RSSignal: um Arcabouço para Evolução de Técnicas de Geração de Chaves Baseadas em RSSI

Leonardo Azalim de Oliveira[1], **Luciano Jerez Chaves**[1], **Edelberto Franco Silva**[1]

[1] Departamento de Ciência da Computação – Universidade Federal de Juiz de Fora
{leonardo.azalim, luciano.chaves, edelberto}@ice.ufjf.br

Abstract. *The increasing number of devices connected to the Internet of Things and the well-known weakness of wireless communication medium demand solid security solutions to mitigate malicious attacks on these networks. Focusing on research reproducibility, this paper presents the RSSignal: a framework for generating and validating keys using the RSSI signal indicator as input. The current framework implementation embraces the generation of symmetric cryptographic keys for LoRaWAN networks using datasets from real scenarios and the subsequent validation of these keys through the NIST 800-22 test suite. The RSSignal is available as a free software so researchers can use it to reproduce the experiments and propose improvements to the process.*

Resumo. *O crescente número de dispositivos conectados à Internet das coisas e a já conhecida fragilidade dos meios de comunicação sem fio demandam por soluções de segurança robustas para mitigar ataques maliciosos nestas redes. Com foco na reprodutibilidade de pesquisas, este artigo apresenta o RSSignal: um arcabouço para geração e validação de chaves a partir do indicador de sinal RSSI. A implementação atual do arcabouço compreende a geração de chaves criptográficas simétricas para redes LoRaWAN a partir de conjuntos de dados reais, e a posterior validação dessas através da suíte de testes NIST 800-22. O RSSignal está disponível como software livre para que pesquisadores possam reproduzir os experimentos e propor melhorias ao processo.*

1. Introdução

As redes de dispositivos conectados à Internet das Coisas (*Internet of Things* – IoT) são cada vez mais comuns em diversos locais e para as mais variadas aplicações. Elas estão presentes em inúmeros ambientes, com destaque para as cidades e fazendas inteligentes [Pasolini et al. 2018, Codeluppi et al. 2020]. É estimado que em 2025 existirão aproximadamente 75 bilhões de dispositivos IoT ativados pelo mundo [Statista 2016]. Esses dispositivos vão gerar cerca de 25 bilhões de conexões, sendo que 11% destas serão exclusivamente utilizadas para comunicação de longa distância e de baixa potência entre os dispositivos [Machina e Gartner 2016]. Considerando a crescente adesão da IoT e a conhecida fragilidade do seu meio de comunicação sem fio, é de suma importância a proposta e validação de métodos seguros para troca de mensagens entre os dispositivos [Jiang et al. 2020, Stellios et al. 2018]. Com a mesma importância, a reprodutibilidade da pesquisa se mostra uma das grandes lacunas na área de segurança [Deelman et al. 2019].

Neste contexto, o presente trabalho tem foco na segurança do promissor cenário de redes sem fio de longo alcance, especificamente, das redes LoRaWAN (*Long Range Wide Area Networks*) que utilizam a tecnologia de camada física LoRa. Este trabalho preenche

a lacuna da reprodutibilidade ao propor o RSSignal: um arcabouço de código aberto para geração e validação de chaves criptográficas baseadas no indicador de intensidade do sinal recebido (*Received Signal Strength Indication* – RSSI) das redes sem fio LoRaWAN. Para a etapa de geração das chaves, a implementação atual do arcabouço se baseia na técnica proposta por [da Cruz et al. 2021], que faz uso da coleta passiva do RSSI como entrada para um processo de seis outras etapas responsáveis pela criação e sincronização de chaves simétricas entre as partes. Esta técnica não necessita de conhecimento prévio entre as partes, o que reduz a fragilidade intrínseca do LoRaWAN quando da entrada de um novo dispositivo na rede. Entretanto, é importante destacar que o arcabouço pode ser instanciado com técnicas de geração alternativas para atender às especificidades de outros tipos de redes, mitigando o problema da transmissão aberta dos pacotes de sincronia, controle e acordo de chaves. Para a etapa de validação das chaves, o RSSignal utiliza diferentes conjuntos de dados com medições reais de RSSI para caracterizar cenários estacionários e com mobilidade de nós LoRaWAN. Por fim, as chaves geradas são analisadas quantitativamente do ponto de vista de segurança com auxílio da suíte de testes do NIST 800-22. Paralelamente, este trabalho também discute sobre o desempenho de algoritmos de *hash* quando executados em um computador pessoal com arquitetura *x86* e em dispositivos IoT Raspberry Pi *3B* e *3B+* com arquitetura *ARM*.

De maneira sucinta, este trabalho apresenta as seguintes contribuições que avançam o estado da arte na área de segurança:

1. A proposta do arcabouço RSSignal, viabilizando a reprodutibilidade de pesquisas em métodos para geração e validação de chaves baseadas em RSSI;
2. A instanciação do RSSignal com base na técnica de geração de chaves proposta por [da Cruz et al. 2021], acompanhado da avaliação das chaves geradas para cenários estacionários e com mobilidade de nós LoRaWAN;
3. De maneira transversal, a avaliação e comparação do desempenho de algoritmos de *hash* em dispositivos IoT.

O restante deste trabalho encontra-se organizado da seguinte maneira: Alguns dos aspectos de segurança em redes LoRaWAN e os trabalhos relacionados ao tema são apresentados na Seção 2. O RSSignal e sua instanciação com base na técnica de geração de chaves proposta por [da Cruz et al. 2021] são descritos na Seção 3. Os resultados obtidos durante a validação do arcabouço proposto são expostos na Seção 4. Por fim, as conclusões deste trabalho são apresentadas na Seção 5.

2. Geração de chaves a partir do RSSI em redes LoRaWAN

Os dispositivos LoRa para comunicação de longo alcance operam na faixa de frequência não licenciada abaixo de $1GHz$ e utilizam a modulação de sinal CSS (*Chirp Spread Spectrum*), de propriedade da empresa *Semtech*® [Sinha et al. 2017]. As especificações para dispositivos LoRa detalham o protocolo da camada física e suas configurações, bem como a arquitetura de uma rede LoRaWAN, mostrando como dispositivos LoRa podem integrar as redes IoT [Yegin et al. 2017]. É importante destacar que redes de dispositivos IoT tendem a possuir um número elevado de nós com limitações em recursos computacionais. Considerando a necessidade de criptografar os dados trafegados no meio sem fio e também as limitações de implementação dos aspectos de segurança nos dispositivos de uma rede LoRaWAN, o problema de distribuição de chaves merece especial atenção neste cenário [Jayasuriya 2021].

