Análise Experimental de Técnicas de Hashing em Dispositivos IoT para Soluções de Segurança

Eduardo E. P. Mosca[1], **Rafael L. Gomes**[1]

[1]Universidade Estadual do Ceará (UECE), Fortaleza, Ceará, Brasil.

eduardo.mosca@aluno.uece.br, rafa.lopes@uece.br

Abstract. *The evolution of micro-controllers has enabled the diffusion of the Internet of Things (IoT) devices in any kind of environment. However, those devices still lack security solutions, once it has lower computational resources, hampering the applying of techniques, for example, Hashing algorithms. In this context, this paper aims to identify the viability of Hashing techniques in IoT devices to increase the reliability and evolve the security level, such as the do not burden the limited resources which are already limited. This identification was realized through real experimentation with two IoT devices (ESP8266, and ESP32) and an analysis of collected data. These experiments were realized the hashing algorithm in security solutions: SHA2-256, SHA2-512, SHA3-256, SHA3-512 e Blake2B-512. Through the experiments realized it was possible to identify the applicability limit of these hashing techniques to support security solutions when IoT devices are considered.*

Resumo. *A evolução dos microcontroladores possibilitou a difusão de dispositivos de Internet das Coisas (IoT) em quaisquer ambientes. Contudo, estes dispositivos ainda carecem de soluções de segurança dentro deles, visto que estes possuem poucos recursos computacionais, dificultando o uso de técnicas a serem aplicadas, como por exemplo algoritmos de Hashing. Dentro deste contexto, este artigo visa identificar a viabilidade de técnicas de Hashing em dispositivos IoT a fim de aumentar a confiabilidade e evoluir o nível de segurança destes dispositivos, bem como não onerar seus recursos computacionais já limitados. Esta identificação foi realizada por meio de experimentação real com dois dispositivos IoT (ESP8266 e ESP32) e uma análise de dados coletados. Nestes experimentos, foram analisados os principais algorítimos de hashing aplicados em soluções de segurança: SHA2-256, SHA2-512, SHA3-256, SHA3-512 e Blake2B-512. A partir dos experimentos realizados foi possível identificar o limite de aplicação das técnicas de hashing para soluções de segurança quando considerados os dispositivos IoT.*

1. Introdução

A modernização dos aspectos de hardware dos microcontroladores trouxe, além de uma evolução na capacidade computacional, componentes eletrônicos mais baratos, habilitando assim a difusão massiva de dispositivos nos mais diversos ambientes. Assim, foi possível a automatização de tarefas e criação de novos serviços mais complexos a darem suporte as atividades do dia-a-dia da sociedade. Este contexto criou a era da Internet das Coisas (IoT) [Li et al. 2018]. IoT inclui utensílios domésticos como lâmpadas,

veículos, etiquetas de endereçamento, dispositivos médicos, sensores, câmeras de vigilância e quaisquer outros objetos conectados com a internet que possam ter envolvimento com o ambiente a qual está inserido.

IoT pode ser considerada como uma extensão da Internet tradicional, a qual permite que diversos dispositivos como televisões, sensores, ares-condicionados, etc, possam conectar-se entre si criando uma interação mútua, de modo que os consumidores possam se aproveitar desses benefícios para ter uma melhor qualidade de vida. Assim, IoT ingressou em um cenário de tecnologias modernas e de telecomunicações que atuam em diversos contextos, como casas inteligentes [Stojkoska and Trivodaliev 2017], [Zemrane et al. 2020], cidades inteligentes [Samih 2019], e indústria [Garrido-Hidalgo et al. 2019]. Consequentemente, a introdução de dispositivos IoT nestes mais variados ambientes gera um grande volume de dados que circulam entre estes dispositivos para a Internet, visto que todos esses objetos são capazes de coletar e de transmitir informações auxiliando na execução de várias tarefas.

Alguns desses dados que circulam possuem um teor sigiloso, o que pode gerar situações as quais as informações críticas usadas de forma maliciosa podem vir a comprometer o valor dos serviços prestados [da Rocha 2019]. Adicionalmente, a exposição desses dados para entidades não desejadas, pode ferir as leis de privacidade, como a Lei Geral de Proteção de Dados (LGPD - 13709/2018) no Brasil e *General Data Protection Regulation* (GDPR - 2016/679) na Europa [Li and Palanisamy 2018, Rizou et al. 2020, de Oliveira 2019].

