses dados, consegue reestabelecer a sessão com o cliente como se tivesse sido o próprio dispositivo a estabelecer

[2]Endereço IPv6 ou outra informação que o dispositivo.

a primeira sessão DTLS. Esse processo é chamado de transferência de sessão DTLS.

Em [Jara et al. 2011] é descrito um esquema de gerenciamento de mobilidade segura para dispositivos que fazem uso do protocolo 6LoWPAN [Montenegro et al. 2007] e que atravessam diferentes domínios administrativos. Os autores propõem uma solução para prover a autenticação entre domínios para os dispositivos móveis. Cada domínio administrativo contém um *gateway* que é responsável por manter informação sobre a localização dos dispositivos que ingressam em seu domínio e remover tal informação quando estes saem do domínio. Assim, quando um dispositivo troca de domínio, este envia uma mensagem de ingresso para o *gateway* do novo domínio, denominado *Foreign Gateway* (F-GW), para que este atualize as informações de localização do dispositivo, acionando o *gateway* do domínio anterior, denominado *Home Gateway* (H-GW).

Para evitar que a mensagem enviada pelos dispositivos para troca de domínio seja forjada, [Jara et al. 2011] propõem o uso de uma chave compartilhada entre o dispositivo e seu H-GW. Essa chave é usada para assinar todas as mensagens sobre atualização da localização, trocadas entre dispositivo e o H-GW. Para assinar tais mensagens, os autores propõe o uso de do HMAC ou o CBC MAC, já que estes são adequados para ambientes com restrições de recursos, ou ainda um método de assinatura digital baseado em criptografia de curvas elípticas (ECC), proposto pelos autores.

[Bonetto et al. 2012] propõem um método leve que permite a proteção de dispositivos de Internet das coisas através de criptografia forte e técnicas de autenticação, de modo que os dispositivos com restrição da IoT podem se beneficiar das mesmas funcionalidades de segurança que são típicos de domínios sem restrições (Internet), sem, contudo, ter que executar operações, computacionalmente intensivas para estes dispositivos. Para tornar isso possível, os autores propõem o uso de um nó confiável sem restrição (*gateway*) para assumir as tarefas intensivas de computação. A solução proposta faz uso do protocolo *Extensible Authentication Protocol* (EAP) [Aboba et al. 2004], para realização da autenticação de dispositivos. O *gateway* faz um papel ativo intermediando o processo de autenticação, viabilizando assim o uso do protocolo EAP no cenário da IoT.

O protocolo *Host Identity Protocol* (HIP) [Moskowitz et al. 2008], concebido para autenticação de *hosts*, tem recebido diversas variantes para operar com dispositivos que possuem restrições computacionais, tais como LHIP [Heer 2006], DEX [Moskowitz 2012], TEX [Saied e Olivereau 2012b] e D-HIP [Saied e Olivereau 2012a], que podem ser utilizados para autenticação de dispositivos na IoT.

4.4.3. Autorização

Mecanismos de controle de acesso são necessários para garantir que recursos estejam disponíveis somente para sujeitos autorizados pela política de controle de acesso. Um sujeito pode ser um processo, uma pessoa ou um dispositivo que deseja executar alguma ação sobre um recurso [Hu e Scarfone 2012]. Na IoT, a implementação deste tipo de mecanismo deve levar em consideração a dinamicidade do ambiente, com grande número de dispositivos e usuários, bem como a presença de dispositivos com recursos computacionais restritos [Liu et al. 2012, Rotondi et al. 2011].

No contexto da IoT, dentre os trabalhos analisados, observa-se que os mecanismos

de controle de acesso implantam modelos conhecidos e já empregados na Internet clássica, a saber:

- **Modelo discricionário**: por exemplo, [Guinard et al. 2010] propõem um mecanismo baseado em lista de controle de acesso (*Access Control List – ACL*) para possibilitar que pessoas compartilhem seus dispositivos da WoT com outros usuários por meio das redes sociais existentes. O dono do dispositivo necessita configurar as permissões para cada dispositivo e para cada usuário com quem queira compartilhá-lo. Uma abordagem utilizando ACLs é custosa para um usuário manter quando este possui muitos dispositivos e muitos usuários com quem deseje compartilhar;

- **Modelo baseado em papéis** (*Role Based Access Control* – RBAC): [Liu et al. 2012, Jindou et al. 2012, De Souza et al. 2008] adotam o modelo RBAC que é amplamente aceito na Internet e conhecido por sua simplicidade para gerenciar permissões e usuários. Porém, [Mahalle et al. 2013a] apontam que o RBAC possui granularidade limitada e a forma com que este lida com a delegação de direitos não é adequada para ambientes de larga escala, como o ambiente da IoT;

- **Modelo baseado em habilidades** (*Capability Based Access Control – CapBAC*): o detentor da habilidade (*token* de autorização) é capaz de interagir com um objeto por meio de operações bem definidas. A informação sobre a identidade do usuário ou dispositivo é transformada em uma habilidade, que ainda combina os direitos de acesso deste usuário/dispositivo. Esse modelo oferece boa escalabilidade, uma vez que não há a necessidade de confrontar a identidade do usuário com uma lista de controle de acesso, ou com listas de papéis e de permissões. Neste modelo, tem-se um número menor de informações armazenadas na entidade responsável por aplicar o controle de acesso. [Rotondi et al. 2011, Mahalle et al. 2012, Mahalle et al. 2013a] seguem este modelo em suas infraestruturas de autorização;

- **Modelo baseado em atributos** (*Attribute Based Access Control* – ABAC): a decisão de autorização é tomada a partir de um conjunto de atributos do sujeito, do objeto, das operações requisitadas e das condições do contexto frente às políticas de controle de acesso, regras ou relações que descrevam as operações permitidas para um determinado conjunto de atributos [Hu et al. 2013]. [Han e Li 2012] fazem uso do modelo ABAC na IoT, adaptando-o para tratar da delegação de atributos neste cenário. Segundo os autores, é possível perceber alguns benefícios do uso do ABAC em cenários como IoT, quando o sujeito faz o acesso a um objeto fora de seu domínio administrativo. Nesse caso, as listas de controle de acesso (*Access Control List* – ACL) ou os papéis do RBAC não são aplicáveis, pois estes estão fortemente ligados ao contexto do detentor do recurso. [Zhang e Liu 2011] também adaptam o modelo ABAC para o cenário de IoT, combinando uma abordagem orientada a *workflow* (WABAC). Neste modelo, para que seja tomada uma decisão de controle de acesso, são considerados os atributos de três atores: (i) o sujeito, aquele que deseja realizar uma ação sobre um recurso, que pode ser um usuário, uma aplicação ou um telefone móvel, tendo atributos como um identificador, um endereço IP ou endereço de e-mail, etc; (ii) o recurso, que pode ser, por exemplo, um serviço, um dado ou um dispositivo inteligente, tendo atributos como localização geográfica, identificador ou

data de criação, etc; e (iii) o ambiente, que se refere ao contexto em que o acesso à informação acontece, tendo atributos como a data ou o nível de segurança da rede.

4.5. Infraestruturas de Autenticação e de Autorização Aplicadas à IoT

Em [Wangham et al. 2010, Nogueira et al. 2011, Feliciano et al. 2011, Silva et al. 2013] foram descritos os principais padrões e soluções para prover gestão de identidade para a Internet clássica, para Internet do Futuro, para Nuvens e para Redes Experimentais, respectivamente. Esta seção apresenta os principais padrões e soluções que estão sendo empregados no contexto da Internet das Coisas.

4.5.1. Especificações de Segurança para Serviços Web

A *Security Assertion Markup Language* (SAML) [OASIS 2008], baseada na linguagem XML, define sintaxe e regras para criação, requisição e transporte de informações sobre autenticação, autorização e atributos através de asserções de segurança . A *eXtensible Access Control Markup Language* (XACML) [OASIS 2003] tem por objetivo descrever políticas de controle de acesso em um formato interoperável. Na especificação da XACML, também é descrito um protocolo para realizar requisições sobre decisões de controle de acesso. Os padrões SAML e XACML são amplamente usados em Serviços *Web*. O uso destes na IoT também é possível, conforme pode ser visto nos trabalhos, a seguir.

Conforme citado na Seção 4.4.3, [Zhang e Liu 2011] apresentam um modelo de controle de acesso baseado em atributos e orientado a *Workflow* (WABAC), no qual permissões são geradas para usuários de acordo com seus atributos, atributos dos recursos, do ambiente e da tarefa atual. Na solução proposta, o SAML é usado para o transporte dos atributos do sujeito e o XACML é usado como linguagem para descrição das políticas de acesso e para tomada de decisão sobre quais usuários, baseado nas asserções SAML, podem acessar quais recursos.

Inicialmente, antes do sujeito solicitar o acesso ao sistema, ele deve possuir uma asserção SAML de atributos , emitida por uma autoridade de atributos. Em seguida, essa asserção SAML é inserida no cabeçalho de uma requisição SOAP enviada ao sistema. Ao receber a requisição, o sistema gera as tarefas relacionadas à requisição e as coloca em um estado *pronto*. Assim que uma das tarefas é ativada, o *Policy Enforcement Point* (PEP) obtém os atributos do sujeito, as informações da tarefa e monta uma requisição de autorização XACML, a qual é enviada para o *Policy Decision Point* (PDP). Cabe ao PDP tomar a decisão de autorização baseado nas políticas de autorização, no estado da tarefa e, caso precise de mais atributos, irá obtê-los através do *Policy Information Point* (PIP).

Em [Domenech e Wangham 2013] é proposta uma infraestrutura de autenticação e de autorização (IAA) para IoT que faz uso dos padrões SAML e XACML. O trabalho, em andamento, tem como objetivo prover autenticação e autorização de usuários e de dispositivos em domínios diferentes de segurança e que utilizam tecnologias de comunicação e de autenticação diferentes. A IAA proposta segue o modelo de identidades federadas, sendo o SAML usado para a troca de dados de atributos de usuários e de dispositivos. Cada domínio de segurança possui uma IAA, que contribui para a autenticação e o controle de acesso dos serviço web *RESTful*, disponibilizados pelos dispositivos (seguindo uma arquitetura orientada a recursos). O XACML é usado para expressar as políticas de controle

de acesso baseado em atributos e para troca de informações de autorização entre PDPs, PEPs e PIPs.

A IAA é composta de duas partes: uma disponibilizada como um Serviço *Web RESTful*, que contempla o provedor de identidade (IdP) para autenticação e o PDP para tomada de decisão de autorização de um dado domínio; e a outra embarcada em cada dispositivo (ou *smart gateway*), que deve ser usada por estes para redirecionar as funções de autenticação e de autorização para a IAA oferecida como um serviço e para prover a implementação do PEP (monitor de referência). A IAA irá prover um IdP SAML com suporte a diferentes técnicas de autenticação de dispositivos e de usuários, oferecendo, quando necessário, a transposição de credenciais de autenticação para o padrão SAML. A asserção de atributos resultante do processo de autenticação será apresentada para o provedor de serviço (do dispositivo) para que então o processo de autorização se inicie.

4.5.2. Autenticação de Usuários com OpenID e Windows CardSpace

O OpenID é um protocolo de autenticação única (SSO - *Single Sign On*) que permite que os usuários se autentiquem em sites (provedor de serviços), utilizando o identificador OpenID (conta) que desejarem. O OpenID também permite ao usuário controlar as informações que serão compartilhadas com as aplicações [Recordon e Reed 2006]. No OpenID, quando um usuário fornece o seu identificador, este é imediatamente redirecionado para o seu provedor OpenID, que realiza a autenticação utilizando o método de autenticação, suportado no provedor OpenID indicado. Após a confirmação dos dados, o usuário é redirecionado para o provedor de serviços, junto com seus atributos [OpenID 2007].

O Windows CardSpace é um metasistema que permite aos usuários escolherem, diante de um portfólio de identidades que possuem, aquela que melhor se adequa ao contexto de um dado provedor de serviços, independente do sistema que originou tal identidade [Chappell 2006]. O CardSpace é um componente da plataforma .Net da Microsoft, projetado para oferecer aos usuários uma experiência consistente do uso de múltiplas identidades digitais, a partir do uso de um agente (user-agent) especializado, chamado seletor de identidades. Quando um provedor de serviços requisita a autenticação e atributos de um usuário, o seletor de identidades do CardSpace transmite as informações requisitadas em um *token* de segurança digitalmente assinado, sendo que esse conjunto de atributos pode ser gerado e assinado pelo próprio usuário ou por um provedor de identidades externo, que gerencia a identidade selecionada pelo usuário [Maler e Reed 2008].