O conceito de usar características da camada física para geração de chaves foi apresentado inicialmente por [Hershey et al. 1995] e, desde então, tem recebido contribuições significativas em diferentes sistemas e cenários. A utilização da entropia do indicador RSSI como entrada para geração distribuída de chaves simétricas de criptografia em redes LoRaWAN se mostra uma opção atraente para driblar os desafios da distribuição de chaves entre as partes [Kitaura et al. 2007, da Cruz et al. 2021].

Em [Kitaura et al. 2007], os autores se norteiam pelo objetivo de garantir a segurança das informações trafegadas em meios sem fio, demonstrando que é possível utilizar medidas de RSSI como entrada para métodos de geração de chaves privadas em sistemas de rádio comunicadores móveis. Já o trabalho de [Badawy et al. 2016] estuda e compara diferentes formas de obter dados randômicos para geração de chaves criptográficas seguras. Os autores discutem sobre algumas limitações no uso de medidas de RSSI como fonte de dados randômicos. Eles destacam que, principalmente para ambientes onde os valores de RSSI são próximos do ruído do canal, pode ocorrer uma alta taxa de discrepância de *bits* nas chaves geradas pelas partes.

O trabalho de [Han et al. 2020] propõe a utilização de medidas de RSSI em dispositivos LoRa para geração de chaves seguras em redes veiculares de alta mobilidade. O ambiente de alta mobilidade traz algumas particularidades, como o tempo curto em que deve ser gerada a chave, já que a transmissão deve ocorrer dentro da janela de tempo em que há conexão entre os veículos. Os autores analisam medidas de RSSI coletadas e concluem que, mesmo com as limitações do processo de medição e a suscetibilidade à interferências, os valores medidos pelos dispositivos ainda se mantém recíprocos e altamente correlacionados. Os resultados obtidos através de testes práticos são promissores e confirmam a efetividade da proposta.

Em [Goldoni et al. 2022], os autores relacionam as medições de RSSI obtidas por dispositivos LoRa em uma rede em campo, com as características climáticas e ambientais locais, como a temperatura do ar, umidade, pressão e outros. Dentre as principais conclusões, destaca-se a relação praticamente linear entre a temperatura ambiente e o RSSI medido nos experimentos. Por fim, o trabalho de [da Cruz et al. 2021] propõe um mecanismo para geração distribuída de chaves simétricas a partir das medidas de RSSI em uma rede LoRaWAN. Os autores avaliam a relação do método com o nível de randomização da chave gerada, e confirmam, por meio de testes em ambiente real, que o método pode ser utilizado por dispositivos de maneira segura.

3. O Arcabouço RSSignal

Considerando a relevância do tema envolvendo a geração e distribuição de chaves em redes LoRaWAN, e também a importância da reprodutibilidade dos resultados de pesquisa, este trabalho apresenta o RSSignal: um arcabouço de código aberto que implementa, de forma modular, as etapas de geração e validação de chaves que utilizam medidas de RSSI como fonte de dados randômicos. Na Seção 3.1 é detalhado o método de geração de chaves atualmente implementado no RSSignal. Já os aspectos relacionados à validação das chaves são discutidos na Seção 3.2. O código e a documentação para utilização do RSSignal estão disponíveis em seu repositório público no GitHub[1].

[1] `https://github.com/oliveiraleo/RSSignal-LoRa`

3.1. Geração de chaves

A implementação atual da geração de chaves no RSSignal se baseia na arquitetura proposta por [da Cruz et al. 2021]. A Figura 1 ilustra esta arquitetura, que está dividida nas seguintes etapas: coleta de dados, pré-processamento dos dados coletados, quantização, troca de índices para descarte, reconciliação das chaves, amplificação de privacidade e geração da chave. Cada uma destas etapas será detalhada na sequência.

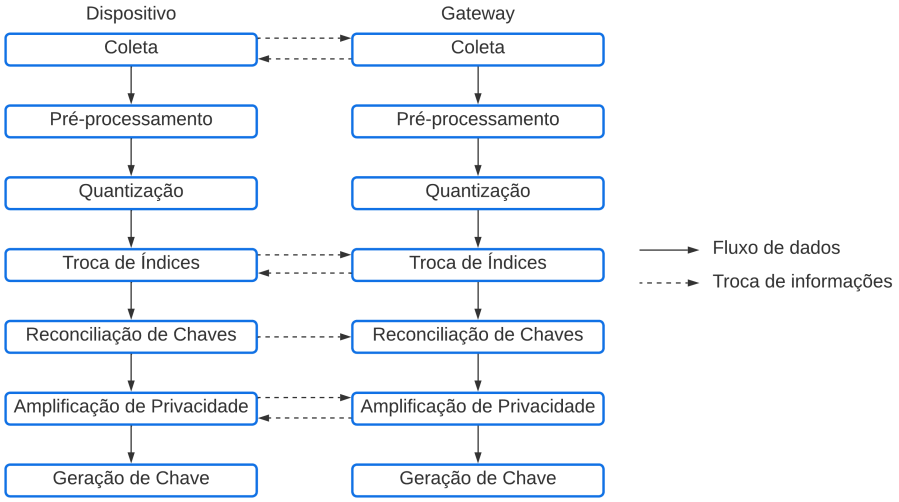

Figura 1. Arquitetura para geração de chaves descrita por [da Cruz et al. 2021].

3.1.1. Coleta de dados

A etapa de coleta de dados consiste na troca de pacotes entre o dispositivo e o *gateway*, de forma que a medida de RSSI possa ser extraída diretamente do cabeçalho dos pacotes LoRaWAN e armazenada de maneira independente por ambas as partes. No RSSignal, a troca de pacotes não ocorre de fato, sendo substituída por valores de RSSI previamente coletados simultaneamente no dispositivo e no *gateway*. Estes valores de entrada estão organizados em diferentes conjuntos de dados que serão melhor detalhados na Seção 3.2.1.

