https://eventos.utfpr.edu.br//sicite/sicite2020

Comparação de uma comunicação segura com AES em sistema embarcado e computador

Comparison of secure communication with AES between embedded system and computer

Augusto Cézar Ferreira
augustof@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Natássya B. F. Silva
natassyasilva@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

RESUMO

Sistemas embarcados são a combinação entre hardware e software designado a realizar funções específicas. Estes, em geral, utilizam a rede para se comunicar com outros dispositivos. Por esta razão, a comunicação segura é importante para garantir a confidencialidade das informações, provida por meio de algoritmos de encriptação. Não obstante, atualmente, poucas técnicas de segurança são implementadas para este tipo de dispositivo. Este artigo apresenta uma avaliação em tempo de execução do AES utilizando a biblioteca criptográfica Relic desenvolvida para sistemas embarcados em uma comunicação entre dois processos de um computador, entre um computador e um sistema embarcado e entre dois sistemas embarcados. Foi considerado o tamanho das chaves suportadas, o tamanho dos dados e a arquitetura usada para mensurar a relação do tempo gasto pela criptografia no tempo total da comunicação. Os resultados mostraram que a arquitetura tem grande impacto na criptografia junto com o tamanho dos dados. No entanto, mesmo na comunicação entre sistemas embarcados a porcentagem do tempo de criptografia em toda a comunicação segura é menor que 14%.

PALAVRAS-CHAVE: Criptografia. Segurança. Sistemas Embarcados.

ABSTRACT

Embedded systems are the combination between hardware and software designed to perform specific functions. An embedded system is generally a part of a larger system that uses the network to communicate with other devices. For this reason, secure communication is of great importance to guarantee the confidentiality information, which can be provided by using cryptography. The security is not a requirement when designing an embedded system, causing few techniques to be implemented for the security of these devices. This paper presents an evaluation of run-time performance of AES using a cryptography library developed for embedded systems, called Relic, in a communication between two processes in a general purpose computer, between a general purpose computer and an embedded system and between two embedded systems. It is considered the key sizes, the data size and the architecture, also measuring the relation of the time spent by cryptography and the total time for the communication. The results demonstrated that the architecture has a greater impact on the cryptography times along with the message sizes. However, even for the communication between embedded systems, the percentage of the encryption and decryption time on the entire secure communication is smaller than 14%.

KEYWORDS: Cryptography. Security. Embedded Systems.

Recebido: 19 ago. 2020.
Aprovado: 01 out. 2020.

INTRODUÇÃO

Sistemas embarcados são a combinação de software e hardware que executa uma determinada tarefa continuamente (BAAR, 1999). Muitos destes sistemas estão presentes no cotidiano em alguns automóveis, *drones* e atualmente até mesmo dentro de muitas residências realizando a automação de diversas tarefas.

O fato de um sistema embarcado ser compacto, tanto em relação a software quanto para hardware, fez com que sua utilização fosse cada vez mais cogitada, tendo em sua maioria suporte para diversos tipos de conexões como Wi-fi, Bluetooth, Ethernet e entradas seriais para conexões de sensores e atuadores aumentando ainda mais suas possibilidades de uso.

Paralelamente a todos os benefícios que um sistema embarcado traz aos seus usuários, a preocupação em relação a segurança de software e hardware é igualmente crescente. Por exemplo, a facilidade de manuseio e a utilização de redes *wireless* faz com que este tipo de sistema seja alvo de ataques maliciosos, com vulnerabilidades que podem dar acessos a usuários não autorizados. Atacantes podem violar a confidencialidade dos dados e a privacidade dos usuários e causar danos físicos e lógicos ao sistema. Estas preocupações devem ser enfatizadas, uma vez que a maioria das soluções de segurança propostas atualmente não são projetadas para este tipo de sistema com menores capacidades de processamento e memória e fonte restrita de energia (SILVA, JR e SOUZA, 2013).

A criptografia é um dos métodos mais utilizados para garantir a segurança dos dados em sistemas computacionais. Ela fornece diferentes níveis de segurança com base nas necessidades da aplicação. E além disso é capaz de garantir integridade, autenticidade e confidencialidade para o sistema (STALLINGS, 2012).