O uso do OpenID, do Windows CardSpace e do padrão SAML no cenário de Internet das Coisas, apenas para autenticação de usuários, é tratado no *middleware Hydra* [Akram e Hoffmann 2008c]. A proposta dos autores é que haja um complemento entre as tecnologias para prover uma solução de gestão de identidades seguras, na qual uma tecnologia complementa a outra.

4.5.3. OAuth e OpenID Connect

O OAuth é um *framework* de autenticação e de autorização que permite que um usuário/aplicação compartilhe recursos na web (delegue acesso a um recurso) com terceiros sem ter que compartilhar sua credencial de autenticação. Com o protocolo OAuth é possível

autorizar o acesso a esses recursos por um tempo determinado [Hardt 2012].

Na versão 2.0 do protocolo o OAuth, são definidos quatro papéis: proprietário do recurso, servidor de recursos, cliente e servidor de autorização. Uma das interações possíveis entre os papéis possui os seguintes passos [Hardt 2012]:

1. O cliente solicita a autorização do proprietário do recurso;
2. O proprietário do recurso verifica os dados do cliente e retorna a permissão de autorização, representada por uma credencial de autorização do proprietário do recurso;
3. O cliente utiliza a credencial de autorização para solicitar o *token* de acesso ao servidor de autorização;
4. O servidor de autorização autentica o cliente e valida a credencial de autorização e, se válidos, emite um *token* de acesso;
5. O cliente solicita o recurso (aplicação) ao servidor de recursos e se autentica utilizando o *token* de acesso;
6. O servidor de recursos verifica o *token* de acesso, se válido, disponibiliza o recurso ao cliente.

O OpenID Connect 1.0 é uma camada de identidade sobre o protocolo OAuth 2.0. Esta integração OpenID com OAuth permite que um cliente verifique a identidade do usuário final baseada na autenticação executada pelo *Authorization Server*, assim como para obter informações do perfil do usuário, a partir de uma solução interoperável e baseada em REST [Sakimura et al. 2013].

Segundo [Sakimura et al. 2013], o OpenID Connect 1.0 permite que clientes de diversos tipos, incluindo clientes Web, móveis e JavaScript, requisitem e recebam informações sobre sessões de autenticação de usuários finais. A especificação é extensível, permitindo, por exemplo, a cifragem de dados de identidade e a descoberta de provedores OpenID Connect.

Um trabalho em andamento que envolve o uso do OpenID Connect no cenário da WoT está sendo desenvolvido por [Santos et al. 2013]. Este trabalho tem por objetivo avaliar os impactos causados em um sistema de assistência médica pelo uso de um sistema de IdM centrado no usuário. A autenticação de usuários e de dispositivos e o estabelecimento das relações de confiança entre usuários, servidor OAuth (IdP) e o servidor de recurso [3] são providos pela infraestrutura de autenticação e autorização (IAA) OpenID Connect 1.0.

Uma característica importante do uso do OpenID Connect neste trabalho é que por incluir o OAuth 2.0 em sua arquitetura, é possível que clientes sejam não apenas navegadores web, mas também outros tipos de aplicações ou dispositivos, possibilitando ao IdP OpenID Connect ser utilizado não só para autenticar usuários, mas também dispositivos inteligentes e aplicações que enviam dados para o sistema de assistência médica remota.

[3] Equivale aos provedores de serviços (SPs).

Um usuário, através de seu navegador web e por meio da API RESTful oferecida pelo *Smart Gateway*, ao tentar acessar um recurso de um dispositivo médico, é redirecionado para o *OpenID Connect Provider*, para que o usuário se autentique. Após a autenticação, o navegador web é redirecionado ao servidor de recursos munido do *token* de acesso. Com base nos atributos do usuário, o SP concede acesso a ele, retornando uma mensagem com os sinais vitais do paciente obtidos por meio do dispositivo médico.

Na solução proposta, durante o processo de envio dos dados monitorados do paciente, obtidos com o dispositivo médico para uma aplicação web de assistência médica remota, o *smart gateway* também precisará se autenticar. O *Smart Gateway* ao tentar publicar dados na aplicação web (servidor de recurso), é informado pelo servidor de recurso que este precisa se autenticar, solicitando que indique seu *OpenID Connect Provider*. O dispositivo então se autentica no provedor escolhido e recebe um *token*, o qual é enviado ao servidor de recurso para que, baseado nos atributos do dispositivo, este possa conceder ou não o acesso ao serviço solicitado.

Outro trabalho em andamento, [Prazeres e do Prado Filho 2013], propõem uma infraestrutura para disponibilização de dispositivos físicos na Web (WoT) por meio de barramento de serviços. Para controlar e prover autenticação e autorização para acesso a esses dispositivos, a solução proposta pelos autores é a utilização do OpenID Connect. Neste caso, um servidor OpenID Connect, externo ao barramento, é o responsável por prover a autenticação de usuários finais que tentarem acessar os recursos (coisas) disponibilizados no barramento de serviços.

4.6. Iniciativas de Gestão de Identidades para Internet das Coisas

Com o objetivo de fazer um levantamento do estado da arte sobre autenticação e autorização na Internet das Coisas, uma revisão sistemática da literatura foi conduzida. A seguir, tem-se um pequeno resumo do protocolo de busca executado:

- Pergunta de pequisa: Quais mecanismos (soluções) abordam a autenticação ou autorização na IoT?
- *String* de busca em português: (Internet das Coisas OR IoT OR Dispositivos Inteligentes OR Objetos Inteligentes) AND (Autenticação OR Autorização OR Gestão de Identidade);
- *String* de busca em inglês: (Internet of Things OR IoT OR Smart Devices OR Smart Objects OR Machine to Machine) AND (Authentication OR Authorization OR Identity Management);
- Fontes pesquisadas: IEEExplore (`http://ieeexplore.ieee.org`;Springer Link (`http://link.springer.com/`, Google acadêmico (`http://scholar.google.com.br`, BDBComp (`http://www.lbd.dcc.ufmg.br/bdbcomp/bdbcomp.jsp`, ACM Digital Library `http://portal.acm.org`, Periódicos CAPES (`http://www.periodicos.capes.gov.br`).
- Critérios de seleção: data da publicação (2005 a 2013) e análise do título, resumo e conclusões de forma a confirmar o alinhamento com a pergunta de pesquisa;

- Critérios para exclusão: foram excluídos do estudo trabalhos cujos títulos e resumos eram conflitantes (em relação à questão de pesquisa) e soluções de segurança que não estavam alinhadas aos principais padrões de IoT.

- Período de execução da revisão sistemática: abril a agosto de 2013.

Como resultado desta revisão sistemática, trinta e um trabalhos (artigos) foram identificados e tabulados. Destes, alguns foram agrupados, pois são desdobramentos de uma pesquisa ou por serem resultados de projetos de pesquisas. Para seleção dos projetos apresentados nesta seção, utilizou-se como critérios: projetos cujos resultados foram publicados em artigos retornados com o protocolo de busca, que foram financiados por órgãos de fomento e que tiveram mais de um ano de duração. Em relação aos artigos, dentre os que retornaram na revisão sistemática, foram selecionados os trabalhos mais recentes (a partir de 2010) e relevantes em relação às características da IoT.

4.6.1. Trabalhos Acadêmicos

4.6.1.1. Nguyen et al. 2010

[Nguyen et al. 2010] abordam um cenário de saúde eletrônica (*e-health*) no qual enfermeiros e médicos precisam ter acesso aos dados de monitoramento da saúde do paciente em tempo real, quando este está dentro do hospital. Dados como temperatura e batimento cardíaco são monitorados através de sensores móveis que ficam no próprio paciente, além de dados sobre o ambiente em que o paciente está, como temperatura e umidade, que são medidos através de sensores fixos em cada ambiente. Esses sensores fazem parte de uma rede de sensores sem fio (RSSF) e podem se comunicar diretamente com o dispositivo móvel do médico, enviando informações diretamente para ele, sem a necessidade de uma infraestrutura de autenticação e comunicação ou de dispositivos intermediários que transferem dados para a Internet. Para esse ambiente, os autores propõem um esquema de autenticação baseado em identidades (IDs) dinâmicas para que um dispositivo autentique outro (M2M).

No cenário descrito no trabalho, diversas RSSF existem dentro de um hospital, sendo que cada rede é composta por um *gateway* e por diversos sensores móveis (instalados no pacientes) e sensores fixos. Há ainda um provedor de serviços M2M para o hospital. O esquema se divide em três etapas. A primeira e tapa consiste na inicialização da identidades de todos os sensores e *gateways* da rede. As identidades dos sensores são formadas pela concatenação da ID do domínio em que estão com a sua própria ID. Os *gateways*, possuem uma ID igual à ID do domínio seguido por uma sequência de zeros. Já os dispositivos móveis (portados por médicos e emfermeiros) possuem uma ID igual a concatenação de sua própria ID com a ID do provedor de serviços M2M.

A segunda etapa consiste na pré-distribuição de material criptográfico para que a comunicação entre os dispositivos (gateway, sensores fixos e móveis e dispositivos móveis) possa ocorrer. Nesta etapa, os sensores e os *gateways* recebem material criptográfico de modo a poderem se comunicar com qualquer dispositivo móvel, que é portado pelos médicos. Já a comunicação entre os sensores poderá ocorrer de acordo com uma probabilidade de que eles dividam o mesmo espaço de chaves criptográficas, que permitirá que eles, na

etapa seguinte, consigam gerar uma chave compartilhada comum.

A última etapa consiste na autenticação e ocorre de maneira distinta para dispositivos móveis, sensores móveis e sensores fixos. Os dispositivos móveis difundem suas identidades ao ingressarem em um domínio. Quando um sensor fixo recebe esta informação, este consegue determinar que se trata de um dispositivo móvel e envia essa informação para o *gateway* de seu domínio. O *gateway*, por sua vez, consulta o provedor de serviço M2M para confirmar se o dispositivo móvel é válido. Se for móvel, o *gateway* cria uma ID dinâmica para este dispositivo móvel, a qual é válida apenas para esse domínio. Essa ID dinâmica é então enviada ao sensor fixo que recebeu a informação de identidade difundida pelo dispositivo pela primeira vez. Ao receber essa informação, o sensor envia a sua própria ID ao dispositivo móvel, juntamente com um índice do espaço de chaves a que o sensor pertence. De posse disso, o dispositivo móvel e o sensor irão calcular uma chave compartilhada.

O dispositivo móvel cifra sua ID com esta chave compartilhada e a envia ao sensor fixo. O sensor tentará decifrar a mensagem e, caso não consiga, isto indica que se trata de um dispositivo malicioso que está forjando a ID de um dispositivo válido. Caso consiga decifrar com sucesso, então a ID dinâmica é enviada pelo sensor estático ao dispositivo móvel. Ambos irão calcular uma nova chave compartilhada, baseada na nova ID recebida, e poderão se comunicar com segurança. O sensor fixo difunde, aos sensores fixos do mesmo domínio, que há um novo dispositivo móvel no domínio. Caso o dispositivo móvel saia do domínio, o *gateway* avisará aos sensores fixos.

Em um sensor móvel o processo de autenticação ocorre de maneira diferente. Ao invés do *gateway* fazer a verificação de identidade com o provedor de serviço M2M, este a faz com o *gateway* no qual o sensor móvel estava associado anteriormente. Assim, quando um sensor móvel sai de um domínio B e entra num domínio A, este comunica-se com o *gateway* do domínio A, para que este confirme com o *gateway* de B que o sensor móvel não está mais no domínio B. Ao constatar que não está, o *gateway* de B avisa aos sensores de seu domínio que o sensor em questão deixou o domínio e, em seguida, o *gateway* de A recebe a confirmação da saída do sensor móvel do domínio B. Isso permite que o *gateway* do domínio A gere uma ID dinâmica que permitirá ao sensor móvel continuar com o processo de estabelecimento de uma chave compartilhada com algum sensor vizinho, da mesma forma como foi descrito para o dispositivo móvel. Ao fazê-lo, o sensor vizinho irá informar a todos do domínio que o sensor móvel faz parte da rede.

