Increasing cross-correlation in LoRaWAN RSSI based key generation with DCT and PCA

Pedro Ivo da Cruz, Matheus Costa Damasceno, Ricardo Suyama and Murilo Bellezoni Loiola

Abstract— **Physical-layer Key generation for wireless networks has been considered a promising alternative to traditional techniques based on pseudorandom number generators. This work evaluates key generation in a Long Range Wide Area (LoRaWAN) based on Received Signal Strength Indicator (RSSI) values, measured during the reception of LoRaWAN packets. This work uses Discrete Cosine Transform (DCT) and Principal Component Analysis (PCA) to increase reciprocity in the measurements. The usage of PCA for this purpose with RSSI values is a novelty of this work. Simulation results indicate that PCA can produce higher correlation values and lower key disagreement rate (KDR) than DCT.**

Keywords— **physical layer security, wireless communications, signal processing**

I. Introduction

Low energy consumption devices with low computational power are widely used in wireless sensor networks as gateways and, mainly, as end nodes that collect data through sensors or provide control to actuators [1]. New communication protocols are necessary to allow efficient communication through the wireless medium for these devices. One of these protocols is the Long Range Wide Area Network (LoRaWAN) [2], which makes use of the Long Range (LoRa) protocol in the physical layer. Nonetheless, energy and computational constraints, as well as a high number of nodes exchanging information, make the employment of traditional security techniques, such as key-based cryptography advanced encryption system [3] and the RSA [4], challenging in such networks.

In this context, encryption key generation based on the wireless channel information helps to reduce the energy consumption and the computational power necessary to generate secret keys and its distribution in networks with several nodes [1]. It considers the random behaviour of the wireless channel as a source of randomness to generate the encryption key. The channel reciprocity guarantees that legitimate nodes that exchange information between themselves observe the same channel and, therefore, generate the same key. The channel reciprocity can only be considered during a small time window, however, due to the possibility of mobility between legitimate nodes. Also, the channel variability over time allows the legitimate nodes to sample it in different time instants, collecting more information to generate a key long enough to reach encryption algorithms requirements. The channel spatial decorrelation hinders an eavesdropper that is located in a different location – usually with a distance $d > \lambda/2$, where λ is the carrier wavelength – from the authentic nodes to generate the same key since it will not observe the same channel.

It is desirable that the channel measurements made in both legitimate users be highly correlated [5], [6] to reduce the number of mismatched bits in the keys. Therefore, the quality of the keys highly depends on the assumption that the channel between legitimate users is reciprocal. However, differences in the hardware and the presence of additive noise degrade the reciprocity in the received signal. A denoising procedure using a low pass filter has been investigated in [7] under the assumption that the main cause of the non-reciprocity is the noise present in the high frequencies. The use of Discrete Cosine Transform (DCT), also aiming noise removal, is investigated in [8], Discrete Wavelet Transform in [9], and Principal Component Analysis (PCA) in [10]. These works, however, consider systems such as Orthogonal Frequency Division Multiplexing (OFDM) or with multiple antennas, which are not the case for LoRaWAN systems, which usually uses spread spectrum modulation and a single antenna [2].

The key generation has been investigated in LoRaWAN devices in [11], where a quantization algorithm to generate the key from the channel measurements considering the received signal strength indicator (RSSI), that is available in the LoRaWAN packet, is proposed. The key generation in LoRaWAN devices is further investigated in [12], where a sample selection is made before quantization to reduce the bit mismatch in the key. A preprocessing of the RSSI samples can also be performed to remove high frequency noise components and abnormal RSSI values using low pass filters [13] or DCT [12].

