Definindo um Protocolo de Autenticação Utilizando Bluetooth Low Energy para Dispositivos no Conceito de IoT

André R. Eichner[1], **Nathan H. da Silva**[1], **Tiago A. Rizzetti**[1]

[1] Colégio Técnico Industrial de Santa Maria – Universidade Federal de Santa Maria (UFSM)
Av. Roraima, 1000 – 97.105-900 – Santa Maria – RS – Brasil

{eichner, nathanhs}@redes.ufsm.br, rizzetti@ctism.ufsm.br

Abstract. *Because of its energy-saving attributes, Bluetooth Low Energy (BLE) has been a robust candidate to connect different types of sensors and actuators to communicate within the Internet of Things (IoT) environment. When compared to other data communication technologies, BLE provide a simple and low cost architecture, with a reduced size and scope similar to other technologies. In this context, a cryptanalysis and exhaustion-resistant authentication protocol is critical to ensure data integrity so that only previously authenticated communications packets can be accepted by other devices.*

Resumo. *Por conta de seus atributos de economia de energia, o Bluetooth Low Energy (BLE) vem sendo um forte candidato à conectar diferentes tipos de sensores e atuadores para que se comuniquem no âmbito de Internet das Coisas. Quando comparado com outras tecnologias de comunicação de dados, o BLE oferece uma arquitetura simples e barata, com dimensões reduzida e alcance similar às demais tecnologias. Nesse contexto, um protocolo de autenticação resistente a criptoanálise e ataques de força bruta é fundamental para garantir a integridade dos dados, fazendo com que somente os pacotes de comunicações previamente autenticados possam ser aceitos pelos demais dispositivos.*

1. Introdução

O *Bluetooth Low Energy* (BLE), é uma tecnologia desenvolvida e mantida pela Bluetooth Special Interest Group (SIG) [Chang 2014]. Anunciado em 2010, o BLE se popularizou rapidamente por se tratar de um novo modo de desenvolver dispositivos de automação de baixo custo e baixo consumo de energia, principalmente no conceito de *Internet of Things* (IoT).

Esta tecnologia está sendo cada vez mais aceita como meio de troca de dados padrão entre dispositivos de uso cotidiano, como por exemplo a comunicação entre pulseiras inteligentes e smartphones [Santos et al. 2016b]. Para estabelecer a comunicação entre dispositivos BLE, ambos podem utilizar protocolos que especificam regras de cliente e servidor. Segundo [Zhang et al. 2019] o *Generic Attribute Profile* (GATT) define as especificações necessárias para realizar comunicações entre dispositivos BLE, organizando-se em conceito de Serviços e Características.

Considerando o baixo consumo de energia e limitações apresentadas pelo protocolo GATT, o método de autenticação proposto por este trabalho auxilia na eficiência energética dos dispositivos, pois toda segurança envolvida é transmitida em apenas um

frame de dados. Este protocolo visa capacidades de segurança com características multi-plataformas, garantindo a autenticidade dos dados, por meio de algoritmos de geração de senhas de uso único baseadas no tempo – *Time-based One Time Password* (TOTP).

Nas próximas seções deste artigo estão o referencial teórico necessário para seu desenvolvimento, a descrição do protocolo desenvolvido, testes e resultados. E por fim, as conclusões, trabalhos futuros e as referências.

2. Referencial teórico

Esta seção contém todo embasamento teórico, no qual proporcionou a criação do algoritmo de autenticação proposto neste trabalho.

2.1. Internet of Things – IoT

Em [Santos et al. 2016a] os autores definem IoT como sendo a proliferação de objetos inteligentes com capacidade de sensoriamento, processamento e comunicação. Para [Ashton et al. 2009] a IoT é a presença difusa de objetos que permitem a telecomunicação sem fio que através de esquemas de endereçamento exclusivos, são capazes de interagir uns com os outros para alcançar objetivos em comuns. Já para [Gubbi et al. 2013] o IoT é a possibilidade de conectar diversos dispositivos independente do seu tipo, fazendo o uso de sensores e Identificadores de radiofrequência sem a necessidade da intervenção do ser humano.

