

ADRIANO WITKOVSKI

**Um IdM e Método de Autenticação baseado em
chaves para prover autenticação única
em Internet das Coisas**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

CURITIBA

2015

ADRIANO WITKOVSKI

**Um IdM e Método de Autenticação baseado em
chaves para prover autenticação única
em Internet das Coisas**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: *Ciência da Computação*

Orientador: Prof. Dr. Altair O. Santin

CURITIBA

2015

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central

Witkovski, Adriano

W825i Um IdM e Método de Autenticação baseado em chaves para prover
2015 autenticação única em Internet das Coisas / Adriano Witkovski; orientador,
Altair O. Santin. – 2015.
66 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná,
Curitiba, 2015

Bibliografia: f. 62-66

1. Internet das Coisas. 2. Internet – Controle de acesso. 3. TCP/IP
(Protocolo de computação). 4. Informática. I. Santin, Altair Olivo. II. Pontifícia
Universidade Católica do Paraná. Programa de Pós- Graduação em
Informática. III. Título.

CDD 20. ed. – 004



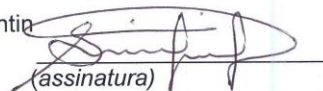
Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

DEFESA DE DISSERTAÇÃO DE MESTRADO Nº 06/2015

Aos 30 dias do mês de Setembro de 2015 realizou-se a sessão pública de Defesa da Dissertação “ **Um IDM e Método de Autenticação baseado em Chaves para prover Autenticação única em Internet das Coisas**” apresentado pelo aluno **Adriano Witkovski**, como requisito parcial para a obtenção do título de Mestre em Informática, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Altair Olivo Santin
PUCPR (Orientador)


(assinatura)

Aprov.
(Aprov/Reprov)

Prof. Dr. Raphael Machado
INMETRO


(assinatura)

Aprov.
(Aprov/Reprov)

Prof. Dr. Luiz Fernando Rust da Costa Carmo
INMETRO/UFRJ


(assinatura)

Aprov.
(Aprov/Reprov)

Conforme as normas regimentais do PPGIa e da PUCPR, o trabalho apresentado foi considerado Aprovado (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora. Este resultado está condicionado ao cumprimento integral das solicitações da Banca Examinadora registradas no Livro de Defesas do programa.



Prof.ª Dr.ª Andreia Malucelli.

Coordenadora do Programa de Pós-Graduação em Informática.



Dedico este trabalho à minha família que sempre me incentivou para continuar nesta caminhada e jamais desistir, em especial à minha esposa Patrícia, minha filha Maria Clara, minha Mãe Sônia e meu irmão Alisson.

Agradecimentos

A Deus

Pela vida, família, saúde e força para esta caminhada.

À minha família

Em especial à minha esposa Patrícia pelo seu companheirismo, incentivo, paciência e dedicação para assumir as responsabilidades da família enquanto eu estava ausente para os estudos e reuniões do Mestrado.

À minha mãe Sônia pelo incentivo e sabedoria, e também pelo exemplo de pessoa, mãe e amiga.

Ao professor Dr. Altair Olivo Santin

Agradecimento pela sua dedicação, postura, experiência e pelo tempo disponibilizado para construção deste trabalho. Agradecimento pela concessão da bolsa. Obrigado Santin.

Aos colegas do SecpLab

A todos os colegas do SecpLab pelo companheirismo em diversos momentos, contribuições e correções. Agradecimento especial ao colega Vilmar Abreu Jr. pela contribuição técnica para este trabalho e contribuição para o artigo. Ao colega João Eugenio Marynowski pela correção e contribuição do artigo. Ao colega Rafael C. Ribeiro pelo incentivo e companheirismo. Aos demais colegas Cleverton e Eduardo, muito obrigado.

Sumário

Agradecimentos	vii
Sumário	viii
Lista de Figuras.....	x
Lista de Tabelas	xi
Lista de Abreviaturas	xii
Resumo.....	xiii
Abstract	xiv
Capítulo 1	1
1.1 Introdução	1
1.2. Objetivo Geral	5
1.3. Objetivos Específicos.....	5
1.4. Contribuições.....	6
1.5. Estrutura do Documento.....	6
Capítulo 2	7
Fundamentação Teórica	7
2.1 Internet das Coisas	7
2.1.1 Características da IoT	9
2.1.2 Pilha de protocolos IoT.....	11
2.1.3 Segurança em IoT.....	12
2.2 Gerenciamento de Identidade	14
2.2.1 Sistemas de Gerenciamento de Identidade	15
2.2.2 Tipos de Sistemas de Gerenciamento de Identidades.....	16

2.2.3 Single Sing-on (SSO)	20
2.2.4 OpenID.....	21
2.3 Padrão ANSI X.9.17	22
Capítulo 3	25
Trabalhos Relacionados	25
3.1 IoT e Gestão de Identidade.....	25
3.2 Outros modos de autenticação em IoT.....	32
3.3 Resumo dos trabalhos relacionados	33
Capítulo 4	36
Proposta.....	36
4.1 Visão Geral.....	36
4.2 Fluxo de Mensagens	39
4.3 Protótipo	42
4.3.1 Implementação	42
4.3.2 Avaliação	44
Capítulo 5	48
Conclusão	48
Referências	50

Lista de Figuras

Figura 1 - Nova dimensão com a IoT - Adaptado de [6]	9
Figura 2 - Visões da IoT - Adaptado de [2]	10
Figura 3 - Comparativo da pilha de protocolos do TCP/IP com IoT - Adaptado de [41].....	12
Figura 4 - Modelo tradicional - Adaptado de [42].....	17
Figura 5 - Modelo Centralizado - Adaptado de [42]	18
Figura 6 - Modelo Federado - Adaptado de [42].....	19
Figura 7 - Modelo Centrado no Usuário - Adaptado de [42]	19
Figura 8 - Fluxo do <i>OpenID</i> - Adaptado de [43].....	22
Figura 9 - Nível de hierarquia de chaves da norma ANSI X9.17 - Adaptado de [28].....	23
Figura 10 - Um exemplo da arquitetura proposta - Adaptado de [10].....	26
Figura 11 - Procedimentos principais do IoT-OAS - Adaptado de [13].....	29
Figura 12 - Integração do SP com IoT-OAS - Adaptado de [13].....	30
Figura 13 - Visão Geral do IdM e esquema de autenticação baseado em chaves para prover SSO para IoT	37
Figura 14 - Fluxo de mensagens para uma requisição iniciado pelo Appliance.	40
Figura 15 - Autenticação e autorização de acesso para o técnico da fabricante.....	41
Figura 16 - Diagrama de sequência do procedimento de <i>Call Back</i>	42
Figura 17 - Arquitetura do protótipo.....	43
Figura 18 - Avaliação do protótipo.....	45
Figura 19 - Comparativo de DTLS com e sem SSO para 50 App.....	46
Figura 20 - Requisições paralelas vs sequenciais	47
Figura 21 - Limite de apps suportados na solução proposta	47

Lista de Tabelas

Tabela 1 - Resumo dos trabalhos relacionados	34
Tabela 2 - Comparação dos trabalhos relacionados com a proposta	35

Lista de Abreviaturas

6LowPAN	IPv6 Over Low Power Wireless Personal Area Networks
App	Appliance
AppTec	Appliance Technician
AS	Authorization Server
ASS	Access Authorization Server
COAP	Constrained Application Protocol
CS	Customer Service
DTLS	Datagram Transport Layer Security
GW	Gateway
IdM	Identity Management
IdP	Identity Provider
IoT	Internet of Things
KD	Data Key
KEK	Key Encrypting Key
KKM	Master Key Encrypting Key
MTU	Maximum Transmission Unit
REST	Representational State Transfer
SP	Service Provider
SSO	Single Sign-On
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URI	Uniform Resource Identifier

Resumo

A Internet das Coisas (IoT) traz desafios significativos para esquemas de autenticação em cenários com múltiplos *appliances* em uma *Smart House*, que devem ser acessados por um técnico para tarefas de manutenção, por exemplo. O Gerenciamento de Identidades (IdM) pode ser aplicado para autenticar um técnico que pretende acessar os *appliances* a partir da Internet. No entanto, o contexto Internet é significativamente diferente da IoT, exigindo adaptação do contexto para trabalhar corretamente. Assim, integrar estes contextos para permitir a autenticação na Internet e fornecer *Single Sign-On* (SSO) em IoT é um desafio. O objetivo é permitir que um técnico possa acessar um *appliance*, que não é acessível diretamente a partir da Internet, usando um IdM e sem que haja um ponto único de vulnerabilidade no gateway que interliga os dois contextos. A proposta interage com dois esquemas baseados em chave, um para a Internet e outro para a Internet das Coisas, para alcançar a integração entre ambos os contextos. O protótipo desenvolvido e a prova de conceito mostram que a proposta é viável e não apresenta um *overhead* significativo para mensagens com tamanho máximo de 4096 bytes e com 50 *appliances*.

Palavras-Chave: Internet das Coisas, Gerenciamento de Identidade, Autenticação baseada em chave, Single Sign-on, Autorização de Acesso, Smart House.

Abstract

The Internet of Things (IoT) brings significant challenges to authentication schemes in a scenario with several appliances for a smart house that should be accessed by a technician for maintenance tasks, for instance. An Identity Management (IdM) can be applied to easily authenticate a technician that intend to access the appliances from the Internet. However, Internet context is significantly different from IoT, demanding context adaptation to work. Thus, integrate these contexts to allow the authentication on the Internet and provide Single Sign-On (SSO) in IoT is a challenge. The goal is to allow a technician to access an appliance that is not reachable from the Internet, using IdM and without creating a single compromising point in the gateway that links the two contexts. The proposal interacts two key-based scheme, one for Internet and another for IoT, to reach integration between both contexts. A proof-of-concept implementation shows the proposal is feasible and presents no significant overhead for messages with up to 4096 bytes and 50 appliances.

Keywords: Internet of Things; Identity Management; Key based Authentication; Single Sign-On; Smart House.

Capítulo 1

Introdução

A Internet tradicional é uma das mais importantes criações para a humanidade e tem trazido inúmeros benefícios nos últimos anos, o próximo passo da revolução tecnológica segue voltada para a Internet das Coisas - *Internet of Things* (IoT). O principal objetivo da IoT é interligar diversas “coisas” ou *appliances*, conectando-os com a Internet para prover diversos tipos de serviços, como monitoramento, gerenciamento e automação [1].

O termo *Internet of Things* - IoT foi cunhado em 1999 por um pesquisador do laboratório AUTO-ID do MIT (*Massachusetts Institute of Technology*) chamado Kevin Ashton. Kevin apontou que em um futuro próximo as coisas poderiam ser capazes de gerar e coletar dados de forma autônoma, sem a necessidade de intervenção humana. Na IoT as coisas geralmente possuem restrição de recursos computacionais como processamento, memória, energia e comunicação (baixa largura de banda). Dessa forma, inserir mecanismos de segurança em dispositivos embarcados com recursos computacionais limitados, pode representar um grande desafio [2].

Nesse contexto, os objetos podem ser passíveis de problemas de segurança como confidencialidade, autenticidade e integridade dos dados trocados entre si ou com usuários. Assim, de acordo com Babar e colaboradores [2], aspectos de segurança são identificados e apresentados como obstáculos a serem observados no processo de amadurecimento da Internet das Coisas, e alguns requisitos de segurança necessitam ser garantidos, sendo: comunicação segura dos dados, acesso seguro à rede, resistência à violação física e Gestão de Identidades.

Um cenário que explora o mundo da IoT é o de uma *Smart House*, na qual diversos *appliances* (e.g. eletrodomésticos) podem ser acessados e configurados para monitorar e auxiliar aspectos da vida diária dos residentes.

Dessa forma, a empresa fabricante dos *appliances* tem a necessidade de acessar remotamente os dispositivos que vendeu e que estão localizados na *Smart House*. A fabricante necessita do acesso para efetuar manutenção (monitoramento, otimizações) ou reparos, ou também para efetuar atualização de *firmware*¹ do *appliance*, corrigindo eventuais vulnerabilidades de segurança.

Embora o fabricante tem a necessidade de acessar remotamente os *appliances* localizados na *Smart House*, é interessante, do ponto de vista de segurança, que os *appliances* não estejam expostos para acesso livre a partir da Internet. É sabido que o IPv6, devido ao grande número de endereços disponíveis, permite que qualquer dispositivo possa ser endereçado de forma única na Internet. Portanto, gerir a identidade dos técnicos do fabricante que acessam os *appliances* localizado na *Smart House* é imprescindível para manter a segurança de acesso aos *appliances*.

A autenticação e autorização de acesso são desafios importantes para a IoT, pois diferentemente dos componentes da Internet tradicional, os *appliances* estão baseados em dispositivos de propósito específico, em geral, com restrição de recursos.

Um sistema de Gestão de Identidade - *Identity Management* (IdM) - provê a autenticação e autorização de acesso na Internet [4]. Entretanto, integrar um IdM da Internet com a IoT não é trivial, devido à restrição de recursos dos *appliances* e da falta de garantia de segurança de um canal de comunicação entre os dois mundos. Assim, desenvolver uma solução que integre autenticação e autorização de acesso do fabricante na Internet com os *appliances* da IoT de maneira segura e fazendo uso das tecnologias atuais é um desafio importante.

As abordagens apresentadas para autenticação e autorização de acesso em IoT normalmente utilizam chave única em todos os *appliances*, e também um serviço de autenticação [2, 3]. A utilização de uma chave única tem como vantagem a redução de recursos, porém, é inviável devido à possibilidade de a chave ser descoberta, além da falta de controle de qual usuário está acessando um *appliance*, e a dificuldade da manutenção da chave em todos os *appliances* [3].

¹ Firmware is a software embedded in an appliance to operate it, and it can offer an interface with external environment.

A utilização de um serviço de autenticação possibilita a autorização de acesso que resolve as deficiências da chave única. As abordagens existentes se limitam à autenticação e autorização de acesso localizado na Internet ou exclusivamente na IoT. Algumas propostas buscam integrar os dois mundos, mas não apresentam implementação e não consideram um canal de comunicação seguro fim-a-fim [11, 12, 13]. Além do mais, nenhuma proposta aborda a autenticação única, *Single Sign-On* (SSO), para acesso temporário a diversos *appliances* a partir de uma única autenticação de um técnico, por exemplo.

A proposta traz como subproduto (*byproduct*) a autenticação única (SSO), herdada de IdM, mas provida para IoT com base no esquema de chave simétrica partilhada entre o fabricante e o *appliance*. Este mecanismo funciona bem porque o fabricante e o gateway (interface com o *appliance*) partilham o mesmo servidor de autenticação (IdP) e autorização de acesso, o *appliance* e a fabricante partilham o mesmo servidor de chaves simétricas, dessa forma, é possível fazer a integração dos dois universos sem exigir que senhas adicionais para acessar o IdM sejam necessárias. O esquema proposto é baseado em padrões, e o protótipo utiliza tecnologias consolidadas da Internet e protocolos da pilha da IoT.

O protótipo desenvolvido, juntamente com os resultados dos testes obtidos, mostra que a proposta é viável e não apresenta um *overhead* significativo para mensagens com um tamanho máximo de 4096 bytes e 50 *appliances*.

1.1. Motivação

A IoT possui ubiquidade e interconexão de múltiplas coisas que podem se conectar a internet em qualquer lugar, a qualquer tempo, e de diferentes modos. Uma contribuição importante para a Internet das Coisas é o protocolo IPv6, devido à quantidade de endereços disponível, algo em torno de 2^{128} , muito superior ao disponível no IPv4, cujo espaço de endereçamento é 2^{32} , possibilitando que cada *appliance* seja endereçável na Internet.

Com esta facilidade muitos fabricantes podem monitorar, coletar informações dos dispositivos e oferecer serviços de manutenção e reparos remotos, etc. Por exemplo, uma máquina de lavar inteligente conectada à Internet pode submeter os dados de rotação e de funcionamento para um servidor que armazena estatísticas no domínio do fabricante. Estes dados podem ser processados e analisados, possibilitando que a empresa fabricante forneça um

serviço pró-ativo de substituição de peças em função do envelhecimento e desgaste do produto, além do ajuste de parâmetros de funcionamento das partes mecânicas do *appliance*.