Deve-se ressaltar que o processo de estabelecimento de chave compartilhada entre um sensor móvel (colocado no paciente) e um sensor fixo (colocado no ambiente) é probabilístico, ou seja, é possível que dois sensores não consigam encontrar uma espaço de chaves comum para o cálculo de uma chave compartilhada. Isso se dá devido às restrições computacionais de tais dispositivos. Caso isto ocorra, os dois sensores devem encontrar uma maneira de concordar com uma chave comum ou utilizar outros sensores para encaminhamento de mensagens entre eles. Já com os dispositivos móveis, não há esse problema, haja vista que possuem mais recursos computacionais e podem carregar em sua memória toda a informação criptográfica necessária para se associar em qualquer domínio da rede hospitalar. Desse modo, é possível garantir que o dispositivo móvel sempre estará conectado em qualquer ambiente, enquanto o sensor móvel não possui tal garantia.

4.6.1.2. Alam et al. 2011

[Alam et al. 2011] abordam a provisão de acesso seguro a serviços na IoT e a interoperabilidade semântica de atributos de segurança entre diferentes domínios administrativos. No *framework* proposto, usa-se o conceito de regras semânticas para expressar restrições de autorização de acesso, que são usadas para inferir decisões de acesso. Este processo é chamado pelos autores de *security reasoning* (raciocínio de segurança).

Atributos de segurança de domínios administrativos diferentes normalmente são diferentes. Tratando-se de organizações diferentes, o Diretor da organização A não terá os privilégios do Diretor da organização B, quando um estiver atuando em recursos/serviços do outro. Este deve ser mapeado para um outro papel dentro da organização B, com menor nível hierárquico e mais restrições de acesso.

Os autores descrevem um cenário para esclarecer a motivação para o *framework* proposto. No cenário, deseja-se monitorar constantemente trens de uma infraestrutura ferroviária, com o objetivo de: (i) detectar anomalias, como temperatura dos componentes e vibrações elevadas, e (ii) tornar tais informações disponíveis para os diferentes atores da infraestrutura, como o operador do trem, o dono da infraestrutura de trilhos e o consumidor do serviço de transporte, que estão em domínios administrativos diferentes.

Para tratar o problema da interoperabilidade semântica dos aspectos de segurança, os autores propõem o uso de ontologias que levam em consideração as seguintes situações:

- Organizações mantém papéis/responsabilidades de modos diferentes (nomes de papéis, significados, hierarquias). Deve ser possível mapear, assim, o papel de um usuário em uma organização para o papel correspondente em outra organização;
- Manutenção de níveis de segurança diferentes pelas organizações. Mapear, conforme o papel, os níveis de segurança aplicáveis.

Os autores utilizam um *framework* arquitetural do ETSI para M2M chamado TS 102 690 [ETSI 2011], o qual é estendido para atender ao requisito de segurança entre domínios administrativos distintos. Para que o processo de raciocínio de segurança seja possível, é necessário que o sistema tenha conhecimento completo e formal do domínio em que atua, contendo a identificação de sensores, dados de sensores, identificação de usuários e atributos de usuários (como papéis).

Regras semânticas especificam as restrições de autorização de acesso e a execução das regras irá gerar as decisões de autorização. Basicamente, são utilizadas três ontologias na base de conhecimento: (i) a de sensores, que descreve sensores e dados recuperados por estes; (ii) de eventos, que descreve falhas e suas características; e (iii) de controle de acesso, que descreve os atores envolvidos na provisão de acesso seguro.

O *framework* proposto permite que aspectos relacionados ao controle de acesso em uma organização, possam ser influenciados pelos mecanismos e modelos de controle de acesso utilizados em outros domínios administrativos. A interoperabilidade semântica de aspectos de segurança, abordada com ontologias em uma plataforma M2M, permite que atributos de segurança em diferentes domínios possam ser compreendidos sem ambiguidades, visando assim garantir uma operação integrada [Alam et al. 2011].

4.6.1.3. Fu et al. 2011

[Fu et al. 2011] apresentam um sistema de gestão de identidades para um cenário *Machine-to-Machine* (M2M) em que os dispositivos possuem múltiplas funcionalidades e atendem a diferentes aplicações ao mesmo tempo, podendo atender cada uma com uma funcionalidade diferente. A identidade (ID) é definida pelos autores como um conjunto de funções do dispositivo e de opções de configuração. Atributos de uma identidade se referem, portanto, à descrição de uma função do dispositivo e as opções de configuração desta função.

Visando garantir a privacidade do dispositivo, para cada aplicação que utiliza uma função (ou um conjunto de funções) do dispositivo, este pode apresentar uma identidade diferente. Assim, para este cenário assume-se dois requisitos: (i) o requisito de autorização, no qual a aplicação exige que o dispositivo prove sua identidade; e (ii) o requisito de privacidade, no qual o dispositivo deseja manter privadas as informações não pertinentes à aplicação.

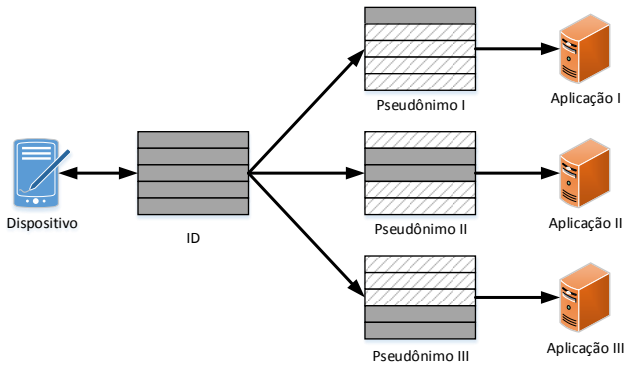

Figura 4.5: Relação do dispositivo, ID, pseudônimo e aplicações – [Fu et al. 2011]

Antes que um dispositivo possa acessar uma aplicação, este deve acessar um provedor de identidades (IdP) para obter um ID, tendo como base as funções e propriedades que este possui e a aplicação que este deseja acessar. O IdP armazena um índice de registro global das funções que cada dispositivo do domínio pode oferecer. Uma aplicação M2M, ao ingressar no domínio, busca no índice do IdP pelo dispositivo mais adequado para atender a necessidade desta aplicação. Para não fornecer funcionalidades e informações que não são relevantes para o acesso pleiteado, o dispositivo faz uso de pseudônimos, definido a partir de um subconjunto do seu ID (funções e configurações que o dispositivo deseja compartilhar com a aplicação alvo). O dispositivo apresenta suas credenciais à aplicação (ID completo ou pseudônimo), a qual o autentica e recebe direito de acesso às opções de configuração no dispositivo.

A Figura 4.5 mostra um dispositivo fornecendo um pseudônimo diferente para cada uma das aplicações, tendo como base o ID gerado pelo IdP. Apresentando diferentes pseudônimos para diferentes aplicações, o dispositivo consegue atender diversas aplicações

simultaneamente e ainda manter sua privacidade, já que as aplicações só poderão atuar sobre as configurações/funções que foram apresentadas pelo dispositivo. Ao utilizar um pseudônimo, o dispositivo deve ainda provar sua identidade para a aplicação, processo que é feito através do método *Zero Knowledge Proof* [Quisquater et al. 1989].

4.6.1.4. Hecate [Graf et al. 2011]

Hecate, proposto em [Graf et al. 2011], define um *framework* de autorização centralizado e flexível. O Hecate usa do conhecimento da estrutura dos recursos oferecidos pelos dispositivos para tomar decisões de autorização. A flexibilidade do framework de autorização se baseia em conjuntos de permissões que estão relacionados aos métodos do HTTP (*HTTP-verbs*) usados pelo REST.

O mecanismo de autorização do Hecate tem como base um modelo de usuário, que faz o mapeamento do usuário (suas credenciais, IDs) para um conjunto de regras, as quais são definidas em documentos XML de permissões (*Permission XML Documents* - PXDs). As requisições de acesso ao recurso contêm informações sobre o usuário requisitante e o URI. Essas duas informações ajudam a encontrar as regras aplicáveis à requisição (modelo do usuário). O documento PXD está relacionado à representação de diferentes regras e seus mapeamentos para funcionalidades do HTTP. Cada regra no PXD pode, opcionalmente, suportar filtragem baseado no conhecimento dos recursos, de acordo com o *HTTP-verb* usado na requisição, bem como baseado nas características desse recurso que são conhecidas pelo mecanismo de autorização. Cada regra no PXD refere-se a uma operação específica do HTTP sobre um determinado *Uniform Resource Identifier* (URI), sendo que esta regra está ligada à URI e está também ligada a uma permissão. Desta maneira, o PXD está baseado nos seguintes aspectos:

- Uma URI registrada pode ser protegida por muitas regras. Isto garante um conjunto de permissões em função das diversas maneiras de acessar uma URI (diversos *HTTP-verbs* do REST);
- Cada regra refere-se a um *HTTP-verb*;
- Além da autorização baseada *HTTP-verbs* do REST, filtros de permissões, baseados no conhecimento do recurso, podem ser aplicados;

Os filtros podem ser usados para alterar a requisição a um recurso (antes da requisição chegar ao dispositivo que provê o recurso) ou a resposta a uma requisição (antes de ela ser enviada ao requisitante). Em uma resposta a uma requisição, quando um filtro é aplicado, só é retornado ao requisitante a parte da informação que o filtro permite. Ou seja, é possível aplicar o filtro para realizar um controle de acesso mais granular ao recurso, a partir do conhecimento da estrutura desse recurso.

O Hecate permite que o controle de acesso se mantenha independente da representação dos recursos, baseando-se nas operações do HTTP. Com conhecimento da estrutura dos recursos, é possível ainda prover controle de acesso com alto grau de granularidade.

4.6.1.5. Rotondi et al. 2011

[Rotondi et al. 2011] apresentam uma abordagem de controle de acesso que utiliza o modelo de autorização baseada em habilidades (CapBAC), no âmbito do projeto IoT@Work[4]. Alguns elementos da abordagem proposta são descritos a seguir [Rotondi et al. 2011]:

- Recurso: pode ser um serviço de informação (que fornece uma medição), um serviço de aplicação ou ainda uma com de serviços. O recurso deve ser identificado de forma única em seu contexto;
- Habilidade de autorização: detalha os direitos de acesso concedidos, os recursos nos quais estes direitos podem ser aplicados, os sujeitos que pode usufruir desses direitos (habilidade) e outras informações adicionais, tais como validade da habilidade e restrições de uso;
- Serviço de Revogação de Habilidade: é utilizado para revogar uma ou mais habilidades. Uma revogação é criada por um sujeito que tem direitos específicos, sendo utilizada para informar ao serviço que faz a gestão do recurso que uma habilidade não é mais válida;
- Requisição de Operação: é uma requisição de serviço usual com uma característica adicional para referenciar ou incluir uma habilidade, que concede direitos ao requisitante;
- *Policy Decision Point* (PDP) do Recurso: responsável por validar e decidir acerca de uma requisição de acesso a um recurso. Avalia-se a habilidade presente na requisição de acesso (direitos que a habilidade garante) e frente às políticas de acesso ao recurso;
- Gerente de Recursos: responsável por gerenciar as requisições de acesso ao recurso, agindo também como *Policy Enforcement Point* (PEP), aplicando as decisões tomadas pelo PDP;
- Serviço de Revogação: responsável por gerenciar as revogações de habilidades e as políticas de acesso aos recursos aplicadas pelo PDP.

Um exemplo do uso desta abordagem e de seus elmentos é descrita em [Rotondi et al. 2011]. Bob é dono de um carro e precisa que outras pessoas possam ter acesso às informações de seu carro, porém a cada pessoa específica é desejado compartilhar somente informações específica. Assim, Bob gera habilidades que são associadas ao identificador da pessoa e que contém um conjunto de direitos de acesso, bem como o prazo de validade desta habilidade. Alice, sua esposa, recebe uma habilidade (C1) para obter informações sobre a localização geográfica do carro de Bob. O serviço de tráfego da cidade também recebe a habilidade (C2) sobre a localização geográfica, porém sem outras informações que possam identificar que o carro é de Bob. Assim, toda requisição enviado ao carro de Bob contém os dados do requisitante, sua assinatura digital e a habilidade que Bob delegou

[4] `https://www.iot-at-work.eu`

ao requisitante. A habilidade C2 foi delegada para uma determinada aplicação, a qual solicita acesso ao recurso determinado pela habilidade. O Gerente de Recursos solicita ao PDP para que tome uma decisão de controle de acesso baseada na habilidade apresentada. Com base em suas regras, o PDP informa ao Gerente de Recursos (PEP) sobre sua decisão, a qual é aplicada por ele.