2.2. Bluetooth Low Energy

Segundo [Reck 2017], ao utilizar a tecnologia BLE para trocar dados entre dispositivos, o consumo de corrente de pico pode chegar a ser até 33% menor que nas demais tecnologias, além do consumo de energia poder variar entre 50% e 90% se comparado com o Bluetooth Clássico. Entretanto há uma limitação nos pacotes GATT, no qual [Bluetooth 2013] define seu *payload* como (*ATT_MTU - 3*) octetos, no qual o acrônimo *ATT_MTU* referencia o número de octetos fornecidos pela camada inferior, ou seja, igual a 23.

Um enorme problema encontrado na comunicação BLE, é o fato de haver um *delay* de aproximadamente 5 segundos, a cada interação entre os dispositivos. Por exemplo, caso um dispositivo envie uma mensagem no tempo x segundos, outra mensagem somente poderá ser encaminhada no tempo $x + 5$ segundos, sendo este um dos principais motivos para a construção deste protocolo de autenticação.

2.3. One Time Password - OTP

Um algoritmo de geração de senhas descartáveis depende de duas informações, um *token* e um evento. Em 2011, [M'Raihi et al. 2011] propuseram o *Time-based One Time Password algoritm*, no qual é feita uma combinação entre um *token* e a hora atual do dispositivo que roda o algoritmo. Portanto, vários dispositivos calculam exatamente o mesmo valor se tiverem o mesmo *token* e seus relógios sincronizados.

2.4. Métodos de Autenticação

Segundo [Maziero 2019], a técnica de *login* e senha pode ser usada para autenticar de forma inequívoca um usuário em uma rede, entretanto, se o *login* e senha vincula a um

usuário forem conhecidas por outros, todos poderão autenticar-se normalmente, caracterizando assim um ataque de personificação.

Atualmente, a literatura está saturada de soluções de autenticação para serviços IoT. Em [Borgohain et al. 2015] e [Gupta 2015] os autores abordam soluções utilizando o método tradicional (usuário e senha) junto com um segundo fator, podendo ser uma senha de uso único (OTP). Porém esta abordagem implicaria em um número maior de dados trafegados na rede.

Os autores de [Vieira and Ruggiero 2007] abordam um estudo sobre autenticação baseada em desafio-resposta, porém a utilização deste protocolo implicaria em um número maior de interações entre os dispositivos.

Em [Zhang et al. 2019], os autores propuseram uma camada adicional (*middleware*) entre a aplicação e a camada *Generic Attribute Profile* (GATT), usando certificação digital como método de autenticação. Esta solução implica na necessidade de haver uma entidade certificadora para gerenciar os certificados.

2.5. Métodos de Criptografia

Para [Oliveira 2012] o *Advanced Encryption Standard* (AES) é o algoritmos mais popular para criptografia de chave simétrica. Com tamanho de bloco fixo (128 bits) e chave de tamanho variável (128, 192 ou 256 bits), seu processo é rápido tanto em software quanto em hardware, além de relativamente fácil de executar em dispositivos IoT.

3. Apresentação do Protocolo Proposto e Aplicações

O protocolo proposto atua na camada de aplicação do BLE, garantindo a autenticidade e a integridade dos dados gerados pelos sensores e atuadores de uma rede IoT. Uma vez que os dados gerados são encaminhados a outros dispositivos, estes podem ser interceptados e adulterados antes mesmo de chegar ao destino final, gerando uma série de consequências indesejadas. Como mostra a Figura 1, um atacante pode interceptar os pacotes e adulterá-los, ou mesmo, passar-se por um sensor.

