UM ESTUDO COMPARATIVO SOBRE A PERFORMANCE DE MECANISMOS DE SEGURANÇA EM DISPOSITIVOS DA INTERNET DAS COISAS

A COMPARATIVE STUDY OF SECURITY MECHANISMS PERFORMANCE IN INTERNET OF THINGS DEVICES

UN ESTUDIO COMPARATIVO SOBRE EL RENDIMIENTO DE MECANISMOS DE SEGURIDAD EN DISPOSITIVOS DEL INTERNET DE LAS COSAS

UNE ÉTUDE COMPARATIVE DE LA PERFORMANCE DES MÉCANISMES DE SÉCURITÉ DANS LES DISPOSITIFS DE L'INTERNET DES OBJETS

MOSER ZEFERINO VICENTE JOSÉ

https://orcid.org/0000-0002-9374-2091

MESTRE. UNIVERSIDADE JOSÉ EDUARDO DOS SANTOS - INSTITUTO POLITÉCNICO DO HUMABO. HUAMBO. ANGOLA

moser.jose@outlook.com

DATA DA RECEPÇÃO: Fevereiro, 2023 | DATA DA ACEITAÇÃO: Maio, 2023

RESUMO

A segurança de dados é um aspecto fundamental a ser considerado em sistemas de recolha de informações da Internet das Coisas (IoT), uma vez que a IoT é uma rede de dispositivos interconectados que colectam e compartilham dados em tempo real, tornando-se cada vez mais presente em nossas vidas. No entanto, a segurança de dados em sistemas IoT apresenta desafios únicos, devido a grande quantidade de dispositivos e pontos de acesso que estão envolvidos. No presente artigo, realizamos uma revisão da literatura sobre segurança na IoT. Utilizando esses estudos como base para uma análise do desempenho de mecanismos de segurança em plataformas de desenvolvimento actuais, especificamente num Raspberry Pi 3. A compreensão do conhecimento apresentado neste artigo é imprescindível para a elaboração de sistemas e softwares seguros na IoT, sendo isto de grande importância para o desenvolvimento de tecnologias nessa área. O artigo contém ainda os resultados da performance para algumas funções da biblioteca *OpenSSL*, incluindo as funções de *hash* e cifra mais populares actualmente, comparando-os com os resultados obtidos em um computador pessoal.

Palavras-chave: Internet das coisas; Performance; Requisitos; Segurança; Dispositivos.

ABSTRACT

Data security is a fundamental aspect to be considered in information collection systems of the Internet of Things (IoT), as IoT is a network of interconnected devices that collect and share data in real time, becoming increasingly present in our lives. However, data security in IoT systems presents unique challenges, due to the large number of devices and access points involved. This article reviews the literature on security in IoT, using these studies as a basis for an analysis of the performance of security mechanisms in current development platforms, specifically on a Raspberry Pi 3. Understanding the knowledge presented in this article is essential for the development of secure systems and software in IoT, being of great importance for the development of technologies in this area. The article also contains performance results for some functions of the OpenSSL library, including the most popular hash and cipher functions currently, comparing them with the results obtained on a personal computer.

Keywords: Internet of Things; Performance; Requirements; Safety; Devices.

RESUMEN

La seguridad de los datos es un aspecto fundamental a considerar en sistemas de recolección de información de Internet de las cosas (IoT), ya que IoT es una red de dispositivos interconectados que recopilan y comparten datos en tiempo real, cada vez más presente en nuestras vidas. Sin embargo, la seguridad de los datos en los sistemas IoT presenta desafíos únicos debido al gran número de dispositivos y puntos de acceso involucrados. Este artículo revisa la literatura sobre seguridad en IoT, utilizando estos estudios como base para un análisis del rendimiento de los mecanismos de seguridad en las plataformas de desarrollo actuales, específicamente en una Raspberry Pi 3. La comprensión del conocimiento presentado en este artículo es imprescindible para la elaboración de sistemas y software seguros en IoT, siendo de gran importancia para el desarrollo de tecnologías en esta área. El artículo también contiene resultados de rendimiento para algunas funciones de la biblioteca *OpenSSL*, incluyendo las funciones de *hash* y cifrado más populares actualmente, comparándolas con los resultados obtenidos en una computadora personal.

Palabras clave: Internet de las Cosas; Actuación; Requisitos; La seguridad; dispositivos

RESUMÉ

La sécurité des données est un aspect fondamental à prendre en compte dans les systèmes de collecte d'informations de l'Internet des objets (IoT), car l'IoT est un réseau de dispositifs interconnectés qui collectent et partagent des données en temps réel, devenant de plus en plus présent dans nos vies. Cependant, la sécurité des données dans les systèmes IoT présente des défis uniques en raison du grand nombre de dispositifs et de points d'accès impliqués. Cet article passe en revue la littérature sur la sécurité dans l'IoT, en utilisant ces études comme base pour une analyse de la performance des mécanismes de sécurité dans les plates-formes de développement actuelles, spécifiquement sur un Raspberry Pi 3. La compréhension des connaissances présentées dans cet article est essentielle pour le développement de systèmes et de logiciels sécurisés dans l'IoT, étant d'une grande importance pour le développement de technologies dans ce domaine. L'article contient également des résultats de performance pour certaines fonctions de la bibliothèque *OpenSSL*, y compris les fonctions de *hash* et de chiffrement les plus populaires actuellement, les comparant avec les résultats obtenus sur un ordinateur personnel.