The works in [11]–[13] consider that legitimate nodes obtain the RSSI values through the exchange of LoRa packets, i.e., only the physical layer packets are exchanged and the LoRaWAN packets, which runs in the MAC layer, are not used. It is, however, interesting to investigate methods that allow the devices to acquire RSSI values during the exchange of LoRaWAN packets and use them to generate secret keys. The use of LoRaWAN, however, increases the time between RSSI measurements at the end node and the gateway. A requirement of the LoRaWAN Class A, for instance, is that the downlink occurs 1 or 2 seconds after the end node sends the uplink packet. This decreases the cross-correlation between the RSSI values obtained in both legitimate nodes, which will, therefore, increase the key disagreement rate (KDR), i.e. the number of mismatched bits in the keys generated at the end node and the

The authors are with the Engineering, Modeling and Applied Social Sciences Center, Federal University of ABC, Santo André, SP, Brazil, E-mails: pedro.cruz@ufabc.edu.br, matheus.costa@aluno.ufabc.edu.br, ricardo.suyama@ufabc.edu.br, murilo.loiola@ufabc.edu.br. This work was funded in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 - and the National Council for Scientific and Technological Development – CNPq.

gateway increases.

In this work, this problem is addressed by employing the exchange of LoRaWAN packets to acquire RSSI values and using the discrete cosine transform (DCT) and the principal component analysis (PCA) to process the RSSI data obtained in each legitimate node to improve their cross-correlation. The goal is to use the DCT and PCA to remove components from the RSSI signals that might be causing discrepancies and, thus, reducing the cross-correlation. To the best of the authors' knowledge, the usage of PCA in RSSI values aiming the increase of correlation in the data used to generate encryption keys was still not investigated in the literature.

The paper is organized as follows: section II describes the proposed procedure to acquire the RSSI values; the DCT and the proposed PCA techniques to remove components that causes non-reciprocity in the channel RSSIs are described in section III, together with the quantization method used in this work to generate bits from the RSSI values that can be used as encryption key; results are shown and discussed in section IV; finally, conclusions are presented in section V.

II. Data Acquisition

In LoRaWAN, the end device is not always awake, i.e., it is not always able to receive packets from the gateway. The gateway, on the other hand, is connected to a LoRaWAN server and is always able to receive messages from the end device, which can, therefore, initiate the uplink transmission by sending a "*begin*" packet. The end device, configured as a LoRaWAN Class A device, will, then, open two receive windows to receive a downlink packet from the gateway, which can only occur after 1 or 2 seconds after receiving the uplink packet.

The RSSI measurements are taken in the following way:

1) The end node sends the initial uplink transmission. Upon receiving the packet, the gateway extracts the RSSI.
2) The gateway sends a packet to the end node, which extracts the RSSI.
3) The end node can, then, proceeds to send another packet to the gateway and opens the receiving windows so that the above procedure can be repeated several times until both nodes collect enough measurements.

During operation, packet loss might happen some times. Two workarounds are considered to prevent errors due to packet loss depending on the direction of the transmission:

- **Gateway → End Node:** upon not receiving a message from the gateway, the end node will send a message of "*no answer*" so that the gateway can resend the previous packet. The gateway extracts the RSSI from this packet and discards the previous one. If the end node still does not receive a packet, it will keep sending the "*no answer*" packet indefinitely.
- **End Node → Gateway:** if the gateway does not receive a response from the end node, it will not send another packet. In this case, neither nodes obtain RSSI values. Upon not receiving a packet, the end node will send a message of "*no answer*" to the gateway, that can extract the RSSI from this packet and restart the measurement procedure.

In this work, the above procedure is employed and the RSSI values are stored. These values are then processed in an external computer to simplify the analysis.

III. Preprocessing and Key Generation

In order to increase the cross-correlation between the measurements made at the gateway and the end node, the DCT and the PCA techniques are employed. The following subsections describe these techniques.

A. Discrete Cosine Transform

Given a signal $x(n)$ containing N samples, its DCT is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \quad k = 0, \cdots, N-1 \tag{1}$$

On the assumption that the high frequency components, i.e., high values of k, causes the most discrepancies in the RSSI obtained in the gateway and the end node, these values can be substituted by zero and keeping intact the first N_c components. In other words, both nodes perform

$$\tilde{X}(k) = \begin{cases} X(k), & 0 \le k \le N_c - 1 \\ 0, & N_c \le k \le N-1 \end{cases}. \tag{2}$$

Differently from [8], in this work the result of the operation in (2) is brought back to time domain by using the Inverse Discrete Cosine Transform (IDCT), obtaining a reconstructed signal $\tilde{x}(n) = \mathrm{IDCT}\{\tilde{X}(k)\}$ that is used as input to the quantizer.