Portanto, a popularização dos dispositivos IoT trouxe também a preocupação com aspectos de segurança da informação, devido a características como número massivo de conexões e vulnerabilidades existentes nesses dispositivos. Essas características tornam as redes IoT uma poderosa ferramenta na expansão de ataques cibernéticos, o que justifica a necessidade de estudos e abordagens para a implantação de soluções que deem suporte às redes IoT e que implementem sistemas de detecção de ameaças e proteção para as aplicações executadas sobre essas redes.

Uma abordagem bastante difundida para atender aspectos de segurança é aplicação de algoritmos de Hashing, que permitem transformar uma grande quantidade de dados em um identificador único (não apresente colisão ou que tenha uma probabilidade de tal baixa) e irreversível. Assim, algoritmos de Hashing são utilizados para assegurar a integridade de dados transmitidos e impedir acesso indevido, visto que violações de segurança não resultariam em acesso aos dados do usuário em si.

A aplicação de algoritmos de Hashing torna-se então uma alternativa interessante para mitigar vulnerabilidades de segurança existentes em dispositivos IoT. Contudo, a implantação desses algoritmos de Hashing dentro de dispositivos IoT ainda é um desafio em aberto, visto que os algoritmos de Hashing existentes possuem características e complexidades diferentes, as quais afetam diretamente a disponibilidade dos, já limitados, recursos computacionais dos dispositivos IoT.

Dentro deste contexto, buscou-se identificar a viabilidade de técnicas de Hashing em dispositivos IoT a fim de aumentar a confiabilidade e evoluir o nível de segurança destes dispositivos, bem como não onerar seus recursos computacionais já limitados. Esta

identificação foi feita por meio de experimentação realizada com dois dispositivos IoT (ESP8266 e ESP32) e uma análise de dados coletados. Nestes experimentos, foram analisados os principais algorítimos de hashing aplicados em soluções de segurança: SHA2-256, SHA2-512, SHA3-256, SHA3-512 e Blake2B-512.

Os experimentos realizados sugerem que existe a viabilidade de se colocar quaisquer um dos algorítimos apresentados nos dispositivos testados, uma vez que todos os aparelhos conseguiram realizar os testes em menos de 1 milissegundo, devendo ser avaliado apenas os requisitos que melhor satisfazerem o sistema que está sendo desenvolvido.

O restante deste artigo está organizado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados a experimentação em redes IoT e técnicas de Hashing; A Seção 3 descreve uma fundamentação teórica para a análise experimental realizada, enquanto que a Seção 4 descreve a configuração dos experimentos e analisa os resultados obtidos; Por fim, a Seção 5 apresenta a conclusão e os trabalhos futuros.

2. Trabalhos Relacionados

Nesta seção, resumem-se os trabalhos encontrados na literatura relacionados à solução para segurança em redes IoT usando Hashing.

Rao et al.[Rao and Prema 2019b] avaliam as abordagens existentes de autenticação e privacidade do usuário, tais como MD5 e a família SHA para esquemas de assinatura digital, realizam testes com o algorítimo BLAKE2b e propõem uma versão modificada do mesmo com o intuito de trazerem uma versão chamada cBLAKE2b, que traz benefícios como uma melhora no consumo de energia pelos dispositivos, mais resistência a ataques e um aumento na geração e verificação das assinaturas. Entretanto todos os testes e experimentação foram feitos visando o Raspberry Pi-3, que se trata de um dispositivo que possui uma capacidade consideravelmente maior de recursos quando comparado com ESP32 e ESP8266.

Landge et al. [Landge and Satopay 2018] apresentam os conceitos de segurança dentro dos dispositivos IoT, alguns dos problemas que são encontrados para que ocorra a comunicação entre esses dispositivos e o algorítimo de Hash MD5, também apresentado para solucionar os problemas de segurança apresentados. Contudo, o algoritmo apresentado pelos autores como solução para os problemas de segurança é executável com boa performance na maioria dos dispositivos devido sua simplicidade. Entretanto, é um algoritmo ultrapassado, que pode ser facilmente descriptografado, além de ser passível de vulnerabilidade de colisões.