3.1.2. Pré-processamento dos dados coletados

Num ambiente real, o processo de coleta de dados está sujeito às interferências do meio sem fio que podem resultar em perdas de pacotes. Desta forma, a etapa de pré-processamento é a responsável por identificar os valores de RSSI que serão, de fato, utilizados nas próximas etapas da geração das chaves. O arcabouço RSSignal implementa esta etapa através de um filtro com expressões regulares que extrai unicamente os números válidos das medidas de RSSI armazenadas, desconsiderando outras informações ou mesmo dados inválidos.

3.1.3. Quantização

A etapa de quantização é a responsável por transformar os valores de RSSI previamente coletados em sequências de *bits*. Em [da Cruz et al. 2021], os autores propõem duas abor-

dagens: a quantização baseada na média e no desvio padrão (*Mean and Standard Deviation* – MSD) e a quantização baseada na função de distribuição cumulativa com perdas (*Lossy Cumulative Distribution Function* – LCDF). A implementação atual do RSSignal suporta apenas a quantização MSD que, como mostra a equação 1, faz a diferenciação D de uma medida de índice l, de modo que M represente a distância entre l e a medida $l + M$ que será utilizada na diferenciação.

$$D_l = X_l - X_{(l+M)} \tag{1}$$

Como mostra a equação 2, a média μ e o desvio padrão σ são usados para definir dois limiares, um superior (η^+) e outro inferior (η^-), sendo α um fator de ajuste destes.

$$\begin{cases} \eta^+ = \mu_D + \alpha\sigma_D \\ \eta^- = \mu_D - \alpha\sigma_D \end{cases} \tag{2}$$

Então, como mostra a equação 3, os D valores acima do limiar superior são marcados com o *bit* 1, os valores abaixo do limiar inferior são marcados com o *bit* 0 e, por fim, os valores entre os limiares são marcados com o valor 2 para que sejam descartados na etapa seguinte. A marcação de valores como 2 tem como objetivo evitar o uso de medidas agrupadas (as medidas próximas à média), já que isso criaria vários blocos com sequências de 0s e 1s na entrada, o que desfavoreceria a randomização da chave gerada.

$$\begin{cases} K_i = 1, D_i > \eta^+ \\ K_i = 0, D_i < \eta^- \\ K_i = 2, \text{caso contrário} \end{cases} \tag{3}$$

3.1.4. Troca de índices

Depois da etapa de quantização, os nós envolvidos no processo devem enviar um para o outro os índices das medidas que serão descartadas (que foram aqueles marcados com 2 na etapa anterior). Aqui vale destacar que os valores das medidas não são enviados, mas sim os índices que representam as posições delas dentro do conjunto de dados. Isso é importante, pois neste momento o canal de transmissão ainda é considerado inseguro e enviar diretamente os dados coletados poderia acarretar em uma vulnerabilidade de segurança. De fato, o envio em texto plano dos índices de descarte pode fazer com que eles estejam vulneráveis à captura e, então, poderiam ser utilizados para amplificar um ataque de força bruta contra o sistema. Porém, levando em consideração que o atacante não tem acesso físico aos dispositivos, ele não seria capaz de determinar qual o número de medidas utilizadas como entrada para a etapa de quantização, nem quais os parâmetros α e M utilizados, nem mesmo quais foram os efeitos destes sobre a entrada.

Após a troca de índices, os nós executam uma operação de concatenação dos índices que eles mesmos já descartariam com aqueles recebidos do outro nó, reduzindo a disparidade entre os *bits* que serão usados nas etapas seguintes do processo.

3.1.5. Reconciliação de chaves

Mesmo após a coleta simultânea das medidas de RSSI na etapa 3.1.1 e do descarte seletivo das medidas agrupadas na etapa 3.1.4, ainda podem existir *bits* díspares nas sequências de *bits* geradas no dispositivo e o *gateway*. Assim, como sugerido em [da Cruz et al. 2021], o RSSignal implementa um codificador *Reed-Solomon* para diminuir esta disparidade.

Como mostra a Figura 2, este codificador é executado de maneira independente em ambos os nós, tendo como entrada a sequência de *bits* da etapa anterior. Ao fazer a codificação, são gerados símbolos (novos *bits*) de correção. O codificador foi implementado com a taxa de 1 para 2 (1 *byte* de correção para cada 2 *bytes* de dados). O poder de correção t é dado pela equação $t = (n - k)/2$, onde t é o número máximo de *bits* que podem ser corrigidos por este codificador, n representa a quantidade de *bits* de correção, e k é o comprimento da entrada [Fernando et al. 2017].

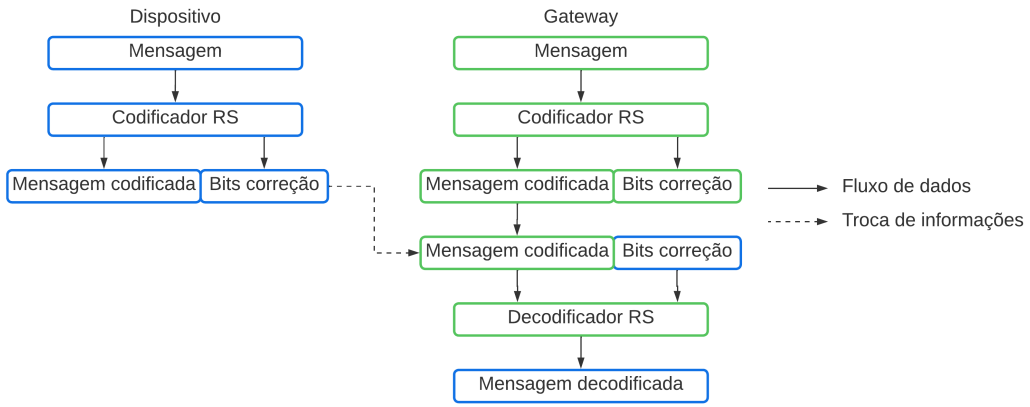

Figura 2. Etapa de reconciliação de chaves.

Os *bits* de correção obtidos pelo dispositivo ao executar o codificador são enviados ao *gateway* que, por sua vez, fará uma operação de decodificação e correção de seus *bits* dados usando as informações recebidas do dispositivo. É importante notar que somente o dispositivo envia seus símbolos de correção para o *gateway*. Como definido por [Fernando et al. 2017], não é possível se obter uma mensagem a partir do conhecimento de seus *bits* de código de correção. Logo, mesmo que o atacante intercepte estes símbolos, já não seria possível recuperar as medidas de RSSI da etapa 3.1.1 de coleta, nem os valores resultantes da etapa 3.1.3 de quantização.