Por outro lado, apesar de os *appliances* da *Smart House* estarem endereçáveis na Internet, é evidente que por questões de segurança esses não devem estar disponíveis para acesso sem restrições, uma vez que podem sofrer ataques e acessos indevidos provenientes da Internet. Segundo Chadwick [30], gestão de Identidade, comunicação segura, e resistência à violação de dispositivos são requisitos no contexto de segurança que devem ser observados na IoT.

Para que o fabricante possa acessar os *appliances* é importante que haja autenticação para saber quem está acessando o dispositivo e se detém permissão suficiente para tal acesso. Um dos grandes desafios é integrar um IdM da Internet com os dispositivos com baixa capacidade de recursos dentro da *Smart House* e com pilha de protocolos diferente. Além disso, é necessário proporcionar comunicação segura fim-a-fim em todo o processo de assistência do técnico da fabricante.

Devido à restrição da capacidade computacional, os dispositivos da IoT não suportam os protocolos tradicionais como HTTP e SSL, utilizados largamente na Internet. Por isso, foram desenvolvidos protocolos como CoAP e DTLS para atender os requisitos da IoT [7, 8]. Assim, temos o mundo da Internet com seus protocolos bem definidos, e do outro lado do dispositivo (*appliance*) temos o mundo da IoT. Para interligar esses mundos de Internet e IoT e proteger os *appliances* na *Smart House*, faz-se necessário o uso de gateway para intermediar a comunicação.

A utilização do gateway traz como necessidade a execução de um *parser* para que os protocolos HTTP no contexto da Internet, e o CoAP no contexto da IoT, possam interagir e trocar dados. Ainda assim, é imprescindível que o elemento intermediador não se apresente como um elemento crítico (com relação a falhas e comprometimento) na solução. Dessa forma, é necessário um mecanismo que promova a integração entre os dois mundos.

Além disso, com múltiplos *appliances*, existe a necessidade de o técnico da fabricante se autenticar em vários *appliances* para efetuar os ajustes, correções e alteração de *firmware*, por exemplo. Esta autenticação em larga escala é um problema para o fabricante, dificultando a gestão, comprometendo a segurança devido ao grande número de senhas, além de tornar o processo demorado para atuação do técnico.

Na literatura, trabalhos tratam de autenticação em IoT, no entanto, a maioria considera apenas o contexto da IoT, e em outros trabalhos assume-se que o gateway é um elemento confiável na solução. Outras implementações não consideram segurança fim-a-fim em todo o processo. Por fim, o recurso de SSO é praticamente inexplorado na literatura.

Este trabalho considera a hipótese de que é possível integrar um sistema de IdM com IoT, levando em conta a restrição de recursos, sem comprometer o desempenho e a segurança do gateway (elemento de ligação entre a Internet e IoT).

1.2. Objetivo Geral

O objetivo geral deste trabalho é a criação de um esquema de segurança baseado em chave simétrica capaz de prover segurança fim-a-fim, envolvendo os *appliances* de Internet das Coisas. Este esquema é projetado para a integração de um IdM proveniente da Internet com o mundo da IoT, levando em consideração a restrição de recursos dos *appliances* ou dispositivos da IoT. Outro requisito é que o mecanismo de integração (gateway) seja robusto e não se apresente como um elemento crítico da solução, ou seja, permitindo que a comunicação ocorra de modo cifrado entre os pares, impedindo a interceptação e manipulação do conteúdo das mensagens.

Como complemento, o esquema visa contemplar e prover um mecanismo de autenticação única (SSO) para que o técnico do fabricante se autentique em múltiplos *appliances* localizados nas *Smart Houses* e os manipule com segurança.

1.3. Objetivos Específicos

- a) Implementação de IdM no ambiente de Internet;
- b) Implementação do mecanismo de integração (gateway) para troca de informações entre os mundos de Internet e IoT;
- c) Estudo e validação dos protocolos utilizados pela IoT;
- d) Customização do *parser* para tradução de HTTP para CoAP e vice-versa;
- e) Implementação do esquema de segurança baseado em chave simétrica entre o Gateway e os *appliances*;
- f) Construção de modelo de autenticação única (SSO) para IoT.

- g) Definição de um cenário de avaliação e implementação de um protótipo.
- h) Testes e validação do esquema proposto;
- i) Avaliação dos resultados.

1.4. Contribuições

O trabalho desenvolvido contribui para a segurança da IoT, fornecendo a integração de um IdM da Internet com a IoT, juntamente com um esquema de segurança baseado em chaves simétricas que utiliza padrões de protocolos de Internet e IoT. Este esquema é capaz de operar de forma integrada entre os mundos de Internet e IoT. A comunicação é segura, opera de modo fim-a-fim, fornece proteção por mensagem (*per-message*) no trânsito entre o *appliance* e o gateway, e entre este e o site do fabricante, sem que o elemento integrador se torne um elemento vulnerável no esquema.

Outro ponto de contribuição relevante é o uso do SSO, provendo autenticação única para que o técnico do fabricante se autentique uma vez por dia e possa acessar múltiplos *appliances*, facilitando a administração e aplicando as devidas correções, ajustes e atualizações.

Na literatura, nenhuma abordagem apresenta um esquema que contemple a integração entre os mundos de Internet e IoT e que promova segurança fim-a-fim no trânsito das mensagens, além do benefício da autenticação única (SSO) herdada do IdM.

1.5. Estrutura do Documento

O capítulo 2 deste documento apresenta os fundamentos teóricos necessários para o entendimento do trabalho. O capítulo 3 apresenta os trabalhos relacionados com o assunto da pesquisa. O capítulo 4 apresenta a proposta deste trabalho juntamente com os resultados obtidos. Por fim, no capítulo 5 é apresentada a conclusão do estudo realizado.

Capítulo 2

Fundamentação Teórica

Nesta seção serão apresentados uma introdução sobre a IoT e suas principais características, a pilha de protocolos desenvolvida para IoT, os problemas de segurança relacionados, além de uma fundamentação sobre Gestão de Identidade e uma breve fundamentação do padrão X.9.17.

2.1 Internet das Coisas

O termo Internet das Coisas direciona para uma visão de máquinas, dispositivos e coisas conectados à Internet. No século XXI as máquinas aprenderão a perceber através de sensores, isto é, monitorar e apresentar uma sensibilidade em tempo real do ambiente físico para, conseqüentemente, reagirem de alguma forma [26]. Desse modo, com a visibilidade da IoT, aumentado à capacidade de capturar e distribuir dados, será possível transformá-los em informações, conhecimento e posteriormente em sabedoria [4].

A ideia básica da IoT está na relação de vários objetos, denominados “coisas”, que cooperam e interagem entre si na tentativa de alcançar um objetivo comum, para isso, utilizam esquema de endereçamento único e padrões de protocolos [2]. Dessa forma, com o avanço da nanotecnologia e a integração de sensores e atuadores, as coisas equipadas com sensores são capazes de capturar dados e o estado de coisas do mundo físico, como por exemplo, dados de umidade, pressão, iluminação, vibração, temperatura, localização [2]. Estes objetos podem variar desde simples sensores, atuadores, roupas, smartphones até dispositivos inteligentes, como uma máquina de lavar, por exemplo.

Com a sensibilidade de obter dados do mundo físico, os objetos que antes não geravam dados e passavam despercebidas pelo ser humano, agora, poderão ser monitorados e proporcionarão dados que levarão ao conhecimento de novas descobertas [3].

O potencial ofertado pela IoT possibilita o desenvolvimento de inúmeras aplicações que estarão presentes em nossa sociedade [2]. Na área da saúde, os remédios poderão ser personalizados, contendo na embalagem uma receita eletrônica capaz de garantir a dosagem correta e alertar ao paciente o horário de ingerir o remédio. Do mesmo modo, o médico terá dados para checar possíveis incompatibilidades dos remédios receitados com os novos que poderão ser receitados.

Ainda, a vida assistida poderá melhorar a qualidade de vida dos idosos, reduzindo os custos com monitoramento. Possíveis quedas poderão ser identificadas em ambientes onde pessoas idosas residem sozinhas (movimentos bruscos ou anormais obtidos por um acelerômetro), além do monitoramento de sinais vitais e alerta sobre a validade de alimentos. Outro exemplo é a vida assistida de bebês, em que a própria roupa poderá possuir sensor de temperatura capaz de monitorar sua febre, além da capacidade de enviar dados para os pais através de uma pulseira com conexão *bluetooth*.

Outra área que explora o mundo da IoT é o de uma *Smart House*, o termo geralmente é aplicado para uma casa inteligente que possui dispositivos conectados à Internet. Nestes ambientes automatizados existem diversos sensores, tais como sensor de iluminação, de ar-condicionado, sensores de presença, câmeras de vídeos e central de controle. Estes dispositivos são integrados e podem ser acessados a partir de qualquer divisão da casa, permitindo monitorar o ambiente interno, e capazes de sugerir redução dos custos de energia com o desligamento de aparelhos onde não há presença de pessoas, além de auxiliar nos aspectos da vida diária dos residentes da casa.

De acordo com Atzori e colaboradores [1], são inúmeras as aplicações que estarão disponíveis para a sociedade, proporcionando uma melhoria no cotidiano. O seu potencial pode ser explorado em diversas áreas:

- Transporte e logística: aplicações relacionadas ao transporte inteligente em que os produtos em transporte são monitorados em tempo real a fim de melhorar a gestão;
- Saúde: no domínio da saúde os pacientes podem receber monitoramento em tempo real, a partir de indicadores de saúde capturados por sensores;

- Ambiente inteligente: aplicações que proporcionem melhorias no ambiente de casas, escritórios, fábricas, através da atuação de sensores distribuídos nos ambientes, possibilitando a personalização, economia e conforto. Ex: aquecimento adaptado às preferências e ao clima;
- Pessoal e social: permitir que usuário seja capaz de construir relações e interagir com outras pessoas, ou seja, coisas podem gerar mensagens automaticamente para notificar os amigos sobre o que se está fazendo;
- Aplicações Futuristas: desenvolvimento de aplicações com visões futuristas, a fim de alavancar discussões, como por exemplo, robô taxi;

2.1.1 Características da IoT

O desenvolvimento e a incorporação de sensores de curto alcance, juntamente com inúmeros dispositivos e itens do cotidiano possibilitam a comunicação entre pessoas e coisas ou entre as próprias coisas [27]. Com isso, uma nova dimensão é visualizada, uma coisa ou dispositivo poderá obter conexão em qualquer tempo (quando), em qualquer lugar (onde) e com qualquer coisa (como), poderá fazer um intercâmbio de dados em tempo real, conforme apresentado na Figura 1.

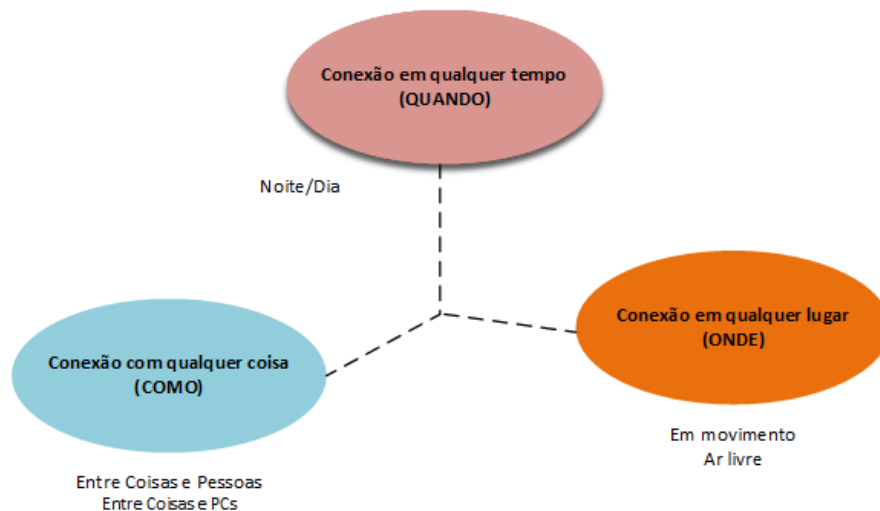


Figura 1 - Nova dimensão com a IoT - Adaptado de [6]

Como característica da IoT, também apresenta-se um novo paradigma com diferentes visões, conforme apresenta Figura 2.

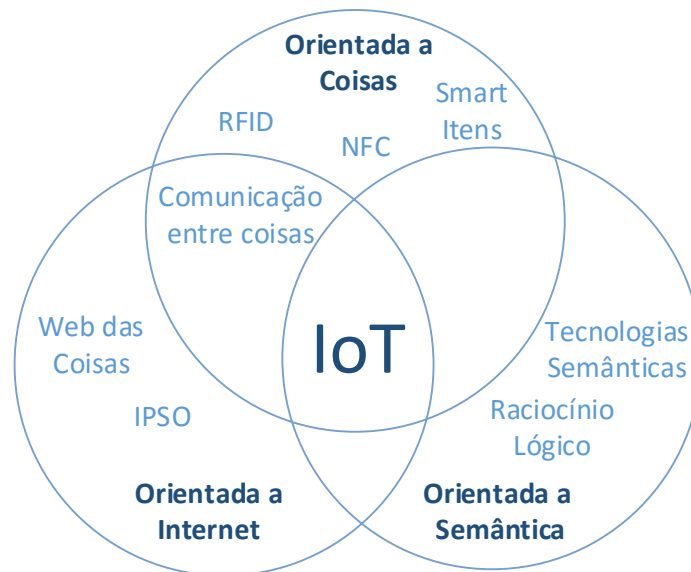


Figura 2 - Visões da IoT - Adaptado de [2]

Dessa forma, temos uma breve descrição das visões:

- **Orientada as Coisas:** Está relacionada com a identificação de coisas simples com endereçamento único, utiliza tecnologias como RFID, NFC;
- **Orientada a Internet:** Intimamente ligada com os protocolos e suas adaptações a fim de permitir a troca de dados entre as coisas e a Internet;
- **Orientada a Semântica:** Com uma grande quantidade de dados que está por vir, justifica-se a criação de modelos de coisas, desde a obtenção e processamento dados da IoT até desenvolvendo ambientes semânticos;

Além disso, a IoT é composta por um ecossistema que dispõe de dispositivos com recursos computacionais restritos. Este tipo de dispositivo geralmente tem capacidade limitada de processamento, memória (RAM e ROM), armazenamento e energia. Outras características são a baixa potência de transmissão e baixa taxa de dados (*throughput*).

Os dispositivos podem possuir um ciclo de vida curto, ou seja, coisas podem entrar e logo desaparecerem da rede, pois sensores são desativados e podem apresentar falhas de operação, esse fato promove estudos de modelos de gestão de coisas. Como consequência disso, a IoT pode estar associada a uma topologia dinâmica em determinados ambientes, visto que os ciclos de vida dos dispositivos são curtos e uma topologia ágil e incerta é apresentada.

A heterogeneidade também é apresentada como um aspecto da IoT, diversos dispositivos geram dados e realizam intercâmbio de dados, sendo necessária uma atenção dos fabricantes em relação a interoperabilidade dos dispositivos e protocolos. Por fim, o ambiente ubíquo e pervasivo, de acordo com as visões da IoT, proporciona um ambiente volumoso de computadores e/ou dispositivos. Todas as características mencionadas contribuíram para o desenvolvimento de uma pilha de protocolos.

2.1.2 Pilha de protocolos IoT

A comunicação na IoT é realizada a partir da utilização de uma pilha de protocolos adequada à restrição de recursos dos *appliances*. O padrão IEEE 802.15.4 é utilizado na camada física e enlace, pois permite a comunicação sem fio com baixo consumo de energia, embora apresente curto alcance e baixas taxas de transmissão (*throughput*) [6].