Parmar et al. [Parmar et al. 2021] apresentam um estudo sobre a rede Blockchain e os problemas de segurança dos dispositivos IoT e discutem como os dispositivos podem se integrar e lucrar com a descentralização da Blockchain. Todavia, não são apresentados resultados práticos sobre a utilização e segurança dos dispositivos dentro da rede.

Rao et al. [Rao and Prema 2019a] fazem uma análise de diversos algorítimos de Hash, comparando seus tempos de geração de chave, tempo de assinatura, tempo de verificação e de Hash em diversos sistemas operacionais usando como base um Raspberry-Pi-Model-3b. Entretanto, os testes gerados por Rao et al. [Rao and Prema 2019a] são feitos dentro de dispositivos que possuem uma quantidade considerável de recursos quando comparados à um ESP32 ou ESP8266.

Chien et al. [Chien 2021] propõe um esquema de autenticação anônima usando Hashing composto, alcançando 4 destaques, propor um novo esquema de autenticação IoT anônimo, usar um sistema de pré-cálculo de vetores para lidar com ataques de negação de serviço ou conexões não confiáveis, apresentar bons desempenhos em computação, comunicação, segurança e descrever uma instância da aplicação do esquema proposto no protocolo padrão MQTT. Porém os testes foram realizados utilizando dois tipos de autenticação entre os dispositivos e o servidores, além de não serem especificados os dispositivos utilizados para realizar os testes.

Islambouli et al. [Islambouli et al. 2020] formulam um modelo de Hash distribuído baseado em uma distribuição eficiente e com descarregamento de computação em Hashing ao utilizar como base os algorítimos de Hash da família SHA. Todavia, os autores apresentam um modelo que depende também de dispositivos móveis, o que traz uma eficiência em termos de recursos, mas ainda pode ser suscetível a falhas, além de não levar em consideração o algorítimo Blake.

Percebe-se que nos trabalhos existentes na literatura, com o objetivo de aplicar técnicas de Hashing em redes IoT, nenhum desses trabalhos descreve o levantamento deste contributo: um cenário de avaliação experimental utilizando dispositivos IoT heterogêneos. Portanto, este artigo busca agregar novos aspectos de pesquisa em relação ao estado da arte no que se refere a soluções de segurança para redes IoT.

3. Fundamentação Teórica

Nesta seção iremos descrever os algoritmos de Hashing considerados na análise experimental realizada, bem como serão detalhados os aspectos de hardware dos dispositivos IoT usados. Estas informações irão embasar os aspectos que serão analisados nos experimentos e a discussão dos resultados.

3.1. Algoritmos de Hashing

Foram escolhidos cinco algoritmos de Hashing já consolidados em soluções tradicionais: SHA2-256, SHA2-512, SHA3-256, SHA3-512 e Blake2B-512. Estes são conhecidos na comunidade científica, onde neste trabalho, a análise é feita quanto à viabilidade de execução em dispositivos IoT heterogêneos.

- SHA2-256 / 512: São funções Hash computadas a partir de 8 palavras podendo cada uma ter 32 bits para o SHA2-256 ou 64 bits para o SHA2-512, utilizando diferentes quantidades de rodadas, podendo ser 64 ou 80 rodadas, e constantes aditivas, possuindo uma estrutura virtualmente idênticas.
- SHA3-256 / 512: É baseado no algorítimo *Keccak* que por sua vez é baseado em um algorítimo de esponja que consiste em um ampla função randômica ou permutação aleatória a permitir inserir qualquer quantidade de dados e emitir qualquer quantidade de dados, enquanto age como uma função pseudo-aleatória em relação as entradas anteriores garantindo uma grande flexibilidade.
- Blake2B-256 / 512: É um algorítimo baseado no BLAKE, mas removendo a adição de constantes às palavras da mensagem da função de rodadas do BLAKE, altera para duas rodadas de rotação constantes, simplifica os paddings, adiciona o bloco de parâmetro (XOR'ed) com os vetores de inicialização (mesmos do sha2-512) e reduz o numero de rodadas de 14 para 10 (no caso do BLAKE2b).