Mots-clés: Internet des objets; Performance; Exigences; Sécurité; Dispositifs.

1. INTRODUÇÃO

A Internet das Coisas (IoT) é um novo paradigma (Mohd Aman et al., 2020) que permite interligar objectos físicos (“coisas"”) inteligentes do nosso quotidiano à rede mundial de computadores (Internet) e entre si, ampliando fortemente a sua utilidade no contexto em que se inserem, aumentando, deste modo, a sua utilidade para os humanos. A designação inglesa *Internet of Things (IoT)* foi inventada no final da década de 90 por Kevin Ashton (Ashton & others, 2009) para identificar um conjunto de sensores e dispositivos ligados em ambientes inteligentes que partilham informações entre si e não só (Al-Fuqaha et al., 2015).

Existem várias tecnologias que, pela sua especificidade, surgem normalmente associadas à IoT, como a *Radio Frequency Identification (RFID), Wi-Fi, ZigBee, Sigfox,*

Low Power Wide Area Network (LoRaWAN), Z-Wave, Bluetooth LE, 6LoWPAN, e protocolos de comunicação como *Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT), Extensible Messaging and Presence Protocol (XMPP)*, distribuídos nas várias camadas da arquitectura da IoT.

O crescimento de aplicações IoT faz com que uma grande quantidade de dados seja gerada em diversas áreas da vida humana como saúde, indústrias, agricultura, educação, comércio, cidades inteligentes, casas inteligentes, etc.

Com esse pensamento, a *International Data Corporation (IDC)* realizou um estudo em 2021 onde apontou que, até o ano de 2025, haveria mais de 55,7 mil milhões de dispositivos IoT em uso, gerando quase 80 *zettabytes (ZB)* de dados (Hojlo, 2021). Já a Gartner (Sasaki, 2022) espera que o número de dispositivos IoT se mantenha na faixa dos 20 mil milhões até 2025. Portanto, a demanda está aumentando para a análise de grandes quantidades de dados gerados pelos dispositivos IoT. No entanto, uma quantidade tão significante de dispositivos ligados à Internet traz consigo vários problemas e grandes responsabilidades.

A IoT está num estado muito prematuro em termos de segurança e privacidade (M. Samaila et al., 2017). Não existe uma abordagem geral (ou até preocupação) em relação à segurança, e não existe um mecanismo padronizado para a protecção dos dados e dispositivos.

O entusiasmo em torno dos novos sistemas de IoT leva a uma redução no tempo de mercado, o que beneficia a funcionalidade, mas prejudica a engenharia de segurança. As especificações de processamento e memória menos robustas para muitos dos dispositivos da IoT criam desafios adicionais para o design e a integração de mecanismos de segurança. Além disso, a acessibilidade física de muitos dispositivos, sem um controle de acesso rígido, tem facilitado a ocorrência de brechas para ataques maliciosos a dispositivos IoT.

A presença dessas brechas cria diversos desafios para a IoT, desafios difíceis de solucionar devido a um conjunto de restrições. Em primeiro lugar, a falta de controlo físico rigoroso em relação aos dispositivos IoT aumenta as chances de um atacante mal-intencionado obter informações confidenciais dos usuários. Em segundo lugar, como a maioria das comunicações é realizada por meio de tecnologias de redes sem fio, é possível que os atacantes explorem vulnerabilidades inerentes a essas tecnologias para atacar os dispositivos IoT. Por fim, os dispositivos IoT são caracterizados por possuir recursos

limitados em termos de energia, memória e processamento, tornando difícil a implementação de mecanismos robustos de segurança.

Este artigo apresenta um conjunto de testes de performance a algoritmos da criptografia moderna muito utilizados em soluções de segurança informática, efectuados num dispositivo específico para a IoT, o Raspberry Pi 3. Os resultados desta performance são comparados com os resultados obtidos num computador pessoal. Estas contribuições constituem apenas uma parte do longo caminho a percorrer para uma IoT mais segura por desenho. Acreditamos que estudar e relatar o comportamento dos algoritmos e mecanismos de segurança em dispositivos IoT é fundamental para compreender melhor seus problemas e limitações, além de ajudar a pensar sobre a segurança para a IoT no futuro e auxiliar na escolha dos mecanismos a serem implementados. Este artigo também apresenta uma visão geral de outros estudos e trabalhos recentes relacionados, buscando aprofundar a compreensão da segurança dos dispositivos IoT.

