DAFTAR PUSTAKA

Afisiadis, O., Cotting, M., Burg, A., & Balatsoukas-Stimming, A. (2020). On the Error Rate of the LoRa Modulation with Interference. *IEEE Transactions on Wireless Communications*, *19*(2), 1292–1304.

Al Homssi, B., Dakic, K., Maselli, S., Wolf, H., Kandeepan, S., & Al-Hourani, A. (2021). IoT Network Design Using Open-Source LoRa Coverage Emulator. *IEEE Access*, *9*, 53636–53646.

Alliance, L., & Ramon, S. (2017). *LoRaWAN 1.1 Regional Parameters*.

Almeida, N. C., Rolle, R. P., Godoy, E. P., Ferrari, P., & Sisinni, E. (2020). Proposal of a Hybrid LoRa Mesh / LoRaWAN Network. *2020 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2020 - Proceedings*, 702–707.

Astudillo León, J. P., & de la Cruz Llopis, L. J. (2019). Emergency aware congestion control for smart grid neighborhood area networks. *Ad Hoc Networks*, *93*, 101898.

Barrera, A., Cheng, C. W., & Kumar, S. (2019). Improved Mix Column Computation of Cryptographic AES. *Proceedings - 2019 2nd International Conference on Data Intelligence and Security, ICDIS 2019*, 229–232.

Baruffa, G., Rugini, L., Germani, L., & Frescura, F. (2020). Error probability performance of chirp modulation in uncoded and coded LoRa systems. *Digital Signal Processing: A Review Journal*, *106*, 102828.

Capuzzo, M., Magrin, D., & Zanella, A. (2018). Mathematical Modeling of LoRa WAN Performance with Bi-directional Traffic. *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*, 206–212.

Chang, K. H., & Mason, B. (2012). The IEEE 802.15.4g standard for smart metering utility networks. *2012 IEEE 3rd International Conference on Smart Grid Communications, SmartGridComm 2012*, 476–480.

Chen, K. C., Yeh, P. C., Hsieh, H. Y., & Chang, S. C. (2010). Communication infrastructure of smart grid. *Final Program and Abstract Book - 4th International Symposium on Communications, Control, and Signal Processing, ISCCSP 2010, March*, 3–5.

Chou, J. S., & Gusti Ayu Novi Yutami. (2014). Smart meter adoption and deployment strategy for residential buildings in Indonesia. *Applied Energy*, *128*(2014), 336–349.

Daemen, J., & Rijmen, V. (2002). The Design of Rijndael. In *New York* (Vol. 1, Issue 69). © Springer-Verlag Berlin Heidelberg 2002. http://portal.acm.org/citation.cfm?id=560131

De Carvalho Silva, J., Rodrigues, J. J. P. C., Alberti, A. M., Solic, P., & Aquino, A. L. L. (2017). LoRaWAN - A low power WAN protocol for Internet of Things: A review and opportunities. *2017 2nd International Multidisciplinary Conference on Computer and Energy Science, SpliTech 2017*, *August*.

De Moraes, P., & Da Conceicao, A. F. (2021). Protecting LoRaWan data against untrusted network servers. *Proceedings - IEEE Congress on Cybermatics: 2021 IEEE International Conferences on Internet of Things, IThings 2021, IEEE Green Computing and Communications, GreenCom 2021, IEEE Cyber, Physical and Social Computing, CPSCom 2021 and IEEE Smart Data, SmartD*, 99–106.

Elshabrawy, T., & Robert, J. (2018). Closed-Form Approximation of LoRa Modulation BER Performance. *IEEE Communications Letters*, *22*(9), 1778–1781.

Figueiredo, L. M., & Franco Silva, E. (2020). Cognitive-LoRa: Adaptation-aware of the physical layer in LoRa-based networks. *Proceedings - IEEE Symposium on Computers and Communications*, *2020-July*(March).

Frank Jay, Christopher J. Booth, R. M. (1990). IEEE Standard Computer Dictionary A Compilation of IEEE Standard Computer Glossaries 610. In *The Institute of Electrical and Electronics Engineers 345 East 47th Street, New York, NY 10017, USA* (Vol. 246, Issue 3).

Guy-Cedric, T. B. I., & R., S. (2018). A Comparative Study on AES 128 BIT AND AES 256 BIT. *International Journal of Scientific Research in Computer Science and Engineering*, *6*(4), 30–33.

Hwang, L. C., Chen, C. S., Ku, T. T., & Shyu, W. C. (2019). A bridge between the smart grid and the Internet of Things: Theoretical and practical roles of LoRa. *International Journal of Electrical Power and Energy Systems*, *113*(May), 971–981.

Ifanda, Toha Zaki, N. A. A. (2014). *Kajian Perkembangan Sistem Metering Kelistrikan di Indonesia Menuju Advanced Metering Infrastructure (AMI), 2014*. Pusat Teknologi Konversi dan Konservasi Energi - BPPT.

Ismail Aouichak, Anouar Achouri, Jean-Charles Le Bunetel, & Yves Raingeaud. (2016). Narrowband PLC Channel Workbench to Emulate Transmission in Smart Grid. *J. of Control Science and Engineering*, *4*(2), 51–60.

