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

MSc. Adriano Witkovski, Prof. Dr. Altair O. Santin (Orientador)

Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná de Informática – PUCPR
Curitiba – PR – Brasil

wadriano@gmail.com, santin@ppgia.pucpr.br

Abstract. *Abstract. The Internet of Things (IoT) brings significant challenges to authentication schemes in a scenario with several appliances in a smart house. An Identity Management (IdM) can be applied to easily authenticate a technician that intends to access the appliances from the Internet. However, the Internet context is different from the IoT, demanding context adaptation. Therefore, integrating 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 directly reachable from the Internet, using IdM, and without creating a single point of vulnerabilities in the gateway. The proposal works with two key-based schemes, one for Internet and the other for IoT. A proof-of-concept implementation shows the proposal is feasible and does not present significant overhead for messages with up to 4096 bytes and with 50 appliances.*

Resumo. *Resumo. A IoT (Internet das Coisas) traz desafios significativos para esquemas de autenticação em cenários com múltiplos appliances em uma Smart House. O Gerenciamento de Identidades (IdM) pode ser aplicado para autenticar um técnico que pretende acessar os appliances a partir da Internet. O contexto Internet é diferente da IoT, exigindo adaptação do contexto. 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 criar o ponto de vulnerabilidades no gateway. A proposta interage com dois esquemas baseados em chave, um para a Internet e outro para a IoT. O protótipo mostra que a proposta é viável e não apresenta um overhead significativo para mensagens com tamanho máximo de 4096 bytes e com 50 appliances.*

1. Introdução

O principal objetivo da IoT é interligar diversas “coisas” ou *appliances*, provendo conexão à Internet para fornecer serviços como monitoramento, gerenciamento e automação [Atzori, Iera and Morabito 2010]. No contexto de IoT os *appliances* geralmente possuem restrições de recursos computacionais. Por esse motivo, inserir mecanismos de segurança com recursos limitados pode representar um grande desafio [Babar, Mahalle, Stango, Prasad 2010]. Os *appliances* podem ser passíveis de problemas de segurança como confidencialidade, autenticidade e integridade dos dados.

Assim, alguns requisitos de segurança necessitam ser garantidos como comunicação segura dos dados, acesso seguro à rede e Gestão de Identidades [Babar, Mahalle, Stango, Prasad 2010].

Um cenário da IoT é a *Smart House* que contém diversos *appliances* contectados à Internet (e.g. eletrodomésticos). A empresa fabricante dos *appliances* acessa remotamente os dispositivos para efetuar manutenção, atualização de *firmware*, etc. Por questões de segurança é importante que os *appliances* localizados dentro da *Smart House* não estejam expostos na Internet. Assim, gerir a identidade dos técnicos do fabricante é imprescindível para manter gestão de acesso aos *appliances*. Integrar um IdM da Internet com a IoT não é trivial, devido à restrição de recursos dos *appliances* e também pela ausência de um canal de comunicação segura entre os dois contextos.

Abordagens apresentadas para autenticação e autorização de acesso em IoT utilizam chave única em todos os *appliances* [Babar, Mahalle, Stango, Prasad 2010 e Miorandi, Sicari, De Pellegrini, Chlamtac 2012]. A utilização de uma chave única tem como vantagem a redução de recursos, porém, torna-se inviável caso a chave seja descoberta. Outra desvantagem é a falta de controle do usuário está acessando um determinado *appliance*, além da dificuldade de manutenção da chave em todos os *appliances* [Miorandi, Sicari, De Pellegrini e Chlamtac 2012]. A utilização de um serviço de autenticação possibilita a autorização de acesso que resolve as deficiências da chave única. As abordagens se limitam à autenticação e autorização de acesso localizado na Internet ou exclusivamente na IoT. Outras propostas buscam integrar os dois contextos, mas não apresentam implementação e não consideram um canal de comunicação seguro fim-a-fim entre os dois contextos [Van Thuan, Butkus 2014 e Fremantle, Aziz, Kopecky, Scott 2014 e Leo, Battisti, Carli e Neri 2014]. Nenhuma proposta aborda a autenticação única utilizando SSO (*Single Sign-On*).