2. TRABALHOS RELACIONADOS

A literatura na área da IoT especificamente no quesito segurança tem aumentado bastante nos últimos anos. Assim, esta secção foca-se apenas na descrição de trabalhos relacionados (posteriores a 2016).

Os Autores em (Kaliya & Hussain, 2017) propõem um mecanismo de segurança que classifica os dados dos utilizadores de acordo com a sua sensibilidade, baseando-se no controlo de acesso e, para elevar o nível de segurança, mecanismos de autenticação forte são incorporados para garantir a segurança do mecanismo proposto. Os autores analisam os requisitos de segurança voltados para a IoT e utilizam estes requisitos para garantir a máxima privacidade do utilizador.

Os autores em (Pal et al., 2017) propõem uma lista de requisitos de segurança para ambientes da IoT, que são atendidos por meio de um mecanismo de controle de acesso. Além disso, eles abordam de forma abrangente as vulnerabilidades e ameaças de segurança no ecossistema da IoT. Vale ressaltar que os autores não detalham como exactamente esse mecanismo de segurança aborda as ameaças e ataques mencionados. Eles afirmam que, embora seja praticamente impossível um único mecanismo garantir total segurança neste ambiente, o mecanismo de segurança proposto pode elevar os níveis de segurança para os dispositivos e para o ambiente em geral, com base em testes realizados e discussões feitas.

Em (Daud et al., 2017), é feita uma análise genérica da segurança na IoT, com

enfoque nas tecnologias de comunicação *Radio Frequency IDentification (RFID* e *Wireless Sensor Networks (WSN)*, amplamente utilizadas nesse ambiente. Com base nos requisitos de segurança para cada camada da arquitectura IoT, os autores propõem um modelo de segurança, mas sem a criação específica de um mecanismo seguro para as diversas questões de segurança enfrentadas pela IoT. Em vez disso, os autores focam em soluções existentes para ataques na rede.

Em (Josyula & Gupta, 2017), os autores apresentam uma nova metodologia de segurança para ambientes IoT que leva em consideração a engenharia de requisitos de segurança proposta em um trabalho anterior (Chatterjee et al., 2013). Eles argumentam que as abordagens de segurança tradicionais utilizadas em redes convencionais não são adequadas para a IoT e propõem a utilização de algoritmos criptográficos leves para garantir a segurança. A metodologia é projectada para permitir a análise e adaptação dos requisitos de segurança para a IoT em cada fase do desenvolvimento do sistema.

Os autores em (M. G. Samaila et al., 2019) propuseram um Framework denominado *IoT Hardware Platform Security Advisor (IoT-HarPSecA)*, para resolver o desafio de escolher os algoritmos criptográficos correctos para garantir a segurança em uma plataforma IoT. Esse Framework ajudaria a seleccionar algoritmos de segurança específicos com base em requisitos específicos, como objectivos de segurança, especificações de hardware, tamanho da carga útil da mensagem, área de aplicação e requisitos de energia. A ferramenta poderia ajudar engenheiros electrónicos e de computação, bem como desenvolvedores de aplicativos, que não são especialistas em segurança, a tomar decisões informadas sobre quais algoritmos de segurança usar em suas aplicações.

O estudo realizado pelos autores (Wahab et al., 2021) apresenta uma revisão dos padrões de segurança e estruturas de avaliação existentes, incluindo várias publicações do *National Institute of Standards and Technology (NIST)* sobre técnicas de segurança. O objectivo da revisão é destacar as principais áreas de foco dos padrões e estruturas de avaliação existentes para encontrar soluções que possam atender às necessidades de segurança de dispositivos IoT. O artigo visa identificar as técnicas e metodologias de segurança mais adequadas para garantir a segurança de dispositivos IoT, através da análise de diferentes padrões e *frameworks* existentes.

3. ALGORITMOS ESTUDADOS

Os algoritmos mencionados são comumente utilizados na construção de

aplicações tanto para a web quanto para dispositivos IoT. No entanto, é importante destacar que alguns desses algoritmos não são facilmente adaptáveis a dispositivos com restrições de memória, processamento e consumo de energia, o que torna relevante a análise comparativa de seu desempenho em plataformas típicas de IoT e em computadores convencionais.

As subsecções a seguir apresentam brevemente os algoritmos criptográficos que foram considerados nos testes de performance. A subsecção 3.1, Algoritmos de Chave Simétrica, apresenta os algoritmos conhecidos como algoritmos de chave simétrica ou de chave secreta, e a subsecção 3.2, Algoritmos de Chave pública, apresenta os algoritmos conhecidos como algoritmos criptográficos de chave assimétrica ou de chave pública.