O protocolo 6LowPAN é usado na camada de rede, pois aplica mecanismos de compressão e encapsulamento e permite que pacotes da Internet (IPv6) sejam recebidos e enviados a partir do IEEE 802.15.4 [6], ou seja, o 6LowPAN se apresenta como uma camada de adaptação entre a camada de rede e a camada de enlace. Tal adaptação é necessária, motivada pelo valor mínimo do MTU suportado pelo protocolo IPv6 que é de 1280 bytes. No entanto, para dispositivos que atuam com o protocolo IEEE 802.15.4, a camada de enlace suporta apenas 127 bytes, devido ao MTU da camada de enlace [6].

A camada de transporte utiliza o protocolo UDP justamente pela necessidade de obter um protocolo mais simples e enxuto em relação ao *overhead* do TCP. O algoritmo de controle de congestionamento e o *Three-Way Handshake* podem apresentar excessos desnecessários para dispositivos da IoT em que processamento, bateria e baixa largura de banda são restritos, sendo assim, utilizar o protocolo TCP pode ser considerado custoso em relação ao protocolo UDP [33].

O protocolo HTTP foi projetado principalmente para operar com o protocolo TCP, sendo ineficiente para dispositivos com restrições. Para a camada de aplicação da IoT o IETF vem trabalhando no desenvolvimento de um protocolo para a comunicação dos *appliances*, denominado protocolo *Constrained Application Protocol* (CoAP). O CoAP é baseado na arquitetura *Representational State Transfer* (REST), e projetado para atender às necessidades específicas, tais como simplicidade, baixo custo operacional e suporte a *multicast*, além disso, os recursos controlados por um servidor são identificados e acessados por meio de

identificadores de recursos universais (*Uniform Resource Identifier - URI*) [7]. O CoAP utiliza o UDP na camada de transporte.

Para obter serviços de segurança o CoAP propõe o uso do protocolo *Datagram Transport Layer Security* (DTLS), um protocolo baseado no TLS capaz de oferecer segurança equivalente, fornecendo serviços que garantam integridade, autenticação e confidencialidade, no entanto, é capaz de utilizar como transporte o UDP, oportuno para dispositivos com poucos recursos [8]. Em comparação ao protocolo HTTP protegido por TLS, denominado HTTPS, a integração entre CoAP e DTLS é também denominado CoAPs. Dessa forma, um *appliance* na IoT permite ser acessado através de um canal seguro como: *coaps://ip_appliances:porta/temperatura*.

Por fim, a figura 3, apresenta um comparativo da pilha TCP/IP com a pilha desenvolvida para ser utilizada na Internet das Coisas.

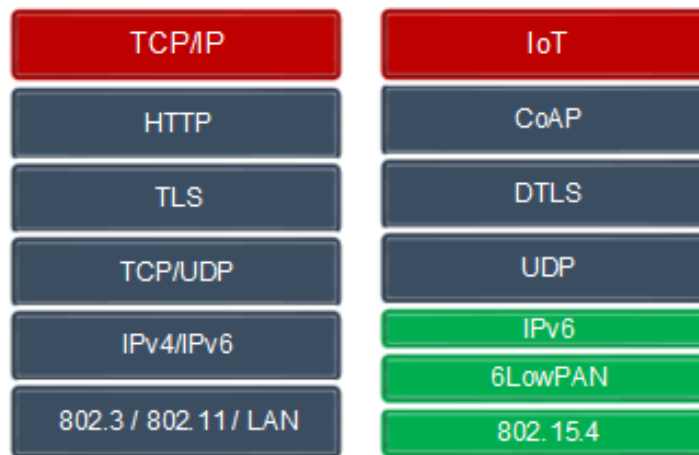


Figura 3 - Comparativo da pilha de protocolos do TCP/IP com IoT - Adaptado de [41]

2.1.3 Segurança em IoT

Uma das maiores preocupações dos pesquisadores com o surgimento da Internet das Coisas é com a segurança, devido a sua ampla utilização em várias áreas como saúde, vida assistida, *Smart House* e objetos pessoais. Dessa forma, a IoT deve seguir alguns requisitos de segurança da informação [2], tais como:

- **Confidencialidade:** assegurar a confidencialidade dos dados;
- **Integridade:** impedir que dados transmitidos sejam adulterados;

- **Disponibilidade:** proporcionar a continuidade dos serviços oferecidos pela IoT;
- **Autenticidade:** promover a autenticação mútua, tanto para os usuários quanto para o provedor de recurso;
- **Privacidade:** prover meio para garantir a privacidade dos dados;
- **Irrefutabilidade:** prover um meio para que os envolvidos em uma transação possam provar o que aconteceu;

Além destes requisitos comumente utilizados na tecnologia, são considerados outros requisitos que necessitam de garantia para o bom funcionamento na Internet das Coisas. A Gestão de Identidades é fundamental para a identificação e autenticação dos usuários, dispositivos ou *appliances*, objetos no ecossistema da IoT, uma vez que diversos dispositivos estarão conectados em qualquer lugar, qualquer tempo e com qualquer dispositivo.

A comunicação segura também deve ser observada dependendo do tipo de aplicação, informações médicas podem apresentar problemas de privacidade sendo necessária uma comunicação cifrada, sem permitir a visibilidade dos dados em texto claro. O acesso seguro e autorizado à rede também deve ser observado através de mecanismos para garantir a possibilidade de conexão na rede ou o acesso a um serviço apenas para dispositivos autorizados.

Por fim, é razoável que os dispositivos possam se auto recuperar de possíveis violações, assegurando que eles continuem trabalhando mesmo que seja necessário resistir alguns ataques.

Em relação aos ataques que estão envolvidos no contexto de Internet das Coisas, podemos citar [2]:

- **Ataques físicos:** violações de hardware do dispositivo, mesmo sendo considerado um ataque caro, pelo fato de exigir outros mecanismos além de software;
- **Ataques no meio da comunicação:** ataques realizados a partir de dados recolhidos ou recuperados de dispositivos responsáveis por operações criptográficas. Dados obtidos através de análise de temporização, potência consumida;
- **Ataques de análise de criptografia:** ataques de *Man-in-the-Middle*, procurando encontrar a chave ou meios para obter os dados em texto claro;

- **Ataques de rede:** Ataques no meio sem fio, tais como negação de serviço e ataques de roteamento;
- **Ataques de software:** baseados em exploração de vulnerabilidades encontradas em softwares, incluindo *buffer overflow* e vírus, *worms*, a fim de inserir códigos maliciosos no sistema;

Na próxima seção será apresentada a fundamentação sobre a gestão de identidade.

2.2 Gerenciamento de Identidade

O termo Gestão de Identidade é amplo, e no contexto de Tecnologia da Informação está relacionado ao estabelecimento de papéis e privilégio dentro de rede, possibilitando a administração dos acessos dos seus subordinados e parceiros. De acordo com Chadwick [30], a Gestão de Identidades (*Identity Management - IdM*) é um conjunto de serviços ou funções com objetivo de administrar, descobrir e trocar informações, a fim de garantir a consistência das identidades, possibilitando transações seguras.

Gerenciar a identidade é vital para proteger a privacidade do usuário, contribuir para sua experiência e para suportar as transações e interações, de acordo com os controles estabelecidos entre os envolvidos na comunicação.

Para representar digitalmente no mundo virtual, utiliza-se uma identidade que pode ser associada às informações conhecidas como um objeto, pessoa ou uma organização. Essas informações são utilizadas para que as pessoas possam atestar a sua identidade, como por exemplo, um documento de passaporte. Além disso, uma identidade pode conter atributos complementares como características pessoais, impressões digitais e biométricas. Usualmente, a identidade virtual também pode compreender identificadores como nomes de usuário e apelidos, utilizados para acessar aplicações e sistemas ou interagir com outros objetos representados digitalmente [4].

Dessa forma, o gerenciamento de identidade aparece com uma solução de sistema integrado de políticas, tecnologias e processos de negócios, que possibilitam às entidades tratar e manipular os atributos de identidades dos usuários envolvidos interagindo entre si [31]. Além disso, o gerenciamento de identidades também está relacionado com o gerenciamento das identidades digitais, definição, certificação e infraestrutura para trocar e validar informações

[31].

2.2.1 Sistemas de Gerenciamento de Identidade

Um sistema de Gerenciamento de Identidades fornece ferramentas para o gerenciamento de identidades no mundo digital [4]. De acordo com Chadwick [30], o gerenciamento de identidades é formado por funções de administração, descoberta e intercâmbio de informações, a fim de garantir a identidade e seus atributos, fornecendo artifícios para que uma comunicação comercial seja segura, por exemplo. Fazendo um paralelo com o mundo digital, onde as pessoas revelam suas informações para outras conforme o contexto ou interesse, esta tarefa é de responsabilidade do sistema de gerenciamento de identidade.

Na sociedade e em nosso cotidiano encontramos diariamente sistemas de gerenciamento de identidade, como por exemplo, a carteira de estudante. Um usuário de posse de uma carteira pode entrar na universidade, emprestar livros na biblioteca e receber os benefícios provisionados para sua identidade. No contexto empresarial, os sistemas são importantes para administrar um volume alto de usuários e as tarefas vinculadas à criação ou exclusão de identidades, modificação de atributos e quesitos relacionados às senhas de acesso, como redefinição de senhas.

Um sistema de gerenciamento de identidade é composto pelos seguintes elementos:

- **Usuário:** indivíduo que acessa o sistema ou serviço.
- **Identidade:** conjunto de atributos pertencentes a um usuário. Ex: nome, CPF, naturalidade.
- **Provedor de Identidades (Identity Provider - IdP):** gerencia as identidades de usuários e emite credenciais.
- **Provedor de Serviços (Service Provider - SP):** oferece recursos aos usuários autorizados de acordo com a identidade e os seus atributos.

Além disso, visando garantir uma experiência interessante para o usuário do sistema de gerenciamento de identidade, foram propostos e descritos alguns requisitos para o sistema [32].

- **Suporte para Anonimato:** Possuir uma infraestrutura para garantir que o usuário tenha assegurado o direito de ficar no anonimato.
- **Privacidade:** O usuário necessita de poder para decidir quais informações serão reveladas em sua identidade.
- **Interoperabilidade:** Sistema deve apresentar mecanismos para que as identidades possam ser utilizadas em diferentes domínios.
- **Gerenciamento de confiança:** Provedores de serviços e de identidade devem possuir uma relação de confiança, a fim de que as credenciais fornecidas em diferentes domínios sejam aceitas.
- **Revogação de identidades:** Possibilidade e opções para que o usuário possa revogar informações apresentadas em sua identidade.

2.2.2 Tipos de Sistemas de Gerenciamento de Identidades

Os sistemas de gerenciamento de identidades podem ser classificados de acordo com alguns modelos, sendo: tradicional, centralizado, centrado no usuário e federado.

a) Modelo Tradicional ou isolado

Este modelo é o mais utilizado na Internet e nos sistemas computacionais. Através deste, a identificação do usuário é realizada de forma isolada para cada provedor de serviços que também terá a função de provedor de identidade. A partir desta forma de trabalho, os usuários deverão criar uma identidade digital para cada provedor de serviços que necessita interagir, pois neste caso, não há compartilhamento das identidades dos usuários nos provedores de serviço.

Mesmo sendo amplamente utilizado, o modelo possui desvantagens, pois apresenta custos adicionais tanto no provedor de serviço como para os usuários, uma vez que o usuário necessita criar várias identidades para usufruir dos serviços dos portais, e-mails, comércio de eletrônicos. Existindo essa independência, existe uma vantagem, cada provedor de serviços tem condições de criar o conjunto de atributos que lhe achar conveniente para caracterizar a identidade do indivíduo.

Outro problema relacionado com o modelo é que os usuários são responsáveis por gerenciar várias identidades e que isso pode se tornar algo custoso, visto que se faz necessário

apresentar as informações repetidas vezes e também pela preocupação em criar usuário e senha diferente para cada provedor de serviços. Neste caso, usuários tendem a ser displicentes na forma de preenchimento de atributos que não são cruciais para o negócio em questão. A figura 4 mostra o modelo tradicional.

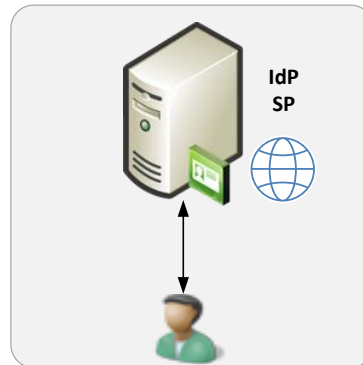


Figura 4 - Modelo tradicional - Adaptado de [42]

b) Modelo Centralizado

O modelo centralizado nasceu na tentativa de apresentar uma flexibilidade maior no modelo tradicional, tendo como principal ideia o compartilhamento de identidade dos usuários entre os provedores de serviços envolvidos e também no conceito de autenticação única SSO (*Single Sign-on*) [29]. A ideia de autenticação única proporciona que o processo de autenticação ocorra uma única vez, onde o usuário poderá utilizá-la em todos os provedores de serviços até que suas credencias expirem.

A plataforma Microsoft Passport foi o sistema precursor para tentar evitar inconsistências e duplicidade de informações existentes no modelo tradicional e para apresentar uma experiência melhor para o usuário [29], contribuindo para que os indivíduos que interagem com o sistema não mais necessitassem de um processo manual e repetitivo de fornecimento da identidade em cada provedor de serviços. Neste modelo, é estabelecido apenas um único provedor de identidade cuja responsabilidade é autenticar os usuários e compartilhar as informações com os demais provedores de serviços.

No entanto, o sistema possui a fragilidade de ficar refém do provedor de identidades, uma vez que ele tem controle sobre as informações de seus usuários, o que permite utilizá-las da forma que desejar. A figura 5 mostra o modelo centralizado.

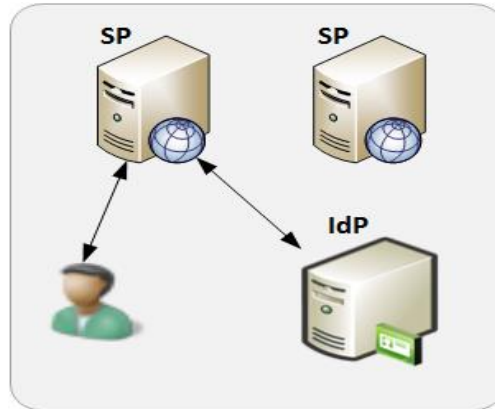


Figura 5 - Modelo Centralizado - Adaptado de [42]

c) Modelo Federado

Com novos avanços, o modelo de identidade federada veio com o objetivo de aprimorar o modelo centralizado. Este modelo de identidade procura distribuir os procedimentos de autenticação para diversos provedores de identidade separados por domínios administrativos. A ideia de domínio administrativo é representar uma entidade, órgão, governo (municipal, estadual, federal) com usuários envolvidos por diversos provedores de serviços, mas com um único provedor de identidade.

Neste modelo, o provedor de identidade possui a capacidade de compartilhar a identificação do usuário (ID) entre os servidores participantes do elo de confiança. Todo o círculo de confiança é estabelecido a partir de acordos relacionados à autenticação e segurança na relação com os membros de provedores de identidade e provedores de serviços.

Uma das vantagens do modelo é proporcionar uma autenticação única, evitando o desgaste do usuário para passar repetidamente pelo processo de autenticação. Como exemplo destes sistemas destaca-se o projeto *Liberty Alliance* e *Shibboleth*. A figura 6 mostra o modelo federado.

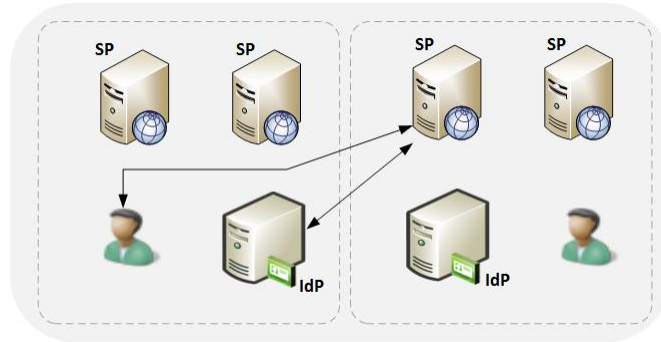


Figura 6 - Modelo Federado - Adaptado de [42]

d) Modelo Centrado no usuário

O modelo centrado do usuário transmite a ideia de o usuário possuir a capacidade de controlar suas identidades, tendo capacidade de criar, manter e utilizar suas informações. Neste caso, o modelo necessita de um aceite explícito do usuário antes de compartilhar suas informações de autenticação.

