Trabalho publicado na VI ERSI-RJ
Como citar:
ROCHA, Igor; SCHOTT, Richard; VERLY, Pedro; LAZARIN, Nilson. Análise de desempenho e conformidade em bibliotecas criptográficas para Internet das Coisas. In: ESCOLA REGIONAL DE SISTEMAS DE INFORMAÇÃO DO RIO DE JANEIRO (ERSI-RJ), 6. , 2019, Duque de Caxias. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020.

Análise de desempenho e conformidade em bibliotecas criptográficas para Internet das Coisas

Igor Rocha, Richard Schott, Pedro Verly, Nilson Lazarin

[1]Bacharelado em Sistemas de Informação – Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (Cefet/RJ) – Nova Friburgo, RJ – Brazil

{igor.freitas23, pedrofverly, richard.bfrj, nilsonmori}@hotmail.com

Abstract. *In the context of the Internet of Things (IoT) there are major challenges related to data security, mainly due of devices with low computational power present in this scenario. This paper presents a survey and performance comparison between cryptographic libraries specially optimized for Arduino. Experiments were performed considering performance, memory consumption, storage and compliance.*

Resumo. *No contexto de IoT percebe-se grandes desafios relacionados a segurança de dados, principalmente devido ao baixo poder computacional dos dispositivos presentes neste cenário. Este artigo apresenta uma pesquisa e um comparativo de desempenho entre bibliotecas criptográficas otimizadas especialmente para a plataforma Arduino. Foram executados testes de desempenho, consumo de memória, armazenamento e conformidade.*

1. Introdução

A forma mais comum de se garantir sigilo na transferência de dados é a adoção de técnicas criptográficas. Entretanto, seu uso pode ser comprometido uma vez que no contexto de IoT os dispositivos sofrem limitações de espaço e processamento [Zhang et al. 2014]. Optar por algoritmos que utilizam chave e bloco menor que 128bits, não é uma opção, pois, [Blaze et al. 1996] recomenda fortemente o uso chaves criptográficas com no mínimo 90bits.

No contexto de IoT é comum o uso de criptografia simétrica, dada a facilidade de implementação e melhor desempenho, se comparado com a criptografia assimétrica. Os modos de operação mais utilizados são: 1) ECB *(Eletronic Codebook Mode)*, no qual, a informação é dividida em blocos e cada bloco é submetido ao processo de criptografia individualmente. A desvantagem deste método é que a partir de uma determinada chave, o resultado cifrado será sempre o mesmo; 2) CBC *(Cipher Block Chaining)* envolve um encadeamento dos blocos de texto em claro com os blocos cifrados anteriormente. O primeiro bloco da informação sofre uma operação XOR com um vetor de inicialização (IV) que deve ser gerado aleatoriamente. Dessa forma é garantindo que o o texto cifrado final seja sempre diferente, mesmo utilizando a mesma chave criptográfica [Dworkin 2001].

Este trabalho apresenta uma pesquisa sobre bibliotecas disponíveis para implementação de sigilo na troca de informações entre dispositivos com baixo poder computacional, acompanhado de testes de embarcação, testes de conformidade e testes de desempenho entre as bibliotecas encontradas, além de uma implementação própria para atender as especificações de [NIST 2001] e [Dworkin 2001].

2. Metodologia

Escolhemos, para execução dos testes, a placa Arduino UNO por ser um hardware muito utilizado em projetos de computação embarcada. Durante a pesquisa, buscamos encontrar bibliotecas criptográficas que implementassem o algoritmo AES *(Advanced Encryption Standard)*, que fossem compatíveis com o Arduino e que tivessem o código fonte disponíveis no GitHub. Utilizamos um buscador de internet com a seguinte expressão de pesquisa: *(library AND arduino) AND (AES OR "Advanced Encryption Standard"OR Rijndael) site:https://github.com*. Foram encontradas 10 bibliotecas, listadas na Tabela 1.

Tabela 1. Resumo das bibliotecas analisadas

Biblioteca	Chave	Modo	Licença	Teste de Embarcação	NIST FIPS 197	NIST SP 800-38A
AES for microcontrollers (Arduino & Raspberry pi) https://github.com/spaniakos/AES	128 192 256	CBC	Copyright	OK	X	X
AES_128bit_Arduino https://github.com/beranm14/AES_128bit_Arduino	128	ECB	-	OK	OK	X
AESLib https://github.com/DavyLandman/AESLib	128 192 256	ECB CBC	GPL	OK	OK	X
AES library for Arduino Board https://github.com/indrabagus/arduino-aes	128	CBC	Apache	OK	OK	X
ArduinoAES256 https://github.com/qistoph/ArduinoAES256	256	ECB	Copyright	OK	X	X
Arduino-AES https://github.com/DanielVukelich/Arduino-AES	128 92 256	ECB	GPL	X	-	-
Arduino-AES https://github.com/edogaldo/Arduino-AES	128 192 256	ECB CBC	Copyright	OK	OK	X
Arduino Cryptography Library https://github.com/rweather/arduinolibs	128 192 256	ECB CBC	MIT	OK	OK	X
Micro-aes https://github.com/DarkCaster/Micro-AES-Arduino	128 192 256	ECB CBC	MIT	OK	OK	X
Ptolemy-XV https://github.com/octaviovieira/Ptolemy-XV	128 192 256	ECB	-	X	-	-
Securino https://github.com/nilsonmori/securino	128	ECB CBC	GPL	OK	OK	OK