Bob decide revogar a habilidade C1 que sua esposa possui. Para isso, ele cria uma nova habilidade de revogação, a qual é utilizada para informar ao Serviço de Revogação que uma determinada habilidade foi revogada. Essa habilidade contém informações sobre o ID do recurso ao qual a habilidade a ser revogada se aplica, qual a habilidade a ser revogada, quem está revogando esta habilidade, o período a partir do qual a revogação é válida, qual a habilidade que garante direitos a Bob de solicitar essa revogação e, por fim, a assinatura digital de Bob. O Serviço de Revogação, após verificar que Bob está autorizado a realizar a operação de revogação, aplica a revogação da habilidade e atualiza as regras do PDP [Rotondi et al. 2011].

4.6.1.6. Hanumanthappa e Singh 2012

[Hanumanthappa e Singh 2012] apresentam uma técnica de autenticação de usuários que visa garantir, além da autenticidade do usuário, a posse do dispositivo pelo usuário correto, o dono do dispositivo. Através disso, operações que demandam maior segurança, como o acesso à uma conta bancária, poderão ser feitas pelo usuário no dispositivo, desde que tanto o usuário quanto o dispositivo sejam autenticados e que o usuário tenha se registrado previamente como dono do dispositivo.

A solução proposta passa por quatro etapas. A primeira etapa é feita pelo fabricante do dispositivo (antes de sua venda), que o registra em um *Key Distribution Center* (KDC) exclusivo para fabricantes de dispositivos. Esse registro é feito através do envio do identificado (ID) e do número do modelo do dispositivo (*Device Model Number* – DMN). O KDC gera um *token* (T) composto pelo *hash* da ID do fabricante do dispositivo e um *timestamp* e os envia ao fabricante. Esse *token* é armazenado, juntamente com o ID do dispositivo e seu DMN, em um *Central Key Server* (CKS) para uso futuro no processo de autenticação do dispositivo. Esse *token* é criptografado com o algoritmo RSA e então gravado no *Trusted Platform Module* (TPM) embarcado no dispositivo.

A segunda etapa, referente ao registro do usuário como dono, é realizada após a venda do dispositivo. O dono do dispositivo se registra no CKS, enviando o ID do dispositivo e o *token* (T). O CKS compara estas informações com o ID do dispositivo e o *token* recebidos do fabricante na primeira etapa. Se esta verificação for bem-sucedida, o CKS envia ao usuário uma mensagem de *acknowledgement* (ACK) informando que a autenticação do dispositivo foi bem-sucedida e uma senha OTP (*one time password*). Em seguida, o usuário precisa se registrar, e para isso informa seu ID de usuário e a senha OTP ao CKS. Finalizada a etapa de registro, o CKS envia um ACK ao usuário, junto com um *Temp ID*. A partir desse momento, o usuário pode acessar qualquer serviço com seu dispositivo.

A terceira etapa trata de como usuários (não donos) se conectam nos dispositivos, etapa esta realizada por meio da Internet. Inicialmente, o usuário (A) que deseja se

conectar ao dispositivo de outro usuário (B) envia ao CKS seu *Temp ID*, o *token* (T) do seu dispositivo e os detalhes do dispositivo que este deseja se conectar. O CKS compara então o *Temp ID* e o *token* (T) com os armazenados na fase de registro do usuário e, caso a verificação seja bem sucedida, o CKS solicita que B envie seus dados para o CKS (*token* (T)) para que uma verificação também seja conduzida. Se a verificação de B for bem sucedida, então o CKS envia uma chave de sessão (*one time session key*) para ambos os dispositivos, que podem, então, se comunicar. Caso alguma das verificações não seja bem sucedida, o CKS envia uma mensagem ao outro usuário informando que ele está tentando realizar uma conexão com uma entidade não confiável.

A última etapa, de transações de alto nível, é aquela em que o usuário envia, através do dispositivo, informações importantes e sensíveis para provedores de serviço (como o acesso a uma conta bancária). Caso o usuário deseje realizar uma transação desse tipo, este enviará ao CKS a requisição de serviço, uma senha (*pass-phrase* P2) conhecida por ele e pelo CKS (informada manualmente pelo usuário), o *token* (T), seu *Temp ID* e um *nonce*. O CKS então autentica o usuário com base nas informações armazenadas nas etapas anteriores e envia como resposta o ID de um provedor de serviços adequado para atender à requisição do usuário. Junto com o ID do provedor de serviços, o CKS envia ao usuário uma chave de sessão (K), a senha P2, uma senha (*pass-phrase* P1, conhecida apenas pelo CKS) encriptada com um *nonce* do CKS. O usuário verifica se P2 é igual ao P2 enviado anteriormente e, em caso positivo, armazena K. Por fim, o usuário envia ao provedor de serviço o ID desse provedor e a senha P1 cifrada, conforme veio do CKS.

O provedor de serviço envia ao CKS a P1 cifrada e o ID do usuário. Após receber essa mensagem, o CKS a confronta com as informações armazenadas das mensagens anteriores e autentica o provedor de serviços. O CKS então responde ao SP com duas mensagens, (i) uma contendo o P2, K e o ID do provedor de serviço, cifradas com o *nonce* do usuário e (ii) outra contendo K e a ID do usuário. Por fim, o provedor de serviço envia ao usuário uma mensagem contendo o P2, K e a ID do provedor de serviço, cifrados com o *nonce* do usuário, conforme recebido do CKS no passo anterior. O usuário então decifra essa mensagem e verifica se P2 é o mesmo que foi mandado no início para o CKS e se K corresponde ao valor armazenado anteriormente. Se sim, o usuário autentica o provedor de serviço com a certeza de que não é uma entidade maliciosa. A partir desse momento, usuário e provedor de serviço trocarão mensagens cifradas com K, que é comum a ambos.

4.6.1.7. Liu et al. 2012

[Liu et al. 2012] apresentam uma arquitetura para Internet das Coisas que contempla a autenticação e controle de acesso para dispositivos e usuários. Nesta arquitetura, os dispositivos são nós finais da arquitetura da Internet, tendo endereços únicos globais (como IPv6) e podem-se comunicar entre si através da Internet.

A fim de gerenciar e organizar recursos massivos, o dispositivo faz seu pré-registro em um *gateway* confiável, denominado Autoridade de Registro (*Registration Authority* – RA). O RA auxilia o processo de autenticação e pode também ser usado para fins de auditoria.

O protocolo de autenticação proposto é mostrado na Figura 4.6. Inicialmente,

o usuário solicita acesso a um dispositivo (passo 1), o qual envia uma solicitação de autenticação do usuário para seu RA (passo 2). O RA então solicita ao usuário a sua identidade (ID) de usuário (passo 3), o qual responde com informações sobre o seu *Home Registration Authority* (HRA) e a sua ID (passo 4). No passo 5, o RA solicita ao HRA a verificação da ID do usuário. Para isto, o HRA, autentica o usuário utilizando oo método de autenticação que melhor se adeque às suas necessidades (passos 5.1 e 5.2). Na sequência, o HRA responde ao RA que a ID do usuário é válida ou não (passo 6). Por fim, o RA responde ao dispositivo sobre a ID do usuário e gera uma chave de sessão para ambos, utilizando um protocolo de estabelecimento e distribuição de chaves baseado em curvas elípticas (ECC) (passo 7).

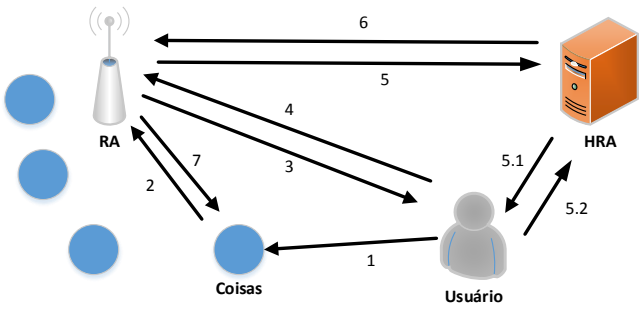

Figura 4.6: Protocolo de Autenticação – [Liu et al. 2012]

[Liu et al. 2012] também sugerem que o controle de acesso aos dispositivos seja feito pelo RA, usando o modelo RBAC. O algoritmo de controle de acesso decide quando uma nova conexão será aceita com base também nas informações de qualidade da comunicação. Caso a qualidade seja garantida, a conexão é aceita. Caso contrário, a conexão é descartada ou colocada em uma lista de espera. As conexões são classificadas em dois tipos: (i) a solicitação de um novo serviço que é disparada por usuários móveis dentro de um mesmo domínio e (ii) a solicitação de troca de domínio, feita por usuários móveis que estão mudando de um domínio para outro. O segundo caso tem mais prioridade na aceitação da conexão, haja vista que para os usuários é pior interromper um serviço que está sendo prestado, do que ser impossibilitado de ter acesso a um serviço.

Para a aceitação de uma conexão pelo mecanismo de controle de acesso também são levadas em consideração os requisitos de Qualidade de Serviço da conexão solicitada, baseado nas diferentes características tecnológicas das redes disponíveis. Por exemplo, um domínio pode ter duas redes, uma rede local sem fio, que possui largura de banda maior, porém tem um atraso maior, e uma rede típica de Internet das Coisas, com baixo atraso e largura de banda limitada. Baseado nessas informações, o mecanismo de controle de acesso julga se uma nova conexão será aceita ou não, permitindo a otimização do uso da rede. Por fim, aos usuários são atribuídos papéis dentro do domínio, baseado nas políticas configuradas para cada aplicação. De posse de todas essas informações, o mecanismo de controle de acesso baseado no modelo RBAC toma a decisão de aceitação ou não de uma nova conexão.

4.6.1.8. VIRTUS Middleware [Conzon et al. 2012]

Em [Conzon et al. 2012] é apresentado o *middleware* VIRTUS, o qual é orientado a eventos e tem como base o protocolo XMPP (*eXtensible Messaging and Presence Protocol*) [Saint-Andre 2004], amplamente utilizado como protocolo para troca de mensagens instantâneas e que possui mecanismos de segurança que contribuem com a interoperabilidade, como a federação de servidores e o mapeamento sobre o HTTP. No VIRTUS, os protocolos TLS (*Transport Layer Security*) e SASL (*Simple Authentication and Security Layer*) são utilizados para garantir a integridade e confidencialidade das mensagens e para a autenticação das partes envolvidas, respectivamente.

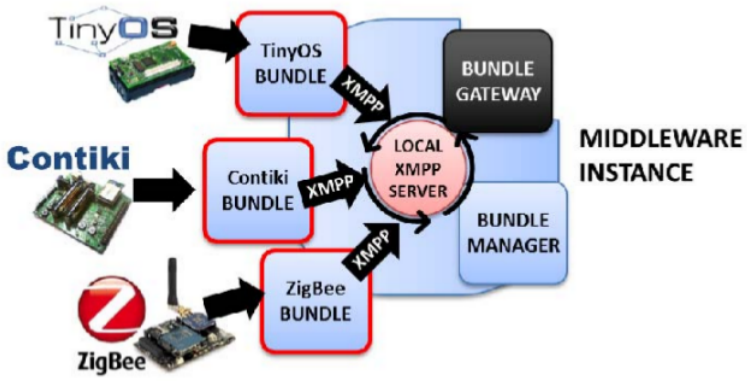

Figura 4.7: Módulos do *middleware* VIRTUS – [Conzon et al. 2012]

A Figura 4.7 ilustra os módulos presentes no *middleware* VIRTUS, conforme descritos a seguir:

- Módulos personalizados: também chamados de *bundles* (p.ex. *Zigbee bundle*), são usados para permitir a comunicação entre dispositivos com restrições computacionais e o Servidor VIRTUS, traduzindo as mensagens específicas da tecnologia do dispositivo para XMPP. Geralmente, são implementados em um intermediador entre o dispositivo e o resto do *middleware*;
- Servidor VIRTUS: utilizado para gerenciar a comunicação entre os componentes do *middleware*, sendo que há um servidor para cada rede (domínio) na qual o *middleware* é implementado;
- *Manager*: gerencia a conexão entre os diversos módulos do *middleware*. Este também provê uma lista dos módulos disponíveis e faz o gerenciamento de dependências;
- *Gateway*: se comunica com a instância local do Servidor VIRTUS, assim como com instâncias remotas do *middleware*. Este componente é o intermediário entre qualquer aplicação que deseje se comunicar com qualquer dispositivo dentro do *middleware*.