Nas propostas encontradas na literatura, as identidades de um usuário são utilizadas em diferentes provedores de serviços e ficam armazenadas em um dispositivo físico de posse do usuário, como por exemplo, um *smartphone*. Assim, o usuário autentica-se no dispositivo que possui e logo libera as informações que o provedor de serviços acessará, de acordo com a preferência de privacidade do usuário.

Desse modelo destacam-se as soluções *OpenID*, *Higgs* e *Microsoft CardSpace*. A figura 7 mostra o modelo Centrado no usuário.

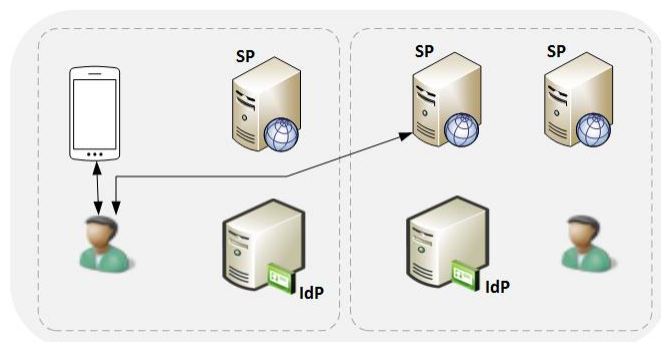


Figura 7 - Modelo Centrado no usuário - Adaptado de [42]

2.2.3 Single Sing-on (SSO)

O *Single Sing-on* é uma tecnologia cujo objetivo é facilitar e melhorar a experiência do usuário no processo de autenticação em sistemas, permitindo que após o primeiro logon o usuário possa autenticar-se em outros sistemas (desde que o usuário possua permissão para isso) sem a necessidade de realizar um novo logon, ou seja, vários sistemas são acessados através de uma única autenticação do usuário.

Na visão da IoT com múltiplos serviços e dispositivos que podem ser acessados pelo usuário no ambiente da *Smart House* ou outro qualquer, o SSO pode ser um grande facilitador para melhorar esta experiência no ambiente da IoT, pois retira a necessidade de efetuar logon individualmente em cada *appliance* ou serviço que é disponibilizado na IoT. Além disso, os dispositivos geralmente possuem restrições de recursos computacionais e não contém recursos ou potencial para gerenciar base de dados de usuários, e também não possuem mecanismos sofisticados para validação de credenciais de acesso dos usuários.

Ainda, dentre as vantagens do SSO podemos destacar:

- Aumento de produtividade devido à redução no tempo de acesso aos sistemas, devido ao usuário não ter que se autenticar em cada acesso ou a cada sistema (poupar tempo).
- Melhor experiência para o usuário que acessa vários sistemas.
- Facilidade na administração das contas de usuários através de um ponto único e centralizado.
- Aumento da segurança, pois o usuário não necessita memorizar todas as senhas necessárias para acessar os diversos sistemas.

Por outro lado, algumas desvantagens também devem ser destacadas:

- Repositório central de autenticação capaz de proporcionar um ponto único de falha.
- Se um atacante descobrir as credenciais de um usuário X ele terá acesso a todos os sistemas que o usuário tem acesso.

O sistema de SSO utiliza como base o protocolo OAuth, dessa forma, o uso deste protocolo permite a comunicação entre os servidores de autenticação e aplicação de maneira robusta e controlável, fazendo uso de mecanismos de controle de acesso já existentes.

Uma das implementações mais conhecidas de SSO é o Kerberos, que possibilita que os usuários utilizarem vários serviços com as mesmas credenciais em diferentes Sistemas Operacionais, como Windows e Linux, e que suporta múltiplas plataformas de hardware.

2.2.4 OpenID

OpenID é um sistema de Gestão de Identidade baseado no modelo centrado no usuário, uma plataforma livre e descentralizada. O sistema é descentralizado, pois não obriga o usuário a se cadastrar no sistema que deseja acessar.

O *OpenID* traz benefícios aos usuários e aos sistemas que utilizam a plataforma. Por diversas vezes os usuários são forçados a efetuar cadastros redundantes em vários sistemas na Internet que desejam obter acesso, dessa forma, muitas credenciais são repetidas ou são inseridas senhas consideradas fáceis com o objetivo de serem lembradas pelos usuários, como por exemplo 123mudar. Do lado dos sistemas, cada desenvolvedor tem a necessidade de garantir privacidade, segurança e integridade dos dados e oferecer o seu mecanismo de proteção, mas, é possível terceirizar esta autenticação e autorização de acesso utilizando um provedor de identidade. Assim, o *OpenID* auxilia na resolução dos problemas acima mencionados.

Um exemplo de IdP é *OpenID Connect* [5], que permite a integração de autenticação e autorização de acesso para que um aplicativo ou sistema não tenha que gerenciar as identidades, senhas e autorizações de acesso de usuários. *OpenID Connect* é uma camada de gerenciamento de identidade que usa o protocolo OAuth [5] para a autorização de acesso.

Pela facilidade de implementação e pelo uso de padrões abertos, muitas empresas têm se apresentado como um provedor de identidade, como por exemplo: *Facebook*, *Google*, *Paypal*, *Yahoo* e *Salesforce*.

Na figura 8, é apresentado um fluxo do *OpenID*.

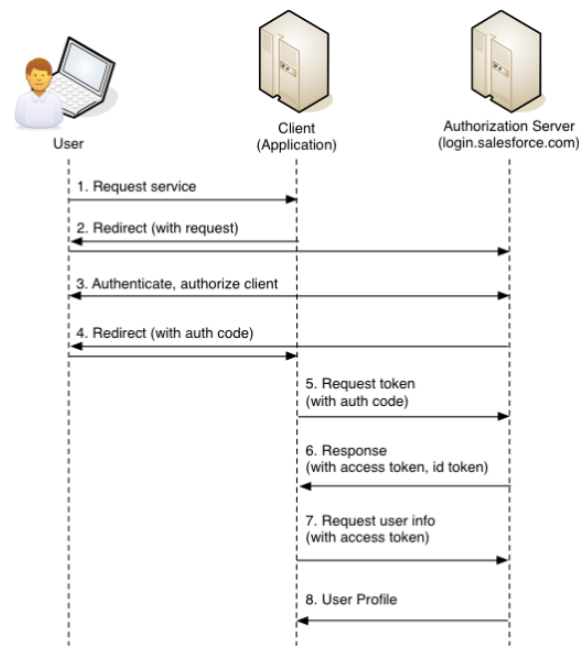


Figura 8 - Fluxo do *OpenID* – Adaptado de [43]

O usuário irá utilizar algum serviço ou aplicação que possua autenticação através do *OpenID*, dessa forma, a aplicação redireciona o usuário com a requisição pretendida para a página do servidor onde serão informadas as credenciais. Após o *login*, o usuário será redirecionado novamente para a página da aplicação com um código de autorização recebido pelo provedor de identidade. Logo, a aplicação requisita para o IdP um *token*, passando o código de autorização do usuário. O IdP responde com um identificador do *token* e um *token* de acesso.

Por fim, a aplicação solicita informações do usuário com base no *token* de acesso, resta ao IdP entregar o profile do usuário.

Além disso, como questão de segurança, o servidor não envia diretamente as informações do usuário para o serviço, pois as informações são enviadas através da URL, sendo que a troca de mensagens com o provedor de serviço e *OpenID* é realizada através de criptografia com chave assimétrica.

2.3 Padrão ANSI X.9.17

A utilização de chaves de criptografia é comumente empregada na Internet, pois permite realizar a comunicação de maneira cifrada, impedindo a visualização do conteúdo das mensagens. A norma ANSI X.9.17 [9] é um padrão para o gerenciamento de chaves de criptografia criado inicialmente para atender instituições financeiras, cujo objetivo é apresentar

como a distribuição de chaves simétricas deve acontecer, no entanto, seu conteúdo é de recomendação.

Além disso, a norma ANSI X9.17 especifica alguns requisitos mínimos, conforme os apresentados abaixo [28]:

- Controle e gestão das chaves durante o tempo de vida, a fim de evitar a divulgação não autorizada, alteração ou substituição.
- Distribuição da chave com objetivo de permitir a interoperabilidade entre os equipamentos e sistemas.
- Garantir a integridade da chave durante todas as fases de sua vida, incluindo a sua produção, distribuição, armazenamento, acesso, utilização e destruição.
- Recuperação no caso de uma falha do processo de gestão de chave ou quando a integridade do material de chave é comprometida.

A norma também define uma hierarquia de três níveis para a distribuição de chaves entre os pares envolvidos na comunicação. O nível mais alto é realizado com a chave *Master Key Encrypting Key* (KKM), que é distribuída manualmente e *off-line* entre os pares. O nível intermediário é realizado com a chave *Key Encrypting Key* (KEK), que é distribuída de modo *on-line*, ou seja, durante a comunicação. Já o nível mais baixo é realizado com a chave *Data Key* (KD), que também é distribuída de maneira *on-line*. As chaves KEK e KD são alteradas periodicamente e criptografadas com a chave mestra KKM que são distribuídas manualmente, *off-line*, e de forma segura. A figura 9, apresenta os níveis das chaves, segundo a norma.

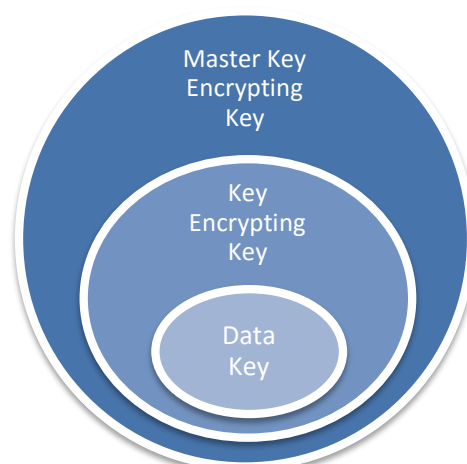


Figura 9 - Nível de hierarquia de chaves da norma ANSI X9.17 [28]

Finalizada a fundamentação teórica dos assuntos pertinentes ao trabalho proposto, o próximo capítulo apresenta uma análise dos trabalhos relacionados.

Capítulo 3

Trabalhos Relacionados

Os trabalhos relacionados foram divididos em 02 partes, sendo a primeira para agrupar trabalhos que envolvem IoT e Gestão de Identidade, já a segunda parte agrupa os trabalhos que utilizam outros modos de autenticação em IoT.

3.1 IoT e Gestão de Identidade

Na proposta de et al. [10] é apresentada uma arquitetura para Internet das Coisas que segue o modelo federado de identidades e que considera autenticação e controle de acesso para usuários e dispositivos. No trabalho, dispositivos são vistos como coisas que recebem endereçamento único e global com protocolo IPv6 e possuem a capacidade de se comunicar através da Internet.

Neste artigo, as chaves de segurança são estabelecidas através de um protocolo com base na criptografia ECC (*Elliptic Curve Cryptosystem*). Já para a política de controle de acesso, foi adotado o método de autorização baseado em RBAC (*Role-Based Access Control*).

Com a intensão de gerenciar e organizar recursos, é necessário que os dispositivos façam um pré-registro em um *gateway* confiável, chamado de Autoridade de Registro (*Registration Authority - RA*). O RA contribui para o processo de autenticação e também para fins de auditoria.

A distribuição e o estabelecimento das chaves são considerados fundamentais para a autenticação de uma entidade. Pode-se utilizar tanto uma SKC (*Secret Key Cryptography*) ou PKC (*Public Key Cryptography*) para uma implementação, no entanto, é oportuno observar as

vantagens e desvantagens de cada opção, mencionam os autores. Esquemas apoiados em SKC exigem uma grande quantidade de memória, proporcionam baixa escalabilidade devido ao modo de distribuição das chaves, adição e revogação de chaves, e exigem modo complicado para pré-distribuição da chave. No entanto, os sistemas baseados em PKC sofrem de alto consumo de energia e de tempo de atraso considerável. Em contrapartida, PKC fornece uma interface mais flexível e simples em comparação com o SKC, e que não necessita de pré-distribuição no compartilhamento de chave de pares. Dessa forma, os autores adotam a solução baseada em PKC.

Como próximo passo existe a necessidade de autenticar o usuário legítimo na Internet das Coisas. Objetos inteligentes e usuários podem estar em diferentes domínios, localizados em diferentes níveis de hierarquia da rede. Assim, o método de autenticação central é válido se o KDC consegue ampla aceitação e se ele estiver disponível. A tecnologia *OpenID* resolve esse problema, pois permite que os usuários tenham uma única conta, sendo possível efetuar login em diferentes domínios autenticando um provedor de identidade único. Uma abordagem para o gerenciamento de identidade é a federada, onde os sites participantes pertencem a um círculo de confiança. Um exemplo da arquitetura é representado na figura 10.

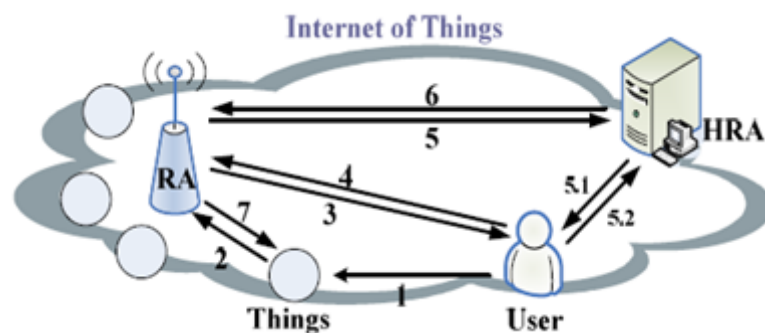


Figura 10 - Um exemplo da arquitetura proposta - Adaptado de [10]

O usuário deseja obter acesso a uma coisa ou dispositivo (passo 1), o dispositivo da IoT envia uma requisição para o RA (passo 2). O RA solicita que o usuário informe a sua Identidade (ID) de usuário (passo 3), assim, o usuário repassa a informação do HRA (*Home Registration Authority*) (passo 4). Logo, o RA verifica a informação do HRA recebida do usuário e valida do ID do usuário (passo 5). HRA desafia o usuário com uma pergunta (passo 5.1) e o usuário responde o desafio (5.2). Como consequência, o HRA responde ao RA que o ID do usuário é válido ou não (passo 6). Por fim, o RA responde para a coisa as informações 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).

No mesmo artigo, são tratadas as questões de controle de acesso, que apesar de não estarem associadas diretamente ao trabalho, é interessante salientar algumas tecnologias envolvidas. Dessa forma, é sugerido que o controle de acesso aos dispositivos seja realizado pelo RA, utilizando-se do modelo RBAC.

A proposta não aborda questões de SSO e também não apresenta resultados e testes que possam validar a proposta de trabalho. Como não apresenta detalhes de implementação, a proposta também não menciona protocolos da IoT utilizados. Por outro lado, o artigo traz uma interessante discussão teórica de possíveis ataques em que a solução proposta está supostamente protegida.

Na proposta de Fremantle et al. [12] os autores examinam o uso de federação de identidade e gerenciamento de acesso em uma abordagem para IoT. Em seu trabalho é explorado o uso dos protocolos OAuth para Internet das Coisas e o protocolo MQTT na versão 3.1 que é baseado em fila de mensagens. O MQTT atua como um componente intermediário entre Internet e Internet das Coisas.