B. Principal Component Analysis

To perform the PCA, the RSSI values are organized in a data matrix $\mathbf{X}$ with dimensions $2N-1 \times N$ columns, defined as

$$\mathbf{X} = \begin{bmatrix} x(0) & 0 & 0 & \cdots \\ x(1) & x(0) & 0 & \cdots \\ x(2) & x(1) & x(0) & \cdots \\ \vdots & \ddots & \ddots & \cdots \\ x(N-1) & x(N-2) & \ddots & \cdots \\ 0 & x(N-1) & \ddots & \cdots \\ \vdots & \vdots & \ddots & x(N-2) \\ 0 & 0 & \cdots & x(N-1) \end{bmatrix}. \tag{3}$$

The mean for each column of this matrix is computed and organized in a vector $\mathbf{u} = [\mu_0, \ \mu_1, \ \cdots, \ \mu_{N-1}]$, where μ_i is the mean of the i-th column. This vector is then subtracted from each row of the data matrix by

$$\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{1}_{2N-1\times 1}\mathbf{u}, \tag{4}$$

where $\mathbf{1}_{2N-1\times 1}$ is a $2N-1 \times 1$ column vector of 1s. The covariance matrix is, then, computed as

$$\mathbf{R_X} = \left(\tilde{\mathbf{X}}^{\mathrm{H}}\tilde{\mathbf{X}}\right) \odot \mathbf{M}, \tag{5}$$

where $(\cdot)^{\mathrm{H}}$ denotes the conjugate transpose of a matrix, $\odot$ denotes the Hadamard product and $\mathbf{M}$ is a $N \times N$ normalization matrix whose element is given by

$$m_{ij} = \frac{1}{N - |i-j|}, \; i,j = 0, \; \cdots, \; N-1. \tag{6}$$

The covariance matrix has an eigenvalue decomposition given by

$$\mathbf{R_X} = \mathbf{V \Lambda V}^{-1}, \tag{7}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues and $\mathbf{V}$ is the matrix of eigenvectors. The columns of $\mathbf{V}$ and $\mathbf{\Lambda}$ are then sorted in order of decreasing eigenvalues.

The assumption here is that the smaller singular values in $\mathbf{\Lambda}$ are responsible for the non-reciprocity in the RSSI measurements obtained at the gateway and the end node. The effects of these components can be removed by projecting the RSSI values into a basis vector $\tilde{\mathbf{V}}$ by

$$\tilde{x}(n) = \tilde{\mathbf{V}}\tilde{\mathbf{V}}^{\mathrm{T}}\mathbf{x}, \tag{8}$$

where $\mathbf{x} = [x(0), \; x(1, \; \cdots, \; x(N-1)]^{\mathrm{T}}$. The basis vector $\tilde{\mathbf{V}}$ is chosen by selecting only the first N_s columns of the sorted eigenvector matrix $\mathbf{V}$ and replacing the remaining columns elements by 0s.

C. RSSI Quantization

The raw RSSI values, $x_u(n)$, or the processed samples, $\tilde{x}_u(n)$, then go through the quantization step, which generates a sequence of bits to be used as encryption key at the user $u = \{EN, \; G\}$, with EN denoting the end node and G denoting the gateway. To simplify further notation, the input to the quantizer will be denoted by $Z_u(n)$, which will be $Z_u(n) = x_u(n)$ for the raw RSSI values and $Z_u(n) = \tilde{x}_u(n)$ for the processed data. The algorithm used here is the mean based quantization, given by

$$K_u(n) = \begin{cases} 0, & Z_u(n) \leq \mu \\ 1, & Z_u(n) > \mu \end{cases}, \tag{9}$$

where μ is the mean of the input values Z_u.

IV. Results and Discussion

The data was collected through a period of 55 minutes and 12 seconds, resulting in 2088 RSSI samples for each user. The gateway was placed inside one of the buildings of the Federal University of ABC and the end node was moving randomly outside the building at walking speed.