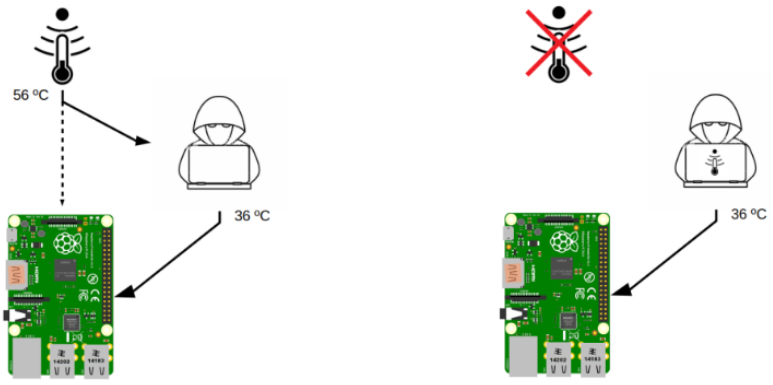

Figura 1. Possíveis ataques.

Seguindo a linha de raciocínio de [Gupta 2015], a autenticação em dois fatores resolve parcialmente o problema. Antes dos dados serem processados por outro dispositivo, os sensores devem provar que os dados são legítimos e foram emitidos por ele.

Cada sensor deve autenticar-se utilizando o método clássico: usuário e senha descartável (TOTP).

A confidencialidade da mensagem é fornecida através da encriptação dos dados, utilizando-se uma chave temporária fornecida pelo TOTP em ambas as partes. A integridade dos dados, é verificada através do resumo (HASH[1]) da mensagem encriptada. Na sequência será exposto um cenário hipotético, criado para expor com maiores detalhes as interações realizadas entre os dispositivos BLE.

Neste cenário hipotético, um dispositivo concentrador é responsável por receber dados críticos de vários sensores, tomando diversas decisões baseadas nestes dados. Para mostrar-se disponível a receber os dados dos sensores, ele emite sinais de advertência por um meio não guiado, aguardando solicitações de conexão dos diversos sensores espalhados pelo cenário hipotético.

Caso um sensor necessite encaminhar dados ao concentrador, ele deve escanear a rede e aguardar por um sinal de advertência do concentrador. Ao perceber este sinal, os sensores poderão solicitar conexão e aguardar o recebimento dos atributos e serviços deste concentrador. Como pode ser visto na Figura 2, após a exposição dos serviços, o dispositivo concentrador estará apto a receber dados dos sensores.

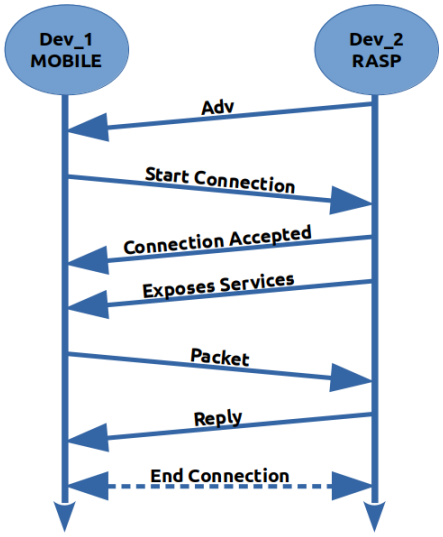

Figura 2. Protocolo de autenticação proposto.

Neste momento o sensor deve iniciar o processo de construção do pacote. Ao receber os dados, o concentrador dará inicio ao processo de autenticação, no qual o pacote é fracionado e submetido a diversos testes, os quais serão vistos com maiores detalhes, na subseção 4.4 e Figura 4.

4. Definições

Cada dispositivo da rede deve ser identificado por um número de identificação único (UUID) e (*seed*), no qual serão apresentados separadamente no decorrer desta seção.

[1]Segundo [Roman et al. 2011] hash é um algoritmo criptográfico que converte uma mensagem de tamanho variável em um resumo de mensagem com tamanho fixo. [Maziero 2019] diz que a função hash é de cálculo rápido e direto, mas o cálculo de sua inversa é impossível ou computacionalmente inviável.

4.1. *Universally Unique Identifier* – UUID

Como em qualquer sistema de autenticação, todo dispositivo cadastrado deve ser unicamente identificado, para que não haja problemas de auditoria. Nesse sistema, a identificação é baseada em 4 octetos, onde cada dispositivo recebe um conjunto de 2^{32} bits (4 bytes) – num total de 4.294.967.296 possibilidades. O UUID deve ser a primeira parte a ser anexada ao pacote, anexado em texto plano, para que o concentrador consiga facilmente encontrá-lo em um banco de dados.