O protocolo OAuth 1.0 [34], e seu sucessor, o OAuth *Framework* 2.0 [35], são protocolos projetados para solucionar problemas de privacidade e questões de controle de acesso relacionados a aplicações interconectadas em grande escala. Já o protocolo MQTT foi originalmente concebido como um protocolo para a telemetria através de redes restritas [36], construído para obter um baixo *overhead* por mensagem. O MQTT é usado em alguns cenários da IoT, e há bibliotecas para sistemas baseados em microcontroladores, como Arduino que se torna mais fácil de utilizar. Outra característica do MQTT é que a troca de mensagens é baseada em um modelo de *publish* e *subscribe* [37].

A proposta dos autores trabalha com 04 componentes principais, sendo *Broker* MQTT, o *Authorization Server*, a *Authorization Web Tool* e o dispositivo ou coisa. O *Broker* MQTT é baseado no Mosquitto (MQTT desenvolvido na linguagem C) [38] que inclui customizações para permitir a autenticação baseada em OAuth. O módulo *Authorization Server* é baseado na ferramenta WSO2 de código aberto e funciona como um IdP, utilizando o *OpenID*. O componente *Authorization Web Tool* permite ao usuário ou desenvolvedor criar um *token* para autorizar o acesso aos dispositivos baseado em seus dados pessoais. Além disso, o dispositivo da IoT foi representado na proposta com base em *hardware* do Arduino. Para validação dos

tokens, os autores implementaram no IdP uma própria API com o protocolo SOAP para consultar a validade e os escopos de um determinado *token*.

Apesar de os autores mencionarem que a solução funciona bem, no entanto, não foram apresentados os resultados de testes, além disso, foi possível observar que a validação mencionada utiliza apenas uma única coisa. Outro ponto importante de notar é que não foram utilizados canais seguros de criptografia entre dispositivo da IoT até o *Broker*.

Como trabalhos futuros, os autores desejam investigar a integração do protocolo OAuth com COAP e demais protocolos utilizados na IoT, além de melhorar os aspectos de segurança.

Battisti et al. [13] propôs um modelo de arquitetura federado no contexto da uma *Smart House*. Os autores propõem a utilização de um componente intermediário entre a Internet e Internet das Coisas com base em *Web Service*, fornecendo segurança de mensagens utilizando recursos de *WS-Security* (Serviços de Segurança de *Web Service*), para assegurar a integridade e confidencialidade das mensagens.

Dessa forma, os autores apontam que os dispositivos são cada vez mais chamados de “inteligentes” devido às suas funções, embora existam grupos de dispositivos que são desprovidos de recursos. Dessa forma, é necessário um *gateway* ou elemento de mediação que foi denominado de *Secure Mediation Gateway* (SMGW), capaz de descobrir informações sobre o status de domínio (*Smart House*) a qual o dispositivo pertence, superar a heterogeneidade de dispositivos e prover comunicações seguras entre os objetos da *Smart House*.

Na abordagem os autores dividem as coisas em dois grupos, o primeiro, designado como *intraSMGW* representa um conjunto interno de coisas que pertencem a um domínio de segurança acessível por um único SMGW. O segundo grupo é constituído pelo restante das coisas. Além disso, SMGW pode conectar-se a uma rede federada de SMGWs, chamada de *interSMGWs*, permitindo o acesso remoto e troca de dados entre outras SMGWs, sendo um elemento que atua na fronteira entre *intradomain* e *interdomain*. Como mecanismo de segurança, o SMGW também pode ser acessado por meio de criptografia de chave pública, e também com esquema de assinatura digital.

Na proposta para a troca de dados entre *interSMGW* é utilizado um *Web Service REST* no SMGW, que possui a função de publicar dados por meio do recurso *WS-Eventing*. Para segurança na troca de dados a solução proposta utiliza a pilha do *WS-Security*, a fim de garantir a segurança dos dados durante a transmissão.

Apesar de a proposta contextualizar a *Smart House*, não foram utilizados protocolos

desenvolvidos para IoT, além disso, não há uma segurança fim-a-fim no modelo.

Cirani et al. [14], apresenta uma arquitetura de autorização de serviço externo com base no protocolo OAuth, chamada IOT-OAS. O proposta trata da integração da Internet das Coisas com a Internet com esquema de autorização usando um canal de comunicação seguro entre os pares.

O principal objetivo da proposta dos autores com o trabalho é aliviar a carga dos *appliances*, terceirizando as funcionalidades de autenticação e autorização para um ambiente externo, mantendo a lógica simples nos dispositivos, fazendo com que os dispositivos não utilizem muitos recursos computacionais. Na figura 11, é apresentado o fluxo principal da arquitetura.

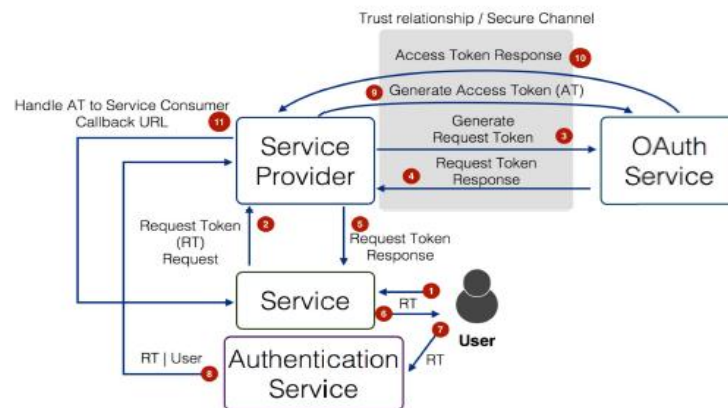


Figura 11 - Procedimentos principais do IOT-OAS - Adaptado de [13]

Como no fluxo normal, o passo 1 é sempre iniciado pelo usuário, o qual solicita um serviço ou aplicação (SC), o SC consulta o *Service Provider* (SP) com o objetivo de receber um *Request Token* (RT), o SP solicita ao IOT-OAS a emissão de um RT para o SC, no passo 4 o IOT-OAS verifica a identidade de SC e emite uma resposta para SP com o RT, o SP processa o RT, que devolve para SC, no passo 6, o SC redireciona o usuário (U) para o AS com o RT recebido, o usuário (U) efetua a autenticação, logo o AS notifica o SP que a autenticação foi bem sucedida e apresenta a RT com a identidade do usuário (U), o SP pede a da IOT-OAS para trocar o RT com um *Access Token* (AT) para o usuário (U), o IOT-OAS gera a AT e retorna para o SP.

Além disso, temos o fluxo de integração do SP com IOT-OAS para autorização de

requisição, conforme apresentado na figura 12.



Figura 12 - Integração do SP com IoT-OAS - Adaptado de [13]

Dessa forma, o SC solicita informações do usuário (U) para o SP usando o *Access Token* (AT) recebido após a autenticação do usuário (U), logo o SP, solicita ao IoT-OAS verificação, informando provedor, requisição e o AT, assim, IoT-OAS verifica a solicitação de SC e informa ao SP sobre a autorização de SC, requisição validada em um base de autorização, por fim, o SP devolve a solicitação do SC de acordo com a resposta do IOT-OEA.

Os autores também utilizaram outros 04 cenários de aplicações, sendo com comunicação através de *Broker*, comunicação baseada em gateway, comunicação fim-a-fim para CoAPs e comunicação híbrida entre HTTP e CoAP.

O protótipo foi implementado com ContikiOS e com hardware físico, e como testes, foi avaliado o consumo de energia dos dispositivos.

No entanto, o trabalho apresentado não trata comunicação fim-a-fim na integração de segurança entre Internet e IoT, permitindo que o intermediário seja um único ponto de falha que pode comprometer a segurança. Além disso, ele não aborda questões de SSO.

Chibelushi et al. [15] propõem um sistema de IdM centrado no usuário para IoT considerando o contexto da saúde e aplicações médicas. No entanto, a proposta concentra-se em Redes Móveis Ad-Hoc (*MANETs*) e não proporciona comunicação segura e recurso do SSO, além de não utilizar o protocolos da IoT.

A proposta do *framework* contém um módulo de identidade que consiste no ID do dispositivo, ID do usuário. Dessa forma, como na IOT não existe a possibilidade de compartilhar informações com um dispositivo ou com outras aplicações, os autores propõem o uso de uma MANET adequada par fornecer meios para uma interação com os dispositivos da

IoT.

Segundo a argumentação do artigo, devido à natureza das informações sensíveis no contexto de Saúde na IoT, é vital criar uma separação de dados individuais dentro do dispositivo compartilhado, essa funcionalidade é fornecida por um módulo chamado *Sandbox*. Este módulo cria uma parede virtual entre os usuários individuais de um dispositivo compartilhado e fornece um mecanismo que segmenta o dispositivo e as informações dos usuários para que não possam ser compartilhadas ou acessadas por outros usuários. Segundo a alegação, isso também fornece um mecanismo extra para ajudar na proteção contra roubo de identidades e informações de uso indevido.

Além disso, a proposta possui um módulo de gerenciamento de privacidade que fornece um meio extra de criação de políticas de privacidade dinâmicas que irão reforçar a segurança do IdM.

O artigo traz apenas um modelo e não apresenta preocupações com segurança, tais como segurança fim-a-fim, criptografia, além de protocolos da IoT. A proposta está concentrada em utilizar um IdM para IoT e criar módulos adicionais de privacidade e contexto de usuário.

Domenech et al. [39] propõem um estudo de caso de uma aplicação da Web das Coisas que utiliza *OpenID*. O contexto está associado ao conceito de ambiente de vida assistida para assistência médica a pacientes em suas casas, tentando mantê-los independentes das infraestruturas de saúde, como hospitais.

No artigo o *framework OpenID Connect* foi utilizado para autenticar os usuários e dispositivos e para estabelecer uma relação de confiança entre os usuários e outras entidades. Com características do contexto, os autores argumentam que para sistemas de saúde a escolha do modelo de IdM centrada no usuário é o mais adequado pelas seguintes razões: capacitar o usuário: os usuários podem ter controle sobre os atributos; escolher o IdP mais apropriado para uma transação; e a privacidade.

A proposta da arquitetura envolve a utilização de dispositivos médicos (por exemplo, dispositivos portáteis), para monitorar continuamente o estado de saúde do paciente. Os dispositivos podem usar diferentes protocolos de comunicação (por exemplo, IEEE 802.15.4, Wi-Fi e Bluetooth). Dispositivos da IoT são incapazes de tratar informações de usuários na web devido a restrições de recursos, por isso, estão ligados por um *gateway* inteligente que atua como uma ponte entre dispositivos e Internet.

A proposta não apresenta implementação e testes, e também não apresenta aspectos de

segurança como criptografia, segurança fim-a-fim e recursos como SSO.

Na próxima seção são apresentados brevemente alguns outros modos de autenticação em IoT.

3.2 Outros modos de autenticação em IoT

Outras propostas visam proporcionar autenticação de diferentes maneiras em IoT. Hummen et al. [16] apresentam um método de autenticação com base em certificados, usando DTLS e com o objetivo de desempenho e redução de *overhead* de comunicação. A proposta não considera SSO no contexto da IoT, por não se tratar de um IdM.

Li et al. [17] propõe usar *Lightweight Directory Access Protocol* (LDAP) e Kerberos para fornecer autenticação e SSO na da IoT. No entanto, a proposta não considera um elemento (gateway) intermediário para adaptar as mensagens entre Internet e IoT.

Yao et al. [18] apresentam um mecanismo leve para autenticação *multicast* em pequena escala para a IoT, porém não abordam o recurso de SSO.

Embora vários trabalhos propostos buscam utilizar um IdM da Internet para IoT e outros esquemas de autenticação e autorização de acesso, muitas questões permanecem em aberto quando se busca um esquema viável para integrar os contextos da Internet e IoT.

Algumas propostas não consideram aspectos de segurança, tais como canais de comunicação segura, uso da pilha de protocolos da IoT e criptografia de chaves simétricas. Outras propostas visam integrar contextos da Internet e da IoT, mas não consideram comunicação fim-a-fim para uma integração segura, permitindo um ponto único de falha, comprometendo a segurança, possibilitando a exploração de vulnerabilidade para interceptar e manipular mensagens. Muitas propostas não consideram o recurso de SSO, além disso, várias propostas são apenas modelos e não apresentaram resultados experimentais, dificultando sua avaliação da viabilidade das hipóteses.

A tabela 1, apresenta um resumo dos trabalhos relacionados de IdM com IoT.

3.3 Resumo dos trabalhos relacionados

Artigo	Proposta	Comparação com trabalho	Comentários
Liu et al. [10]	<p>Os autores propõem uma arquitetura de autenticação e controle de acesso de dispositivos e usuários da Internet das Coisas.</p> <p>Na proposta, os dispositivos podem se comunicar diretamente através de endereços únicos globais, utilizando IPv6.</p> <p>Para autenticação e autorização, os autores propõem utilizar o <i>OpenID</i> e RBAC.</p>	Os autores utilizam a autenticação do <i>OpenID</i> e também focam no controle de papéis com o uso RBAC.	<p>A proposta não aborda questões de SSO e também não menciona os protocolos da IoT.</p> <p>Além disso, não apresenta resultados que possam validar a proposta.</p>
Fremantle et al. [12]	Propõem controlar o acesso de dispositivos da IoT através do protocolo OAuth, além de utilizar um protocolo baseado em fila de mensagens (MQTT 3.1) como um componente intermediário entre Internet e Internet das Coisas.	Utilizam um broker para intermediar a Internet e IoT, porém, fazendo uso de um protocolo MQTT, ao invés de CoAP.	<p>Não utilizam segurança fim-a-fim.</p> <p>Não utilizam protocolos de segurança para o mundo da IoT.</p> <p>Não mencionam a utilização de SSO.</p>
Battisti et al. [13]	<p>Battisti et al. [13] propôs um modelo de arquitetura federado no contexto da <i>Smart House</i>.</p> <p>Os autores propõem a utilização de um componente intermediário entre a Internet e IoT com base em Web Service, fornecendo segurança de mensagens usando WS-Security (Serviços de WS), para assegurar a integridade e confidencialidade das mensagens.</p>	<p>Apesar de a proposta utilizar o cenário da <i>Smart House</i>, não foram utilizados protocolos desenvolvidos para IoT, além disso, não há uma segurança fim-a-fim no modelo.</p> <p>A solução proposta traz uma abordagem baseada em Web Service tradicional para o mundo da IoT.</p>	<p>A proposta não utiliza segurança fim-a-fim.</p> <p>Não utilizam protocolos de segurança para o mundo da IoT.</p> <p>Não mencionam a utilização de SSO.</p>
Cirani et al. [14]	Cirani et al. [14], apresentam uma arquitetura de autorização de serviço externo com base no protocolo OAuth, chamada IOT-OAS.	<p>A proposta apresenta um trabalho robusto e bem desenvolvido. Utiliza a pilha de protocolos da IoT e canais de seguros para troca de mensagens.</p> <p>Contribui para a segurança da IoT, no entanto, a comunicação é</p>	<p>Não trata de comunicação fim-a-fim na integração de segurança entre Internet e IoT.</p> <p>Não aborda SSO.</p>

		fim-a-fim somente no ambiente da IoT. Dessa forma, permite que o intermediador, seja um ponto único de falha que pode comprometer a segurança.	
Chibelushi et al. [15]	Chibelushi et al. [15] propõem um sistema de IdM para IoT considerando o contexto da saúde. No entanto, a proposta se concentra em Redes Móveis Ad Hoc (<i>MANETs</i>) e não proporciona comunicação segura, expondo todos os dispositivos diretamente para a Internet.	O artigo trata apenas de um modelo e não apresenta preocupações com segurança, como: segurança fim-a-fim, criptografia, além de protocolos da IoT. A proposta está concentrada em utilizar um IdM para IoT e criar módulos adicionais de privacidade e contexto de usuário.	Não proporciona comunicação segura e também não leva em consideração o recurso de SSO. Além de não utilizar o protocolos da IoT.
Domenech et al. [39]	Domenech et al. [39] propõem um estudo de caso de uma aplicação da Web das Coisas usando <i>OpenID</i> . O contexto está associado ao conceito de ambiente de vida assistida para assistência médica a pacientes em suas casas, tentando mantê-los independentes da infraestrutura de saúde, como hospitais.	Trata se um modelo que utiliza <i>OpenID</i> com a IoT, porém, não se preocupa com aspectos de segurança. Sua visão é mais funcional do que preocupada com segurança.	A proposta não apresenta implementação e testes, e também não apresenta aspectos de segurança como criptografia, segurança fim-a-fim e recursos como SSO.