O trabalho está estruturado da seguinte forma. A seção 2 apresenta a motivação e contribuição do trabalho. A seção 3 apresenta a proposta. A seção 4 apresenta o protótipo desenvolvido e os resultados obtidos. E por fim a seção 5 traz as conclusões do trabalho.

2. Motivação e Contribuição

Ao acessar remotamente os dispositivos o fabricante dos *appliances* pode monitorar e coletar informações para prover serviços. Uma máquina de lavar conectada à Internet pode submeter os dados de rotação e determinados parâmetros para um servidor na nuvem computacional do fabricante, por exemplo. O fabricante pode fornecer um serviço pró-ativo de substituição de peças. Apesar dos *appliances* da *Smart House* serem endereçáveis e, portanto, estarem acessíveis a partir da Internet, não devem estar expostos para acesso direto, pois vulnerabilidades podem exploradas ou injetadas nos mesmo e toda a segurança da casa pode ser comprometida.

Segundo [Chadwick 2009] 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 técnico do fabricante possa acessar os *appliances* é importante que exista autenticação para o acesso. Um dos desafios é integrar um IdM da Internet com os dispositivos de baixa capacidade de processamento, armazenamento e memória com a pilha de protocolos de IoT, além de fornecer

comunicação segura fim-a-fim. Os dispositivos da IoT geralmente não suportam os protocolos tradicionais como HTTP (*Hypertext Transfer Protocol*) e SSL (*Secure Socket Layer*). Por esta razão foram desenvolvidos o CoAP (*Constrained Application Protocol*) e DTLS (*Datagram Transport Layer Security*) para atender os requisitos da IoT [Shelby, Hartke, Bormann e Rescorla, Modadugu]. O *gateway* necessita de um *parser* para que o protocolo HTTP no contexto da Internet e o CoAP no contexto da IoT possam se comunicar. Na literatura, a maioria das propostas considera apenas o contexto da IoT, outros trabalhos assumem que o *gateway* é um elemento confiável. O recurso de SSO para IoT é praticamente inexplorado na literatura.

A principal contribuição deste trabalho é que o esquema proposto é capaz de operar um IdM de forma integrada entre o contexto de Internet e IoT. A comunicação 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 *gateway* se torne um elemento vulnerável. O uso do SSO, provendo autenticação única para que o técnico do fabricante se autentique uma única vez por dia, por exemplo, e possa acessar múltiplos *appliances*, facilitando a administração.

3. Proposta

A proposta envolve seis componentes, conforme apresentado na Figura 1: *Appliance*, *Customer Service*, *Gateway*, *Appliance Technician*, *Authentication Server* e *Access Authorization Server*. O *Appliance* (App) é uma "coisa" utilizado na IoT com um atributo identificador e uma chave simétrica fornecida em sua fabricação. 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 faz a comunicação entre o App e o técnico da empresa. O *Gateway* (GW) é o elemento responsável pela intermediação da comunicação entre a Internet e os App da *Smart House*.

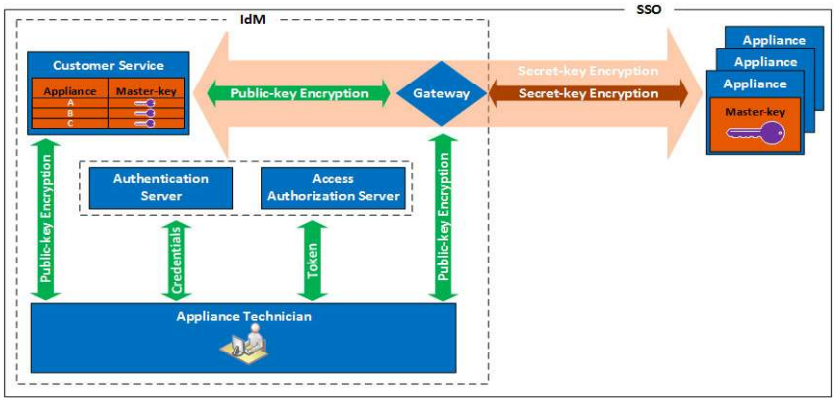

Figura 1 - Esquema de autenticação baseado em chaves para SSO em IoT.