3.1.6. Amplificação de privacidade

A etapa de amplificação de privacidade serve para garantir a igualdade entre as sequências de *bits*, geradas de maneira independente pelos nós, antes que elas sejam usadas na geração da chave. Para isso, um algoritmo de *hash* é aplicado às sequências de modo a obter seus resumos. O dispositivo envia então seu resumo para o *gateway* que, por sua vez, confere a igualdade de seu resumo com aquele gerado pelo dispositivo. Em caso positivo, o *gateway* deve enviar uma mensagem de confirmação ao dispositivo.

Esta etapa é implementada no RSSignal com auxílio do algoritmo SHA3-512. Os algoritmos da família SHA (*Secure Hash Algorithm*) foram projetados com a proprie-

dade de que a entrada pode ser facilmente transformada em um resumo, mas existe um nível de complexidade adequadamente alto para encontrar a entrada a partir do resumo [Preneel 2010, Dworkin et al. 2015]. Para subsidiar a escolha da variante adequada, foram avaliados diferentes algoritmos SHA em três dispositivos distintos: um computador pessoal com processador Intel Core i7 (arquitetura *x86*) e 32GB de memória; um Raspberry Pi 3B com processador BCM2837 (arquitetura *ARM*) e 1GB de memória; e um Raspberry Pi 3B+ com processador BCM2837B0 (arquitetura *ARM*) e 1GB de memória. As Figuras 3 e 4 mostram os resultados das avaliações experimentais.

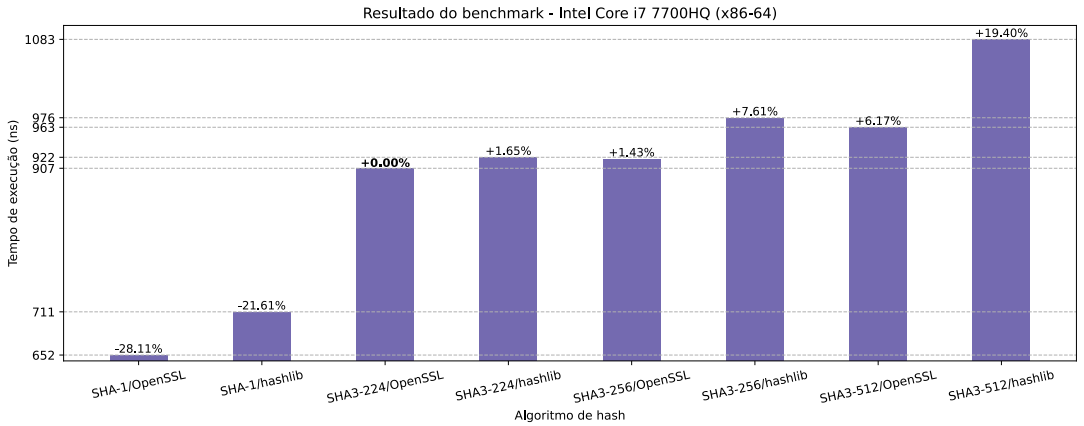

Figura 3. Tempo de execução médio na plataforma x86.

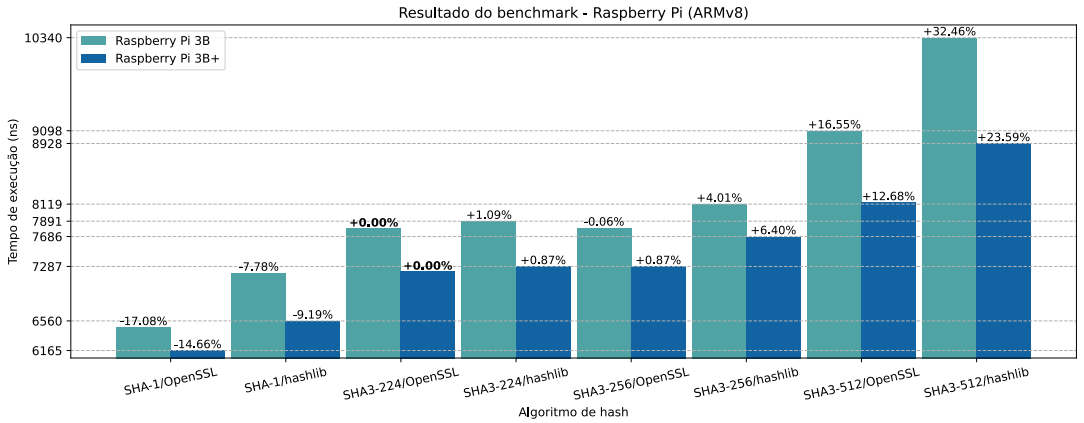

Figura 4. Tempo de execução médio na plataforma ARM.

Apesar do algoritmo SHA-1 ter obtido o menor tempo de execução dentre os avaliados, este foi descartado por ser vulnerável aos ataques de força bruta [Bošnjak et al. 2018]. Por sua vez, o SHA3-512 melhora consideravelmente a segurança do resumo se comparado aos demais algoritmos da família SHA3 [Tchórzewski e Jakóbik 2019]. Por conta disso e da diferença observada entre o tempo de execução dos algoritmos da família SHA3 implementados pelo módulo OpenSSL[2], foi decidido utilizar o algoritmo SHA3-512 na proposta. Analisando os resultados, é possível concluir que, na sua pior média, o SHA3-512 foi 16.55% mais lento que o SHA3-224. Entretanto, como demonstrado por [Tchórzewski e Jakóbik 2019], o SHA3-512 é 128%

[2]A implementação do módulo da etapa 3.1.6 permite que com poucas mudanças no código seja possível alternar entre os algoritmos SHA e, inclusive, eliminar a dependência do módulo OpenSSL.

mais resistente que o SHA3-224, e 100% mais resistente que o SHA3-256 aos ataques por colisão de *hash* e ataques de força bruta por pré-imagem.

3.2. Validação de Chaves

Além da geração de chaves, o arcabouço RSSignal suporta a validação destas com o auxílio de conjuntos de dados provenientes de ambientes reais (descritos na Seção 3.2.1) e de uma suíte de testes estatísticos (descrita na Seção 3.2.2).