3.2. Hardware e Software dos Dispositivos IoT

A seguir descreveremos os dispositivos IoT utilizados na análise de desempenho, ressaltando suas características de Hardware e Software que influenciam diretamente a execução das técnicas de hashing e abordagens de segurança de dados.

- ESP8266/WEMOS: A placa WEMOS D1 R2 é uma placa compatível com o Arduino Uno, mas que é controlada pelo módulo ESP8266EX, oferecendo conectividade *WiFi* nativa, sendo uma ótima opção para projetos IoT. A WEMOS D1, tem um conector micro USB e pode ser programada utilizando a IDE do Arduino. Tem 11 pinos de I/O digitais e 1 entrada analógica com o máximo de 3.3V, com programação compatível com Arduino e *NodeMCU*. No geral, o ESP8266 possui uma CPU 32-bit RISC rodando a 80 MHz de clock com 64 KB de memória RAM de instruções e 96 KB de dados.
- ESP32: O módulo ESP32s é um módulo *WiFi* de alta performance se comparado com os seus semelhantes, com um baixíssimo consumo de energia. É uma evolução do ESP8266, com maior poder de processamento e *Bluetooth Low Energy*(BLE) 4.2 embutido. A placa possui o chip ESP32 com antena embutida, uma interface *usb-serial* e um regulador de tensão 3.3V. A programação pode ser feita em LUA ou usando a IDE do Arduino através de um cabo micro-usb. Possui 4 *Megabytes* de memória flash, uma CPU Xtensa dual-core de 32 bits operando a um clock máximo de 240 MHz e uma memória RAM de 520Kbytes.

4. Experimentos

Esta seção apresenta os experimentos realizados para avaliar o desempenho dos algoritmos de hashing em dispositivos IoT. Para realizar os experimentos, foi implantado um cenário com dispositivos reais para analisar a viabilidade das técnicas, bem como o impacto destas nos dispositivos IoT. A implementação dos algoritmos de Hashing foi feita diretamente dentro dos dispositivos descritos na Seção 3.2, visto que os mesmos não possuem bibliotecas para suporte a tal (por isso faz-se necessário uma abordagem de baixa complexidade e compatível com as limitações de hardware). A seção 4.1 apresenta a configuração do experimento e a seção 4.2 discute os resultados.

4.1. Configuração do Experimento

No cenário de teste montado foram usados: ESP32 [Datasheet 2021] e ESP8266/WEMOS D1 MINI [Datasheet 2015] que apesar de não se encaixarem puramente na categoria de dispositivos IoT, por funcionarem através da instância de um sistema operacional, podem ser considerados dispositivos de baixa capacidade computacional. O principal objetivo dos experimentos era medir o tempo de processamento dos algoritmos de Hashing nos diferentes hardwares, habilitando assim a identificação da viabilidade dessas técnicas em dispositivos heterogêneos.

A realização dos experimentos foi realizada utilizando uma adaptação biblioteca do github[1] de algorítimos Hash para o contexto dos dispositivos IoT usados, testando os algorítimos SHA2-256, SHA2-512, SHA3-256, SHA3-512 e BLAKE2b-512. Com relação aos dados a serem utilizados para as funções de Hash, foram escolhidos, aleatoriamente, dados sensíveis de acordo com as Leis de Privacidade, onde os parâmetros

[1] https://github.com/rweather/arduinolibs

recebem um índice e são sorteados para compor uma tabela de dados para os testes [de Oliveira 2019]. Adicionalmente, variou-se gradativamente o tamanho da entrada de dados de 5 caracteres até 100 (sendo coletadas 100 amostra para cada algorítimo e cada palavra gerada), a fim de possibilitar uma análise mais completa do comportamento dos algoritmos nos dispositivos IoT. É válido ressaltar que o processo de Hash ocorre em anexo a outros processos que são executados nos dispositivos IoT. Portanto, a demanda de recursos não pode inviabilizar e/ou impactar negativamente os demais serviços (comprometendo o QoS/QoE), como por exemplo monitoramento, automação, etc.