O *Appliance Technician* (AppTec) é um sistema operado por um técnico do fabricante para responder às exigências dos consumidores e as necessidades dos Apps no período de pós-venda. O *Authorization Server* (AS) e *Access Authorization Server* (ASS) fazem parte de um IdP (*Identity Provider*). O AS fornece um serviço de autenticação capaz de validar as credenciais do AppTec e também para o fornecer o recurso de SSO. O AAS é um serviço de autorização de acesso que fornece *tokens* para que o AppTec já autenticado possa acessar o CS e também vários GWs. O GW é

acessado de forma segura, caso alguém tente acessá-lo e não esteja autenticado e autorizado, visto que as entidades devem estar previamente registadas no AAS para acessar o GW. No primeiro nível, CS e App utilizaram criptografia de chave simétrica baseada em uma chave mestre, fornecida pelo fabricante durante a fabricação do *appliance*, denominada KKM (*Master Key Encrypting Key* – ANSI X.9.17). Esta chave é utilizada para distribuir posteriormente a chave de sessão KEK (*Key Encryption Key*). 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 mestre KKM que é armazenada manualmente no CS e no firmware do App. A chave KKM está relacionada com o número de série do App registrada no sistema do CS. As mensagens trocadas entre CS e App são criptografadas por KKM, não permitindo qualquer acesso intermediário ao conteúdo da mensagem. A mesma proteção é obtida quando a chave KEK criptografa os dados da mensagem.

3.1 Fluxo de Mensagens

A Figura 2 apresenta a comunicação iniciada por um App. Um sujeito solicita para o App um serviço prestado pelo CS (evento 1) com 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. O App usa a chave mestre KKM para criptografar a chave KEK e um *nonce* (e.g. *timestamp*). O App envia o valor criptografado (*encryptedValue*) ao GW com o número de série (*serialNumber*) e o endereço do CS (*costumerURL*). O GW realiza o *parser* da mensagem da IoT para Internet e encaminha a mensagem para o CS (evento 1.1.1).

O CS recupera a chave KKM do App através do número de série 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 cifragem e decifragem de mensagens futuras do App na mesma sessão. O CS responde ao App o índice (ID) da chave de sessão e um valor do *nonce* incrementado de 1 (*nonce* + 1). O *nonce* é usado para garantir a autenticidade do CS, garantindo que apenas o titular da KKM pode a decifrar e recifrar uma mensagem. O GW recebe a resposta e mapeia o índice da KEK e o endereço do App. O GW traduz a mensagem de Internet para contexto da IoT e encaminha o App. O App recebe uma mensagem cifrada com KEK, decifra a mensagem e valida o *nonce.* O App cifra a requisição com um novo *nonce* e encaminha para o GW (caso 1.2), fornecendo o índice (ID) da chave de sessão que será utilizada para se comunicar com CS. O CS recupera a chave de sessão com base no ID, decifra e armazena a requisição para ser respondida posteriormente pelo técnico. O CS retorna o índice da chave de sessão e o *nonce* incrementado de 1. O App recebe a requisição, valida o *nonce*, e informa ao sujeito o status da mesma.

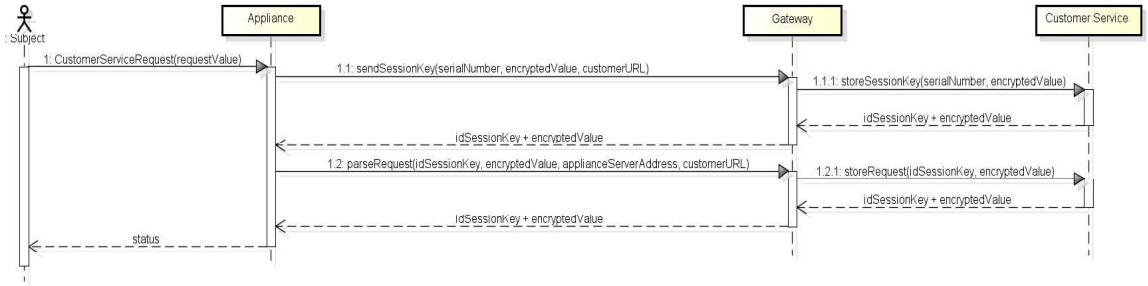

Figura 2 - Fluxo de mensagens para uma requisição iniciado pelo Appliance.

