Desempenho de Algoritmos Criptográficos de Hashing em um Sistema IoT baseado em Névoa Computacional

Franklin Magalhães Ribeiro Jr.[1, 2], Carlos A. Kamienski[1]
[1]Universidade Federal do ABC (UFABC); [2]Instituto Federal do Maranhão (IFMA)
franklin.junior@ufabc.edu.br; carlos.kamienski@ufabc.edu.br

Resumo: Com a computação em névoa é possível lidar com sistemas IoT sensíveis a latência. Porém, o uso da névoa computacional expõe vulnerabilidades de segurança, visto que os dados podem ser maliciosamente corrompidos ou fabricados na borda do sistema IoT. Algoritmos criptográficos de *hashing* podem garantir a integridade e a autenticidade dos dados. Por isso, esse trabalho avaliou o desempenho de 11 algoritmos de *hashing*, para uma carga simulada de 5000 sensores. Foram aferidos o uso de CPU e RAM da névoa, além da vazão e do atraso dos pacotes. Foi observado que o desempenho dos algoritmos foi similar para o ambiente de IoT avaliado.

1. Introdução

Sistemas IoT baseados em névoa computacional são capazes de analisar os dados localmente na borda [1], reduzindo a necessidade da tomada de decisões por um servidor em nuvem. Contudo, por estar mais próxima da borda, a névoa está mais suscetível a ataques. Isso porque dispositivos maliciosos podem transmitir pacotes corrompidos ou não autênticos, com dados que diferem do contexto da aplicação de IoT [2].

Os algoritmos criptográficos de *hashing* podem lidar com a autenticidade e com a integridade dos dados na IoT. Porém, devido às restrições nos recursos computacionais da névoa, é essencial escolher a política de segurança considerando essas limitações [2]. Por isso, este trabalho realizou uma avaliação de desempenho acerca dos algoritmos da família SHA [3, 4] e BLAKE2 [3] na névoa computacional. Este trabalho está organizado da seguinte maneira, a Seção 2 explana a fundamentação e os trabalhos relacionados, a Seção 3 explica a metodologia, a Seção 4 apresenta e discute os resultados e, finalmente, a Seção 5 apresenta a conclusão e os trabalhos futuros.

2. Fundamentação e Trabalhos Relacionados

A névoa é uma camada computacional entre os sensores e a nuvem, por isso ela permite que o sistema IoT tome decisões mais rápidas [1]. Para tornar a névoa mais confiável, é possível utilizar os algoritmos SHA-2, SHA-3 e BLAKE2 [3]. Em [4] foi apresentada uma solução de IoT utilizando o SHA-3, mas o estudo não avaliou o impacto do algoritmo. Em [3] foi realizada uma avaliação de desempenho dos algoritmos de *hashing* na IoT. Contudo, o ambiente de avaliação em [3] usou poucos dispositivos, sendo que um cenário real de IoT tem milhares de sensores.

3. Metodologia

Para simular os sensores foi utilizado o simulador *SenSE* [5]. O *SenSE* gera pacotes que são recebidos pela névoa através do servidor *ChirpStack*[1]. Na névoa é gerado o código de *hash* para o *payload* (de 263 bytes) de cada pacote, além disso, o código de *hash* é concatenado ao *payload*. Em seguida a névoa envia esses dados a nuvem via MQTT[2].

A névoa possui o sistema operacional Linux Ubuntu 18.04, com processador Intel i5 de 1.60GHz e memória RAM de 8GB, já a nuvem possui os mesmos componentes, mas com uma CPU de 4 núcleos de 2.4GHz. No estudo foram avaliadas 4 métricas: (i) o atraso entre o tempo de chegada de um pacote e a criação do mesmo, (ii) a vazão dos dados entre a névoa e a nuvem, (iii) o uso de CPU da névoa e (iv) o uso de memória RAM da névoa. Para os experimentos foram avaliados 11 cenários e cada um foi executado apenas uma vez. Em cada cenário foi utilizado um algoritmo de *hashing* da biblioteca *PyCryptodome*[3] (SHA-1, SHA224, SHA256, SHA384, SHA512, SHA-3 224, SHA-3 256, SHA-3 384, SHA-3 512, BLAKE2s e BLAKE2b), para uma carga simulada de 5000 sensores.

O tempo total do experimento simulado em cada cenário foi de 60 segundos, onde todas as métricas foram coletadas numa periodicidade de 1 segundo. As métricas de CPU e RAM foram coletadas através do comando *ps*

[1] https://www.chirpstack.io/
[2] https://mqtt.org/
[3] https://pycryptodome.readthedocs.io

aux do Linux, a vazão pelo comando *ifstat* e o atraso pela subtração entre o *timestamp* de criação e o *timestamp* de chegada do pacote na nuvem. No experimento foi desconsiderado o uso de CPU e RAM do *ChirpStack* na névoa.

4. Resultados e Discussão

Após a execução dos experimentos para os 11 cenários, foram obtidas as médias das métricas e os intervalos de confiança, para um nível de confiança de 90% (Fig. 1.). Através dos resultados foi possível observar que o algoritmo BLAKE2s apresentou um atraso maior que os algoritmos SHA-2 384, SHA-2 512 e SHA-3 224 (Fig. 1.a). Também foi possível observar que, com exceção do BLAKE2s, todos os demais algoritmos apresentaram um atraso similar dentro do intervalo de confiança (margem de erro). Em relação aos resultados das métricas de vazão (Fig. 1.b), do uso de CPU (Fig. 1.c) e de memória RAM (Fig. 1.d), foi percebido que todos os algoritmos ficaram tecnicamente empatados ao considerarmos o intervalo de confiança.