3.2.1. Caracterização dos conjuntos de dados

Para a etapa de validação das chaves foram selecionados dois conjuntos de dados com valores de RSSI medidos em redes LoRaWAN reais. Um dos requisitos observados para a escolha destes conjuntos foi a de que as medidas de RSSI deveriam ser realizadas de maneira sequencial e simultânea em ambos os lados da conexão (dispositivo e *gateway*), de modo a explorar a coesão do canal. O outro requisito observado foi o de que as medidas de RSSI deveriam refletir cenários com diferentes padrões de mobilidade dos nós.

O primeiro conjunto de dados foi gentilmente cedido por [da Cruz et al. 2021]. Este conjunto possui três coletas distintas: coleta 1 com 377 medições de RSSI variando entre -9 e -115, representando um ambiente de alta mobilidade; coleta 2 com 676 medições de RSSI variando entre -13 e -65, representando um ambiente com alguma mobilidade; e coleta 3 com 2087 medições de RSSI variando entre -61 a -116, também representando um ambiente com alguma mobilidade. Já o segundo conjunto de dados foi disponibilizado em [Simka e Polak 2022] e a única coleta utilizada possui 200 medições de RSSI variando entre -44 e -68, representando um ambiente sem mobilidade. As Figuras 5, 6, 7 e 8 mostram as medidas de RSSI em cada coleta.

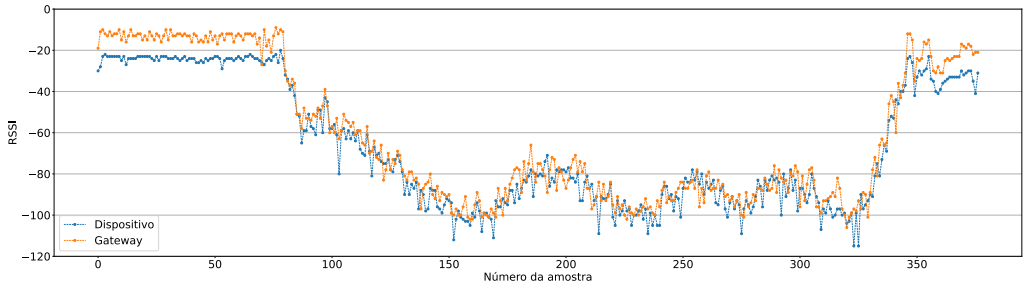

Figura 5. Medidas de RSSI da coleta 1 do conjunto de dados 1.

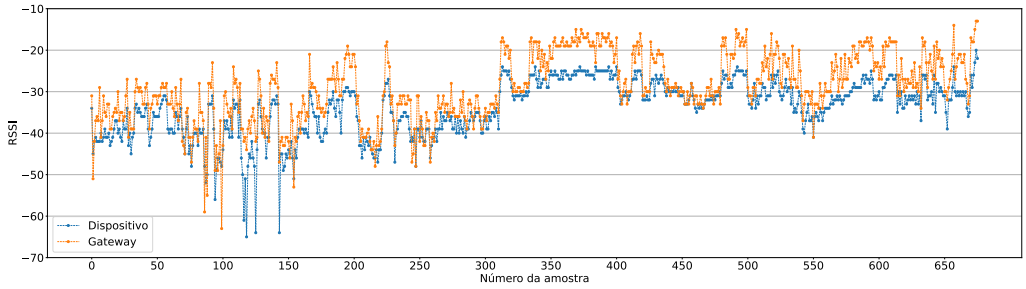

Figura 6. Medidas de RSSI da coleta 2 do conjunto de dados 1.

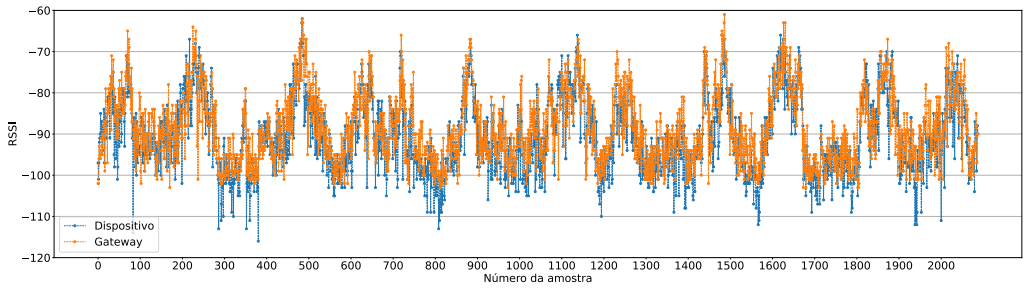

Figura 7. Medidas de RSSI da coleta 3 do conjunto de dados 1.

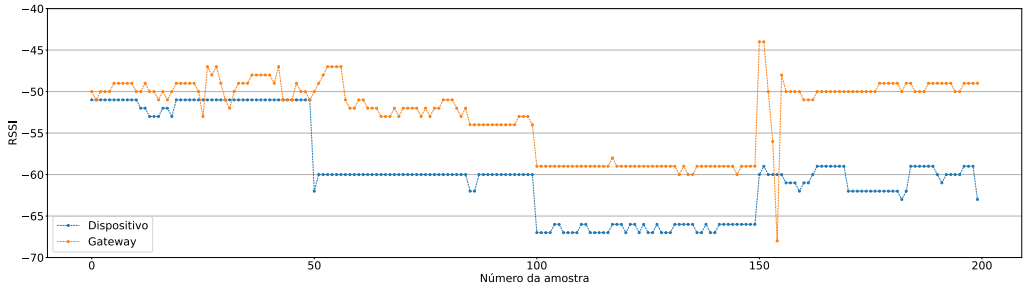

Figura 8. Medidas de RSSI da coleta 1 do conjunto de dados 2.

3.2.2. Suíte de testes do NIST 800-22

Para validar as chaves geradas anteriormente, o RSSignal submete as sequências de *bits* para a suíte de testes NIST 800-22. O NIST (*National Institute of Standards and Technology*) é bem conhecido por seus trabalhos de padronização, e sua suíte de testes estatísticos 800-22 realiza diversas análises sobre a aleatoriedade de uma sequência de *bits*. Os resultados dos testes são expressos através do chamado *p-valor*, e são considerados como bem sucedidos aqueles que obtiverem *p-valor* acima de 0.01. Como destacado por [Marton e Suciu 2015], uma chave é considerada suficientemente randômica caso seja aprovada em 7 ou mais dos 15 grupos de testes disponíveis na suíte. Para este trabalho, foram cuidadosamente escolhidos os testes mostrados nas tabelas da seção 4, já que os demais testes não trariam resultados estatisticamente relevantes por conta do comprimento das chaves geradas [Bassham et al. 2010].