O *middleware* considera a existência de três tipos distintos de dispositivos: dispositivos com muitos recursos – como servidores, que implementam todo o *middleware*;

dispositivos restritos – como smartphones, que implementam módulos cliente XMPP para interagir com outros módulos; e dispositivos simples – como sensores e etiquetas RFID, que tem suas mensagens encapsuladas em formato XMPP por um outro dispositivo com mais recursos computacionais. A comunicação intra-domínio e inter-domínio é feita através de troca de mensagens XMPP, contudo há módulos que permitem a comunicação com outras arquiteturas, como por exemplo com Serviços *Web*.

4.6.1.9. Seitz et al. 2013

[Seitz et al. 2013] propõem um *framework* para IoT que permite o controle de acesso flexível e com granularidade fina para dispositivos com recursos computacionais restritos. O padrão XACML é utilizado visando permitir a aplicação de diferentes regras de autorização para diferentes clientes, além de permitir o controle de acesso em granularidade igual a de serviços RESTful. As funções desempenhadas no processo de decisão de autorização são tomadas fora do dispositivo que oferece o recurso, em um PDP, sendo que o dispositivo fica responsável pela tomada de decisão final de autorização.

Apesar da decisão de autorização ser tomada, basicamente, fora do dispositivo, as condições locais ao dispositivo também são importantes para a tomada de decisão (localização, por exemplo), sendo que estas não são encaminhadas para a tomada de decisão do PDP, mas avaliadas localmente após a decisão ter sido tomada. Neste trabalho, estas condições são expressas através de *XACML Obligations*, isto é, restrições perante as quais a decisão de autorização do PDP é válida. Para o transporte das decisões de autorização do PDP ao dispositivo, os autores propõem o o uso de asserções SAML de decisão de autorização.

O *framework* de autorização considera três entidades:

- Dispositivo: armazena recursos;
- Usuário: deseja acessar um recurso, enviando ao dispositivo, além da requisição de acesso, a asserção gerada pelo AE;
- Motor de Autorização (*Authorization Engine* - AE): faz a avaliação de políticas de autorização e gera as asserções de autorização para que o usuário acesse o recurso. Este atua em nome do dono do dispositivo que configurou as políticas de acesso.

Para que seja possível gerar uma asserção baseada na identidade do usuário, o motor de autorização deve autenticar o usuário, gerar a asserção de autorização e assiná-la utilizando uma chave conhecida e confiável para o dispositivo, permitindo que este possa verificar se a asserção é proveniente ou não de uma fonte confiável. Deste modo, ao receber do usuário um pedido de acesso a um recurso de um dispositivo, o motor de autorização, verifica se o usuário tem direito de acesso e gera, em caso positivo, uma asserção para este usuário. Junto da asserção é informada a chave de criptografia que deve ser usada na comunicação do dispositivo com o usuário.

O dispositivo, ao receber a asserção do usuário junto com a requisição de acesso, verifica as condições locais (através das *XACML Obligations*) e se os direitos da asserção

de autorização correspondem à requisição. Se o acesso for confirmado para as condições locais especificadas, o dispositivo garante o acesso ao recurso. De modo a facilitar o processamento das respostas XACML e asserções SAML no dispositivo, os autores definiram um subconjunto das especificações originais. Com o mesmo objetivo, a notação XML desse subconjunto foi substituída por uma notação baseada em JSON (*JavaScript Object Notation*), mais leve para o cenário, reduzindo a razão dez vezes o tamanho das mensagens.

4.6.1.10. Mahalle et al. 2013a

[Mahalle et al. 2013a] apresentam um modelo de controle de acesso baseado em habilidades e autenticação de identidades (*Identity Authentication and Capability Based Access Control* – IACAC) para Internet das Coisas. A inovação do modelo é que ele apresenta uma abordagem integrada de autenticação e controle de acesso para dispositivos em IoT, por meio de um novo método de autenticação de dispositivos e controle de acesso para recursos. O esquema proposto (IACAC) é compatível com diferentes tecnologias de acesso como Bluetooth, 4G, Wimax e WiFi, sendo que no trabalho a implementação é realizada em um ambiente WiFi.

No modelo, o algoritmo proposto é dividido em três partes: (i) geração de chave secreta baseado no algoritmo *Elliptical Curve Criptography – Diffie Hellman* (ECCDH), (ii) estabelecimento de identidade e (iii) criação de habilidade para controle de acesso.

O dispositivo ao ingressar em um domínio recebe um par de chaves. Essas chaves são geradas por um ou mais *Key Distribution Center* (KDC) considerados confiáveis. Em cada domínio, há um acordo acerca de um parâmetro de domínio relacionado ao protocolo ECC, comum a todos os dispositivos daquele domínio. Quando dois dispositivos desejam se comunicar, estes dão início à primeira etapa do algoritmo. Nesta etapa, os dispositivos enviam mensagens para troca de suas chaves públicas, permitindo o estabelecimento de uma chave secreta (X) para comunicação entre estes.

A segunda etapa possibilita a autenticação em uma via (*one way*)ou mútua. Na autenticação de uma via, o dispositivo A autentica o dispositivo B. Neste processo de autenticação, o dispositivo A gera um numero (r) e um *timestamp* (t). Uma chave de sessão (s) é então gerada pelo dispositivo A, a partir do *hash* de uma operação *XOR* entre a chave secreta (X) e o *timestamp* (t). Na sequência, o dispositivo A cifra o número (r) com a chave de sessão (s), gerando (R), e encripta também o *timestamp* (t) com a chave secreta (X), gerando (T). Em seguida, o dispositivo A gera um código de autenticação de mensagem (*Message Authentication Code MAC)* contendo a chave secreta (X), R e um *token* de habilidade baseado na identidade (chamado de ICAP). Após esse processo, o dispositivo A envia ao dispositivo B uma mensagem contendo R, T e o MAC gerado.

O dispositivo B, ao receber a mensagem vinda do dispositivo A, gera um *timestamp* local e confere se o *timestamp* (t) é menor que o local. Se sim, o dispositivo B tenta calcular a chave de sessão (s) para poder decifrar R, conseguindo assim acesso ao número (r) gerado pelo dispositivo A. O dispositivo B possui uma cópia da ICAP que supostamente deverá vir do dispositivo A. Assim, se a ICAP fornecida pelo dispositivo A for igual a ICAP armazenada no dispositivo B, este último calculará o MAC. Se o MAC gerado pelo

dispositivo B corresponder ao MAC que foi recebido, então o dispositivo B autentica o dispositivo B.

Para que ocorra a autenticação mútua (dispositivo A autentica o B), o dispositivo B cifra o número (r) com a chave secreta (X), gerando R1. Também é gerado por B um MAC contendo o número (r) e a ICAP armazenada por B. Em seguida, R1 e o MAC gerado são enviados para o dispositivo A. Quando a mensagem chega no dispositivo A, este decifra R1 e acessa o número (r) enviado por B. Se (r) for igual ao número (r) gerado no início da autenticação, o processo de autenticação mútua foi bem sucedido e o acesso será garantido de acordo com a ICAP do dispositivo B.

Na terceira etapa, de criação de habilidades para o controle de acesso, considera-se que a habilidade é composto por (1) um *token* que contém um conjunto de direitos de acesso, (2) um identificador do usuário ou dispositivo que é dono desta habilidade e (3) um *hash* dos dois elementos anteriores (para evitar que a habilidade seja forjada). Essa habilidade é utilizada na segunda etapa, sendo enviada no processo de autenticação.

4.6.2. Projetos

4.6.2.1. Middleware Hydra

O objetivo principal do projeto Hydra (*Link Smart Middleware*)[5] foi o desenvolvimento de um *middleware* baseado em uma arquitetura orientada a serviços (SOA), para a qual a camada de comunicação subjacente é transparente. O *middleware* deve incluir suporte para arquiteturas distribuídas e centralizadas seguras. O *middleware* foi concebido para operar em dispositivos que possuem limitações de recursos em termos de poder computacional, energia e uso de memória. Este deve permitir o desenvolvimento de aplicações seguras, confiáveis e tolerantes a faltas através do uso de componentes de segurança distribuídos.

Em [Akram e Hoffmann 2008b] são apresentados os requisitos de gestão de identidades (*Identity Management* - IdM) e são introduzidas as recomendações para a arquitetura do *middleware*. Neste trabalho, foram extraídos 10 requisitos para IdM, tendo como referência um cenário de automação residencial do Projeto Hydra.

[Akram e Hoffmann 2008c] é proposto um metasistema de identidades, independente de tecnologia, que possui três papéis: o sujeito, ao qual se refere a identidade; a *Relying Party* (RP), que requisita informações de identidade relacionadas ao sujeito, e o *Identity Provider* (IdP), que fornece informações de identidade do sujeito. No Hydra, uma identidade tipicamente compreende: (1) Identificadores virtuais temporários; (2) Atributos que especificam a entidade; (3) Histórico de acessos feito a esta entidade e a partir desta entidade.

No projeto foram definidos alguns requisitos para o gestor de identidades Hydra (*Hydra Identity Manager* - HIM), a saber:

- Permitir que o usuário seja o responsável por controlar os dados enviados e recebidos;
- Liberar o mínimo de informação (somente o suficiente), para que uma determinada ação seja executada;

[5] `http://www.hydramiddleware.eu/`

- Garantir a irretratabilidade (não repúdio dos dados trocados;
- Suportar diferentes tecnologias de gerenciamento de identidade de diversos fabricantes, além de prover a interoperabilidade dessas diferentes tecnologias;
- Desacoplamento da camada de identidade da camada de aplicação, permitindo que as políticas e componentes de IdM sejam alterados sem que as aplicações sofram alterações, e vice-versa;
- Preocupação com aspectos de usabilidade do usuário, nos processos de seleção e divulgação da identidade;
- Experiência consistente entre contextos, considerando o fato de que uma entidade pode ter muitas identidades (e vice-versa) dependendo do contexto (para prover mais privacidade aos usuários);
- Escalabilidade no gerenciamento das identidades, em função da dinamicidade do ambiente (entidades entrando e saindo do ambiente frequentemente).

De modo a atender esses requisitos, o Hydra Middleware constituiu a arquitetura da HIM de modo a integrar diversas tecnologias e especificações, dentre elas WS-*, Windows Cardspace, OpenID e SAML. A ideia é que haja um complemento entre as tecnologias para prover uma solução de gestão de identidades, de modo que uma tecnologia complemente aspectos de segurança falhos na outra.

4.6.2.2. Butler - SmartLife

BUTLER, acrônimo de *uBiquitous, secUre inTernet-of-things with Location and contExtawaReness*, é um projeto europeu em andamento que começou suas atividades em outubro de 2011. O projeto aborda aspectos de pervasividade, consciência de contexto e segurança na Internet das Coisas. Integra tecnologias existentes e desenvolve novas tecnologias a fim de formar um conjunto de aplicações, serviços e características das plataformas que permitirá trazer a Internet das Coisas para a vida diária [BUTLER 2011].

O projeto apresenta soluções para cenários como *Smart Home/Office*, *Smart Shopping*, *Smart Mobility/Transport*, *Smart Health* e *Smart Cities*. Foram definidos papéis de segurança no nível de aplicação, que refletem os *stakeholders* participantes das interações em cada cenário. Os papéis definidos no projeto são [Hennebert et al. 2013]:

- Usuário: entidade que ganha acesso a um recurso. Normalmente é um humano, mas pode ser também uma aplicação;
- Provedor de Recurso: entidade que provê um recurso e opcionalmente o atualiza. Ele deve conferir o *token* de acesso apresentado para que possa prover/atualizar um recurso;
- Consumidor de Recurso: aplicação cliente recuperando e consumindo recursos em nome do usuário;

- Servidor de Autorização: é a entidade que implementa a gestão de controle de acesso. É responsável pela autenticação do usuário e autorização do consumidor de recurso através da geração de um *token* de acesso relacionado ao recurso que se deseja acessar. Opcionalmente, pode delegar a tarefa de autenticação para o servidor de autenticação;

- Servidor de Autenticação: esta entidade pode ser utilizada pelo servidor de autorização de modo a confiar em um protocolo de autenticação que não é implementado nativamente no servidor de autorização. Isso significa que o servidor de autenticação e o servidor de autorização precisarão fazer a federação de identidades de usuários.