Tabela 1 - Resumo dos trabalhos relacionados

A tabela 2, apresenta uma comparação dos recursos que puderam ser observados nos artigos em relação a proposta apresentada.

	IdM ou OpenID	Integração fim-a-fim Internet e IoT	Criptografia	SSO	Pilha de protocolos IoT	Implementação e Testes
Liu et al. [10]	✓		✓			
Fremantle et al. [12]	✓					
Battisti et al. [13]	✓		✓			
Cirani et al. [14]	✓		✓		✓	✓
Chibelushi et al. [15]	✓					

Domenech et al. [39]	✓					
Hummen et al. [16]			✓		✓	✓
Li et al. [17]				✓		
Yao et al. [18]			✓		✓	✓
Proposta deste trabalho	✓	✓	✓	✓	✓	✓

Tabela 2 - Comparação dos trabalhos relacionados com a proposta

Capítulo 4

Proposta

Nesta seção, será apresentado um esquema de autenticação e autorização de acesso baseado em chaves com segurança fim-a-fim para IoT. A abordagem integra de forma segura um IdM proveniente da Internet com a IoT, considerando as limitações de recursos, além de oferecer o benefício do SSO para que o técnico da empresa fabricante dos *appliances* se autentique em vários *appliances* na *Smart House* com uma autenticação única.

O Gateway, que é um elemento intermediador entre a Internet e a Internet das Coisas, foi concebido para não ser um elemento crítico de segurança, pois a comunicação ocorre de modo cifrado através de chave simétrica entre o *appliance* e o sistema da empresa fabricante dos *appliances*, evitando a interceptação e manipulação do conteúdo da mensagem.

Nas seções seguintes, o esquema proposto será apresentado de forma mais detalhada, a começar pela visão geral da arquitetura, e depois pelo fluxo de troca de mensagens entre os componentes da solução.

4.1 Visão Geral

A proposta de trabalho envolve seis componentes principais, conforme apresentado na figura 13, sendo: *Appliance*, *Customer Service*, *Gateway*, *Appliance Technician*, *Authentication Server* e *Access Authorization Server*. O *Appliance* (App) é uma "coisa" ou objeto utilizado na Internet das Coisas dentro de uma *Smart House*, por exemplo. O App geralmente possui recursos limitados e não possui acesso direto à Internet. Neste caso, a solução busca preservar

a privacidade e impedir o acesso não autorizado ao App, tornando inacessível diretamente da Internet.

O *Appliance* tem um atributo identificador como, por exemplo, o número de série e também uma chave simétrica, a qual foi fornecida pelo fabricante durante a sua produção na linha de montagem. O número de série e a chave simétrica estão armazenados também no *Customer Service* (CS). O CS é um serviço fornecido pelo fabricante do App e que realiza a interface de comunicação entre o App e o técnico da empresa fabricante, a fim de responder às demandas solicitadas pelos Apps, tais como ativação do produto, monitoramento, manutenção pró-ativa e corretiva, e também para atualização de firmware, por exemplo.

O *Gateway* (GW) é o elemento responsável pela ligação da Internet com os App da *Smart House*, permitindo a troca de mensagens entre App e CS.

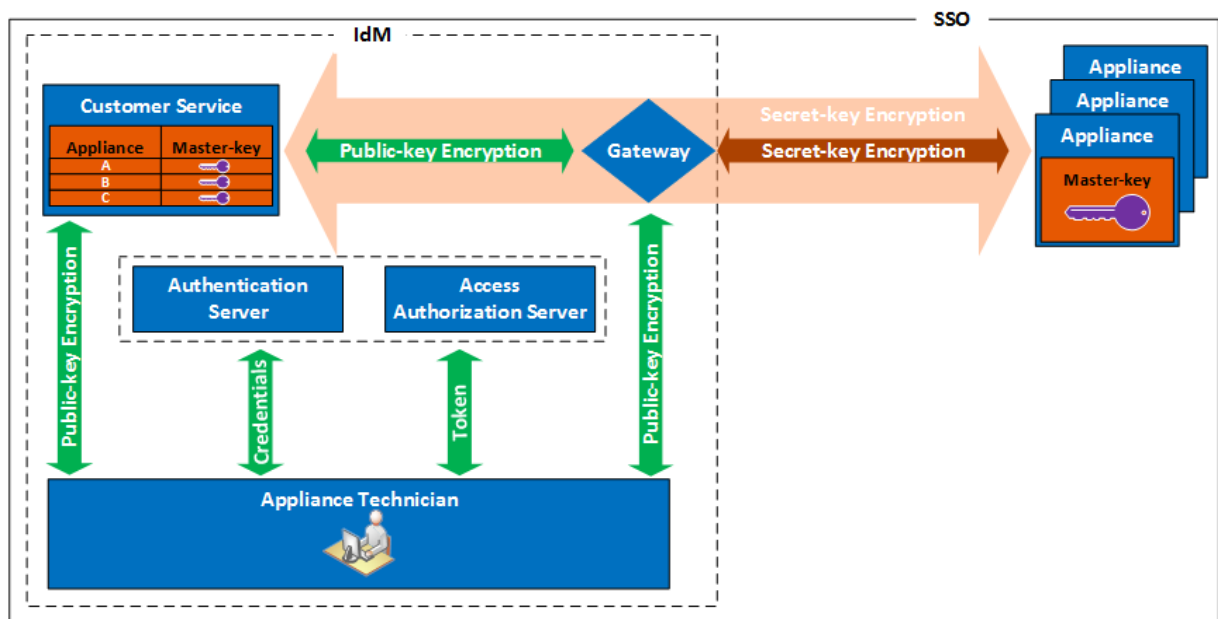


Figura 13 - Visão Geral do IdM e esquema de autenticação baseado em chaves para prover SSO para IoT

Conforme visualizado na figura 13, o *Appliance Technician* (AppTec) é um sistema operado por um técnico da fabricante para responder às exigências dos consumidores e as necessidades dos Apps no período de pós-venda. Dessa forma, a fim de garantir a segurança no acesso, o AppTec não tem acesso direto ao App, para isso, é utilizado o CS como um elemento intermediador entre o CS e o App.

O *Authorization Server* (AS) e *Access Authorization Server* (ASS) fazem parte de um IdP. O AS fornece um serviço de autenticação capaz de validar as credenciais do AppTec e

também para o fornecer o recurso de SSO para o esquema proposto. Já o AAS é um serviço de autorização de acesso que fornece *tokens* para que o AppTec já autenticado (com as credenciais validadas) possa acessar o CS e também vários GWs.

Dessa forma, o GW é acessado de forma segura, minimizando as possibilidades de ataques, caso alguém tente acessá-lo e não esteja devidamente autenticado e autorizado, o acesso não será possível, uma vez que as entidades devem estar previamente registradas no AAS para chegar ao GW.

Para prover a comunicação segura fim-a-fim entre o CS e o App foram utilizados 02 níveis de criptografia baseado em chaves. No primeiro nível, CS e App utilizaram criptografia de chave simétrica baseada em uma chave mestra, denominada KKM, esta chave é utilizada para distribuir posteriormente a chave de sessão KEK. No segundo nível, CS, AppTec, GW, AS e AAS utilizam a criptografia de chave pública. O GW e App utilizam a criptografia de chave secreta para proteger os dados transmitidos, incluindo KEK, por mensagem. Dessa forma, CS e App compartilham a chave mestra KKM que são armazenados manualmente no sistema CS e no firmware do App, seguindo como referência a padronização da norma ANSI X.9.17 para distribuição de chaves.

A chave KKM está relacionada com o número de série do App registrada no sistema do CS, informado no processo de produção do App, conforme mencionado anteriormente. Assim, as mensagens trocadas entre CS e App são criptografadas por KKM, não permitindo qualquer acesso intermediário ao conteúdo da mensagem, além do transporte da chave de sessão KEK.

A mesma proteção é obtida quando a chave KEK criptografa os dados da mensagem. Para a solução proposta assume-se que a chave KKM é imutável, no entanto, ela pode ser facilmente atualizada tanto no CS como no App, sem afetar o esquema e o trabalho técnico, caso isso seja necessário.

A proposta apresenta proteção considerável do IdM no contexto da Internet usando uma chave assimétrica, protegendo as mensagens trocadas entre CS, AppTec, e GW. No contexto da IoT, é utilizada uma chave simétrica para criptografar os dados da mensagem, porque isto é apropriado para os recursos limitados dos Apps, embora exista uma proteção adicional por mensagem.

4.2 Fluxo de Mensagens

Na proposta apresentada são considerados duas possíveis comunicações fim-a-fim entre App e CS. A primeira é iniciada pelo App e que será respondida por um técnico da empresa fabricante. A segunda é iniciada pelo AppTec e respondida por um App. A comunicação iniciada pelo App implica em uma requisição de serviço ao CS, por exemplo, a ativação do *appliance* ou pedido de tarefa de manutenção. A comunicação iniciada por AppTec implica em um serviço solicitado por App ou uma intervenção necessária, por exemplo, uma atualização de firmware.

A Figura 14 apresenta um diagrama de sequência de uma comunicação iniciada por um App. De modo geral, a sequência das mensagens é a mesma para qualquer requisição. Um sujeito requisita para o App um serviço prestado pelo CS (evento 1), informando o conteúdo de uma requisição (*requestValue*). O App gera e utiliza uma chave de sessão KEK para criptografar o conteúdo das mensagens trocadas durante a vida útil da requisição. Além disso, o App também usa a chave mestra KKM para criptografar a chave KEK e um valor de *nonce* (por exemplo, um *timestamp*). Então, o App envia o valor criptografado (*encryptedValue*) ao GW, juntamente com o número de série (*serialNumber*) e o endereço do CS (*customerURL*). O GW, por sua vez, realiza o *parser* da mensagem da IoT para Internet e encaminha a mensagem para o CS (evento 1.1.1).

Na sequência, o CS recupera a chave KKM do App a partir do número de série presente na mensagem e decodifica os dados (*encryptedValue*). O CS valida a KEK do App pelo *nonce*, cujo objetivo é evitar o ataque de *replay*. O CS armazena a KEK do App e a utiliza para criptografar e descriptografar mensagens futuras do App na mesma sessão. O CS responde ao App o índice (ID) da chave de sessão (um valor numérico que identifica a KEK armazenada) e um valor do *nonce* incrementado ($nonce + 1$). O *nonce* é usado para garantir a autenticidade do CS, garantindo que apenas o titular da KKM pode descriptografar e recriptografar uma mensagem. O GW, por sua vez, recebe a resposta e mapeia o índice da KEK com o endereço App. Então, GW traduz a mensagem de Internet para uma mensagem da IoT e encaminha a mensagem para o App.

Dessa forma, o App recebe uma mensagem criptografada com KEK, decodifica a mensagem e valida o *nonce*, através do seu incremento. O App criptografa a requisição com um novo *nonce* e encaminha para o GW (caso 1.2), fornecendo o índice (ID) da chave de sessão

que será utilizado para se comunicar com CS. O CS recupera a chave de sessão com base no ID, decodifica e armazena a requisição para ser respondida pelo técnico posteriormente. O CS retorna o índice da chave de sessão e o *nonce* incrementado. O App recebe, valida o *nonce*, e informa ao sujeito o status da requisição.

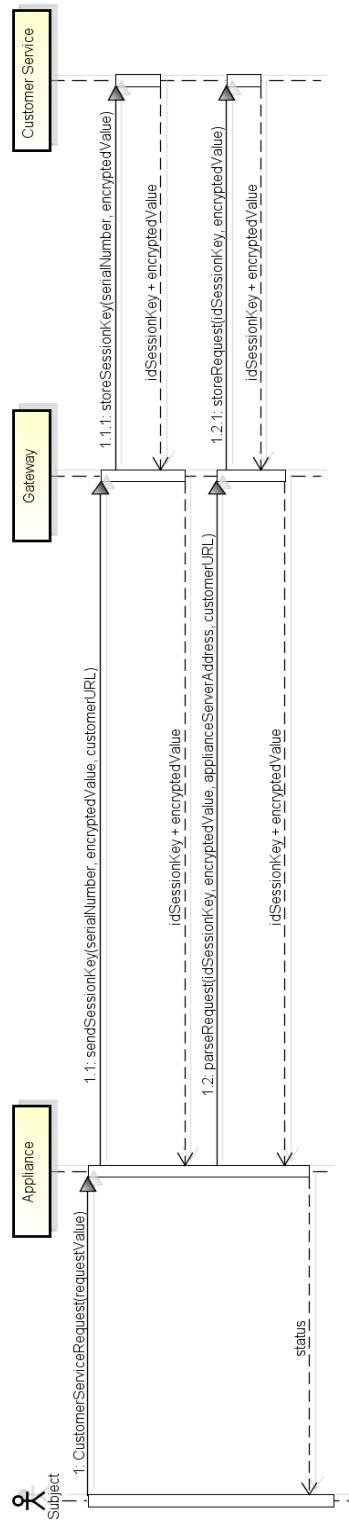


Figura 14 - Fluxo de mensagens para uma requisição iniciado pelo Appliance.

A Figura 15 mostra um diagrama de sequência que representa o processo de autenticação e autorização de acesso de um técnico da fabricante para acessar o AppTec. O técnico requisita acesso ao AppTec (evento 1) sendo redirecionado para o AS com suas credenciais solicitadas (evento 2). Assim, o AS valida as credenciais do técnico e responde com um código (com validade de tempo) a ser utilizado para solicitar um *token* de acesso ao AAS (evento 3.1). O AAS devolve um *token* para ser utilizado pelo técnico para responder ou atender a uma requisição do App.

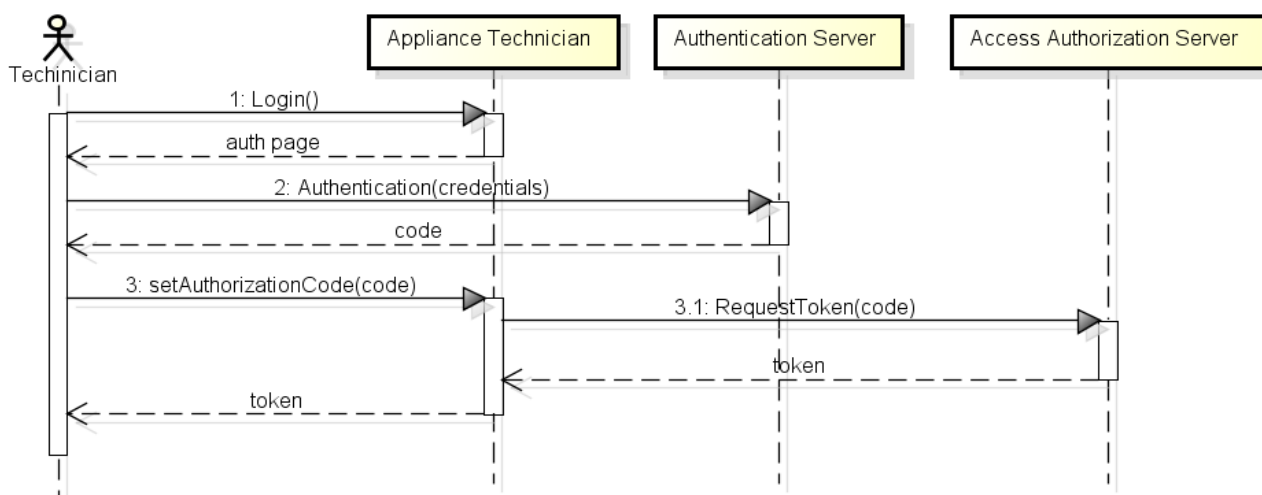


Figura 15 - Autenticação e autorização de acesso para o técnico da fabricante

Depois de o técnico estar devidamente autenticado e autorizado, ele possui acesso através do AppTec aos dados da requisição do App, já enviada anteriormente. O AppTec permite recuperar a KEK através de seu índice (ID) e possui a capacidade de decifrar a requisição, processar e encaminhar a resposta criptografada ao GW. O GW recebe a resposta criptografada fim-a-fim juntamente com um *token* de acesso, que após o processo de validação, permite ao GW analisar a mensagem e transmitir a resposta criptografada para o App. O App recebe a resposta criptografada com a KEK e baseado em seu índice (ID), decodifica a resposta, e processa a requisição.