4. Resultados experimentais

Para validar a implementação do arcabouço RSSignal, foram realizados experimentos combinando diferentes valores nos parâmetros de quantização α e M. A Tabela 1 mostra os *p-valores* obtidos na suíte de testes NIST 800-22 para as chaves geradas a partir dos dados da coleta 1 do conjunto 1. Já a Tabela 2 mostra os resultados para os dados da coleta 3 do conjunto 1. Em ambos os casos, os *p-valores* destacados em vermelho indicam testes não aprovados. As tabelas com os *p-valores* para as outras duas coletas não são exibidas aqui por restrições de espaço, mas seus resultados são comentados a seguir.

Os valores de $\alpha > 1.0$ geram uma sequência de *bits* de tamanho muito reduzido. Conforme sugerido por [Marton e Suciu 2015], a sequência deve ter um mínimo de 100 *bits* para ser considerada segura, já que não respeitar esses limites poderia resultar na geração de uma chave não muito resistente à ataques de força bruta.

Tabela 1. Resultados obtidos da suíte de testes NIST 800-22 para diferentes parâmetros de quantização α e M na coleta 1 do conjunto de dados 1.

	α	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
$M=1$	**Frequency**	0.015	0.818	0.251	0.251	0.479	0.917	0.917	0.705	1.000	1.000	0.577
	Block Frequency	0.220	0.071	0.479	0.479	0.479	0.917	0.917	0.705	1.000	1.000	0.577
	Run	**0.000**	**0.000**	**0.002**	**0.002**	0.046	0.029	0.029	0.016	**0.003**	**0.003**	0.081
	Run (Longest run of ones)	**0.010**	0.140	**0.060**	0.060	0.107	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	Serial	0.247 0.314	**0.000** **0.009**	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498	**0.000** **0.498**	**0.000** **0.498**	0.499 0.498
	Entropy	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Cum. Sums (Forward)	0.026	0.269	0.444	0.444	0.737	0.691	0.691	0.513	0.601	0.601	0.529
	Cum. Sums (Backward)	0.030	0.411	0.302	0.302	0.223	0.596	0.596	0.853	0.601	0.601	0.976
	Input length	375	302	194	194	128	**93**	**93**	**63**	**46**	**46**	**29**
$M=2$	**Frequency**	0.030	0.527	0.382	0.429	0.429	0.446	0.292	0.446	0.500	0.446	0.869
	Block Frequency	0.368	0.182	0.052	0.111	0.111	0.446	0.292	0.446	0.500	0.446	0.869
	Run	**0.000**	0.019	0.409	0.222	0.222	0.528	0.990	0.399	0.458	0.708	0.413
	Run (Longest run of ones)	0.985	0.340	0.181	0.161	0.161	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	Serial	0.086 0.685	0.046 0.498	**0.000** **0.051**	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498
	Entropy	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Cum. Sums (Forward)	0.052	0.169	0.213	0.266	0.266	0.364	0.320	0.408	0.355	**0.000**	0.278
	Cum. Sums (Backward)	0.060	0.548	0.683	0.604	0.604	0.762	0.483	0.733	0.807	0.711	1.000
	Input length	374	303	221	160	160	110	**73**	**62**	**55**	**43**	**37**
$M=3$	**Frequency**	**0.006**	0.687	0.841	0.323	0.867	0.792	0.770	0.662	0.622	0.599	0.655
	Block Frequency	0.265	0.458	0.479	0.052	1.000	0.724	0.770	0.662	0.622	0.599	0.655
	Run	**0.000**	**0.000**	**0.001**	**0.002**	0.012	0.043	**0.008**	0.084	0.228	0.122	0.027
	Run (Longest run of ones)	**0.006**	0.138	0.169	0.334	0.687	0.213	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	Serial	**0.000** **0.073**	0.046 0.724	0.872 0.789	0.931 0.852	0.965 0.899	0.977 0.920	0.993 0.999	0.499 0.498	0.499 0.498	0.499 0.498	0.499 0.498
	Entropy	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Cum. Sums (Forward)	**0.007**	0.704	0.691	0.080	0.307	0.318	0.286	0.203	0.129	1.000	0.272
	Cum. Sums (Backward)	0.012	0.373	0.513	0.507	0.416	0.503	0.482	0.459	0.351	0.297	1.000
	Input length	373	303	225	173	142	129	105	**84**	**66**	**58**	**45**
$M=4$	**Frequency**	0.213	0.365	0.127	0.376	0.678	0.510	0.841	0.518	0.907	0.796	0.376
	Block Frequency	1.000	0.032	**0.003**	0.021	0.596	0.510	0.841	0.518	0.907	0.796	0.376
	Run	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.004**	**0.009**	0.011	0.026	0.020	0.022
	Run (Longest run of ones)	0.297	0.017	0.056	0.524	**0.003**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
	Serial	**0.000** **0.000**	**0.000** **0.000**	**0.000** **0.000**	**0.000** **0.162**	**0.000** **0.106**	**0.000** **0.946**	0.499 0.999	0.499 0.999	0.999 0.994	0.499 0.498	0.499 0.498
	Entropy	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Cum. Sums (Forward)	0.323	0.054	**0.006**	0.037	0.112	0.061	0.178	0.322	0.203	0.141	1.000
	Cum. Sums (Backward)	0.174	0.385	0.222	0.280	0.270	0.264	0.115	0.081	0.158	0.243	0.476
	Input length	372	312	227	184	145	113	100	**86**	**73**	**60**	**46**

Tabela 2. Resultados obtidos da suíte de testes NIST 800-22 para diferentes parâmetros de quantização α e M na coleta 3 do conjunto de dados 1.