A Figura 4.8 ilustra o fluxo de mensagens, baseado no protocolo OAuth 2.0, para acesso a um recurso, o qual deve ser autorizado pelo usuário. Utilizando um agente de usuário (como um navegador web), o usuário requisita um serviço a um Provedor de Serviço, que neste caso fará o papel de Consumidor do Recurso. Através do agente de usuário, a aplicação do Provedor de Serviço requisita um código de autorização para acesso ao recurso. O Servidor de Autorização valida a aplicação e retorna um *token* de aplicação.

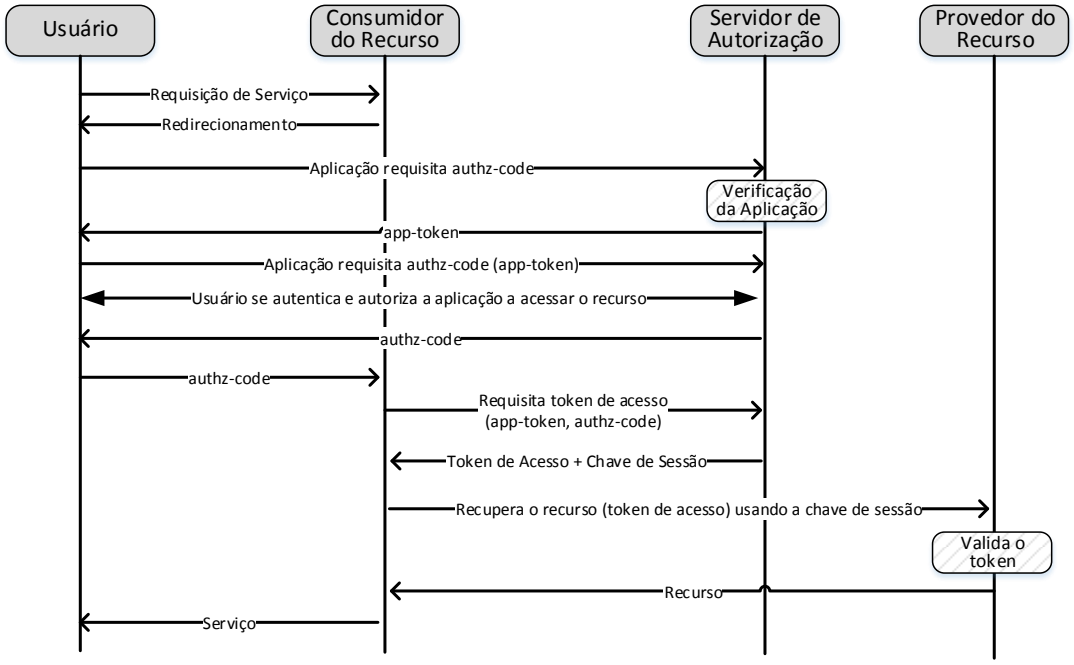

Figura 4.8: Fluxo de mensagens para acesso a um recurso - [Hennebert et al. 2013]

De posse do *token*, a aplicação requisita novamente o código de autorização e o usuário realiza o processo de autenticação junto ao Servidor de Autorização e autoriza a aplicação a acessar o recurso desejado, recebendo o código de autorização. Em seguida, em nome do usuário (e não mais através do agente de usuário), a aplicação do provedor de serviço, através do código de autorização obtido no passo anterior, requisita ao Servidor de Autorização um *token* de acesso. O Servidor de Autorização gera o *token* de acesso

e também gera uma chave de sessão, que será usada entre a aplicação do provedor de serviço e o Provedor de Recurso. Utilizando o *token* de acesso e a chave de sessão gerada, a aplicação requisita o recurso ao Provedor de Recurso. O Provedor de Recurso, por sua vez, valida o *token* de acesso e provê o recurso. Por fim, a aplicação do provedor de serviço consome o recurso e provê o serviço ao usuário [Hennebert et al. 2013].

4.6.2.3. Middleware SMEPP

SMEPP, acrônimo de *Secure Middleware for Embedded Peer-to-Peer systems*, é um projeto de *middleware* para sistemas embarcados em uma arquitetura P2P (*Peer-to-Peer*) que tem como foco a segurança. O *middleware* visa facilitar o desenvolvimento de aplicações, escondendo detalhes das plataformas dos sistemas e detalhes relacionados à escalabilidade, adaptabilidade e interoperabilidade entre tais sistemas. Desse modo, uma premissa deste *middleware* é prover mecanismos que garantam interações seguras entre os pares e abstraiam para os desenvolvedores de aplicações os problemas como falta de infraestrutura e vulnerabilidades de segurança. Também tem como premissa que o *middleware* seja altamente personalizável e adaptável a diferentes dispositivos e domínios [Caro et al. 2009, Roman et al. 2011a].

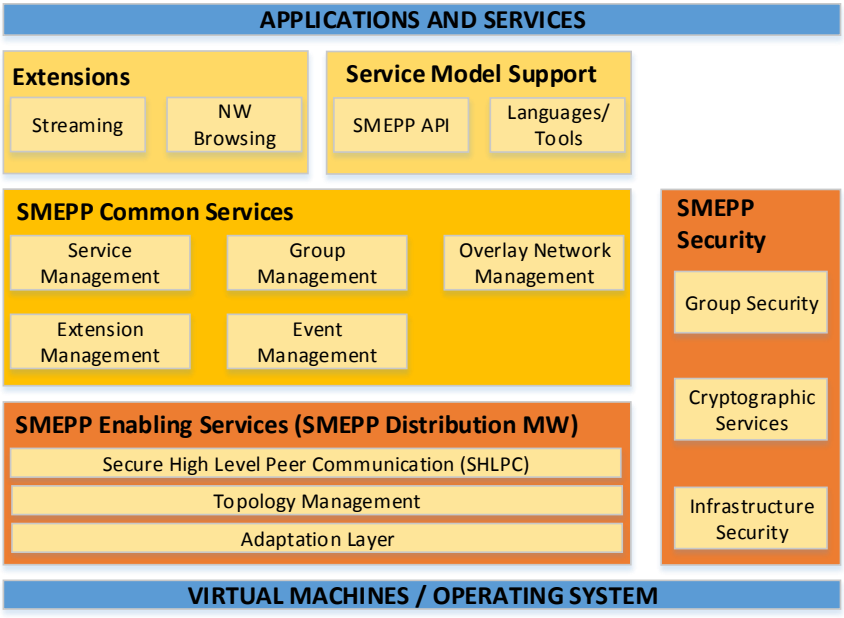

Figura 4.9: Arquitetura do Middleware SMEPP - [Roman et al. 2011a]

Na arquitetura do SMEPP os componentes são divididos em camadas, como pode ser visto na Figura 4.9. A camada superior consiste em uma API para os desenvolvedores de aplicações, a qual permite o acesso aos serviços do *middleware*, além de serviços específicos para cada domínio de aplicação. A camada intermediária oferece serviços comuns para qualquer aplicação que seja executado sobre o *middleware*, como gestão de eventos, gestão de grupos e monitoramento de entrada e saída de dispositivo. A

camada inferior, por sua vez, trata das implementações de funcionalidades oferecidas pelo *middleware* na arquitetura em que ele está embarcado, se preocupando com detalhes dos dispositivos como a capacidade computacional e de energia. Contudo, a arquitetura pode ser simplificada contendo apenas os componentes úteis para um determinado domínio e contexto de aplicação [Caro et al. 2009, Roman et al. 2011a].

Dentro de uma rede SMEEP, os serviços são oferecidos para membros de um mesmo grupo. Os componentes da camada de segurança da arquitetura SMEEP são apresentados a seguir: [Caro et al. 2009, Roman et al. 2011a].

- Segurança de Grupo (*Group Security*): Responsável por estabelecer e manter aspectos de segurança dentro dos grupos, desempenhando tarefas como autenticação de pares que tentam entrar em um grupo e garantindo a comunicação segura entre eles. É nesse componente que é definido, por exemplo, se um membro deve utilizar criptografia simétrica ou assimétrica;
- Serviços de Criptografia (*Cryptographic Services*): provê primitivas criptográficas que serão usadas por outros componentes, como a cifragem, decifragem, assinaturas digitais, código de autenticação de mensagem (MAC), etc. Também provê funcionalidades para geração de números aleatórios, armazenamento de chaves e gestão de certificados;
- Infraestrutura de Segurança (*Infrastructure Security*): faz uso de características de segurança específicas dos sistemas em que o *middleware* está embarcado, facilitando a implementação dos componentes de segurança. Como exemplo dessas características dos sistemas tem-se os conjuntos de instruções específicas para operações criptográficas que um processador pode ter, agilizando os processos de segurança em todo o *middleware*.

Para um sistema ingressar em um grupo é necessário que o mesmo apresente credenciais de segurança válidas. Uma vez dentro do grupo, este sistema poderá se comunicar de forma segura com os demais membros. Compete ao desenvolvedor da aplicação, que é executada sobre o *middleware*, o provimento de credenciais de segurança e ao *middleware* compete o processo para admissão de membros e a comunicação segura dentro do grupo [Caro et al. 2009].

4.6.2.4. Internet of Things - Architecture (IoT-A)

O projeto europeu *Internet of Things - Architecture* (IoT-A) tem como objetivo a criação de um modelo arquitetural de referência para a Internet das Coisas [IoT-A 2009]. Um ponto interessante da arquitetura proposta é que esta define o que se deve ter em uma solução de Internet das Coisas, mas não define através de qual tecnologia isso deve ser implementado. O núcleo da arquitetura é composta por [Gruschka e Gessner 2012] :

- Autorização (*AuthZ*): responsável pelo controle de acesso aos serviços e à infraestrutura de resolução de serviços;

- Autenticação (*AuthN*): responsável pela autenticação de usuários dos serviços;
- Gestão de Identidades (*Identity Management* - IM): responsável pela gestão de identidades, pseudônimos e políticas de acesso relacionadas;
- Gestão e Troca de Chaves (*Key Exchange and Management* - KEM): responsável pela troca de chaves criptográficas;
- Confiança e Reputação (*Trust and Reputation* - TRA): responsável pela coleta de pontuação sobre reputação e pelo cálculo do nível de confiança dos serviços.

O componente AuthZ é responsável por tomar decisões de controle de acesso baseado em políticas de controle de acesso. De maneira abstrata, uma decisão de controle de acesso pode ser modelada dessa forma: *authZ(s, r, o) -> true, false*. Neste caso, *s* é o sujeito tentando executar uma operação *o* sobre um recurso *r*. A decisão de controle de acesso leva em consideração as informações do usuário requisitante, do recurso e da ação que o usuário deseja realizar , resultando em um valor *booleano* acerca da permissão ou negação do acesso. Dessa forma, o componente AuthZ atua como um *Policy Decision Point* (PDP) [Gruschka e Gessner 2012].

O componente AuthN garante que a identidade de um usuário ou serviço é válida. A funcionalidade básica deste componente é oferecida da seguinte maneira: *Assertion: authenticate(UserCredential)*. *Assertion* é o componente de dados que garante que uma autenticação de um usuário ocorreu em determinado momento através de um método de autenticação específico. *UserCredential* é a entrada para o processo de autenticação, que permite que o mecanismo tenha certeza de que o usuário é quem diz ser ou não, sendo normalmente baseado naquilo que o usuário *tem*, naquilo que ele *sabe* ou naquilo que e *é* [Gruschka e Gessner 2012].