4.2. Semente ou *seed* do usuário

Além do UUID, cada dispositivo recebe um conjunto de 2^{256} bits (32 bytes), denominado de semente ou *seed*. A este conjunto de bytes deve ser aplicada uma função de HASH, usando o *Secure Hash Algorithm* com 256 bits de comprimento (SHA-256), no qual seu resultado serve de entrada no algoritmo TOTP.

4.3. Algoritmo para geração do TOTP

O cálculo é realizado seguindo [M'Raihi et al. 2011], no qual: HMAC(*seed*, evento[2]) utilizando função de hash criptográfica (SHA256). O resultado deve ser obtido da seguinte forma: quatro primeiros octetos e em seguida os quatro últimos. Ex: TOTP=1234[...]5678; TOTP=12345678.

4.4. Estrutura do Pacote

O pacote é montado de acordo com a Figura 3, no qual os primeiros 4 bytes é o número único que identifica o sensor na rede (UUID). Os próximos 8 bytes são obtidos através do processo de truncamento dos primeiros octetos do hash do pacote cifrado. E os próximos 8 bytes são obtidos através do processo de truncamento do *Time-based One Time Password* (TOTP), no qual são os 4 primeiros bytes e os 4 últimos bytes do cálculo.

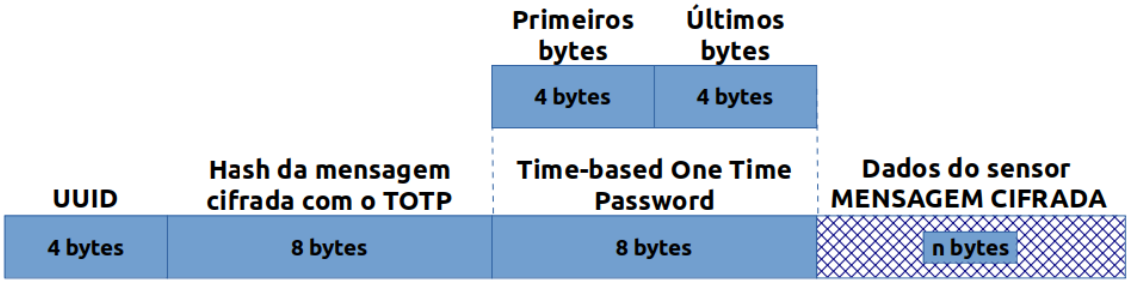

Figura 3. Estrutura do pacote.

A estrutura do pacote foi desenvolvida de acordo [Bluetooth 2013], no qual especifica a unidade máxima de transmissão (MTU) para o GATT, como sendo de 20 octetos. Ao passo que, com os primeiros 20 bytes do pacote é possível verificar a autenticidade e integridade do pacote como um todo.

[2]Ex(C++): tempoAtual={time()/[*t*]}; substituir o *t* pelo tempo de vida da senha (se *t*=30, a senha se torna inválida em até 30 segundos)

5. Cenário de Testes

Para desenvolver este trabalho, foi utilizado um Raspberry Pi3 Model B como dispositivo concentrador, com sistema operacional Raspbian (baseado em Linux Debian), instalado em um cartão microSD Sandisk-Ultra, de 32GB de classe 10 e com Bluetooth v4.1. Como nó escravo, foi utilizado um celular Motorola Moto-G7 Plus com sistema operacional Android 9 (Pie) e com Bluetooth 5.0. Para realizar a comunicação utilizando o *Bluetooth Low Energy*, foram desenvolvidos dois softwares, um para o Raspberry – escrito na linguagem C++ e utilizando bibliotecas nativas do Linux, além de outras – e outro para o Android – escrito na linguagem Java.

Para esse estudo são considerados os tempos total da comunicação (conexão e desconexão do dispositivo), e tempos total e parcial de cada bloco do protocolo (análise e desmembramento do pacote, cálculo de hash e TOTP). Para tal, um cronômetro foi desenvolvido na linguagem C++ e incorporado junto ao software. A Figura 4 mostra os locais das aferições dos dados – desconsiderar o campo de procura do usuário no bando de dados – bem como, todos os passos que o dispositivo concentrador realizou.