Estes métodos podem ser divididos em algoritmos assimétricos ou de chave pública, onde os maiores exemplos são RSA, Diffie-Hellman, e Criptografia de Curvas Elípticas, e simétricos onde os maiores exemplos são DES (*Data Encryption Standard*), Blowfish e AES (*Advanced Encryption Standard*), também conhecido como Rijndael. Os métodos e as configurações devem ser escolhidos de acordo com os requisitos da aplicação e os recursos disponíveis do sistema.

Hyncica *et al.* (2011) avaliaram 15 diferentes algoritmos simétricos em três microcontroladores com instruções de 8, 16 e 32 bits. Entre eles estava incluso o AES com uma chave de 128 bits no modo de operação EBC (*Electronic CodeBook*). Os experimentos utilizando a biblioteca criptográfica TomCrypt mostraram que os algoritmos com melhor performance foi o AES, Twofish e o SAFER, com uma significante margem se comparado com as outras cifras avaliadas.

Silva *et al.* (2016) compararam o AES com diferentes tamanhos de chave e com variação do tamanho dos dados a serem encriptados. O experimento foi conduzido em um sistema embarcado com instruções de 32 bits e em um computador para verificar o impacto da arquitetura no processo. Os resultados mostraram que o tamanho da chave teve uma influência de aproximadamente 10% no tempo da criptografia, enquanto o tamanho da mensagem teve uma influência de aproximadamente 89%. Apesar de comparações do AES já terem sido apresentadas na literatura, nenhuma pesquisa anterior mediu o impacto de suas diferentes configurações na comunicação segura para sistemas embarcados.

Este artigo apresenta a avaliação de desempenho do algoritmo simétrico de criptografia AES em um computador e um sistema embarcado (Raspberry Pi 3), para prover confidencialidade dos dados em uma transmissão feita em um canal não seguro. A implementação do algoritmo foi baseada na biblioteca criptográfica Relic para sistemas embarcados. Os experimentos foram executados com mensagens cifradas sendo trocadas entre dois processos em um computador de, entre um computador e um sistema embarcado e entre dois sistemas embarcados.

MATERIAIS E MÉTODOS

Neste trabalho, uma comunicação segura é considerada como uma comunicação com confidencialidade que ocorre em um canal não seguro, que pode ser alvo de ataques maliciosos podendo causar riscos a todo o sistema. A comunicação segura pode ser obtida aplicando a encriptação com o AES na mensagem antes que ela seja enviada através do canal. Portanto, a mensagem original pode apenas ser obtida se o receptor conhecer a mesma chave para realizar a decriptação.

Para que a comunicação segura ocorra, a biblioteca Relic foi utilizada, um moderno conjunto de ferramentas criptográficas, com ênfase em flexibilidade e eficiência. A biblioteca é desenvolvida principalmente na linguagem C, fazendo-a portável para diferentes plataformas se recompilada e com o reuso do código que não é dependente da arquitetura (ARANHA, 2020). A biblioteca contém protocolos de cifra de bloco que inclui o algoritmo AES, que foi utilizado neste trabalho.

O AES é um algoritmo de encriptação de dados que utiliza o conceito de cifra de blocos onde os blocos usualmente tem tamanho de 64 ou 128 bits de texto claro e produzem blocos de texto cifrado com o mesmo tamanho inicial.

Na Relic, o AES utiliza o modo de operação CBC (*Cipher Block Chaining*), que provê a confidencialidade, utilizando uma chave criptográfica de tamanho 128, 192 ou 256 bits para encriptação e decriptação. O modo CBC cria uma dependência na encriptação, fazendo com que o próximo bloco a ser encriptado necessite do bloco anterior para realizar o processo. Devido ao encadeamento do modo de operação utilizado, ele se torna apropriado para mensagens grandes (STALLINGS, 2012).