	α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$M = 1$	Frequency	0.002	0.601	0.636	0.636	0.815	0.631	0.686	0.762	0.762	0.573	0.530
	Block Frequency	0.753	0.992	0.972	0.972	0.986	0.638	0.960	0.824	0.824	0.596	0.377
	Run	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Run (Longest run of ones)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.012	0.012	0.197	0.801
	Serial	0.001 0.083	0.057 0.093	0.038 0.200	0.038 0.200	0.126 0.418	0.018 0.688	0.798 0.981	0.026 0.322	0.026 0.322	0.978 0.999	0.499 0.499
	Entropy	0.039	0.814	0.999	0.999	0.999	1.000	1.000	1.000	1.000	1.000	1.000
	Cum. Sums (Forward)	0.004	0.906	0.833	0.833	0.991	0.700	0.694	0.763	0.763	0.571	0.586
	Cum. Sums (Backward)	0.003	0.937	0.953	0.953	0.999	0.896	0.906	0.977	0.977	0.857	0.898
	Input length	2087	1774	1291	1291	893	733	612	394	394	255	205
$M = 2$	Frequency	0.007	0.437	0.625	0.544	0.594	0.394	0.832	1.000	0.874	0.903	0.898
	Block Frequency	0.299	0.869	0.959	0.932	0.751	0.271	0.365	0.880	0.969	0.984	0.860
	Run	0.000	0.011	0.962	0.326	0.217	0.039	0.082	0.251	0.791	0.902	0.897
	Run (Longest run of ones)	0.000	0.000	0.003	0.011	0.034	0.075	0.426	0.418	0.588	0.757	0.289
	Serial	0.089 0.271	0.100 0.157	0.223 0.824	0.922 0.629	0.498 0.038	0.240 0.894	0.322 0.165	0.691 0.760	0.500 0.306	0.499 0.499	0.499 0.499
	Entropy	0.142	0.462	0.972	0.999	0.999	0.999	1.000	1.000	1.000	1.000	1.000
	Cum. Sums (Forward)	0.010	0.514	0.947	0.851	0.855	0.499	0.978	0.923	0.995	0.919	0.968
	Cum. Sums (Backward)	0.008	0.834	0.894	0.769	0.828	0.414	0.847	0.923	0.999	0.979	0.997
	Input length	2086	1799	1360	1086	1015	727	557	512	359	270	242
$M = 3$	Frequency	0.001	0.284	0.578	0.902	0.902	0.772	0.965	0.965	0.715	0.522	0.403
	Block Frequency	0.224	0.883	0.967	0.410	0.410	0.952	0.997	0.997	0.820	0.860	0.860
	Run	0.000	0.000	0.000	0.000	0.000	0.171	0.692	0.692	0.953	0.681	0.663
	Run (Longest run of ones)	0.000	0.000	0.003	0.023	0.023	0.060	0.720	0.720	0.273	0.660	0.356
	Serial	0.000 0.000	0.835 0.837	0.024 0.081	0.000 0.043	0.000 0.043	0.500 0.405	0.840 0.961	0.840 0.961	0.760 0.500	0.500 0.498	0.500 0.498
	Entropy	0.136	0.292	0.659	0.999	0.999	0.999	1.000	1.000	1.000	1.000	1.000
	Cum. Sums (Forward)	0.001	0.505	0.796	0.992	0.992	0.907	0.985	0.985	0.783	0.894	0.507
	Cum. Sums (Backward)	0.001	0.383	0.697	0.950	0.950	0.931	0.993	0.993	0.874	0.842	0.640
	Input length	2085	1840	1425	1048	1048	766	517	517	367	244	173
$M = 4$	Frequency	0.004	0.778	0.810	0.443	0.443	0.254	0.240	0.402	0.402	0.640	0.679
	Block Frequency	0.132	0.998	0.996	0.809	0.809	0.304	0.248	0.158	0.158	0.969	0.724
	Run	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.005	0.002	0.055
	Run (Longest run of ones)	0.000	0.000	0.000	0.052	0.052	0.015	0.349	0.076	0.076	0.383	0.017
	Serial	0.000 0.217	0.992 0.999	0.000 0.000	0.027 0.003	0.027 0.003	0.000 0.000	0.673 0.973	0.499 0.329	0.499 0.329	0.190 0.498	0.499 0.498
	Entropy	0.026	0.135	0.982	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Cum. Sums (Forward)	0.007	0.837	0.998	0.731	0.731	0.351	0.386	0.641	0.641	0.849	0.948
	Cum. Sums (Backward)	0.005	0.974	0.975	0.438	0.438	0.351	0.284	0.252	0.252	0.743	0.948
	Input length	2084	1816	1409	1061	1061	788	568	411	411	292	210

Para o conjunto de dados 1, os resultados encontrados foram os seguintes: na coleta 1, para $M = 1$, os valores de $\alpha = \{0.3, 0.4\}$ foram aprovados; para $M = 2$, os valores de $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5\}$ foram aprovados; quando $M = 3$, os valores aprovados foram $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5\}$; e, por fim, para $M = 4$, nenhum valor de α foi aprovado. Já na coleta 2, para $M = 1$, os valores de $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.6\}$ foram aprovados; para $M = 2$, os valores de $\alpha = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ foram aprovados; quando $M = 3$, os valores aprovados foram $\alpha = \{0.1, 0.2, 0.5, 0.6, 0.7, 0.8, 0.9\}$; e, por fim, para $M = 4$, os valores de $\alpha = \{0.5, 0.8, 0.9\}$ foram aprovados. Por fim, na coleta 3, para $M = 1$, os valores de $\alpha = \{0.7, 0.8, 0.9, 1.0\}$ foram aprovados; para $M = 2$, os valores de $\alpha = \{0.1, 0.2, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ foram aprovados; quando $M = 3$, os valores aprovados foram $\alpha = \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$; e, por fim, para $M = 4$, os valores de $\alpha = \{0.6, 0.7, 0.8, 0.9, 1.0\}$ foram aprovados. Para a única coleta do conjunto de dados 2, somente os valores obtidos com $\alpha = 0.0$ produziram sequências com 100 *bits* ou mais. Mesmo considerando apenas $\alpha = 0.0$, todos os 4 valores de M foram reprovados na maioria dos testes. Em resumo, esse ambiente de baixa mobilidade pode ser considerado como inapto para a técnica atualmente implementada no arcabouço RSSignal.