3.1. Algoritmos de Chave Simétrica

Os algoritmos criptográficos de chave simétrica *Data Encryption Algorithm (DES)* e o *Triple Data Encryption Algorithm* (3DES) são algoritmos de cifra por blocos de tamanho fixo (José, 2018). O DES, com chave de 56 bits (7 bytes) e bloco de 64 bits (8 bytes), é considerado inseguro actualmente, mas é histórico por ser o primeiro algoritmo de chave simétrica com norma internacional. Já o 3DES combina a cifra-decifra-cifra DES com 3 chaves para aumentar sua força criptográfica. Ambas as cifras funcionam com rondas e utilizam redes de *Feistel* na sua operação de cifra e decifra. Nos testes realizados, as cifras foram operadas nos modos de cifra *Electronic Codebook (ECB)*, *Cipher Block Chaining (CBC)*, *Output Feedback (OFB)* e *Cipher Feedback (CBF)*. Apesar de ser desaconselhada a utilização do DES, ele foi considerado nos testes devido à sua importância histórica.

O *Advanced Encryption Standard (AES)* é uma cifra de chave simétrica por blocos que se tornou a norma internacional para a segurança da cifra de chave simétrica, substituindo o DES. O AES é capaz de utilizar três tamanhos de chave diferentes, que fornecem diferentes níveis de segurança: 128, 192 e 256 bits. O tamanho do bloco sobre o qual o AES opera é sempre fixo, com 128 bits (Mohurle & Panchbhai, 2016). Além disso, a AES é capaz de operar em vários modos de cifra, como ECB, CBC, Counter (CTR), CFB e OFB. Todos esses modos foram testados no contexto do trabalho. Quando utilizada correctamente, a AES é considerada altamente segura e eficiente em termos de desempenho, tornando-se uma escolha ideal para a cifra de chave simétrica em dispositivos e sistemas IoT.

O *Rivest Cipher* 4 (RC4) é um algoritmo de cifra de fluxo, também conhecido como cifra de chave simétrica contínua. Desenvolvido por Ron Rivest em 1987, inicialmente como um gerador de números pseudo-aleatórios criptograficamente seguros, o RC4 tem sido amplamente utilizado em vários ambientes e protocolos de segurança (Jindal & Singh, 2014). Apesar de ter sido identificada a existência de chaves fracas e fragilidades, o RC4 ainda é utilizado em alguns sistemas, principalmente devido à sua baixa eficiência computacional. O algoritmo utiliza chaves de 40 a 128 bits e foi amplamente utilizado em padrões de segurança, como o *Transport Layer Security (TLS)*, *Wired Equivalent Privacy (WEP)* e *Wi-Fi Protected Access (WPA)*, até 2015 (Mohurle & Panchbhai, 2016). Embora tenha sido considerado no contexto deste trabalho por seu peso histórico, também foi utilizado para fins comparativos em relação aos demais algoritmos abordados neste artigo.

3.2. *Algoritmos de Chave Pública*

O *Rivest Shamir Adleman (RSA)* é um algoritmo criptográfico de chave pública é amplamente utilizado para cifrar e decifrar pequenas quantidades de dados, como cadeias de bits menores que um dos parâmetros da chave pública, conhecido como módulo. Criado por Ronald Rivest, Adi Shamir e Leonard Adleman em 1978, é considerado um dos maiores avanços da criptografia de chave pública (Rivest et al., 1978). O RSA é ideal para cifrar e trocar segredos criptográficos e chaves de cifra simétricas, bem como para assinar valores de *hash* menores que o módulo. Elaborado na Teoria dos Números, o RSA baseia-se na complexidade do problema matemático de factorização de um número composto extremamente grande em primos para garantir a sua segurança (Wahab et al., 2021). A cifra e decifra são operações de exponenciação modular, tornando a implementação relativamente simples e fácil de entender.

O RSA também faz uso de funções de sentido único com alçapão, que não podem ser invertidas em tempo útil, a menos que se saiba a chave privada. Essa abordagem de chave pública é amplamente utilizada para garantir a privacidade e segurança de dados sensíveis em várias aplicações, incluindo a criptografia de mensagens de e-mail, autenticação de usuário em sistemas online e segurança de transacções financeiras.

Para cifrar uma mensagem $M \in \mathbb{Z}_n$, usa-se a chave pública $\boldsymbol{pk}$ constituída por dois números $(\boldsymbol{N}, \boldsymbol{e})$, e calcula-se $\boldsymbol{C} \leftarrow \boldsymbol{M^e}\ (\boldsymbol{mod\ N})$. Para decifrar o criptograma, usa-se a chave privada $\boldsymbol{sk}$ no cálculo $\boldsymbol{M} \leftarrow \boldsymbol{C^d}\ (\boldsymbol{mod\ N})$, onde $\boldsymbol{M}$ é a mensagem, $\boldsymbol{C}$ é o

criptograma, $\boldsymbol{e}$ e $\boldsymbol{d}$ são números inteiros e $\boldsymbol{N}$ é o resultado da multiplicação de dois números primos considerados grandes ($\geq \mathbf{2^{1024}}$). De salientar ainda que os tamanhos de chaves apresentados acima foram utilizados neste trabalho para assinar e verificar ficheiros com tamanhos de 100MB e 1GB e 2GB, utilizando a função de *hash Secure Hash Algorithm* 256 (SHA256) (**José, 2018**).