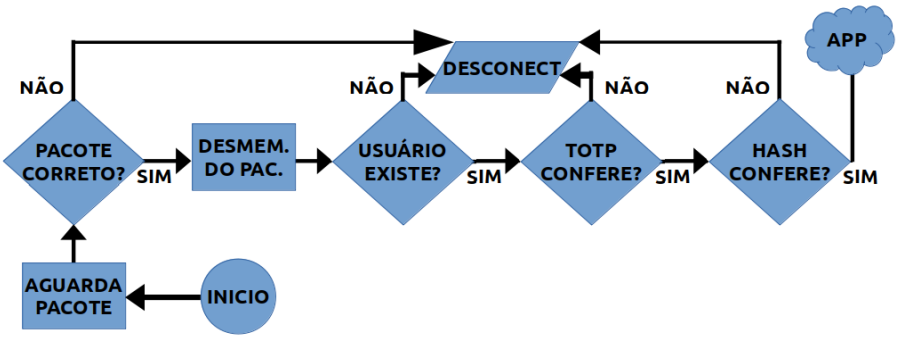

Figura 4. Fluxograma do concentrador pós recebimento do pacote.

Após o recebimento do pacote, o dispositivo concentrador verifica seu tamanho e inicia o processo de separação dos dados em partes, cada qual com uma função e tamanho especificados no item 4.4 deste trabalho. Na sequência, é apresentado como o dispositivo concentrador manipula os estes dados.

- O primeiro passo é a busca em um banco de dados, pelo UUID recebido. No qual, neste momento a *seed* do usuário torna-se conhecida.
- De posse da *seed*, o dispositivo calcula seu hash, captura o *timestamp* [3] podendo dar início ao computo do TOTP.
- O próximo passo é concatenar os 4 primeiros e 4 últimos octetos obtidos pela função TOTP e compará-los aos 8 últimos octetos do pacote recebido, para aferir que os dados são autênticos.
- Para verificar a integridade dos dados recebidos, o concentrador deve calcular o hash dos n bytes da mensagem cifrada, comparando os 8 primeiros octetos do resultado, com os 8 octetos (entre o 5º e o 12º) do pacote recebido.

Caso ocorra algum resultado inválido na verificação de qualquer dos passos anteriores, o concentrador desconecta automaticamente e mantém um *log* da conexão, como visto na Figura 4. Em caso de exito em todas as verificações, os dados validados são finalmente encaminhados à aplicação.

[3] Em C++ a função *timestamp* retorna o tempo em segundos, desde 00:00:00 UTC, 1 de Janeiro de 1970.

6. Resultados Finais e Conclusão

A Figura 5 mostra um gráfico referente ao tempo necessário para processar os dados recebidos, utilizando este protocolo. Nele pode ser visto que a diferença entre o menor e maior tempo foi de apenas 130 µs. Já a Tabela 1 conta com o tempo médio de cada processamento, utilizando este protocolo, no qual o tempo médio para rodar todos os passos do protocolo foi de 2,852ms e seu desvio padrão foi de 36,47.