A Figura 3 mostra o processo de autenticação e autorização de acesso de um técnico do fabricante para acessar o AppTec. O técnico requisita acesso ao AppTec (evento 1) sendo redirecionado para o AS com suas credenciais solicitadas (evento 2). 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 atender a uma requisição do App.

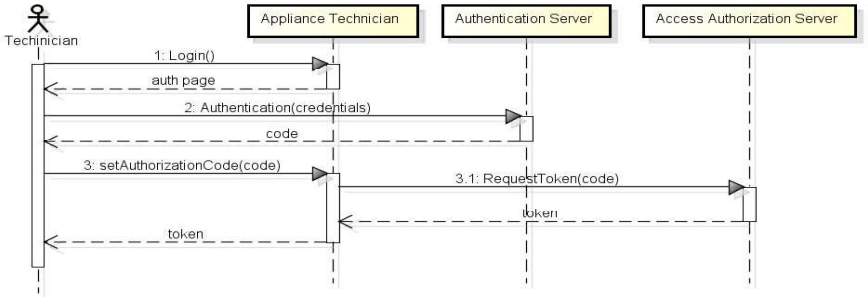

Figura 3 - Autenticação e autorização de acesso para o técnico da fabricante

O AppTec recupera a KEK através de seu índice (ID) e também possui a capacidade de decifrar a requisição, processar e encaminhar a resposta cifrada ao GW. O GW recebe a resposta cifrada juntamente com um *token* de acesso que permite ao GW fazer o *parser* e retransmitir a mensagem criptografada para o App. O App recebe a resposta cifrada com a KEK, usando o índice, decifra a resposta e processa a requisição.

O segundo tipo de comunicação iniciada por um técnico é para coletar informações ou realizar uma tarefa de manutenção, por exemplo, uma atualização de *firmware* no App, então é executado o procedimento de *Call Back* (Figura 4).

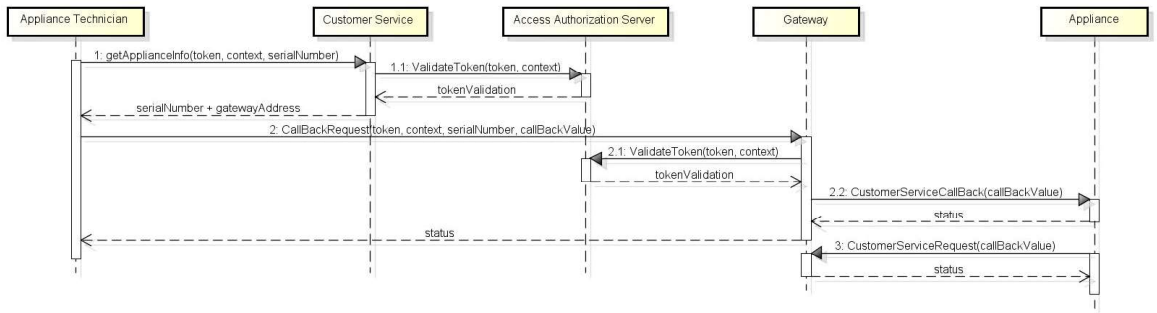

Figura 4 - Diagrama de sequência do procedimento de *Call Back*

4. Protótipo

O protótipo utiliza padrões de tecnologias conhecidas e bibliotecas de código aberto. O *Manufactor Domain* consiste em dois componentes, AppTec e CS. O AppTec foi implementado usando o framework Vaadin. O CS foi implementado como um *Web Service RESTful* usando a API JAX-RS. O *Customer Domain* consiste em um GW e vários Apps. As interfaces com o GW que representam o contexto Internet foram implantadas usando Java em um servidor HTTP. A interface da IoT foi construída num servidor CoAP usando a biblioteca Californium, que também foi usada para o App. Para cifragem foi utilizado o algoritmo AES de 128 bits, a fim de obter as chaves KKM e KEK, usando o Scandium, um subprojeto do Californium. O Scandium com suporte ao DTLS na versão 1.2 no contexto da IoT. O servidor de autenticação (AS) foi implementado seguindo a especificação do *OpenID* 2.0 usando a biblioteca Nimbus. O Nimbus fornece um IdM para AppTec, CS e GW, e também assegura que somente usuários autenticados e autorizados acessam os App. 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.