Ao analisar estes resultados é possível concluir que, se não houver descarte (quando $\alpha = 0.0$), somente a diferenciação não é capaz de proporcionar uma geração de chaves seguras. Outra observação a ser feita é a de que ambientes de baixa mobilidade precisam de mais medidas para que o método funcione corretamente, pois ao executar o descarte de medidas neste tipo de cenário, geralmente o comprimento das chaves é demasiadamente reduzido. Por fim, outros métodos de geração de chaves devem ser adotados para ambientes sem mobilidade, pois a pouca variabilidade temporal nas medidas de RSSI não foi capaz de produzir uma sequência de *bits* minimamente segura em nenhuma das configurações de parâmetros avaliadas.

5. Conclusões

Este trabalho apresenta o arcabouço RSSignal para geração e validação de chaves a partir do indicador de sinal RSSI. A possibilidade de medir o RSSI está presente em diversas tecnologias sem fio e, em conjunto com técnicas de quantização, elas podem servir como entrada para algoritmos de acordo de chaves. A implementação atual do arcabouço é baseada no trabalho de [da Cruz et al. 2021]. A técnica avaliada não necessita do prévio estabelecimento de um canal seguro de comunicação, eliminando o problema da eventual transmissão da chave de criptografia por um canal inseguro, e também garante uma resistência aos ataques de força bruta já que há uma redução na quantidade de parâmetros enviados durante o processo. Para a etapa de validação foram selecionados conjuntos de dados reais como entrada.

Com relação às características de recursos limitados para dispositivos IoT, foram realizados diversos experimentos com o objetivo de apoiar a decisão sobre qual algoritmo de *hash* utilizar na etapa de amplificação de privacidade. Por fim, os resultados baseados na suíte do NIST 800-22 demonstram que cenários com alta mobilidade e variância de RSSI são mais propensos à geração de chaves criptograficamente seguras.

Agradecimentos

Os autores gostariam de agradecer ao Sr. Pedro Ivo da Cruz pelo apoio ao compartilhar saberes e ao Sr. Rodrigo Oliveira Silva pela assistência na etapa de implementação. Os autores também agradecem à UFJF, FAPEMIG (APQ-00999-18), e FAPESP (2018/23062-5) pelo apoio financeiro para o desenvolvimento desta pesquisa.

Referências

Badawy, A., Elfouly, T., Khattab, T., Mohamed, A., e Guizani, M. (2016). Unleashing the secure potential of the wireless physical layer: Secret key generation methods. *Physical Communication*, 19:1–10.

Bassham, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., et al. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards and Technology - NIST.

Bošnjak, L., Sreš, J., e Brumen, B. (2018). Brute-force and dictionary attack on hashed real-world passwords. Em *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics*, páginas 1161–1166.

Codeluppi, G., Cilfone, A., Davoli, L., e Ferrari, G. (2020). LoRaFarM: A LoRaWAN-based smart farming modular IoT architecture. *Sensors*, 20(7).

da Cruz, P. I., Suyama, R., e Loiola, M. B. (2021). Increasing key randomness in physical layer key generation based on RSSI in LoRaWAN devices. *Physical Communication*, 49:101480.

Deelman, E., Stodden, V., Taufer, M., e Welch, V. (2019). Initial thoughts on cybersecurity and reproducibility. Em *Proceedings of the 2nd International Workshop on Practical Reproducible Evaluation of Computer Systems*, páginas 13–15.

Dworkin, M. J. et al. (2015). SHA-3 standard: Permutation-based hash and extendable-output functions.

Fernando, M., Jayalath, D., Camtepe, S., e Foo, E. (2017). Reed Solomon codes for the reconciliation of wireless PHY layer based secret keys. Em *Proceedings of the 86th Vehicular Technology Conference*, páginas 1–6.

Goldoni, E., Savazzi, P., Favalli, L., e Vizziello, A. (2022). Correlation between weather and signal strength in LoRaWAN networks: An extensive dataset. *Computer Networks*, 202:108627.

Han, B., Peng, S., Wu, C., Wang, X., e Wang, B. (2020). LoRa-based physical layer key generation for secure V2V/V2I communications. *Sensors*, 20(3):682.

Hershey, J., Hassan, A., e Yarlagadda, R. (1995). Unconventional cryptographic keying variable management. *IEEE Transactions on Communications*, 43(1):3–6.

Jayasuriya, E. N. (2021). *ECDH Based Key Management for LoRaWAN Considering Sensor Node Limitations*. Tese de doutorado, University of Colombo.

Jiang, X., Lora, M., e Chattopadhyay, S. (2020). An experimental analysis of security vulnerabilities in industrial IoT devices. *ACM Transactions on Internet Technology*, 20(2):1–24.

Kitaura, A., Iwai, H., e Sasaoka, H. (2007). A scheme of secret key agreement based on received signal strength variation by antenna switching in land mobile radio. Em *Proceedins of the 9th International Conference on Advanced Communication Technology*, volume 3, páginas 1763–1767.

Machina e Gartner (2016). Global Internet of things market to grow to 27 billion devices, generating USD 3 trillion revenue in 2025.

Marton, K. e Suciu, A. (2015). On the interpretation of results from the NIST statistical test suite. *Science and Technology*, 18(1):18–32.

Pasolini, G., Buratti, C., Feltrin, L., Zabini, F., De Castro, C., Verdone, R., e Andrisano, O. (2018). Smart city pilot projects using LoRa and IEEE802.15.4 technologies. *Sensors*, 18(4):1118.

Preneel, B. (2010). Cryptographic hash functions: theory and practice. Em *Proceedings of the 11th International Conference on Cryptology in India*, páginas 115–117.

Simka, M. e Polak, L. (2022). On the RSSI-based indoor localization employing LoRa in the 2.4 GHz ISM band. *Radioengineering*, 31(1):135–143.

Sinha, R. S., Wei, Y., e Hwang, S.-H. (2017). A survey on LPWA technology: LoRa and NB-IoT. *ICT Express*, 3(1):14–21.

Statista (2016). Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions).

Stellios, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., e Lopez, J. (2018). A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services. *IEEE Communications Surveys & Tutorials*, 20(4):3453–3495.

Tchórzewski, J. e Jakóbik, A. (2019). Theoretical and experimental analysis of cryptographic hash functions. *Journal of Telecommunications and Information Technology*, 1(11):125–133.

Yegin, A., Sornin, N., e the LoRa Alliance Technical Committee (2017). LoRaWAN™ 1.1 Specification.