Both end node and gateway obtain their respective RSSI data, which are extracted and processed offline in another machine to allow an easier evaluation of different parameters, such a window size for DCT and PCA and the number of DCT components and number of principal components used.

The results will be evaluated in terms of the cross-correlation between the RSSI values obtained in end node and the gateway and the KDR of the generated keys. The cross-correlation will be given by the correlation coefficient

$$\rho = \frac{\mathrm{E}\{Z_a Z_b^{\mathrm{H}}\}}{\sigma_{EN}\sigma_G}, \tag{10}$$

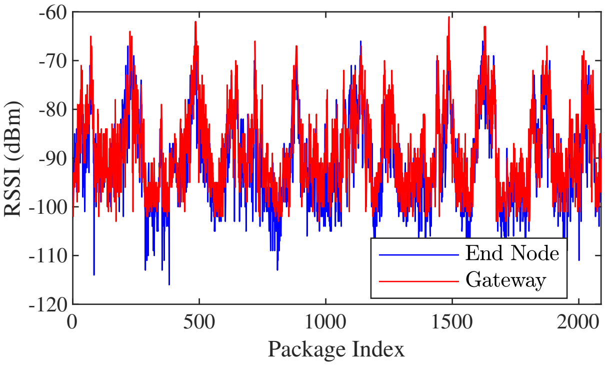

Fig. 1. Collected RSSI values, with $\rho = 0.73$.

where σ_{EN} and σ_G are the standard deviation of Z_{EN} and Z_G, respectively.

The KDR is computed as

$$KDR = \frac{\sum_i |K_{EN}(i) - K_G(i)|}{N_k}, \tag{11}$$

where N_k is the length of the keys and gives the ratio of mismatched bits between the keys generated at end node and the gateway.

A. Raw data

Fig. 1 shows the collected RSSI. These values have a Pearson correlation of 0.73, and the keys generated obtained a KDR around 0.214.

B. Processed data

The DCT and the PCA are computed blockwise, i.e., instead of preprocessing the whole data at once, it is divided into smaller blocks containing N_b samples. The case where the whole data is processed at once, $N_b = N = 2088$, is also evaluated.

Fig. 2 shows the correlation obtained at the output of the IDCT. It is possible to observe that as the number of DCT components used reduces, the correlation increases. This happens since more DCT components responsible for non-reciprocity are removed when N_c decreases. Furthermore, processing the data blockwise decreases the correlation, which is not desired.

It is possible to observe, however, that when $N_b = 2088$, the correlation obtains a maximum when $N_c = 25$. Beyond this point, reducing N_c also decreases the correlation. This indicates that some low-frequency DCT components might also be responsible for some level of non-reciprocity. This is not observed for lower N_c values, where processing the data blockwise produces higher correlated data. In this case, some high-frequency components responsible for reciprocity are not removed due to blockwise processing.

The KDR obtained after quantization is shown in Fig. 3. Here, it is important to note that the lower values of KDR are of the order of 10^{-2} and were obtained from a number of samples of the order of 10^3. This explains the oscillations observed and that increases for lower KDR values. More data should be collected to reduce these oscillations.

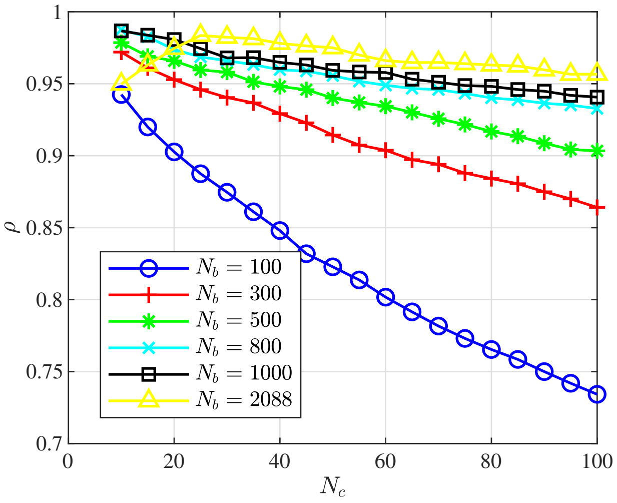

Fig. 2. Correlation with DCT processing in function of the number of components used (N_c) and block size (N_b).