O AES consiste em N rodadas, onde o número de rodadas depende do tamanho da chave: 10 para uma chave de 128 bits, 12 para uma chave de 192 bits e 14 para uma chave de 256 bits. Com esta afirmação, espera-se que para uma chave de 256 bits o tempo de execução da encriptação utilizando uma chave de 256 bits seja 40% maior e a de 192 bits 20% maior quando comparado ambos com uma chave de 128 bits.

A função de encriptação do AES na Relic é *bc_aes_cbc_enc(ciphertext, out_len, plaintext, in_len, key, key_len, iv)*. Ela recebe como parâmetro o endereço de memória onde o texto cifrado deve ser armazenado, um inteiro onde o número de bytes escritos da saída deve ser armazenado, o texto a ser criptografado, o número de bytes a serem criptografados, a chave, o tamanho da chave em bytes e o bloco inicializador. A função retorna 1 se algum erro ocorrer no processo ou 0 caso contrário.

A função de decriptação é *bc_aes_cbc_dec(ciphertext, out_len, plaintext, in_len, key, key_len, iv)*. Onde, analogamente à função de encriptação, recebe o

endereço de memória onde o texto claro deve ser armazenado, um inteiro onde o número de bytes escritos da saída deve ser armazenado, o texto cifrado , o número de bytes a serem descriptografados, a chave, o tamanho da chave e o bloco inicializador.

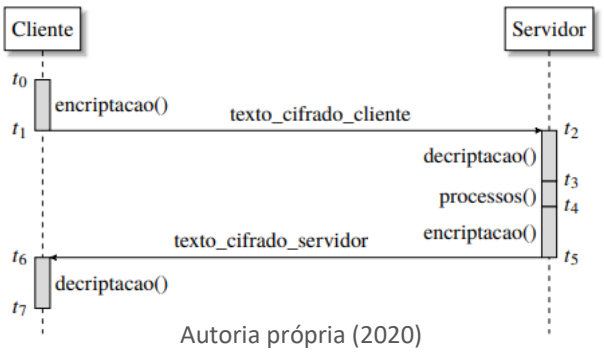
Para a comunicação da aplicação de troca de mensagens criptografadas, o modelo adotado foi o cliente-servidor. O seguinte modelo consiste em uma aplicação distribuída envolvendo provedores de recursos e/ou serviços, chamados servidores, e os requisitantes desses recursos, os clientes (TANENBAUM e WETHERALL, 2010) . A comunicação pode ocorrer em um único *host* através de diferentes processos, ou em diferentes *hosts* através de uma rede de computadores.

Foi utilizado o protocolo TCP para garantir a entrega das informações e o fluxo do processo de comunicação dá-se da seguinte maneira: o cliente verifica se o servidor está disponível e caso esteja, o mesmo aceita a conexão e retorna ao cliente o informando da sua disponibilidade, assim o cliente pode iniciar a comunicação com a mensagem encriptada pela função *bc_aes_cbc_enc.* Ao receber a mensagem o servidor realiza a decriptação com a função *bc_aes_cbc_dec,* e para realizar a comparação nos dois nós é realizada a encriptação da mesma mensagem que é enviada novamente ao cliente onde é decriptada antes da conexão ser encerrada. As mensagens decriptadas são gravadas localmente no cliente e no servidor para verificar se as mensagens recebidas estão legíveis e corretas.

Os experimentos foram realizados em três cenários: em um computador com um processo como cliente e um processo como servidor, entre um computador como cliente e um Raspberry Pi como servidor e entre dois Raspberry Pi. A variação dos hardwares descritos se destina a avaliar o impacto da criptografia em diferentes arquiteturas. O computador utilizado tem um processador i7-7500U com 2.7 GHz e 8 GB de RAM. O Raspberry Pi é a versão 3 modelo B com um processador Bradcom BCM2837 com 1.2 GHz e 1 GB de RAM. A comunicação ocorre nos dois hardwares com Wi-Fi IEEE 802.11n.

Para obter precisamente a performance em ambas plataformas o tempo de cada processo envolvido na comunicação segura foi coletado de acordo com a Figura 1.