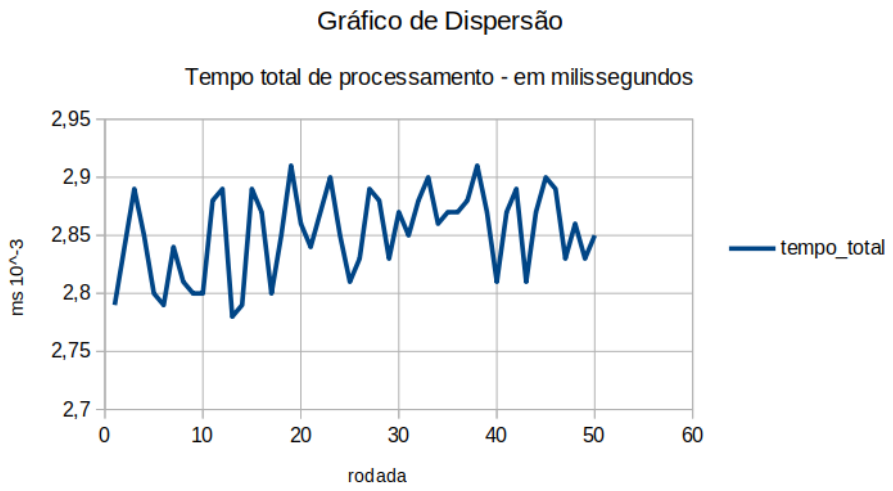

Figura 5. Gráfico de dispersão do tempo total de processamento.

O quadro 1 conta com a média das 50 rodadas de testes, junto com seu desvio padrão. Uma importante observação, a última linha do quadro leva em consideração todo processo, carregando consigo os tempos entre troca de funções, retorno das funções, entre outros processamentos do aplicativo.

Tabela 1. Resultados obtidos

OPERAÇÃO	TEMPO MÉDIO	DESVIO PADRÃO
Análise do pacote	3,661 µs	18,08
Desmembramento do pacote	3,610 µs	20,34
Cálculo de hash	193 ms	25,86
Cálculo do TOTP	209 ms	31,8
Tempo total	2,852 ms	36,47

Na maioria dos protocolos de autenticação encontrados na literatura, os autores não se preocupam com a quantidade de dados trocados entre os dispositivos. Ao passo que este protocolo está sendo projetado com a finalidade de prover um método de autenticação para dispositivos IoT, em que junto ao protocolo existe um método para aferir se os dados recebidos são íntegros.

Como trabalho futuro, pretende-se explorar falhas e/ou vulnerabilidades que este protocolo de autenticação eventualmente possa ter, além de aplicá-lo em um ambiente real, para de verificar sua estabilidade.

Referências

Ashton, K. et al. (2009). That 'internet of things' thing. *RFID journal*, 22(7):97–114.

Bluetooth, S. (2013). Core Specification v4.1 . `https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=282159`. [Online; accessed 23-August-2019].

Borgohain, T., Borgohain, A., Kumar, U., and Sanyal, S. (2015). Authentication systems in internet of things. *arXiv preprint arXiv:1502.00870*.

Chang, K.-H. (2014). Bluetooth: a viable solution for iot?[industry perspectives]. *IEEE Wireless Communications*, 21(6):6–7.

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

Gupta, U. (2015). Application of multi factor authentication in internet of things domain. *arXiv preprint arXiv:1506.03753*.

Maziero, C. A. (2019). Sistemas operacionais: Conceitos e mecanismos. *Livro aberto*.

M'Raihi, D., Machani, S., Pei, M., and Rydell, J. (2011). Totp: Time-based one-time password algorithm. *Internet Request for Comments*.

Oliveira, R. R. (2012). Criptografia simétrica e assimétrica-os principais algoritmos de cifragem. *Segurança Digital [Revista online]*, 31:11–15.

Reck, M. S. (2017). Beacons ble-bluetooth low energy-design e análise de um sistema de localização indoor. .

Roman, R., Najera, P., and Lopez, J. (2011). Securing the internet of things. *Computer*, 44(09):51–58.

Santos, B. P., Silva, L. A., Celes, C., Borges, J. B., Neto, B. S. P., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N., and Loureiro, A. (2016a). Internet das coisas: da teoria à prática. *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC)*.

Santos, D. F. d. S. et al. (2016b). Controle de fluxo adaptativo para gateways bluetooth low-energy aplicado a sistemas de monitoramento remoto de pacientes. .

Vieira, G. Y. M. and Ruggiero, W. V. (2007). Algoritmos para tokens de autenticação. In *Proceedings, Conferência IADIS Ibero-Americana www/internet, Vila Real, Portugal*, pages 1–7.

Zhang, Q., Liang, Z., and Cai, Z. (2019). Developing a new security framework for bluetooth low energy devices. *Computers, Materials & Continua*, 58:457–471.