A pilha de protocolos usada na comunicação entre os contextos de Internet e IoT, feita usando HTTPS e CoAPs, é mostrada na Figura 5.

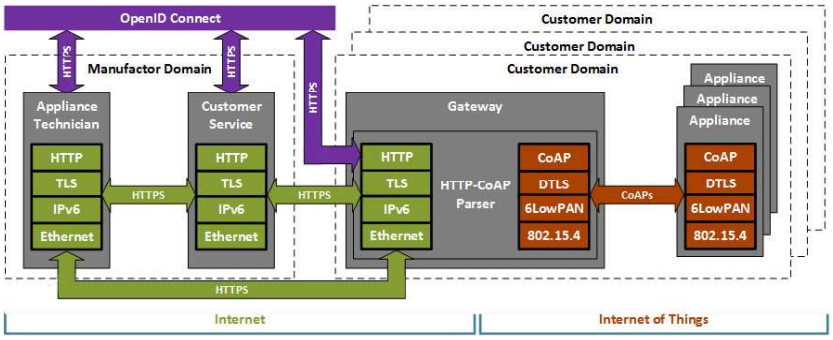

Figura 5 - Arquitetura do protótipo

4.1. Resultados

Os testes mediram o impacto do tamanho da mensagem na requisição e o tempo de resposta com e sem cifragem no ambiente de IoT. Também foi medido o tempo de resposta com e sem ativação do recurso de SSO com 50 Apps. Foi feito teste de stress para avaliar a quantidade de Apps suportados pelo protótipo. Na parte IoT utilizou-se Traffic Management (CBQ) para reduzir a banda de rede e imitar as limitações de 6LoWPAN.

A Figura 6 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. É possível observar um aumento de *overhead* para requisições com mensagens maiores de 1024 bytes, atingindo um *overhead* de 350 ms, para mensagens de 2048 bytes e 10 Apps. O tempo de resposta permanece abaixo de 1250 ms nas demais requisições, e as requisições CoAPs estão abaixo de 200 ms com 4096 bytes, 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. 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 IoT. Ou seja, não existe uma diferença significativa no tempo de resposta de 10 a 50 Apps, que é de cerca de 7% para CoAPs (CoAP + DTLS) 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, CoAP ou CoAPs, a proposta apresentou um pequeno *overhead* sobre 4096 bytes, sem afetar o tempo médio de resposta.

Para validar o SSO foram executados testes com CoAPs com e sem autenticação utilizando 50 App. Devido aos scripts de automação não foram visualizadas diferenças em relação ao tempo de respostas para mensagens até 2048 bytes. É possível concluir que não se ganha quantitativamente, mas sim qualitativamente. Com objetivo de avaliar o limite operacional da proposta, foram realizados testes com CoAPs e autenticação SSO (mensagens de 32 a 4096 bytes), iniciando com 10 Apps e incrementando o mesmo número, concluímos que 120 Apps é o limite suportado.