Figura 1 - Comunicação segura entre cliente e servidor



Autoria própria (2020)

Baseado nos tempos coletados, os tempos de encriptação no cliente e no servidor são dados por (1) e (2) e os tempos de decriptação no cliente e no servidor são dados por (3) e (4), respectivamente.

$$t_{cliente,enc} = t_1 - t_0 \quad (1)$$

$$t_{servidor,enc} = t_5 - t_4 \quad (2)$$

$$t_{cliente,dec} = t_7 - t_6 \quad (3)$$

$$t_{servidor,dec} = t_3 - t_2 \quad (4)$$

Os tempos de encriptação e decriptação foram calculados diretamente pois são medidos no mesmo hardware. Entretanto, o tempo para a transmissão das mensagens não pode ser obtido da mesma maneira. Logo, o tempo estimado da transmissão é dado por (5) onde t_{diff} é a duração para a decriptação, processamento e encriptação no servidor.

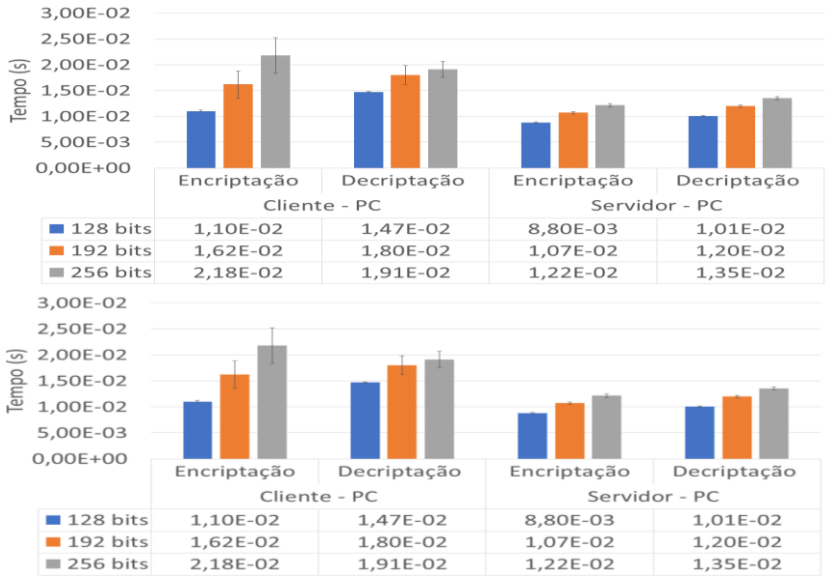
$$t_{transm} = \frac{t_6 - t_1 - t_{diff}}{2} \quad (5)$$

As avaliações são feitas para os diferentes tamanhos de chave do AES e com diferentes tamanhos de mensagens (uma *string* com 1215 bytes e uma imagem com aproximadamente 2 MB). A proposta do experimento é analisar o impacto na comunicação com diferentes configurações do AES, diferentes tamanhos de dados além de diferentes arquiteturas.

RESULTADOS

Conforme mostra a Figura 2 tanto para a *string* quanto para a imagem, os tempos coletados para a encriptação e decriptação tanto no processo cliente quanto no processo servidor, tiveram valores muito próximos do valor teórico, onde o processo com uma chave de 256 bits tem um tempo 40% e uma chave de 192 bits tem um tempo 20% maior se comparados ambos com uma chave de 128 bits.

Figura 2 - Tempo de encriptação e decriptação de uma *string* à esquerda e uma imagem à direita utilizando o AES-128, AES-192 e AES-256 em uma comunicação entre dois processos em um computador.