O segundo tipo de comunicação iniciada por um técnico é quando ele necessita coletar informações ou realizar uma tarefa de manutenção, por exemplo, uma atualização de *firmware* para o App. Esta comunicação segue o procedimento de *Call Back* (Figura 16).

Assumindo que um técnico está autenticado e autorizado, conforme mostrado na Figura 15, utiliza o AppTec para solicitar ao CS informações do App (evento 1), fornecendo um *token* e um número de série (serialNumber). O CS valida o *token* (evento 1.1) e responde os dados do

App, incluindo o endereço do GW (gatewayAddress) associado ao App. O AppTec requisita ao GW de uma *Smart House* para que o App inicie uma comunicação com o GW (evento 2). O GW valida o *token* de acesso (caso 2.1) e notifica o App (evento 2.2). Dessa forma, o App irá iniciar uma sessão seguindo os passos de comunicação mencionados na Figura 15.

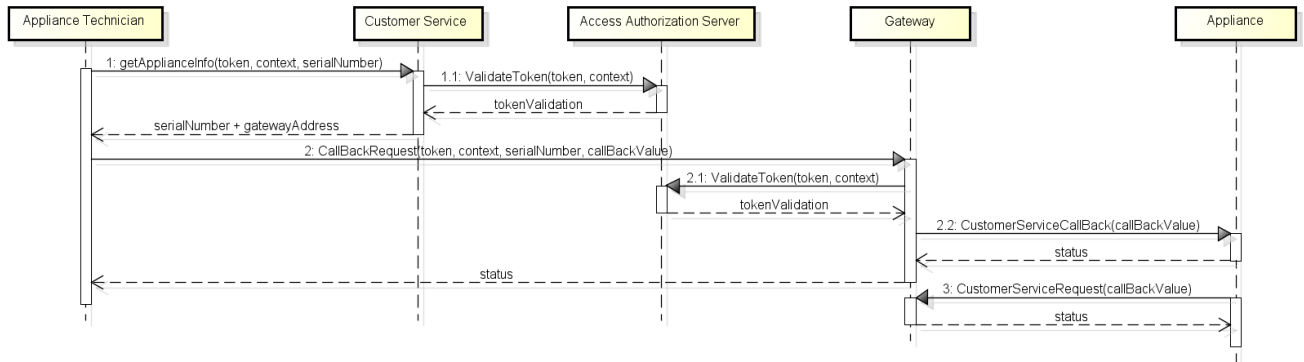


Figura 16 - Diagrama de sequência do procedimento de *Call Back*

4.3 Protótipo

Nesta seção, é apresentado um protótipo que implementa o esquema de autenticação baseado em IdM e autorização de acesso. O protótipo utiliza padrões de TI, tecnologias conhecidas e bibliotecas de código aberto.

4.3.1 Implementação

O *Manufactor Domain* consiste em dois componentes, AppTec e CS. O AppTec foi implementado usando a estrutura do Vaadin [19], um framework Java para desenvolvimento de aplicações Web. O CS foi implementado como um *Web Service RESTful* usando a API JAX-RS [20].

O *Customer Domain* consiste em um GW e vários Apps. As interfaces com o GW que representam o contexto Internet foram implantados através de um servidor HTTP, implementado em Java. Por outro lado, a interface da Internet das Coisas foi construída através de um servidor CoAP implementado utilizando a biblioteca Californium [21]. Com essa implementação o GW é capaz de analisar mensagens entre protocolos HTTP e CoAP e vice-versa.

O App foi implementado em Java baseado no projeto Californium, e pode ser executado em ContikiOS (Sistema Operacional para IoT) [22]. Para a implementação da criptografia foi utilizado o algoritmo AES de 128 bits para executar a criptografia utilizada para as chaves KKM e KEK. O Scandium, um subprojeto do Californium, foi utilizado para fazer a comunicação segura entre App e GW no contexto da IoT, uma vez que suporta DTLS na versão 1.2.

O servidor de autenticação (AS) foi implementado seguindo a especificação do *OpenID* 2.0, usando a biblioteca Nimbus [23], uma biblioteca Java que além de implementar *OpenID* 2.0 implementa a especificação do protocolo OAuth 2.0. O Nimbus fornece um IdM para AppTec, CS e GW, e assegura que somente usuários autenticados e autorizados acessam os App localizados na *Smart House*. A autorização de acesso (AAS) foi implementada seguindo especificação do protocolo OAuth 2.0 e Nimbus, a fim de emitir *tokens* de acesso para o AppTec autenticado para possibilitar que o CS acesse vários GWs.

No nível da rede, a solução assume que os endereços de IPv6 não mudam, no entanto, podem ser atualizados dinamicamente sem afetar o esquema proposto. No contexto da IoT, utilizou-se o software de restrição de banda, chamado WonderShaper [40], com o objetivo de reduzir a largura de banda, a fim de simular ou aproximar a utilização do 6LoWPAN em IEEE 802.15.4.

A Figura 17 mostra um protótipo da arquitetura destacando uma comunicação segura entre componentes e a pilha do protocolo usada. Do ponto de vista da Internet, a comunicação é feita usando HTTPS e de Internet das Coisas usando CoAPs.

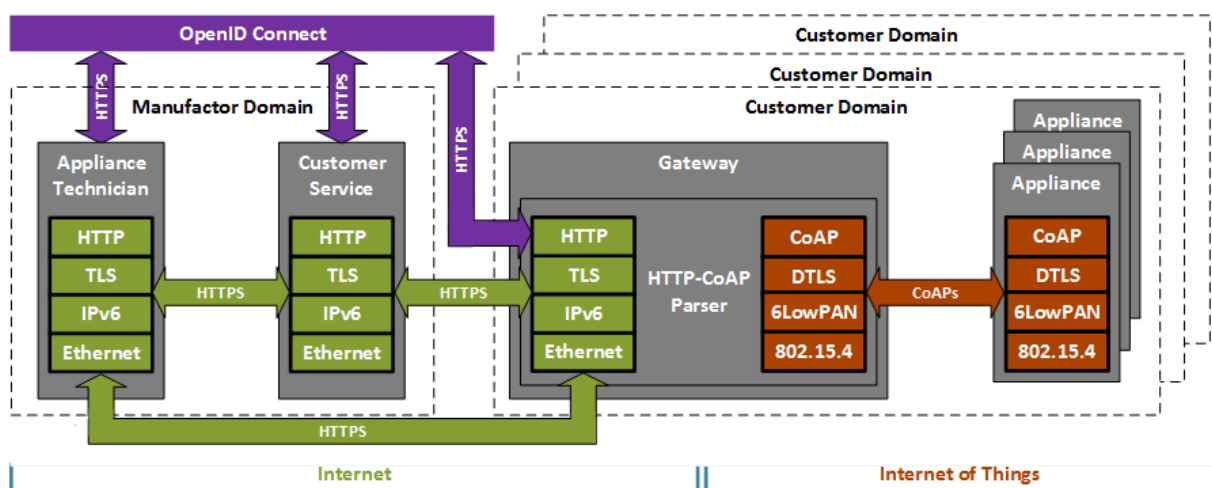


Figura 17 - Arquitetura do protótipo

4.3.2 Avaliação

A avaliação foi realizada utilizando duas máquinas em uma rede local para obter um ambiente controlado e para evitar a interferência com medições de tempo. Uma máquina hospeda o servidor *OpenID Connect*, CS, AppTec e GW, e outra máquina hospeda instâncias de Apps. Cada máquina tem dois núcleos virtuais, memória de 8 GB, 60 GB de disco, rodando Ubuntu 14.04.2 LTS, Java 1.7.0_75 64 bits. Na máquina para simulação dos Apps a largura de banda foi reduzida a 40 Kbps na frequência de 915 MHz, a fim de simular de forma mais precisa a comunicação do App, após a especificação do protocolo 6LoWPAN.

Os testes possuem o objetivo de medir o impacto do esquema de autenticação fim-a-fim, levando em conta questões como: (i) o impacto do tamanho da mensagem na requisição e o tempo de resposta utilizando criptografia e sem criptografia no ambiente de IoT; (ii) medição do tempo de resposta com ativação ou não do recurso do SSO com 50 Apps; (iii) comparativo do tempo de resposta para implementação sequencial e paralela; (iiii) validação da quantidade de Apps suportados pelo protótipo desenvolvido.

Nos testes, foram utilizados tamanhos de mensagens que variam de 32 até 4096 bytes, e o número de *Appliances* de 10 até 50, que representam valores próximos a realidade no contexto *Smart House*. O tempo de resposta mensurado foi medido utilizando os protocolos COAP e CoAPs.

A Figura 18 mostra que para requisições de tamanho de 32 a 1024 bytes, o tempo médio de resposta fica abaixo de 900 ms por requisição e o *overhead* pelo uso do CoAPs permanece abaixo de 200 ms. Esta observação fornece dados para afirmar que a proposta funciona bem, pois apresenta um *overhead* quase constante, mesmo para um grande número de Apps.

No entanto, é possível observar um aumento de *overhead* para requisições com mensagens que possuem mais de 1024 bytes, atingindo *overhead* de 350 ms, quando o tamanho de mensagens é de 2048 bytes e 10 Apps. Além disso, o tempo de resposta permanece abaixo de 1250 ms nas demais requisições, e as requisições CoAPs estão abaixo de 200 ms com 4096 bytes, mantendo-se adequado para o número Apps. Esta observação indica que a proposta funciona bem, com quase 15% de *overhead*, mesmo para 50 Apps e com tamanho de mensagens de 4096 bytes.

Além disso, é possível observar também que utilizando CoAP ou CoAPs com tamanho de mensagem maior do que 1024 bytes, o tempo médio de resposta foi maior para 10 apps do

que para 30 ou 50 Apps. Tal resultado sugere que o SSO fornece alguma vantagem de tempo quando o número de Apps aumenta em 10.

Com o resultado do teste é possível concluir que o GW tem um bom desempenho quando se leva em conta um número realista de Apps, no contexto de uma *Smart House*. Além disso, os resultados mostram a integração adequada entre os dois contextos, sem impacto significativo para a Internet das Coisas. Ou seja, não existe uma diferença significativa no tempo de resposta de 10 a 50 Apps, cerca de 7% para CoAPs e 6% para CoAP.

Considerando o tamanho das requisições, é possível observar que existe um *overhead* considerável para mensagens entre 1024 e 2048 bytes. No entanto, desde que foi utilizada a chave simétrica, adequada para a Internet das Coisas, CoAP ou CoAPs, a proposta apresentou um pequeno *overhead* sobre 4096 bytes, sem afetar o tempo médio de resposta, mostrando a viabilidade da solução no mundo real contexto da *Smart House*.

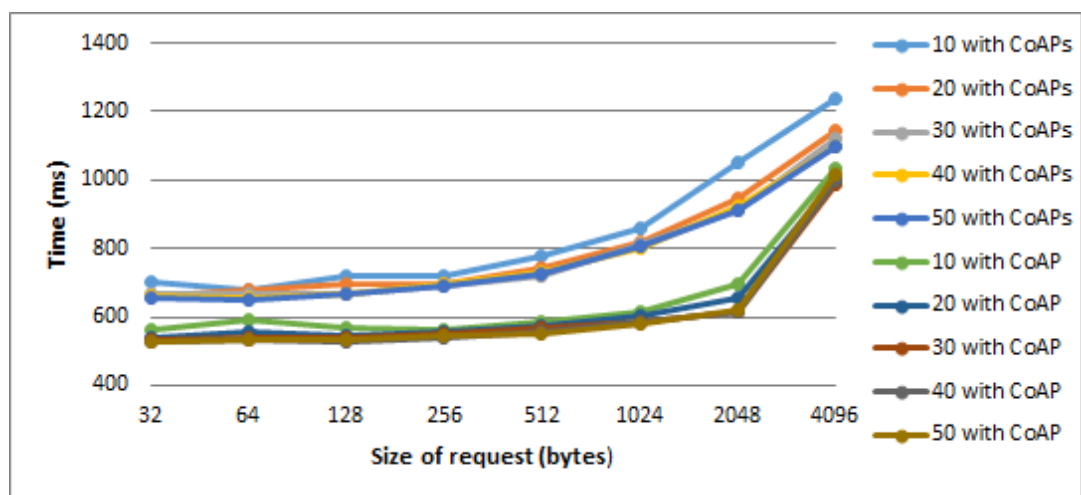


Figura 18 - Avaliação do Protótipo

A fim de avaliar a implementação do mecanismo de autenticação única (SSO), foram executados testes com CoAPs com e sem autenticação SSO, utilizando 50 App. Para este teste foram utilizadas requisições com tamanho de 32 a 4096 bytes. Como resultado do ambiente simulado, devido aos scripts de automação, não foram visualizadas diferenças em relação ao tempo de respostas para mensagens até 2048 bytes. Ou seja, o tempo de resposta das requisições é praticamente o mesmo. No entanto, para mensagens de 4096 bytes houve uma diferença perceptível entre o uso do mecanismo do SSO. Conforme mostra figura 19.

Dessa forma, no teste é possível concluir que não se ganha quantitativamente, mas sim qualitativamente. No ambiente real, o recurso do SSO é adequado para Internet das Coisas, pois retira a função extra do App de validação de autenticações, além disso, o técnico pode acessar múltiplos Apps com uma única autenticação, sem a necessidade de conhecer uma senha diferente para cada *appliance*, evitando também o uso da mesma senha para todos os Apps - prática que submete os *appliances* ao risco, caso a senha seja descoberta em algum momento.

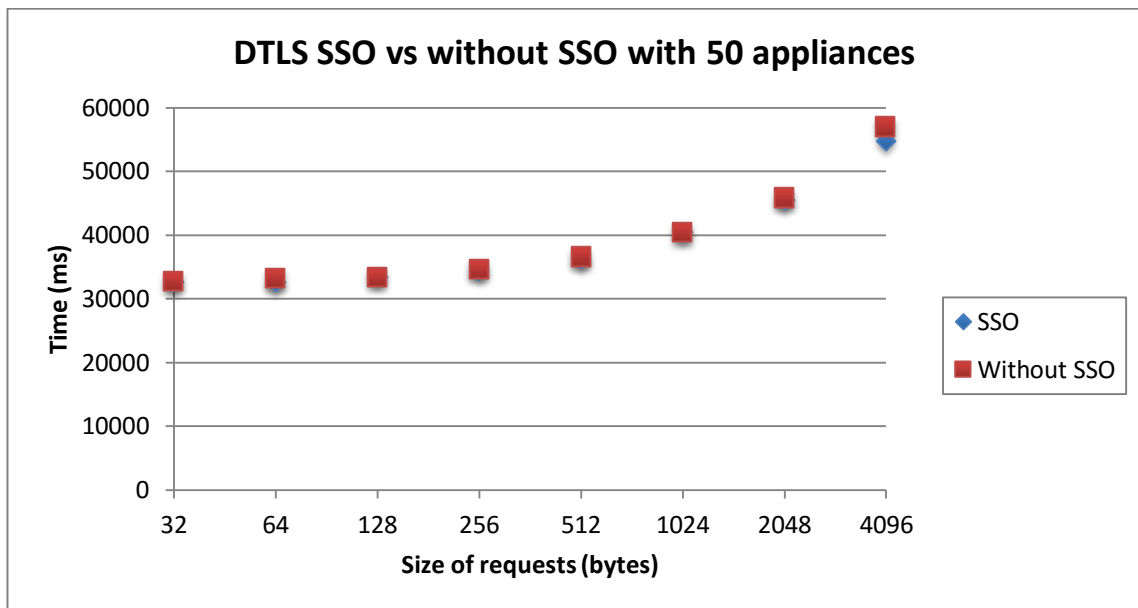


Figura 19 - Comparativo de DTLS com e sem SSO para 50 App

Outro teste realizado foi um comparativo entre as implementações com CoAPs em que as requisições são tratadas em paralelo ou em sequência. Para este teste, foram utilizados 50 Apps para validação com mensagens de 32 a 4096 bytes, como resultado foi possível avaliar que para mensagens consideradas pequenas, ou seja, até 512 bytes, a solução não apresenta diferença considerável em utilizar requisições de modo paralelo ou sequencial. No entanto, para mensagens acima de 512 bytes é possível notar uma diferença considerável no tempo de resposta. Dessa forma, com 4096 bytes o tempo de resposta sequencial é o dobro em relação a implementação paralela, conforme apresentado na figura 20.