O componente IM é responsável por gerar pseudônimos para as IDs dos usuários e serviços, garantindo assim o anonimato destes. Pseudônimos são identidades temporárias de sujeitos fictícios (ou grupos de sujeitos) que podem ter suas credenciais usadas para interações entre sujeitos ou grupos de sujeitos, ao invés de se usar a identidade e as credenciais reais. De um ponto de vista abstrato, a funcionalidade que deve ser provida pelo componente IM é a seguinte: *createPseudo (s1 [,s2, s3, ..., sn], p) -> s**. Neste caso, *s* representa um sujeito ou um conjunto de sujeitos que requisitam um pseudônimo *s**. Há também um conjunto opcional de especificações chamado *p*, podendo ser, por exemplo, o tamanho da chave, algoritmo a ser usado, validade do pseudônimo, direitos de acesso, etc [Gruschka e Gessner 2012].

A geração de pseudônimos obedece as seguintes regras: (i) garantir que os direitos de acesso do pseudônimo estejam contidos no conjunto de direitos do sujeito real; (ii) garantir que o período de validade do pseudônimo seja menor ou igual ao do ID do sujeito real; (iii) garantir que os direitos de acesso de um grupo que solicita um pseudônimo seja igual ou menor que o de qualquer um dos sujeitos do grupo; e (iv) garantir que o período de validade do pseudônimo seja menor ou igual que o de qualquer um dos sujeitos do grupo. Os passos (iii) e (iv) são opcionais, haja vista que quem solicita o pseudônimo para o grupo deverá ser responsabilizado pelo uso que é feito deste por qualquer um dos componentes do grupo [Gruschka e Gessner 2012].

Também, é responsabilidade do componente IM prover a interoperabilidade entre diferentes *frameworks* de autenticação e autorização, o que permite que a autenticação feita em um *framework* resulte em uma asserção que possa ser usada em outro *framework* [Gruschka e Gessner 2012]. Dessa forma, os autores sugerem o uso do XACML para implementar o AuthZ e do SAML para implementar o AuthN.

4.7. Considerações finais

As possibilidades de aplicações para Internet das Coisas são inúmeras e, dentre estas, há potencial para criar ambientes inteligentes através dos *smart objetcts*, que são objetos que tem a capacidade de sentir e atuar sobre o meio em que estão inseridos. As características diferenciadas e muitas vezes restritivas da IoT, como a sua natureza distribuída, a facilidade de acesso físico aos objetos e os objetos com recursos computacionais restritos, tornam o provimento da segurança um desafio.

Este capítulo analisou a segurança na Internet das Coisas, dando foco aos aspectos de autenticação e autorização neste cenário. Os dispositivos na IoT geram, transmitem, modificam e armazenam dados constantemente, sendo que estas informações muitas vezes são confidenciais para seus usuários. Estes dispositivos podem pertencer a mais de uma rede (domínio) e podem de se deslocar por mais de um domínio, o que afeta as abordagens de autenticação e de controle de acesso.

Nos trabalhos apresentados, é possível notar a opção por não conceber mecanismos de autenticação e autorização que estejam condicionados a um determinado domínio de aplicação, nem à determinadas tecnologias utilizadas na IoT, tais como IEEE 802.15.4, WiFi ou Zigbee.

Dos principais modelos de controle de acesso usado em cenários convencionais, o modelo CapBAC (ver Seção 4.4.3) foi apontado como o mais adequado para o cenário da IoT, por permitir um controle mais granular, sem exigir que os dispositivos tenham que lidar com a complexidade de manter listas de controle acesso. O modelo ABAC também se mostrou adequado para o cenário da IoT, como pode ser constatado na adaptação deste modelo feita por [Zhang e Liu 2011].

Nos trabalhos analisados, constatou-se a tendência em se utilizar *gateways* ou servidores dedicados como mecanismos de apoio aos dispositivos quando estes realizam operações relacionadas à segurança que exigem um grande poder computacional, como as operações criptográficas. Contudo, poucos trabalhos exploraram a questão sobre autenticação única de usuários e dispositivos e a transposição destas credenciais de autenticação por diferentes domínios administrativos.

Apesar de existirem diversos trabalhos na literatura que descrevem soluções de segurança para IoT, ainda é necessário superar uma série de desafios científicos e tecnológicos para que estas soluções sejam utilizadas e difundidas em sua forma plena. A implementação e avaliação dos mecanismos dos trabalhos apresentados neste capítulo em cenário reais, já que grande parte dos trabalhos carecem de implementação ou provas formais, é uma oportunidade de pesquisa.

Autenticação e autorização para sistemas distribuídos pervasivos e ubíquos como a IoT são temas de pesquisas atuais e ativos e, provavelmente, diante das suas complexidades

e relevâncias, continuarão assim por muito anos. Esta constatação decorre das inúmeras questões que os sistemas de gestão de identidades devem considerar, tais como: privacidade e anonimato do usuário, uso de algoritmos criptográficos fortes em dispositivos com restrições computacionais, autenticação única (SSO) de dispositivos diante de diferentes tecnologias, controle de acesso de granularidade fina e interoperabilidade semântica entre regras de autorização.

Referências

[Aboba et al. 2004] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., e Levkowetz, E. H. (2004). Extensible authentication protocol (eap). `http://tools.ietf.org/html/rfc3748`.

[Ahson 2012] Ahson, Syed A; Ilyas, M. (2012). *Near Field Communications Handbook*. CRC Press.

[Akram e Hoffmann 2008a] Akram, H. e Hoffmann, M. (2008a). Laws of identity in ambient environments: The hydra approach. In *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08. The Second International Conference on*, pages 367–373. IEEE.

[Akram e Hoffmann 2008b] Akram, H. e Hoffmann, M. (2008b). Requirements analysis for identity management in ambient environments: The hydra approach. *Context Awareness and Trust 2008*, page 17.

[Akram e Hoffmann 2008c] Akram, H. e Hoffmann, M. (2008c). Supports for identity management in ambient environments-the hydra approach. In *Systems and Networks Communications, 2008. ICSNC'08. 3rd International Conference on*, pages 371–377. IEEE.

[Akyildiz et al. 2002] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., e Cayirci, E. (2002). Wireless sensor networks: a survey. *Comput. Netw.*, 38(4):393–422.

[Alam et al. 2011] Alam, S., Chowdhury, M. M., e Noll, J. (2011). Interoperability of security-enabled internet of things. *Wireless Personal Communications*, 61(3):567–586.

[Alliance 2013] Alliance, I. (2013). Ipso. `www.ipso-alliance.or`.

[Atzori et al. 2010] Atzori, L., Iera, A., e Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54(15):2787–2805.

[Babar et al. 2010] Babar, S., Mahalle, P., Stango, A., Prasad, N. R., e Prasad, R. (2010). Proposed security model and threat taxonomy for the internet of things (iot). In Meghanathan, N., Boumerdassi, S., Chaki, N., e Nagamalai, D., editors, *CNSA*, volume 89 of *Communications in Computer and Information Science*, pages 420–429. Springer.

[Babar et al. 2011] Babar, S., Stango, A., Prasad, N., Sen, J., e Prasad, R. (2011). Proposed embedded security framework for internet of things (iot). In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pages 1–5. IEEE.

[Baronti et al. 2007] Baronti, P., Pillai, P., Chook, V. W. C., Chessa, S., Gotta, A., e Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer communications*, 30(7):1655–1695.

[Bonetto et al. 2012] Bonetto, R., Bui, N., Lakkundi, V., Olivereau, A., Serbanati, A., e Rossi, M. (2012). Secure communication for smart iot objects: Protocol stacks, use cases and practical examples. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–7. IEEE.

[Buettner et al. 2008] Buettner, M., Greenstein, B., Sample, A., Smith, J. R., e Wetherall, D. (2008). Revisiting smart dust with rfid sensor networks. In *ACM Workshop on Hot Topics in Networks (HotNets-VII), 2008 7th*. ACM.

[BUTLER 2011] BUTLER (2011). About the butler project. `http://www.iot-butler.eu/about-butler`.

[Caro et al. 2009] Caro, R. J., Garrido, D., Plaza, P., Roman, R., Sanz, N., e Serrano, J. L. (2009). Smepp: A secure middleware for embedded p2p. In *ICT Mobile and Wireless Communications Summit (ICT-MobileSummit'09)*, Santander (Spain).

[Cavoukian 2012] Cavoukian, A. (2012). Mobile near field communications: Keep it secure and private. *Information Systems Security Association Journal*, 10(8):12–17.

[CERP-IoT 2010] CERP-IoT (2010). Vision and challenges for realising the internet of things. `http://www.theinternetofthings.eu/sites/default/files/Rob%20van%20Kranenburg/Clusterbook%202009_0.pdf`.

[Chappell 2006] Chappell, D. (2006). Introducing windows cardspace. Msnd technical articles, Microsoft Corporation. `http://msdn.microsoft.com/en-us/library/aa480189.aspx`.

[COMMUNITIES 2008] COMMUNITIES, C. C. O. T. E. (2008). Future networks and the internet: Early challenges regarding the internet of things. Technical report, CTEC.

[Conzon et al. 2012] Conzon, D., Bolognesi, T., Brizzi, P., Lotito, A., Tomasi, R., e Spirito, M. A. (2012). The virtus middleware: An xmpp based architecture for secure iot communications. In *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pages 1–6. IEEE.

[De Souza et al. 2008] De Souza, L. M. S., Spiess, P., Guinard, D., Köhler, M., Karnouskos, S., e Savio, D. (2008). Socrades: A web service based shop floor integration infrastructure. In *The internet of things*, pages 50–67. Springer.

[Domenech e Wangham 2013] Domenech, M. C. e Wangham, M. S. (2013). Uma infraestrutura de autenticação e de autorização para internet das coisas baseada no saml e xacml. In *Segurança da Informação e de Sistemas Computacionais (SBSeg), 2013 13o Simpósio Brasileiro em*. SBC.

[Eronen e Tschofenig 2005] Eronen, E. P. e Tschofenig, E. H. (2005). Pre-shared key ciphersuites for transport layer security (tls). `http://tools.ietf.org/html/rfc4279`.

[ETSI 2011] ETSI (2011). Etsi ts 102 690 v1.1.1 – machine–to–machine communications (m2m); functional architecture. Technical report, ETSI.

[Feliciano et al. 2011] Feliciano, G., Agostinho, L., Guimarães, E., e Cardozo1, E. (2011). Gerência de identidades federadas em nuvens:enfoque na utilização de soluções abertas. In *Minicurso - SBSeg 2011 - Brasília - DF*.

[Fielding e Taylor 2002] Fielding, R. T. e Taylor, R. N. (2002). Principled design of the modern web architecture. *ACM Trans. Internet Technol.*, 2(2):115–150.

[Fongen 2012] Fongen, A. (2012). Identity management and integrity protection in the internet of things. In *Emerging Security Technologies (EST), 2012 Third International Conference on*, pages 111–114. IEEE.

[Fu et al. 2011] Fu, Z., Jing, X., e Sun, S. (2011). Application-based identity management in m2m system. In *Advanced Intelligence and Awareness Internet (AIAI 2011), 2011 International Conference on*, pages 211–215. IEEE.

[Gorlatova et al. 2010] Gorlatova, M., Sharma, T., Shrestha, D., Xu, E., Chen, J., Skolnik, A., Piao, D., Kinget, P., Kymissis, J., Rubenstein, D., e Zussman, G. (2010). Prototyping energy harvesting active networked tags (enhants) with mica2 motes. In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, pages 1–3.

[Graf et al. 2011] Graf, S., Zholudev, V., Lewandowski, L., e Waldvogel, M. (2011). Hecate, managing authorization with restful xml. In *Proceedings of the Second International Workshop on RESTful Design*, pages 51–58. ACM.

[Group 2004] Group, W. W. (2004). Web services architecture. `http://www.w3.org/TR/ws-arch/`.

[Gruschka e Gessner 2012] Gruschka, N. e Gessner, D. (2012). Project deliverable d4.2 - concepts and solutions for privacy and security in the resolution infrastructure. `http://www.iot-a.eu/public/public-documents/d4.2/at_download/file`.

[GS1-EPCglobal 2009] GS1-EPCglobal (2009). The epcglobal architecture framework, epcglobal final version 1.3.

[Gubbi et al. 2013] Gubbi, J., Buyya, R., Marusic, S., e Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

[Guinard et al. 2010] Guinard, D., Fischer, M., e Trifa, V. (2010). Sharing using social networks in a composable web of things. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 702–707.