Com relação as métricas de desempenho avaliadas, foi observado que todos os algoritmos apresentaram impacto similar para o uso dos recursos da névoa, porém é válido ressaltar que atualmente o algoritmo SHA-1 é considerado vulnerável e que a criptografia dos algoritmos da família SHA-2, em alguns casos, pode estar suscetível a ataques. Portanto, há evidências de que para um projeto de IoT baseado em névoa computacional, é mais relevante considerar o nível de segurança dos algoritmos criptográficos de *hashing* do que o desempenho dos mesmos.

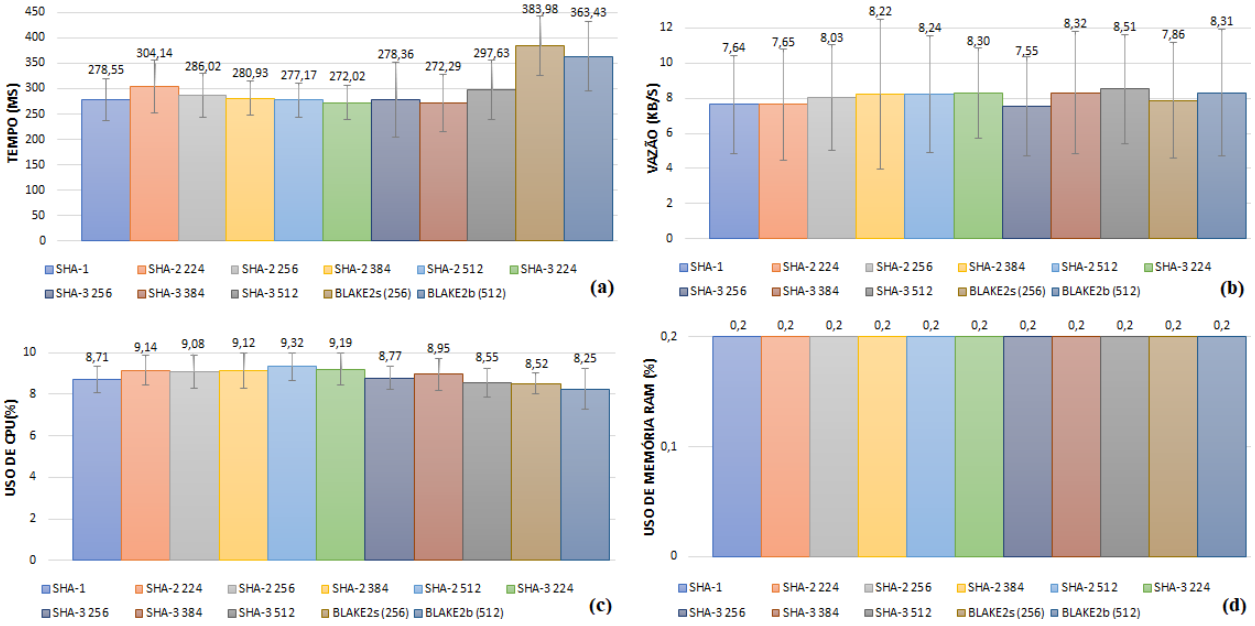

Fig. 1. Desempenho dos algoritmos em relação ao (a) atraso dos pacotes; (b) vazão; (c) uso de CPU *(máximo=800%)* e (d) uso de RAM na névoa.

5. Conclusão

Nessa pesquisa foi realizada uma avaliação de desempenho dos algoritmos de *hashing* em um sistema IoT baseado em névoa, para uma carga simulada de 5000 sensores. No estudo foi observado que o desempenho dos algoritmos da família SHA-1, SHA-2, SHA-3 e BLAKE2 foi similar. Portanto, o projetista de IoT deve escolher o algoritmo com base no nível de segurança requerido pela névoa, já que nesse contexto em específico, não houveram evidências sobre a correlação entre a escolha do algoritmo de *hashing* e o desempenho computacional da névoa. Como trabalho futuro, espera-se avaliar outros algoritmos de segurança para IoT, mas com foco na confidencialidade dos dados.

Referências

[1] S. Yi, Z. Hao, Z. Qin and Q. Li, *Fog Computing: Platform and Applications*, 2015 Third IEEE HotWeb, Washington, DC, 2015.

[2] M. Mukherjee et al., *Security and Privacy in Fog Computing: Challenges*, in IEEE Access, 2017, doi: 10.1109/ACCESS.2017.2749422.

[3] V. Rao and K. V. Prema, *Comparative Study of Lightweight Hashing Functions for Resource Constrained Devices of IoT*, 2019 4th *CSITSS*, Bengaluru, India, 2019, pp. 1-5, doi: 10.1109/CSITSS47250.2019.9031038.

[4] N. Sharma, H P. Sultana, R. Singh, S. Patil, *Secure Hash Authentication in IoT based Applications*, Procedia Computer Science, Volume 165, 2019, Pages 328-335, ISSN 1877-0509, doi.org/10.1016/j.procs.2020.01.042.

[5] I. Zyrianoff, F. Borelli, C. Kamienski (2017). *SenSE? Sensor Simulation Environment: Uma ferramenta para geração de tráfego IoT em larga escala*. Simpósio Brasileiro de Redes e Sistemas Distribuídos (SBRC), 2017.