4. MÉTODO E RESULTADOS

A presente secção, demonstra o método usado e apresenta os resultados obtidos dos testes realizados. De salientar que, devido as limitações de páginas, apenas apresentaremos os resultados obtidos a partir das cifras DES-CBC, 3DES-CBC, AES-CBC, RC4, e os resultados obtidos das operações de assinatura e verificação, com RSA. Posteriormente, foi feita uma comparação entre os resultados obtidos para os diferentes algoritmos. Assim sendo, a subsecção 3.1, Métodos, apresenta o método escolhido, e a subsecção 3.2, Resultados, apresenta os resultados obtidos e a comparação entre os algoritmos.

4.1. Método

Todos os algoritmos de cifra mencionados na Secção anterior foram usados para a realização dos testes de performance, e, nisto, conseguimos analisar certos comportamentos distintos. Estes testes permitiram encontrar ainda os algoritmos de cifra que melhor se adaptam ao mundo da IoT (e.g., dispositivos e sistemas) com nível de recursos semelhantes aos do nosso dispositivo de teste, através da eficiência na resposta dada pela utilização dos seus métodos de cifrar e decifrar. Os testes basearam-se na obtenção do tempo gasto, em segundos, e do consumo de memória em Kilobytes. Além dos cálculos referentes ao tempo e a memória, foram também obtidos valores estatísticos como a variância e o desvio padrão.

Para efectuar os testes, foram utilizadas as implementações dos algoritmos definidas na biblioteca *Openssl*, versão 1.1.0.2g, com chave de cifra e vector de inicialização (quando aplicável) fixos. Para medição de tempo e consumo de memória, foi utilizada a ferramenta time, com os parâmetros $'\boldsymbol{e}'$ e $'\boldsymbol{m}'$, respetivamente. A variância e desvio padrão *(D.P)* foram obtidos a partir das fórmulas $\boldsymbol{var(x)} = \boldsymbol{\Sigma}\frac{(x_i-\bar{x})^2}{x-1}$ e $\boldsymbol{\sigma} = \sqrt{\boldsymbol{\Sigma}\frac{(x_i-\bar{x})^2}{x-1}}$ respectivamente.

Os testes foram efectuados num Raspberry Pi 3 (Cortex A53 Quad Core, ARM Cortex, 1.2 GHz, 16GB de armazenamento, memoria de 1GB com o sistema operativo Ubuntu MATE 16.04.2) e num computador ACER Aspire ES15(AMD Quad-Core, A5-5000 de 1,5 GHz, HDD 1GB, memoria 4GB DDR3, com o sistema operativo Ubuntu 18.04). Foram realizadas um total de 100 repetições para cada ficheiro com tamanhos de 100MB, 1GB e 2GB, tanto para o Raspberry Pi 3 como para o computador.

Figura 1: Trecho de código para a obtenção do tempo gasto e o consumo médio da memória de cifra para ficheiro de 1 GB.

```
#!/bin/sh
soma=0
for i  in {1..100}
do
tempo=$((/usr/bin/time -f '%e' openssl enc -aes-128-cbc -K afcf51195d2aa3dc89fa83857e526fd7
-in 1GB -out ENC -iv ba39aefefad87749f4a9f0e65396a640 1> /dev/null) 2>&1)
soma=$(echo $soma+$tempo | bc -l)
arr[$i]=$tempo
echo $i-$tempo
done
echo "---------------------------------"
media=`echo "scale=2; $soma/100" | bc -l`
echo "Média: " $media
...
```

Fonte: Elaboração do autor (2022).

4.2. Resultados

Os resultados da análise de desempenho dos algoritmos de criptografia DES, 3DES, RC4 e AES foram apresentados nas tabelas 1 e 2, em relação ao tempo gasto para cifrar e decifrar arquivos de 100MB e 1GB entre um Raspberry Pi 3 e um computador. Constatou-se que o AES se mostrou o algoritmo mais eficiente em dispositivos com poder computacional razoável, como o Raspberry Pi 3, tanto para arquivos de menor quanto de maior tamanho, enquanto a cifra 3DES apresentou um desempenho bastante inferior. Embora a cifra RC4 tenha se aproximado do AES em arquivos maiores, a segurança superior do AES o torna a melhor escolha. Não houve diferenças significativas no desempenho em relação ao tamanho das chaves utilizadas, não havendo, portanto, benefício em usar chaves menores. Em relação ao algoritmo de assinatura RSA, os resultados obtidos foram consistentes com o esperado, sendo cerca de três vezes mais rápido no processo do que o AES na sua operação de cifragem. O computador apresentou desempenho superior em ambas as operações em relação ao Raspberry Pi 3, como era de se esperar, executando as mesmas operações em cerca de 15 a 25% do tempo.