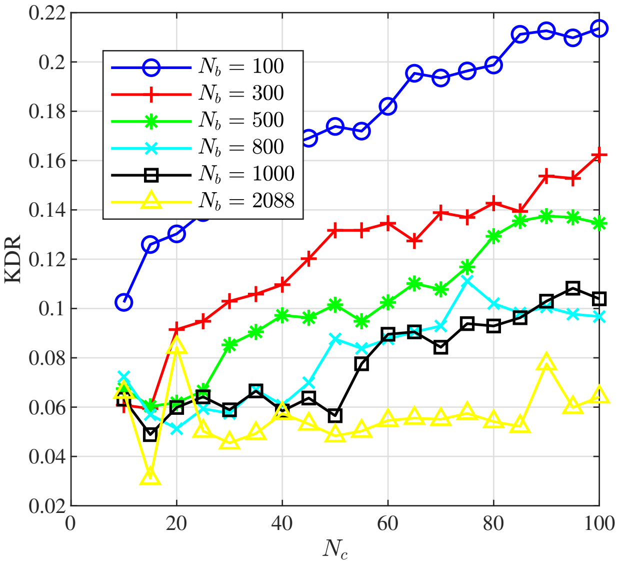

Fig. 3. KDR with DCT processing in function of the number of components used (N_c) and block size (N_b).

Nonetheless, it is possible to observe some trends in the curves that are consistent with what was observed in the correlation curves. First, employing the DCT blockwise increases the KDR, the opposite of what is intended. The best scenario is when the DCT is employed in the whole data at once, providing the lower KDR values, remaining below 0.1. For these values, it is possible to design efficient key agreement code that allow correcting the mismatched bits in the keys [14]. Second, it is possible to observe that the KDR decreases when N_c decreases, due to the removal of the DCT components that are responsible for some of the non-reciprocity in the RSSI values.

The PCA is also employed in the whole data at once and blockwise. The correlation results are shown in Fig. 4. It is possible to observe that the PCA provides an overall performance higher than the DCT, obtaining higher correlation for all values of N_b and N_s. An exception is observed for $N_b = 100$ and $N_s = 100$, since in this case this is equivalent to not perform any modification on the signal.

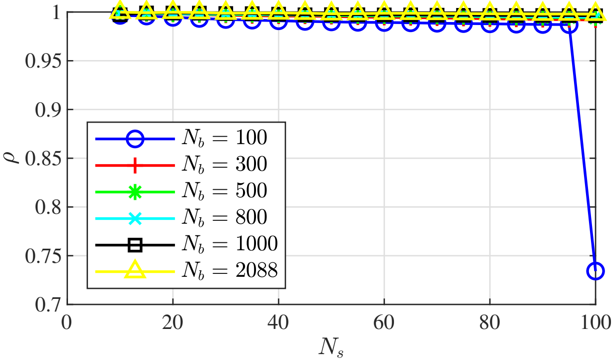

Fig. 4. Correlation with PCA processing in function of the number of principal components used (N_s) and block size (N_b).

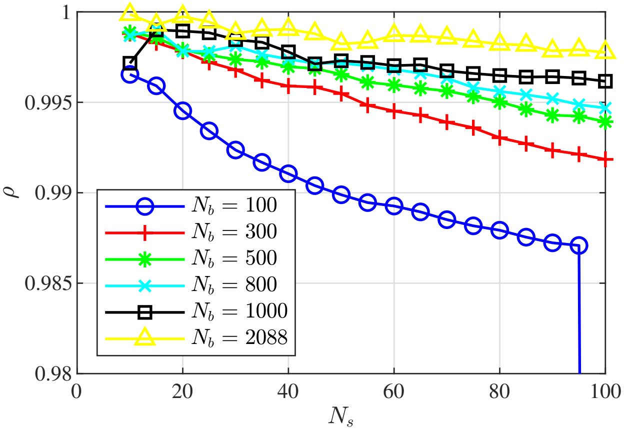

Fig. 5. Correlation with PCA processing in function of the number of principal components used (N_s) and block size (N_b). Only values between 0.98 and 1 are shown to better observe the cross-correlation behaviour.