Além disso, no caso do dispositivo ESP8266 foram geradas dois tipos de testes, sendo o primeiro usando a função eraseConfig(função que apaga as configurações e limpa o cache do dispositivo) utilizada para evitar que o dispositivo guardasse dados na memória cache e consequentemente afetasse o desempenho do teste já que neste primeiro caso a avaliação é feita sob o tempo de execução da primeira vez em que a função é chamada. No segundo caso o equipamento age sem interrupções permitindo que seja armazenado dados na memória cache do dispositivo, fazendo com que tenha um tempo de execução seja um pouco maior nas primeiras execuções e rapidamente consiga ser reduzido.

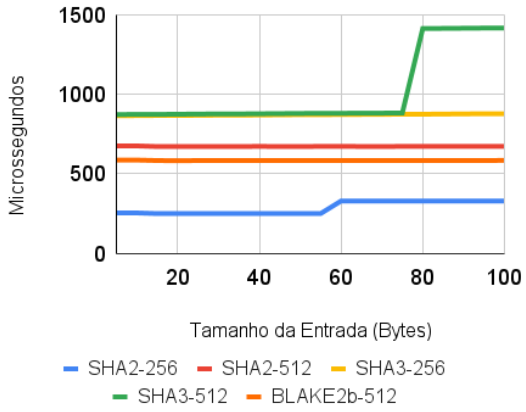
4.2. Resultados

A seguir, as Figuras 1(a), 1(b) e 1(c) apresentam os resultados de tempo de processamento para os casos do dispositivo ESP8266 com eraseConfig, ESP8266 sem eraseConfig e ESP32 além da quantidade de memoria RAM utilizada por cada algorítmo nas condições de teste descritos na Figura 1(d), respectivamente. Os dados nas figuras citadas estão em microssegundos (μs $(10^{-6}s)$ quando referente as figuras de processamento dos algoritmos e em *bytes* quando referente a figura da quantidade de RAM utilizada, a fim de possibilitar uma visualização compatível com o nível de processamento dos dispositivos. Os dados das figuras a seguir estão diretamente relacionados ao desempenho de cada algorítimo e o quanto de tempo foi levado para processarem os códigos de Hash.

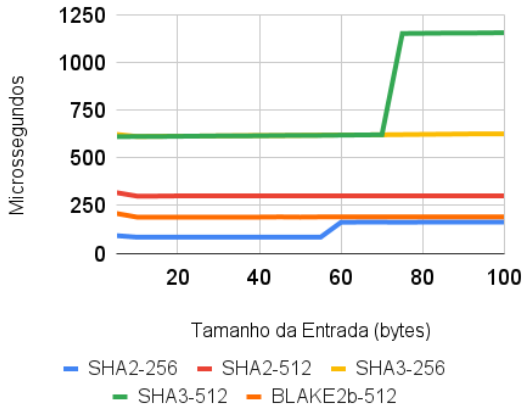
Ao analisarmos os resultados dos dispositivos ESP8266 nas Figuras 1(a) e 1(b) percebe-se o impacto do uso da função eraseConfig na execução das técnicas de hashing. O eraseConfig ao limpar a memória cache do ESP8266 força o dispositivo a refazer todos os cálculos feitos ao processar o hashing, resultando em uma elevação do tempo de processamento. Desta forma, desabilitar a função eraseConfig resulta em uma redução de até três vezes no tempo de processamento, como é o caso do algoritmo SHA2-256. Similarmente, os demais algoritmos também sofrem um aumento no tempo de processamento, como pode ser observado no gráfico da Figura 1(a).

Na Figura 1(c) tem-se um padrão de comportamento similar ao encontrado nas Figuras 1(a) e 1(b), mas, numericamente, o tempo de processamento do dispositivo ESP32 é cerca de três vezes menor que o ESP8266 sem eraseConfig e quatro vezes menor que o ESP8266 com eraseConfig. Desta forma, nota-se que o ESP32 é muito mais adequado para serviços IoT que necessitem de segurança baseado em hashing e de um tempo de resposta diminuto.