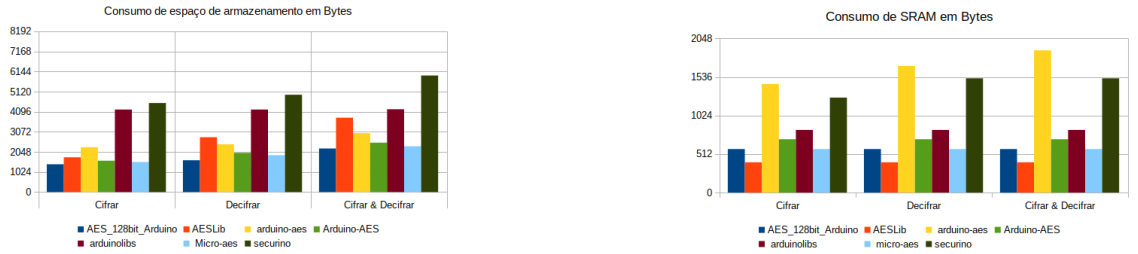
O teste de embarcação é a primeira etapa de exclusão em nosso processo de análise na qual as bibliotecas foram importadas para o Arduino IDE e adicionadas em um *sketch* de teste. Consideramos os dados informados pelo compilador avr-gcc ao término da compilação. O teste de conformidade é a última etapa de exclusão a qual baseia-se na premissa de que o sigilo de uma mensagem não deve depender de seu algoritmo e sim da força da chave, pois as cifras devem ser públicas possibilitando testes e análises, uma vez que, algoritmos privados ou alterados podem conter backdoors ou fragilidades. Neste teste são cifrados e decifrados blocos texto em claro definidos pela *Advanced Encryption Standard* (AES) (FIPS PUB 197) [NIST 2001] e pela *Advanced Encryption Standard Algorithm Validation Suite* (AESAVS) [Bassham III 2002].

Para avaliar o desempenho das bibliotecas consideramos as seguintes métricas: i) Armazenamento: avalia o consumo de espaço de armazenamento considerando as informações fornecidas pelo compilador apenas com a importação da biblioteca; ii) Memória: avalia o consumo de memória SRAM para o processamento de um bloco; iii) Processamento: avalia a quantidade de ciclos de processamento necessários para o processamento de um bloco e a medição foi realizada utilizando um contador de ciclos.

3. Resultados

As bibliotecas *Arduino-AES* e *Ptolemy-XV* não passaram no teste de embarcação, pois suas variáveis globais excederem o espaço de memória SRAM da placa. A primeira usa 2706 bytes (132%) e a segunda usa 5365 bytes (260%) de memória dinâmica. Estas duas bibliotecas não seguiram para a próxima etapa. As oito bibliotecas restantes foram submetidas ao teste de conformidade. As bibliotecas *AES for microcontrollers (Arduino & Raspberry pi)* e *ArduinoAES256* não foram aprovadas, pois o bloco cifrado pelas bibliotecas era diferente da especificação, e não seguiram para a próxima etapa.

Na figura 1a são apresentados os resultados do teste de consumo de espaço de armazenamento. Os testes foram divididos em três partes: i) considera a embarcação apenas da função de cifrar; ii) considera apenas a função de decifrar; iii) considera as duas funções. Foram submetidas as 6 bibliotecas que passaram nas etapas de conformidade e embarcação, além da biblioteca desenvolvida neste trabalho (Securino). Destaca-se o desempenho da biblioteca AES_128bit_Arduino que consome apenas 1416 bytes para cifrar, 1622 bytes para decifrar e 2220bytes para cifrar e decifrar. Na figura 1b são apresentados os resultados do teste de consumo memória SRAM. Os testes foram divididos em três partes: i) considera o consumo de memória da função de cifrar; ii) da função decifrar; iii) considera o consumo das duas funções. Destaca-se o desempenho da biblioteca AESLib que consome em média 402 bytes de SRAM para cifrar, decifrar ou ambas.