To better analyse the results, Fig. 5 only shows the correlation values between 0.98 and 1. The best performance was obtained by employing the PCA in the whole data at once, i.e., $N_b = 2088$. In this case, by keeping the $N_s = 100$ higher eigenvalues, the correlation obtained was $\rho = 0.9970$, while for $N_s = 10$, the correlation obtained was $\rho = 0.999$. Reducing N_b decreased the correlation. The lower correlation obtained was $\rho = 0.987$, for $N_b = 100$ and $N_s = 90$.

The KDR obtained can be seen in Fig. 6. It is possible to see that $N_b = 100$ and $N_s = 100$ produces the KDR value equal to the one obtained by the raw data since this case does not make any changes to the signal. This is consistent with the correlation behaviour observed in Fig. 4.

Once more, to better observe the effects of N_b and N_s on the KDR, only the KDR values between 0 and 0.04 are shown in Fig. 7. As it can be seen, except for the $N_b = 100$ and $N_s = 100$ case, all other KDR values obtained were below 0.035. This allows the design of even more efficient key reconciliation techniques than the ones obtained by employing the DCT.

It is important to highlight that the variation on both correlation and the KDR values, except for the case where $N_b = 100$ and $N_s = 100$, is very small. Thus, although it might

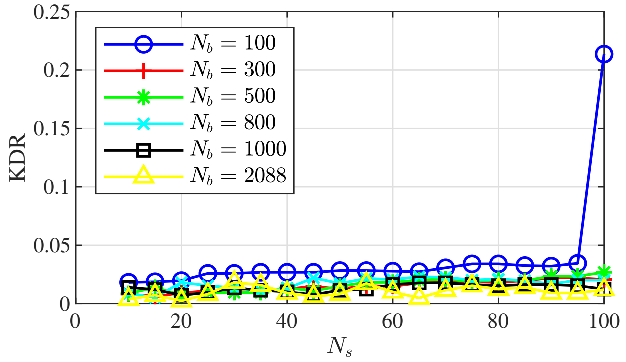

Fig. 6. KDR with PCA processing in function of the number of principal components used (N_s) and block size (N_b).

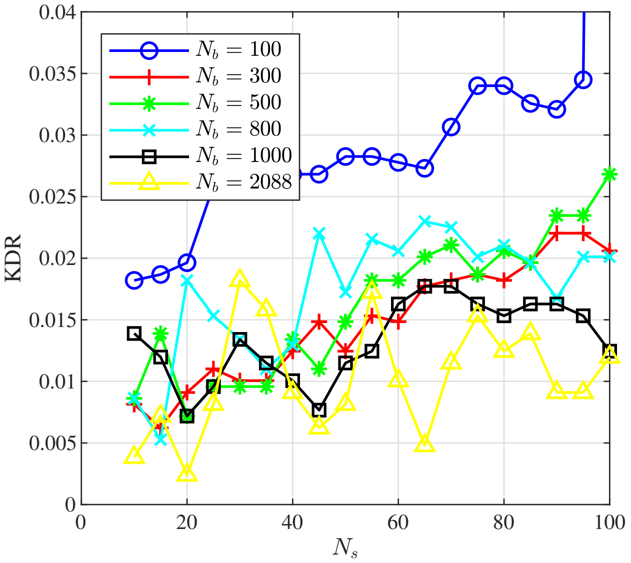

Fig. 7. KDR with PCA processing in function of the number of principal components used (N_s) and block size (N_b). Only values between 0 and 0.4 are shown to better observe the KDR behaviour.

bring small penalties in the correlation and KDR, processing the data blockwise in PCA can benefit the devices in terms of memory, for instance, since it can discard the data after generating the bits from that part. For the DCT, on the other hand, the penalties of the blockwise processing are higher, which impacts the design of the key reconciliation technique.