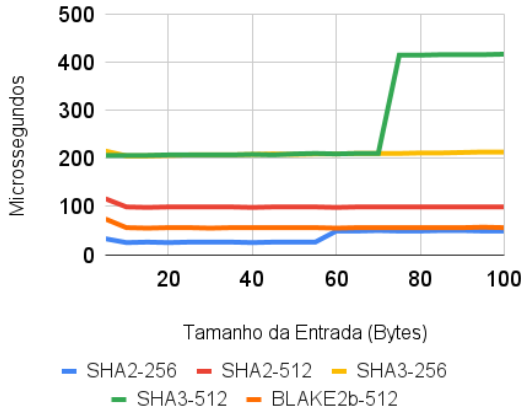
Com relação aos algoritmos de Hashing, tem-se um *trade-off* claro entre ocupação de memória RAM e nível de segurança. O algoritmo SHA3 é mais resistente a ataques de extensão de comprimento (*Length extension attack*) e é baseado no algorítimo Keccak, resultando em um nível de segurança elevado. Por outro lado, o algoritmo SHA2 é mais



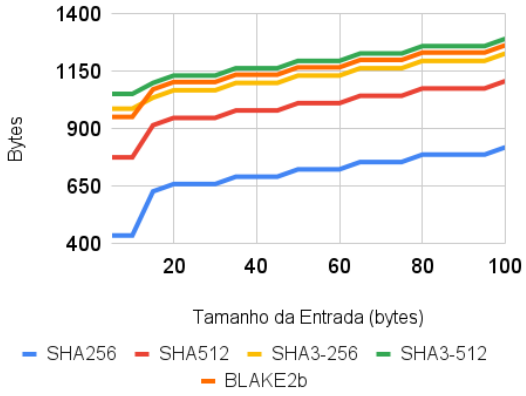
(a) Tempo de Processamento no ESP8266 com eraseConfig.



(b) Tempo de Processamento no ESP8266 sem eraseConfig.



(c) Tempo de Processamento no ESP32.



(d) Quantidade de Memória RAM alocada (em bytes).

Figura 1. Resultados do uso de processamento e memória nos dispositivos
Criado peos autores deste artigo.

rápido e leve (menos alocação de memória RAM), mas menos seguro. Assim, se um certo serviço IoT necessita gerar um alto número de Hashes simultaneamente, é necessário analisar o nível de segurança necessário, bem como a disponibilidade de memória RAM. Por exemplo, o algoritmo SHA2-256, em média, pode processar o dobro de Hashes em comparação aos algoritmos SHA3-256, SHA2-512 e Blake2b.

No geral, os algorítimos baseados no SHA3 mostram-se um pouco mais lento com relação ao seu desempenho, uma vez que consomem mais tempo de processamento devido à maior quantidade de rodadas no processamento interno do algorítimo e do uso de matrizes para realizar o processo, necessitando de mais tempo para realizar o processo, mas sendo algo que não pode ser mudado devido a maneira como é implementada o algorítimo *Keccak* que é base para a implementação do SHA3. A maior diferença pode ser vista quando comparado com os resultados da Figura 1(c). Adicionalmente, vale a pena ressaltar que o algorítimo BLAKE2b-512 possui um nível de eficácia parecido com o SHA3 e leva uma quantidade de tempo menor para o processamento do Hash,

além da quantidade de memória RAM gastada para processar os dados ser maior que a do SHA3-256 e menor que a do SHA3-512, além de consumir menos energia devido à menor quantidade de giros realizadas no processamento interno do algoritmo.

5. Conclusão e Trabalhos Futuros

A sociedade atual vem inovando os serviços existentes nas cidades, onde grande parte desses serviços inovadores é baseado no paradigma de Internet das Coisas, permitindo a evolução da acessibilidade e agilidade nas tarefas diárias dos usuários. Contudo, os dispositivos IoT, em sua maioria, ainda necessitam de soluções de segurança adequadas, pois muitas vezes estão vulneráveis aos mais diversos ataques cibernéticos existentes.