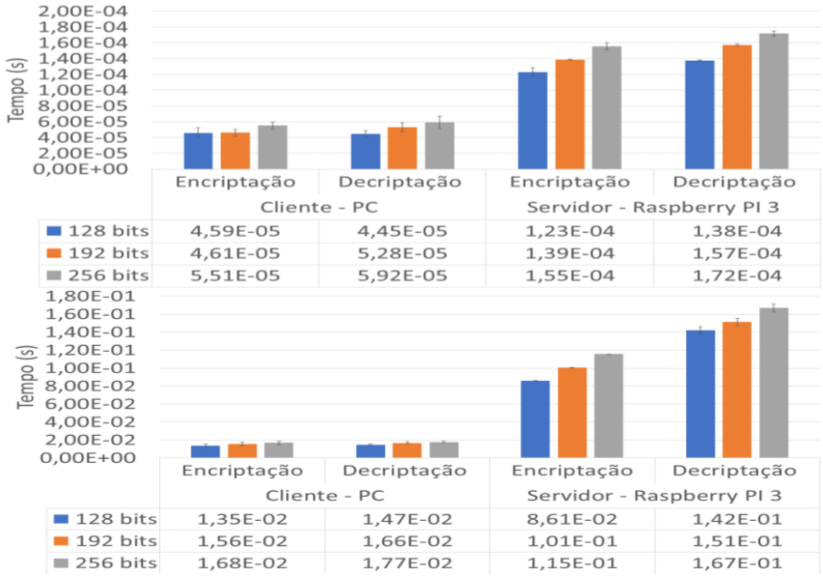
Fonte: Autoria própria (2020)

O tamanho da chave no computador, quando analisados os valores absolutos dos resultados, provou ser uma variável de baixa interferência no processo de criptografia para uma mensagem pequena, como a *string*, tanto para o cliente

quanto para o servidor. Para uma mensagem grande, como a imagem, foi observado um aumento nos valores absolutos como esperado, devido ao tamanho dos dados.

A Figura 3 apresenta os resultados dos experimentos na comunicação entre um computador e um sistema embarcado. Analisando os dois gráficos pode-se observar um aumento do tempo da encriptação no sistema embarcado de aproximadamente 2.7, 3.0 e 2.8 vezes enquanto a decriptação aumentou 3.1, 3.0 e 2.9 se comparado com o computador.

Figura 3 - Tempo de encriptação e decriptação de uma *string* acima e uma imagem abaixo utilizando o AES-128, AES-192 e AES-256 em uma comunicação entre um computador e um sistema embarcado



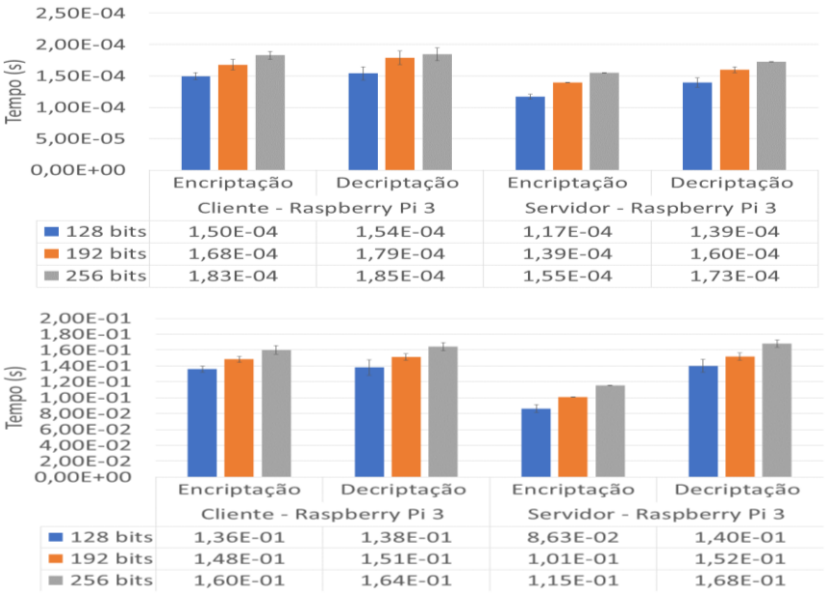
Fonte: Autoria própria (2020)

Segundo à Figura 4, a relação dos tempos de encriptação e decriptação para a *string* e a imagem na comunicação segura entre dois sistemas embarcados quando comparadas as chaves de operação são relativamente abaixo do esperado. Isto reforça a adequação da implementação do AES para este tipo de sistema embarcado, mesmo quando chaves mais fortes com fortes níveis de segurança são necessárias.