As tabelas 1 e 2 apresentam os resultados da performance relativamente ao tempo gasto pelos algoritmos DES, 3DES, RC4 e AES para cifrar e decifrar um ficheiro de 100MB e 1GB entre o Raspberry Pi 3 e o Computador. É possível observar que o algoritmo AES, em dispositivos com poder computacional razoável, como o Raspberry Pi 3 utilizado, se mostra o mais eficiente, tanto para ficheiros de menor como de maior tamanho. A cifra 3DES, por sua vez, apresenta um desempenho bastante inferior, enquanto que, em ficheiros de maior dimensão, a cifra RC4 se aproxima do AES. No entanto, e como explicitado na secção anterior, a cifra AES apresenta uma maior segurança, sendo, por isso, a melhor escolha entre o grupo analisado. Quanto ao tamanho das chaves utilizadas, não existem diferenças significativas (a maior variação é de cerca de 12%), não havendo por isso benefício na utilização de uma chave de tamanho menor. Quanto ao caso do algoritmo de assinatura RSA, os resultados obtidos, em ambos os dispositivos, demonstram uma performance em linha com o esperado, sendo cerca de três vezes mais rápido no seu processo do que o AES no seu processo de cifra. Quando comparando os dois dispositivos, o computador, como esperado, consegue efectuar as mesmas operações em cerca de 15 a 25% do tempo.

As tabelas 3 e 4 apresentam os resultados referentes ao consumo de memória pelos processos dos algoritmos DES, 3DES, RC4 e AES para cifrar e decifrar ficheiros de 100MB e 1GB, nos dois dispositivos utilizados. É possível constatar que, independentemente do algoritmo utilizado, o Raspberry Pi 3 consome cerca de 60% da memória utilizada pelo computador, com valores em torno de 2500 KB. Este baixo consumo de memória é constante independentemente do tamanho do ficheiro, não havendo, por isso, qualquer influência desta métrica. A utilização de chaves de diferentes tamanhos não afecta os resultados obtidos, sendo estes semelhantes entre os diferentes algoritmos.

Tabela 1: Resultados referente ao tempo gasto das cifras DES-CBC, 3DES-CBC, RC4, AES-CBC e RSA com cheiro de 100MB e 1GB para o Raspberry Pi 3.

Ficheiro	Algoritmos		Cifrar/Assinar			Decifrar/Verificar		
	Cifra/modo	Chave(bits)	Tempo(s)	Variância	D. P.	Tempo	Variância	D. P.
100MB	DES-CBC	56	9,86	105	10,25	10,4	107	10,34
	3DES-CBC	168	16,37	275,29	16,59	17,84	324,52	18,01
	RC4	128	8,44	86,40	9,30	8,75	90,27	9,50
	AES-CBC	128	7,89	71,44	8,45	8,80	89,23	9,45
		192	7,94	73,33	8,56	7,90	72,08	8,49
		256	7,46	63,70	7,98	7,78	69,15	8,32
	RSA	2048	1,46	2,17	1,47	1,44	2,07	1,44
		4096	1,58	2,52	1,59	1,44	2,09	1,44

1GB	DES-CBC	56	137,98	19236,86	192,37	138,39	19156,98	191,57
	3DES-CBC	168	216,02	47135,37	217,11	220,46	48606,91	486,07
	RC4	128	129,91	17053,43	130,59	129,80	16855,74	168,56
	AES-CBC	128	129,98	17070,49	170,70	129,33	16732,45	167,32
		192	130,72	17269,09	172,69	130,58	17060,29	170,60
		256	131,10	17365,94	173,66	130,59	17057,44	170,57
	RSA	2048	46,44	2178,68	46,68	46,42	2154,67	46,71
		4096	46,86	2217,80	47,09	46,71	2181,94	46,71

Fonte: Elaboração do autor (2022).

Tabela 2: Resultados referente ao tempo gasto das cifras DES-CBC, 3DES-CBC, RC4, AES-CBC e RSA com ficheiro de 100MB e 1GB para o Computador.