Uma das abordagens aplicadas para lidar com esses aspectos de segurança é a implementação de funções de Hash para garantir a integridade dos dados e das configurações dos dispositivos. Todavia, a implantação desses técnicas de Hashing dentro de dispositivos IoT ainda é um desafio, pois as técnicas existentes possuem características e complexidades diferentes, as quais afetam diretamente a disponibilidade dos, já limitados, recursos computacionais dos dispositivos IoT. Dentro deste contexto, este artigo analisou, a partir de uma experimentação real, a viabilidade de técnicas de Hashing em dispositivos IoT a fim de aumentar a confiabilidade e evoluir o nível de segurança destes dispositivos, bem como não onerar seus recursos computacionais já limitados.

Os resultados obtidos sugerem que a utilização dos algorítimos Hash dentro dos dispositivos abordados é viável, visto que o tempo utilizado para processar os cálculos Hash estão na casa dos μs $(10^{-6}s)$, além de terem sido utilizados algorítimos com uma boa confiabilidade e utilização dentro da área de segurança que são o BLAKE2b e SHA3, que além de serem os mais pesados em termos de processamento ainda são praticáveis sem causarem maiores impactos em outras funções.

Como trabalho futuro, pretende-se desenvolver um novo mecanismo de integridade e segurança para dispositivos IoT baseado em Hashing, a fim de viabilizar sua aplicação em conjunto com os demais serviços que executam nestes. Esta nova proposta irá considerar os aspectos de escalabilidade e recursos computacionais limitados dos sistemas IoT.

Agradecimentos

Os autores agradecem a Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico - FUNCAP (Processo DEP-0164-00242.01.00/19) pelo apoio financeiro.

Referências

Chien, H.-Y. (2021). Highly efficient anonymous iot authentication using composite hashing. In *2021 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 1–7. IEEE.

da Rocha, C. P. (2019). Segurança da informação: A iso 27.001 como ferramenta de controle para lgpd. *Revista de Tecnologia da Informação e Comunicação da Faculdade Estácio do Pará*, 2(3).

Datasheet, E. (2015). Esp8266ex datasheet. *Espr. Syst. Datasheet*, pages 1–31.

Datasheet, E. (2021). Esp32s datasheet. *Espr. Syst. Datasheet*, pages 1–60.

de Oliveira, N. S. (2019). Segurança da informação para internet das coisas (iot): uma abordagem sobre a lei geral de proteção de dados (lgpd).

Garrido-Hidalgo, C., Olivares, T., Ramirez, F. J., and Roda-Sanchez, L. (2019). An end-to-end internet of things solution for reverse supply chain management in industry 4.0. *Computers in Industry*, 112:103127.

Islambouli, R., Sweidan, Z., Mourad, A., and Abou-Rjeily, C. (2020). Towards trust-aware iot hashing offloading in mobile edge computing. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 2216–2221. IEEE.

Landge, I. A. and Satopay, H. (2018). Secured iot through hashing using md5. In *2018 fourth international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB)*, pages 1–5. IEEE.

Li, C. and Palanisamy, B. (2018). Privacy in internet of things: From principles to technologies. *IEEE Internet of Things Journal*, PP:1–1.

Li, Z., Liu, Y., Liu, A., Wang, S., and Liu, H. (2018). Minimizing convergecast time and energy consumption in green internet of things. *IEEE Transactions on Emerging Topics in Computing*.

Parmar, M. et al. (2021). Hashing based data transaction and optimized storage for iot applications. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(5):1206–1215.

Rao, V. and Prema, K. (2019a). Comparative study of lightweight hashing functions for resource constrained devices of iot. In *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, volume 4, pages 1–5. IEEE.

Rao, V. and Prema, K. (2019b). Light-weight hashing method for user authentication in internet-of-things. *Ad Hoc Networks*, 89:97–106.

Rizou, S., Alexandropoulou-Egyptiadou, E., and Psannis, K. (2020). Gdpr interference with next generation 5g and iot networks. *IEEE Access*, PP:1–1.

Samih, H. (2019). Smart cities and internet of things. *Journal of Information Technology Case and Application Research*, 21(1):3–12.

Stojkoska, B. L. R. and Trivodaliev, K. V. (2017). A review of internet of things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140:1454–1464.

Zemrane, H., Baddi, Y., and Hasbi, A. (2020). Internet of things smart home ecosystem. In *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks*, pages 101–125. Springer.