(a) espaço de armazenamento **(b) espaço de memória**

Figura 1. Resultado dos testes

Na figura 2 são apresentados os resultados do teste de consumo de ciclos de processamento. Os testes foram divididos em três partes: i) considera o consumo da função de cifrar; ii) da função decifrar; iii) considera o consumo das duas funções. Novamente destaca-se o desempenho da biblioteca AESLib que consome em média 4567 ciclos para cifrar, 5192 ciclos para decifrar e 9716 ciclos para cifrar e decifrar.

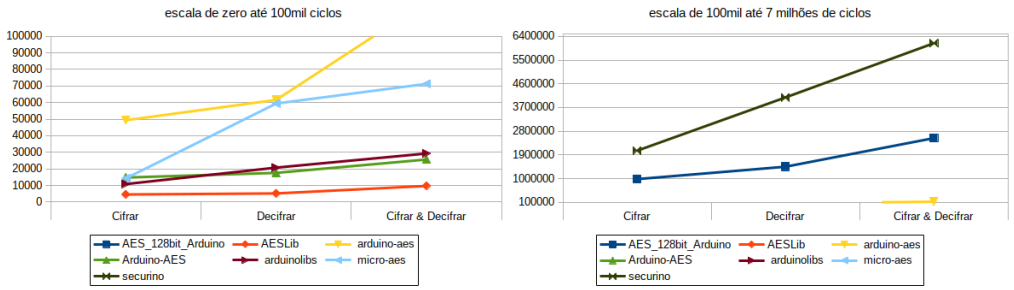


Figura 2. Resultado dos testes de ciclos de processamento.

A maioria das bibliotecas analisadas que implementam o modo CBC, o implementam com vetor de inicialização (IV) fixo. Esta forma de implementação não está em conformidade com a NIST SP 800-38-A [Dworkin 2001] que determina: i) O modo de operação CBC requer um IV como entrada, além do bloco de texto em claro; ii) O IV deve ser imprevisível e gerado para cada execução da operação cifrar; iii) O IV não precisa ser secreto, portanto o IV pode ser transmitido com o texto cifrado. Apenas a biblioteca Securino, desenvolvida neste trabalho, implementa esse nível de segurança. Ressaltamos a importância do uso de CBC com IV aleatório, pois em aplicações de computação embarcada os comandos enviados através de meios inseguros tendem a serem fixos. Dessa forma, o uso de uma biblioteca em modo ECB ou CBC com IV fixo são equivalentes e não devem ser adotados.

4. Conclusão

Foram analisadas 10 bibliotecas disponíveis no GitHub, destas, duas foram descartadas por não poderem ser embarcadas no Arduino UNO, outras duas foram descartadas por não atenderem às especificações do AES. Seguiram para a fase de experimentos 6 bibliotecas obtidas no GitHub e a biblioteca desenvolvida pelos autores deste trabalho. Durante os experimentos podemos destacar o desempenho de processamento e memória da biblioteca AESLib e o desempenho de espaço de armazenamento da biblioteca AES_128bit_Arduino.

Apontamos uma ressalva para o uso de todas as bibliotecas obtidas na pesquisa, visto que algumas só implementam o modo ECB e as outras implementam o modo CBC com vetor de inicialização fixo. Esta forma de implementação não está em conformidade com a NIST SP 800-38-A. Por fim apresentamos a biblioteca Securino, que atende a todas as especificações do AES, além de implementar o modo CBC com vetor de inicialização aleatório, garantindo assim um melhor nível de segurança que as outras bibliotecas. Como trabalhos futuros, será necessária a melhoria no desempenho desta biblioteca, principalmente a questão de ciclos de processamento necessários para cifrar e decifrar.

Referências

Bassham III, L. E. (2002). The advanced encryption standard algorithm validation suite (aesavs). *NIST Information Technology Laboratory*.

Blaze, M., Diffie, W., Rivest, R. L., Schneier, B., and Shimomura, T. (1996). Minimal key lengths for symmetric ciphers to provide adequate commercial security. a report by an ad hoc group of cryptographers and computer scientists. Technical report.

Dworkin, M. (2001). Nist special publication 800-38: Recommendation for block cipher modes of operation. *US National Institute of Standards and Technology,¡ http://csrc. nist. gov/publications/nistpubs/800-38a/sp800-38a. pdf*.

NIST (2001). 197, advanced encryption standard (aes), national institute of standards and technology, us department of commerce, november 2001. *Link in: http://csrc. nist. gov/publications/fips/fips197/fips-197. pdf*.

Zhang, Z.-K., Cho, M. C. Y., Wang, C.-W., Hsu, C.-W., Chen, C.-K., and Shieh, S. (2014). IoT security: ongoing challenges and research opportunities. In *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on*, pages 230–234. IEEE.