Ficheiro	Algoritmos		Cifrar/Assinar			Decifrar/Verificar		
	Cifra/modo	Chave(bits)	Tempo(s)	Variância	D. P.	Tempo	Variância	D. P.
100MB	DES-CBC	56	4,56	20,97	4,58	4,48	20,07	4,48
	3DES-CBC	168	11,50	133,49	11,55	11,56	133,58	11,56
	RC4	128	0,98	0,98	0,99	0,96	0,94	0,97
	AES-CBC	128	1,14	1,35	1,16	0,87	0,78	0,89
		192	1,14	1,33	1,16	0,87	0,78	0,88
		256	1,18	1,43	1,20	0,82	0,68	0,82
	RSA	2048	1,29	167	1,29	1,28	1,63	1,28
		4096	1,32	1,76	1,33	1,28	1,64	1,28
1GB	DES-CBC	56	59,20	3549,42	59,58	76,69	3466,42	58,88
	3DES-CBC	168	128,58	16702,56	129,24	116,45	18394,70	135,63
	RC4	128	31,68	1016,74	31,89	25,12	633,48	25,17
	AES-CBC	128	30,45	940,25	30,66	50,47	651,85	25,13
		192	30,01	910,43	30,17	48,05	535,09	23,13
		256	33,76	1156	34	58,30	1160,95	34,07
	RSA	2048	13	170,74	13,07	12,98	168,56	12,98
		4096	13,03	171,55	13,10	12,99	168,70	12,99

Fonte: Elaboração do autor (2022).

Tabela 3: Resultados referente ao consumo de memória das cifras DES-CBC, 3DES-CBC e RSA com ficheiro de 100MB e 1GB para o Raspberry Pi 3.

Ficheiro	Algoritmos		Cifrar/Assinar			Decifrar/Verificar		
	Cifra/modo	Chave(bits)	Memória	Variância	D. P.	Memória	Variância	D. P.
100MB	DES-CBC	56	2583,16	3984,26	63,12	2585,80	4033,40	63,51
	3DES-CBC	168	2583,32	3575,94	59,80	2586,84	4183,77	64,68
	RC4	128	2577,28	2789,78	52,82	2582,88	2960,03	54,41
	AES-CBC	128	2597,04	3464,93	58,86	2592,24	3214,02	56,69
		192	2601,68	3138,00	56,02	2596,28	3287,76	57,34
		256	2596,12	2663,58	51,61	2598,96	2812,04	53,03
	RSA	2048	2623,56	8223,32	90,68	2579,52	7063,77	84,05
		4096	2629,68	8868,26	94,17	2571,16	7632,73	87,37
1GB	DES-CBC	56	2598,24	3993,96	63,20	2589,16	3982,17	63,10
	3DES-CBC	168	2565,84	4272,30	65,36	2568,68	4166,42	64,55
	RC4	128	2570	2681,37	51,78	2572,32	2446,62	249,46
	AES-CBC	128	2598,36	2883,59	53,70	2595,32	2411,86	49,11
		192	2603,92	3522,90	59,35	2594,68	3367,06	58,03
		256	2595,48	2717,63	52,13	2598,80	3575,52	59,80
	RSA	2048	2615,56	7823,48	88,45	2565,88	7833,43	88,51
		4096	2639,00	7436,73	86,24	2562,48	6869,37	82,88

Fonte: Elaboração do autor (2022).

Tabela 4: Resultados referente ao consumo de memória das cifras DES-CBC, 3DES-CBC, RC4, AES-CBC e RSA com ficheiro de 100MB e 1GB para o Computador.

Ficheiro	Algoritmos		Cifrar/Assinar			Decifrar/Verificar		
	Cifra/modo	Chave(bits)	Memória	Variância	D. P.	Memória	Variância	D. P.
100MB	DES-CBC	56	4452,36	3537,16	59,47	4455,28	5259,32	72,52
	3DES-CBC	168	4489,28	5938,06	77,06	4481,60	5375,36	73,32
	RC4	128	4473,36	5663,10	75,25	4470	7093,28	84,22
	AES-CBC	128	4455,56	4350,03	65,95	4457,28	4785,56	69,18
		192	4461,56	5176,53	71,95	4464,92	4094,03	63,98
		256	4472,48	3997,18	63,22	4465,96	4286,56	65,47
	RSA	2048	4485,88	3060,83	55,32	4546,36	4614,11	67,93
		4096	4524,12	3204,35	56,61	4548,92	4832,59	69,52
1GB	DES-CBC	56	4358,96	10885,3	104,33	4369,40	11395,9	106,7
	3DES-CBC	168	4345,44	5169,30	71,90	4322,28	4717,84	68,69
	RC4	128	4466,48	4957,67	70,41	4457,64	5394,91	73,45
	AES-CBC	128	4301,44	3944,25	39,44	4301,44	5140,65	71,70
		192	4308,04	5067,15	71,18	4318,60	4447,48	66,69
		256	4318,08	5340,60	73,08	4317,32	4632,66	68,06
	RSA	2048	4347,20	3132,12	55,97	4406,80	2278,88	47,74
		4096	4366,04	2199,43	46,90	4413,40	1406,84	37,51

Fonte: Elaboração do autor (2022).