V. Conclusions

This work investigates the encryption key generation using RSSI in LoRaWAN devices. It was proposed the use of DCT and PCA preprocessing to improve the cross-correlation between measurements in both the end node and gateway. Processing the RSSI measurements using PCA is a novelty since it was not investigated in the literature.

It is shown that, although the DCT can increase the correlation and reduces the KDR, the PCA is more efficient. Nonetheless, the PCA allows the blockwise processing of the data, obtaining better performance than the DCT. The advantage of blockwise processing is that it reduces the use of memory in the devices since they do not need to collect all the necessary RSSIs values to start processing.

Although the PCA has shown a good performance in terms of KDR and cross-correlation, the performance of the key in terms of randomness is still to be evaluated in future works. It is also intended to collect more RSSI values to improve the KDR analysis, not only in mobile and outdoor environments but also in static and indoor environments.

Also, the DCT results show that some low frequency components might be causing some level of non-reciprocity. This is also intended to be investigated in future works.

The feasability of these processing techniques operating in real devices is also a good theme for a next work.

Acknowledgments

The authors would like to acknowledge Dimitri Leandro de Oliveria Silva and Jeferson Rodrigues Cotrim for helping setting up the LoRaWAN devices and measuring the RSSI values.

References

[1] J. M. Hamamreh, H. M. Furqan, and H. Arslan, “Classifications and applications of physical layer security techniques for confidentiality: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1773–1828, 2019.

[2] *LoRaWAN Specification v1.1*, LoRa Alliance Std., Accessed in April 2019. [Online]. Available: https://lora-alliance.org/resource-hub/lorawanr-specification-v11

[3] *Advanced Encryption Standard*, Federal Information Processing Standards Publication 197 Std., 2001.

[4] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, Feb. 1978.

[5] J. Zhang, R. Woods, T. Q. Duong, A. Marshall, and Yuan Ding, “Experimental study on channel reciprocity in wireless key generation,” in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, vol. 4, no. 99. IEEE, Jul. 2016, pp. 1–5.

[6] J. Zhang, B. He, T. Q. Duong, and R. Woods, “On the key generation from correlated wireless channels,” *IEEE Communications Letters*, vol. 21, no. 4, pp. 961–964, Apr. 2017.

[7] J. Zhang, A. Marshall, R. Woods, and T. Q. Duong, “Efficient key generation by exploiting randomness from channel responses of individual OFDM subcarriers,” *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2578–2588, 2016.

[8] G. Margelis, X. Fafoutis, G. Oikonomou, R. Piechocki, T. Tryfonas, and P. Thomas, “Physical layer secret-key generation with discreet cosine transform for the Internet of Things,” in *2017 IEEE International Conference on Communications (ICC)*. IEEE, May 2017, pp. 1–6.

[9] F. Zhan and N. Yao, “On the using of discrete wavelet transform for physical layer key generation,” *Ad Hoc Networks*, vol. 64, pp. 22–31, Sep. 2017.

[10] G. Li, A. Hu, J. Zhang, L. Peng, C. Sun, and D. Cao, “High-agreement uncorrelated secret key generation based on principal component analysis preprocessing,” *IEEE Transactions on Communications*, vol. 66, no. 7, pp. 3022–3034, Jul. 2018.

[11] J. Zhang, A. Marshall, and L. Hanzo, “Channel-envelope differencing eliminates secret key correlation: Lora-based key generation in low power wide area networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 462–12 466, Dec. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8519327/

[12] H. Ruotsalainen, J. Zhang, and S. Grebeniuk, “Experimental investigation on wireless key generation for low power wide area networks,” *IEEE Internet of Things Journal*, vol. PP, no. c, pp. 1–1, 2019.

[13] W. Xu, S. Jha, and W. Hu, “Lora-key: Secure key generation system for lora-based network,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6404–6416, 2019.

[14] C. Huth, R. Guillaume, T. Strohm, P. Duplys, I. A. Samuel, and T. Güneysu, “Information reconciliation schemes in physical-layer security: A survey,” *Computer Networks*, vol. 109, pp. 84–104, Nov. 2016.