[Guinard e Trifa 2009] Guinard, D. e Trifa, V. (2009). Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009)*.

[Guinard et al. 2011] Guinard, D., Trifa, V., Mattern, F., e Wilde, E. (2011). From the internet of things to the web of things: Resource-oriented architecture and best practices. In Uckelmann, D., Harrison, M., e Michahelles, F., editors, *Architecting the Internet of Things*, pages 97–129. Springer Berlin Heidelberg.

[Han e Li 2012] Han, Q. e Li, J. (2012). An authorization management approach in the internet of things. *Journal of Information & Computational Science*, 9(6):1705–1713.

[Hanumanthappa e Singh 2012] Hanumanthappa, P. e Singh, S. (2012). Privacy preserving and ownership authentication in ubiquitous computing devices using secure three way authentication. In *Innovations in Information Technology (IIT), 2012 International Conference on*, pages 107–112. IEEE.

[Hardt 2012] Hardt, E. D. (2012). The oauth 2.0 authorization framework. `http://tools.ietf.org/html/rfc6749`.

[Heer 2006] Heer, T. M. (2006). Lhip - lightweight authentication for the host identity protocol. Master's thesis, University of Tübingen.

[Hennebert et al. 2013] Hennebert, C., Denis, B., Gall, F. L., Copigneaux, B., Clari, F., Sottile, F., Mauro, F., Smadja, P., Pascali, S., Preuveneers, D., Ramakrishnan, A., Sancho, J., Shrestha, A., Valla, M., Salazar, M. F., Monjas, M.-A., Macagnano, D., e Korhonen, J. (2013). D2.4 - selected technologies for the butler platform. `http://www.iot-butler.eu/wp-content/plugins/download-monitor/download.php?id=22`.

[Horrow e Sardana 2012] Horrow, S. e Sardana, A. (2012). Identity management framework for cloud based internet of things. In *Proceedings of the First International Conference on Security of Internet of Things*, pages 200–203. ACM.

[Hu et al. 2013] Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., e Scarfone, K. (2013). Guide to attribute based access control (abac) definition and considerations (draft). Technical report, National Institute of Standards and Technology.

[Hu e Scarfone 2012] Hu, V. C. e Scarfone, K. (2012). Guidelines for access control system evaluation metrics. Technical report, National Institute of Standards and Technology.

[Hummen et al. 2013] Hummen, R., Ziegeldorf, J. H., Shafagh, H., Raza, S., e Wehrle, K. (2013). Towards viable certificate-based authentication for the internet of things. In *Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy*, pages 37–42.

[IETF 2007] IETF (2007). Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals. RFC4919. `http://tools.ietf.org/html/rfc4919`.

[IoT-A 2009] IoT-A (2009). Introduction. `http://www.iot-a.eu/public`.

[ITU 2005] ITU (2005). Itu internet reports 2005: The internet of things.

[ITU 2009] ITU (2009). Ngn identity management framework. Recommendation Y.2720.

[Jara et al. 2011] Jara, A. J., Marin, L., Skarmeta, A. F., Singh, D., Bakul, G., e Kim, D. (2011). Secure mobility management scheme for 6lowpan id/locator split architecture. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, pages 310–315. IEEE.

[Jindou et al. 2012] Jindou, J., Xiaofeng, Q., e Cheng, C. (2012). Access control method for web of things based on role and sns. In *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on*, pages 316–321. IEEE.

[Juels 2006] Juels, A. (2006). Rfid security and privacy: a research survey. *Selected Areas in Communications, IEEE Journal on*, 24(2):381–394.

[Konidala et al. 2005] Konidala, D. M., Duc, D. N., Lee, D., e Kim, K. (2005). A capability-based privacy-preserving scheme for pervasive computing environments. In *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, pages 136–140. IEEE.

[Kothmayr et al. 2012] Kothmayr, T., Schmitt, C., Hu, W., Brunig, M., e Carle, G. (2012). A dtls based end-to-end security architecture for the internet of things with two-way authentication. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, pages 956–963. IEEE.

[Li et al. 2010] Li, N., Wang, Q., e Deng, Z. (2010). Authentication framework of iiedns based on ldap & kerberos. In *Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on*, pages 695–699. IEEE.

[Liu et al. 2012] Liu, J., Xiao, Y., e Chen, C. P. (2012). Authentication and access control in the internet of things. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 588–592. IEEE.

[Mahalle et al. 2010] Mahalle, P., Babar, S., Prasad, N. R., e Prasad, R. (2010). Identity management framework towards internet of things (iot): Roadmap and key challenges. In *Recent Trends in Network Security and Applications*, pages 430–439. Springer.

[Mahalle et al. 2012] Mahalle, P. N., Anggorojati, B., Prasad, N. R., e Prasad, R. (2012). Identity establishment and capability based access control (iecac) scheme for internet of things. In *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, pages 187–191. IEEE.

[Mahalle et al. 2013a] Mahalle, P. N., Anggorojati, B., Prasad, N. R., e Prasad, R. (2013a). Identity authentication and capability based access control (iacac) for the internet of things. *Journal of Cyber Security and Mobility*, 1(4):309–348.

[Mahalle et al. 2013b] Mahalle, P. N., Prasad, N. R., e Prasad, R. (2013b). Object classification based context management for identity management in internet of things. *International Journal of Computer Applications*, 63(12).

[Maler e Reed 2008] Maler, E. e Reed, D. (2008). The venn of identity: Options and issues in federated identity management. *IEEE Security & Privacy*, 6(2):16–23.

[Martinez et al. 2008] Martinez, D., Blanes, F., Simo, J., e Crespo, A. (2008). Wireless sensor and actuator networks: Charecterization and case study for confined spaces healthcare applications. In *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on*, pages 687–693.

[Miorandi et al. 2012] Miorandi, D., Sicari, S., De Pellegrini, F., e Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497–1516.

[Montenegro et al. 2007] Montenegro, G., Kushalnagar, N., Hui, J., e Culler, D. (2007). Rfc4944 - transmission of ipv6 packets over ieee 802.15.4 networks. `https://datatracker.ietf.org/doc/rfc4944/`.

[Moskowitz 2012] Moskowitz, R. (2012). Hip diet exchange (dex). `http://tools.ietf.org/html/draft-moskowitz-hip-dex-00`.

[Moskowitz et al. 2008] Moskowitz, R., Nikander, P., Jokela, E. P., e Henderson, T. (2008). Host identity protocol. `http://www.ietf.org/rfc/rfc5201.txt`.

[Nguyen et al. 2010] Nguyen, T.-D., Al-Saffar, A., e Huh, E.-N. (2010). A dynamic id-based authentication scheme. In *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*, pages 248–253. IEEE.

[Nogueira et al. 2011] Nogueira, M., Santos, A., Torres, J., Zanella, A., e Danielewicz, Y. (2011). Gerência de identidade na internet do futuro. In *Minicurso - SBRC 2011 - Campo Grande - MS*.

[OASIS 2003] OASIS (2003). A brief introduction to xacml. `https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html`.

[OASIS 2008] OASIS (2008). Security assertion markup language (saml) v2.0 - technical overview. `https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf`.

[OpenID 2007] OpenID (2007). Openid authentication 2.0 - final. `http://openid.net/specs/openid-authentication-2_0.html`.

[OPENIoT 2012] OPENIoT (2012). Eu ict open iot project. `http://openiot.eu/?q=node/1`.

[Prazeres e do Prado Filho 2013] Prazeres, C. V. S. e do Prado Filho, T. G. (2013). Gestão de identidade, autenticação e autorização na web das coisas - relatório técnico de acompanhamento. Technical report, Rede Nacional de Ensino e Pesquisa.

[Quisquater et al. 1989] Quisquater, J.-J., Guillou, L., Annick, M., e Berson, T. (1989). How to explain zero-knowledge protocols to your children. In *Proceedings on Advances in cryptology*, CRYPTO '89, pages 628–631, New York, NY, USA. Springer-Verlag New York, Inc.

[Recordon e Reed 2006] Recordon, D. e Reed, D. (2006). Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, DIM '06, pages 11–16, New York, NY, USA. ACM.

[Rescorla e Modadugu 2012] Rescorla, E. e Modadugu, N. (2012). Datagram transport layer security version 1.2. `http://tools.ietf.org/html/rfc6347`.

[Roman et al. 2011a] Roman, R., Lopez, J., e Najera, P. (2011a). A cross-layer approach for integrating security mechanisms in sensor networks architectures. *Wireless Communications and Mobile Computing*, 11:267–276.

[Roman et al. 2011b] Roman, R., Najera, P., e Lopez, J. (2011b). Securing the internet of things. *Computer*, 44(9):51–58.

[Rotondi et al. 2011] Rotondi, D., Seccia, C., e Piccione, S. (2011). Access control & iot: Capability based authorization access control system. In *1st IoT International Forum*.

[Saied e Olivereau 2012a] Saied, Y. B. e Olivereau, A. (2012a). D-hip: A distributed key exchange scheme for hip-based internet of things. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–7. IEEE.

[Saied e Olivereau 2012b] Saied, Y. B. e Olivereau, A. (2012b). Hip tiny exchange (tex): A distributed key exchange scheme for hip-based internet of things. In *Communications and Networking (ComNet), 2012 Third International Conference on*, pages 1–8. IEEE.

[Saint-Andre 2004] Saint-Andre, E. P. (2004). Extensible messaging and presence protocol (xmpp): Core. `http://www.ietf.org/rfc/rfc3920.txt`.

[Sakimura et al. 2013] Sakimura, N., Bradley, J., Jones, M. B., de Medeiros, B., e Mortimore, C. (2013). Openid connect basic client profile 1.0 - draft 28. `http://openid.net/specs/openid-connect-basic-1_0.html`.

[Santos et al. 2013] Santos, M. d. L., Domenech, M. C., e Wangham, M. S. (2013). Gestão de identidades na web das coisas: Um estudo de caso em saúde eletrônica. In *Segurança da Informação e de Sistemas Computacionais (SBSeg), 2013 13o Simpósio Brasileiro em*. SBC.

[Schaffers et al. 2011] Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., e Oliveira, A. (2011). Smart cities and the future internet: Towards cooperation frameworks for open innovation. In Domingue, J., Galis, A., Gavras, A., Zahariadis, T., Lambert, D., Cleary, F., Daras, P., Krco, S., Müller, H., Li, M.-S., Schaffers, H., Lotz, V., Alvarez, F., Stiller, B., Karnouskos, S., Avessta, S., e Nilsson, M., editors, *The Future Internet*, volume 6656 of *Lecture Notes in Computer Science*, pages 431–446. Springer Berlin Heidelberg.

[Seitz et al. 2013] Seitz, L., Selander, G., e Gehrmann, C. (2013). Authorization framework for the internet-of-things. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE 14th International Symposium and Workshops on a*, pages 1–6. IEEE.

[Silva et al. 2013] Silva, E. F., Fernandes, N. C., Rodriguez, N., Magalhaes, L. C. S., e Saade, D. C. M. (2013). Gestão de identidade em redes experimentais para a internet do futuro. In *Minicurso - SBRC2013 - Brasília - DF*.

[Wangham et al. 2010] Wangham, M. S., de Mello, E. R., da Silva Böger, D., Guerios, M., e da Silva Fraga, J. (2010). Gerenciamento de identidades federadas. In *Minicurso - SBSeg 2010 - Fortaleza - CE*.

[Xiang e Li 2012] Xiang, C. e Li, X. (2012). General analysis on architecture and key technologies about internet of things. In *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, pages 325–328.

[Xiaohui 2012] Xiaohui, X. (2012). Research on safety certification and control technology in internet of things. In *Computational and Information Sciences (ICCIS), 2012 Fourth International Conference on*, pages 518–521. IEEE.

[Yan et al. 2008] Yan, L., Zhang, Y., Yang, L., e Ning, H. (2008). *The Internet of Things: from RFID to the next-generation pervasive networked systems*. Auerbach Publications.

[Zeng et al. 2011] Zeng, D., Guo, S., e Cheng, Z. (2011). The web of things: A survey (invited paper). *Journal of Communications*, 6(6).

[Zhang e Liu 2011] Zhang, G. e Liu, J. (2011). A model of workflow-oriented attributed based access control. *International Journal of Computer Network and Information Security (IJCNIS)*, 3(1):47–53.