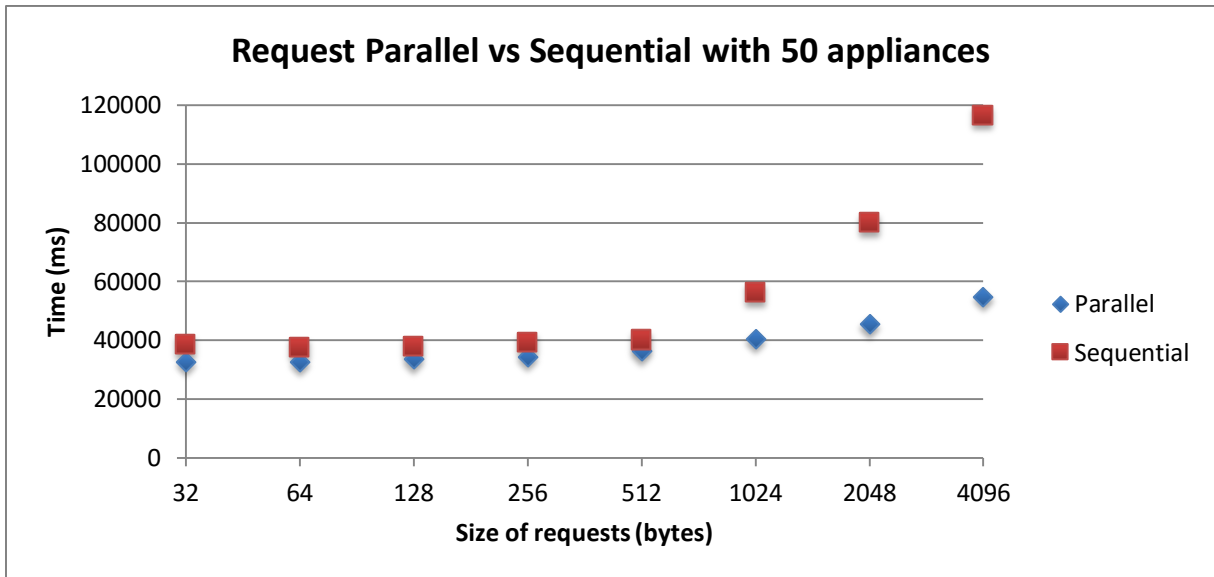


Figura 20 - Requisições paralelas vs sequenciais

Com o objetivo de avaliar qual o limite operacional de funcionamento da solução apresentada, foram realizados testes com CoAPs e autenticação SSO com mensagens de 32 a 4096 bytes, iniciando com 10 Apps variando de 10 em 10 Apps. Como resultado, obtivemos um limite de 120 Apps suportados. A partir deste número, a solução e as bibliotecas utilizadas começaram a apresentar erros e inconsistências, resultando em um comportamento não esperado. Para chegar a este número limitante, foram testadas várias configurações de Apps (e.g. 150, 130 e 125 etc.), sendo o número de 120 Apps o limite aceitável de acordo com *hardware* utilizado na solução. A figura 21 apresenta um gráfico com o número de Apps suportados.

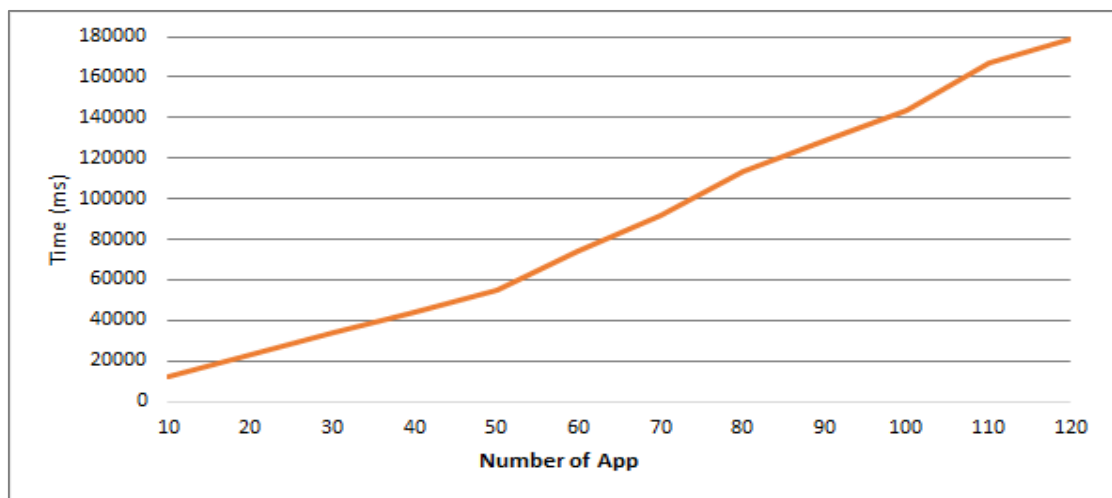


Figura 21 - Limite de apps suportados na solução proposta

Capítulo 5

Conclusão

O trabalho proposto apresentou um método de autenticação para integrar um IdM do contexto da Internet para a Internet das Coisas. A ligação entre os dois contextos é fornecida por um *gateway* incapaz de visualizar o conteúdo da mensagem, mas que atua na análise do contexto da IoT a Internet e vice-versa.

As chaves simétricas são adequadas para a Internet das Coisas para possibilitar a proteção de mensagens fim-a-fim (a partir dos *appliances* até o *Customer Service*), a fim de evitar que o *gateway* seja um ponto único de falha. A autenticação do técnico da fabricante no *gateway* tem o objetivo de mitigar os possíveis ataques provenientes de Internet. Assim, o *gateway* fornece isolamento do *appliance* na Internet, protegendo de ataques para os dispositivos da Internet das Coisas que possuem menos recursos e, conseqüentemente, menor capacidade de proteção.

O recurso do SSO "encapsulado no IdM", é adequado para a Internet das Coisas, pois retira a necessidade de o *appliance* obter um esforço extra para interagir com um servidor de Internet, tal como proposto na literatura. Além disso, o técnico pode acessar múltiplos *appliances* com uma única autenticação, ou seja, sem a necessidade de saber uma senha diferente para cada *appliance*, e também sem a necessidade de utilizar a mesma senha para todos os *appliances* - prática que submete os *appliances* ao risco, caso a senha seja descoberta por alguém mal-intencionado.

O método de proteção adicional, por mensagem, melhora qualitativamente o mecanismo, enquanto se adapta a cada contexto. Além disso, faz com que o conteúdo

difícilmente seja violado, uma vez que uma chave assimétrica fim-a-fim também protege as partes mais sensíveis do conteúdo da mensagem.

O esquema proposto foi baseado em padrões de TI e o protótipo foi implementado utilizando tecnologias consolidadas para a Internet e o contexto de IoT. Assim, mostrou-se a viabilidade da proposta analisando o seu tempo de resposta, variando o número de *appliances* e o tamanho das mensagens. A abordagem proposta apresentada não possui *overhead* significativo para o tempo de resposta de 10 a 50 Apps e com tamanho de mensagem de 32 a 4096 bytes por mensagem. Além disso, o tempo de resposta, em média, fica abaixo de 1250 ms por requisição, um *overhead* aceitável, levando em conta que é apresentado um IdM baseado em chave para a segurança fim-a-fim em IoT.

Como trabalhos futuros se deseja portar a solução apresentada para o ambiente de simulação do ContikiOS, a fim de efetuar testes e simulações considerando outras medições, como por exemplo, consumo de energia [1].

Referências

- [1] L. Atzori, A. Iera, and G. Morabito, “*The Internet of Things: A survey*,” *Computer Networks*, vol. 54, no. 15, 2010, pp. 2787–2805.
- [2] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad, “*Proposed security model and threat taxonomy for the Internet of Things (IoT)*,” in *Proc. of the CCIS - Communications in Computer and Information Science*, 2010, pp. 420–429.
- [3] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “*Internet of things: Vision, applications and research challenges*,” *Ad Hoc Networks*, 2012, pp. 1497–1516.
- [4] A. Belapurkar, A. Chakrabarti, H. Ponnappalli, N. Varadarajan, S. Padmanabhuni, and S. Sundarrajan, “*Distributed Systems Security: Issues, Processes and Solutions*”, John Wiley & Sons, 2009.
- [5] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore, “*OpenID Connect Core 1.0*”. [Online]. Available: http://openid.net/specs/openid-connect-core-1_0.html.
- [6] N. Kushalnagar, G. Montenegro, and C. Schumacher, “*IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*,” RFC 4919.
- [7] Z. Shelby, K. Hartke, and C. Bormann, “*The Constrained Application Protocol (CoAP)*”, IETF RFC 7252.
- [8] E. Rescorla and N. Modadugu, “*Datagram Transport Layer Security Version 1.2*”, IETF RFC 6347.

- [9] ANSI, “X9 Encryption Collection”. [Online]. Available: <http://webstore.ansi.org/RecordDetail.aspx?sku=X9+Encryption+Collection>.
- [10] J. Liu, Y. Xiao, and C. L. P. Chen, “Authentication and Access Control in the Internet of Things,” in Proc. of the ICDCSW - Intl. Conf. on Distributed Computing Systems Workshops, 2012, pp. 588–592.
- [11] D. Van Thuan, P. Butkus, and D. Van Thanh, “A user centric identity management for Internet of things,” in Proc. of the ICITCS - IT Convergence and Security, 2014, pp. 1–4.
- [12] P. Fremantle, B. Aziz, J. Kopecky, and P. Scott, “Federated Identity and Access Management for the Internet of Things,” in Proc. of the Int. Workshop on Secure Internet of Things, 2014, pp. 10–17.
- [13] M. Leo, F. Battisti, M. Carli, and A. Neri, “A federated architecture approach for Internet of Things security,” in Proc. of the EMTC - Euro Med Telco, 2014, pp. 1–5.
- [14] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, G. Ferrari, and S. Member, “IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios,” IEEE Sens. J., vol. 15, no. 2, pp. 1224–1234, 2015.
- [15] C. Chibelushi, A. Eardley, and A. Arabo, “Identity Management in the Internet of Things: the Role of MANETs for Healthcare Applications,” Comput. Sci. Inf. Technol., vol. 1, no. 2, pp. 73–81, 2013.
- [16] R. Hummen, J. H. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, “Towards viable certificate-based authentication for the internet of things,” in Proc. of the HotWiSec - Hot topics on Wireless network Security and privacy, 2013, p. 37.
- [17] N. Li, Q. Wang, and Z. Deng, “Authentication framework of IIEDNS based on LDAP & Kerberos,” in Proc. of the IC-BNMT - Int. Conf. on Broadband Network and Multimedia Technology, 2010, pp. 695–699.

- [18] X. Yao, X. Han, X. Du, and X. Zhou, “A *lightweight multicast authentication mechanism for small scale IoT applications*,” IEEE Sens. J., vol. 13, no. 10, 2013, pp. 3693–3701.
- [19] Vaadin, “*OpenID Integration*.” [Online]. Available: <https://vaadin.com/directory#!addon/openid-integration>.
- [20] “*Java API for RESTful Services*.” [Online]. Available: <https://jax-rs-spec.java.net/>.
- [21] Eclipse Foundation, “*Californium*.” [Online]. Available: <https://www.eclipse.org/californium/>.
- [22] “*The Contiki Operating System*.” [Online]. Available: <http://contiki-os.org/>.
- [23] “*Nimbus OAuth 2.0 SDK with OpenID Connect extensions*.” [Online]. Available: <http://connect2id.com/products/nimbus-oauth-openid-connect-sdk>.
- [24] Recordon, D. and Reed, D. “*Openid 2.0: a platform for user-centric identity management*”. In DIM '06: Proceedings of the second ACM workshop on Digital identity management, pages 11–16, New York, NY, USA. ACM, 2006.
- [25] 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
- [26] CERP-IoT. “*Vision and challenges for realising the internet of things*”. [Online]. Available: http://www.theinternetofthings.eu/sites/default/files/Rob%20van%20Kranenburg/Clusterbook%202009_0.pdf.
- [27] ITU Internet reports 2005: *The internet of things*.
- [28] “*Key Management Using ANSI X9.17*” [Online]. Available: <http://www.cerberussystems.com/infosec/stds/fips171.htm>.
- [29] Bhargav-Spantzel, A., Camenisch, J., Gross, T., and Sommer, D. *User centrality: a*

taxonomy and open issues. Journal of Computer Security, 2007, p. 493–527.

[30] Chadwick, D. W. “*Federated Identity Management. Foundations of Security Analysis and Design*” V. Heidelberg: Springer-Verlag Berlin, 2009. p. 96-120.

[31] Bertino, E.; Takahashi, K. “*Identity Management Concepts, Technologies, and Systems*”. Boston: Artech House, 2011.

[32] Damiani, E.; Vimercati, S. D. C. D.; Samarati, P. “*Managing multiple and dependable identities*”. Internet Computing, IEEE, v.7, n. 6, p. 29-37, dez. 2003. ISSN 10.1109/MIC.2003.1250581.

[33] Kothmayr T., Schmitt C., Wen Hu, Michael Brunig, Georg Carle. *A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-Way Authentication*. 2012. 7Th IEEE Internacional Workshop on Practical Issues in Building Sensor Network Applications, Florida.

[34] E. E. Hammer-Lahav, “*The OAuth 1.0 Protocol*,” IETF, RFC 5849.

[35] D. H. (ed), “*The OAuth 2.0 Authorization Framework*,” IETF, RFC 6749.

[36] “*Mqtt history*.” [Online]. Available: <http://mqtt.org/wiki/doku.php/history>.

[37] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “*The many faces of publish/subscribe*,” ACM Computing Surveys (CSUR), vol. 35, no. 2, pp. 114–131, 2003.

[38] “*An Open Source MQTT v3.1 Broker*”. [Online]. Available: <http://mosquitto.org/>

[39] M. C. Domenech, E. Comunello, and M. S. Wingham, “*Identity Management in E-Health: A Case Study of Web of Things application using OpenID Connect*,” IEEE 16th International Conference on e-Health Networking (Healthcom) pp. 219–224, 2014.

- [40] B. Hubert, “*The Wonder Shaper.*” [Online]. Available: <http://lartc.org/wondershaper>.
- [41] I. Ishaq, D. Carels, G. Teklemariam, J. Hoebeke, F. Abeele, E. Poorter, I. Moerman, and P. Demeester, “*IETF Standardization in the Field of the Internet of Things (IoT): A Survey,*” *J. Sens. Actuator Networks*, vol. 2, no. 2, pp. 235–287, 2013.
- [42] “*Gerenciamento de Identidade*” - Universidade Federal do Rio de Janeiro - Escola Politécnica, Redes de Computadores, 2012. [Online]. Available: http://www.gta.ufrj.br/grad/12_1/gerenc_identicidades/index.php?file=sgi.
- [43] Patterson, P. “*Inside OpenID Connect on Force.com*” - The OpenID Connect Protocol, 2014. [Online]. Available: https://developer.salesforce.com/page/Inside_OpenID_Connect_on_Force.com.