Os dados obtidos são usados para calcular a porcentagem do tempo gasto na encriptação e decriptação em relação ao tempo total da comunicação segura para todos os experimentos.

O envio de uma *string* teve uma porcentagem quase nula para todos os experimentos devido ao tamanho reduzido da mensagem. O pior caso ocorreu para a chave de 128 bits, com um aumento de 0.05% na comunicação entre dois sistemas embarcados. O envio da imagem teve uma maior porcentagem de tempo em todas as arquiteturas e tamanhos de chave, devido ao tamanho ser maior que a *string* e consequentemente levando mais tempo para encriptar e decriptar a mensagem. O pior caso ocorreu com a chave de 128 bits no experimento da comunicação entre dois sistemas embarcados, que causou uma porcentagem de tempo de 13.3% na comunicação.

Figura 4 - Tempo de encriptação e decriptação de uma *string* acima e uma imagem abaixo utilizando o AES-128, AES-192 e AES-256 em uma comunicação entre dois sistemas embarcados



Fonte: Autoria própria (2020)

A arquitetura é um fator que teve grande influência na porcentagem de tempo gasto com a encriptação e decriptação. Isto devido ao fato da capacidade de processamento e memória disponível serem diferentes no sistema embarcado e no computador.

CONCLUSÃO

Neste trabalho, o tempo de execução do algoritmo de criptografia simétrico AES foi avaliado em uma comunicação segura envolvendo sistemas embarcados e um computador. Foi possível verificar que o tempo médio de execução da cifra cresce se o tamanho das chaves é maior em uma proporção muito próxima aos valores teóricos. Isto mostra que a implementação da Relic é apropriada para sistemas embarcados

O tamanho dos dados provou ser uma variável de grande influência nos tempos de encriptação e decriptação. Quanto maior o tamanho dos dados, maior o tempo para o processamento da encriptação e decriptação.

A arquitetura também teve um impacto relevante nos tempos coletados. Isto deve-se ao fato de o computador ter maior processamento e memória do que um sistema embarcado. Esta observação mostra a necessidade de realização de avaliações empíricas sobre soluções de segurança para este tipo de plataforma.

Os tempos coletados que foram analisados mostram que o tempo de envio, além de ser diretamente proporcional ao tamanho dos dados é influenciado pela arquitetura, pelos mesmos motivos que acontecem para a criptografia. Porém para todos os experimentos o percentual de tempo gasto com criptografia em todo o processo de uma comunicação segura é sempre menor que 14%, apresentando baixo impacto para fornecer um bom nível de segurança.

REFERÊNCIAS

ARANHA, D. F. RELIC is an Efficient LIbrary for Cryptography. Disponível em: <https://github.com/relic-toolkit/relic>. Acesso em: mar. 2020.

BAAR, M. **Programming embedded systems in C and C++**. [S.l.]: O'Reilly Media, Inc., 1999.

HYNCICA et al. Performance evaluation of symmetric cryptography in embedded systems, v. 1, p. 277-282, 2011.

MARINI, E. El modelo cliente/servidor, 5, 2012.

PANAGIOTOU, P. et al. Cryptographic system for data applications, in the context of internet of things. **Microprocessors and Microsystems**, v. 72, p. 102921, 2012.

SILVA, B.; JR, D. D. S.; SOUZA, E. M. Segurança de software em sistemas embarcados: Ataques e defesas. **Minicursos do XXIII Simpósio Brasileiro em Segurança da Informação e Sistemas Computacionais**, p. 101-155, 2013.

SILVA, N. B. et al. Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general purpose computer. **Journal of Network and Computer Applications**, v. 60, p. 130-143, 2016.

STALLINGS, W. **Computer security:** principles and practice. [S.l.]: Pearson Education Upper Saddle River, 2012.

TANENBAUM, A. S.; WETHERALL, D. J. **Computer Networks**. 5. ed. USA: [s.n.], 2010.