5. CONCLUSÃO

Neste artigo foram apresentados testes a algoritmos criptográficos entre um Raspberry Pi 3 e um computador pessoal, e os resultados foram comparados entre as plataformas. Como era esperado, o Raspberry Pi 3 apresentou um maior tempo de consumo nas operações criptográficas, devido às suas limitações de recursos. No entanto, os testes demonstraram que é possível implementar estes algoritmos em plataformas com recursos mais limitados, devido à redução no consumo de memória. Entre os algoritmos testados, o RC4, AES e RSA destacaram-se em termos de performance, com os dois últimos sendo os mais indicados para cenários IoT com recursos semelhantes ao Raspberry Pi 3.

Durante o estudo, foi evidenciado que medidas e mecanismos de segurança adicionais são necessários para garantir que a IoT continue avançando e alcance seus objectivos. No entanto, os dispositivos da IoT, são altamente vulneráveis e geralmente carecem de mecanismos e práticas adequados de segurança. Dado o grande aumento na adopção da IoT, é extremamente importante desenvolver técnicas e tecnologias específicas para garantir a segurança desses dispositivos. Isso deve-se não apenas ao ambiente único em que as aplicações da IoT operam, mas também porque não é suficiente

simplesmente transferir os mecanismos existentes em redes ou sistemas actuais para a IoT.

6. REFERÊNCIAS BIBLIOGRÁFICA

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials*, *17*(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

Ashton, K., & others. (2009). That 'internet of things' thing. *RFID Journal*, *22*(7), 97–114.

Chatterjee, K., Gupta, D., & De, A. (2013). ``A framework for development of secure software''. *CSI Transactions on ICT*, *1*(2), 143–157. https://doi.org/10.1007/s40012-013-0010-8

Daud, M., Khan, Q., & Saleem, Y. (2017). A study of key technologies for IoT and associated security challenges. *2017 International Symposium on Wireless Systems and Networks (ISWSN)*, 1–6. https://doi.org/10.1109/ISWSN.2017.8250042

Hojlo, J. (2021, January 6). *Future of Industry Ecosystems: Shared Data and Insights*. https://blogs.idc.com/2021/01/06/future-of-industry-ecosystems-shared-data-and-insights/

Jindal, P., & Singh, B. (2014). Performance analysis of modified RC4 encryption algorithm. *International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2014*. https://doi.org/10.1109/ICRAIE.2014.6909247

José, M. (2018). *Mapeamento de Requisitos de Segurança à Tecnologia na Internet das Coisas* [MasterThesis, Universidade da Beira Interior, Rua Marquês d' Ávila e Bolama, 6201-001 Covilhã, Portugal]. http://hdl.handle.net/10400.6/10019

Josyula, S. K., & Gupta, D. (2017). A new security methodology for internet of things. *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 613–618. https://doi.org/10.1109/CCAA.2017.8229874

Kaliya, N., & Hussain, M. (2017). Framework for privacy preservation in iot through classification and access control mechanisms. *2017 2nd International Conference*

for Convergence in Technology (I2CT), 430–434. https://doi.org/10.1109/I2CT.2017.8226166

Mohd Aman, A. H., Yadegaridehkordi, E., Attarbashi, Z. S., Hassan, R., & Park, Y. J. (2020). A Survey on Trend and Classification of Internet of Things Reviews. *IEEE Access*, *8*, 111763–111782. https://doi.org/10.1109/ACCESS.2020.3002932

Mohurle, M., & Panchbhai, V. V. (2016). Review on realization of AES encryption and decryption with power and area optimization. *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, 1–3. https://doi.org/10.1109/ICPEICES.2016.7853276

Pal, S., Hitchens, M., & Varadharajan, V. (2017). *On the Design of Security Mechanisms for the Internet of Things*. https://doi.org/10.1109/ICSensT.2017.8304476

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, *21*, 120–126.

Samaila, M. G., José, M. Z. V., Sequeiros, J. B. F., Freire, M. M., & Inácio, P. R. M. (2019). Iot-HarpSecA: A framework for facilitating the design and development of secure IoT devices. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3339252.3340514

Samaila, M., Neto, M., Fernandes, D. A. B., Freire, M. M., & Inácio, P. R. M. (2017). ``Challenges of Securing Internet of Things Devices: A Survey''. *Wiley Security and Privacy (SPY)*, *1*(2), 20. https://doi.org/10.1002/spy2.20

Sasaki, Y. (2022). A Survey on IoT Big Data Analytic Systems: Current and Future. *IEEE Internet of Things Journal*, *9*(2), 1024–1036. https://doi.org/10.1109/JIOT.2021.3131724

Wahab, O. F. A., Khalaf, A. A. M., Hussein, A. I., & Hamed, H. F. A. (2021). Hiding data using efficient combination of RSA cryptography, and compression steganography techniques. *IEEE Access*, *9*, 31805–31815. https://doi.org/10.1109/ACCESS.2021.3060317