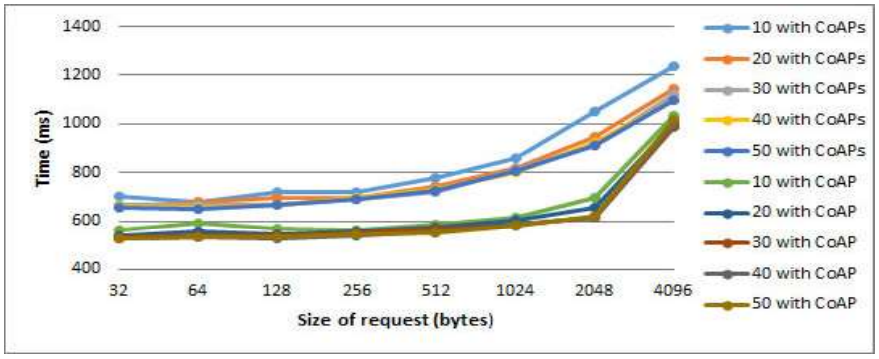

Figura 6 - Avaliação do Protótipo

5. Conclusão

O trabalho apresentou um método de autenticação para integrar um IdM do contexto da Internet à IoT. A ligação entre os contextos é fornecida por um *gateway* incapaz de visualizar o conteúdo das mensagens. As chaves simétricas são adequadas para a IoT para a proteção de mensagens fim-a-fim (a partir dos *appliances* até o *Customer Service*), evitando 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. O *gateway* fornece isolamento do *appliance* na Internet.

SSO "encapsulado no IdM" é adequado para a IoT, pois retira a necessidade do *appliance* precisar um esforço extra para interagir com um servidor de Internet, tal como proposto na literatura. O técnico pode acessar múltiplos *appliances* com uma única autenticação, sem necessitar saber uma senha diferente para cada *appliance,* e também sem necessitar utilizar a mesma senha para todos os *appliances* - prática que submete os *appliances* ao risco. O protótipo mostrou-se viável no seu tempo de resposta, variando o número de *appliances* e o tamanho das mensagens. A abordagem proposta não possui *overhead* significativo para o tempo de resposta de 10 a 50 Apps e mensagens de 32 a 4096 bytes por mensagem. 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.

6. Publicações

Witkovski, A., Santin, A., Abreu, V., Marynowski, J.. **An IdM and Key-Based Authentication Method for Providing Single Sign-On in IoT**. In proc. of IEEE Global Communications Conference, 2015, San Diego, pp.1-6.

Artigo: https://secplab.ppgia.pucpr.br/files/papers/2015-2.pdf

Dissertação: https://www.ppgia.pucpr.br/pt/arquivos/mestrado/dissertacoes/2015/Adriano_Witkovski_Dissertacao.pdf

Patente: Witkovski, A. ; Santin, A. ; Abreu, V. . **Mecanismo baseado em chaves e Gestão de Identidades para autenticação unificada em Internet das Coisas**. 2015, BR1020150306326, INPI - Instituto Nacional da Propriedade Industrial. Depósito: 07/12/2015; Depósito PCT: 07/12/2015. Brasil.

Referências

Atzori L., Iera A., and Morabito G. (2010), "The Internet of Things: A survey," Computer. Networks, vol. 54, pp. 2787–2805.

Babar S., Mahalle P., Stango A., Prasad N., and Prasad R. (2010), "Proposed security model and threat taxonomy for the Internet of Things (IoT)," in Proc. of the CCIS - Communications in Computer and Information Science, pp. 420–429.

Miorandi D., Sicari S., De Pellegrini F., and Chlamtac I. (2012), "Internet of things: Vision, applications and research challenges," Ad Hoc Networks, pp. 1497–1516.

Belapurkar A., Chakrabarti A., Ponnapalli H., Varadarajan N., Padmanabhuni S., and Sundarrajan S. (2009) "Distributed Systems Security: Issues, Processes and Solutions", John Wiley & Sons.

Shelby Z., Hartke K., and Bormann C., "The Constrained Application Protocol (CoAP)", IETF RFC 7252.

Rescorla E., Modadugu N., "Datagram Transport Layer Security Version 1.2.", IETF RFC 6347.

Van Thuan D., Butkus P., and Van Thanh D. (2014) "A user centric identity management for Internet of things," in Proc. of the ICITCS - IT Convergence and Security, pp. 1–4.

Fremantle P., Aziz B., Kopecky J., and Scott P. (2014), "Federated Identity and Access Management for the Internet of Things," in Proc. of the Int. Workshop on Secure Internet of Things, pp. 10–17.

Leo M., Battisti F., Carli M., and Neri A., (2014) "A federated architecture approach for Internet of Things security," in Proc. of the EMTC - Euro Med Telco, pp. 1–5.

Chadwick, D. W. (2009) "Federated Identity Management. Foundations of Security Analysis and Design" V. Heidelberg: Springer-Verlag Berlin, p. 96-120.