Jebroni, Z., Chadli, H., Salmi, K., Saber, M., & Tidhaf, B. (2020). Modeling and simulation of lorawan for smart metering network. In *Lecture Notes in Electrical Engineering: Vol. 684 LNEE* (pp. 687–695).

Jha, A. V., Appasani, B., Ghazali, A. N., Pattanayak, P., Gurjar, D. S., Kabalci, E., & Mohanta, D. K. (2021). Smart grid cyber-physical systems: communication technologies, standards and challenges. *Wireless Networks*, *27*(4), 2595–2613.

Kawle, P., Hiwase, A., Bagde, G., Tekam, E., & Kalbande, R. (2014). Modified

Advanced Encryption Standard. *International Journal of Soft Computing and Engineering (IJSCE)*, *4*(1), 21–23.

Liao, C. H., Zhu, G., Kuwabara, D., Suzuki, M., & Morikawa, H. (2017). Multi-Hop LoRa Networks Enabled by Concurrent Transmission. *IEEE Access*, *5*, 21430–21446.

Luo, P., Sansavini, G., & Zio, E. (2015). Reliability analysis of smart grid communication network by simulation. *Integrated Computer-Aided Engineering*, *22*(2), 119–133.

Ma, R., Chen, H. H., Huang, Y. R., & Meng, W. (2013). Smart grid communication: Its challenges and opportunities. *IEEE Transactions on Smart Grid*, *4*(1), 36–46.

Madaan, A., Bansal, S., Sahu, A., & Kidwai, F. (2020). Peer to Peer Communication in GUI interface using Lora Technology. *Procedia Computer Science*, *173*(2019), 299–304.

Maheswari M., & Gunasekharan S. (2019). *Pengembangan Smart Grid di Indonesia*.

Mai, D. L., & Kim, M. K. (2020). Multi-hop LOra network protocol with minimized latency. *Energies*, *16*(3).

Mishra, R., Pandey, A., & Savariya, J. (2020). Application of internet of things: Last meter smart grid and smart energy efficient system. *2020 1st International Conference on Power, Control and Computing Technologies, ICPC2T 2020*, 32–37.

Muhammad Abdullah, A., & Muhamad Abdullah, A. (2017). *Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data* (Issue June) [Eastern Mediterranean University]. https://www.researchgate.net/publication/317615794

Nafi, N. S., Ahmed, K., & Gregory, M. A. (2016). Modelling software-defined wireless sensor network architectures for smart grid neighborhood area networks. *Security Solutions and Applied Cryptography in Smart Grid Communications*, 267–286.

Nafi, N. S., Ahmed, K., Gregory, M. A., & Datta, M. (2016). A survey of smart grid architectures, applications, benefits and standardization. *Journal of Network and Computer Applications*, *76*, 23–36.

National Institute of Standards and Technology. (2001). 197: Announcing the advanced encryption standard (AES). In *… Technology Laboratory, National Institute of Standards …* (Vol. 2009).

Networks, N. (2012). *ITU-T Y.2060*.

Noreen, U., Ahcenebounceuruniv-brestfr, E., & Clavier, L. (2017). A Study of LoRa Low Power and Wide Area Network Technology Umber. *3rd*

International Conference on Advanced Technologies for Signal and Image Processing.

Nur-A-Alam, Ahsan, M., Based, M. A., Haider, J., & Rodrigues, E. M. G. (2021). Smart monitoring and controlling of appliances using lora based iot system. *Designs*, *5*(1).

Ogbodo, E. U., Dorrell, D., & Abu-Mahfouz, A. M. (2017). Cognitive Radio Based Sensor Network in Smart Grid: Architectures, Applications and Communication Technologies. *IEEE Access*, *5*, 19084–19098.

Osman, N. I., & Abbas, E. B. (2018). Simulation and Modelling of LoRa and Sigfox Low Power Wide Area Network Technologies. *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering, ICCCEEE 2018*, 1–5.

Pau, M., Patti, E., Barbierato, L., Estebsari, A., Pons, E., Ponci, F., & Monti, A. (2018). A cloud-based smart metering infrastructure for distribution grid services and automation. *Sustainable Energy, Grids and Networks*, *15*(2018), 14–25.

Petäjäjärvi, J., Mikhaylov, K., Roivainen, A., Hänninen, T., & Pettissalo, M. (2016). On the coverage of LPWANs: Range evaluation and channel attenuation model for LoRa technology. *2015 14th International Conference on ITS Telecommunications, ITST 2015*, 55–59.

Pramudhita, A. N., Asmara, R. A., Siradjuddin, I., & Rohadi, E. (2018). Internet of Things Integration in Smart Grid. *Proceedings - 2018 International Conference on Applied Science and Technology, ICAST 2018*, 718–722.

Risk assessment tools and techniques according to ISO 31010. (n.d.).

Rothke, B. (2007). Federal Information Processing Standards Publication 197 Announcing the Advanced Encryption Standard (AES). In *Information Security Management Handbook, Sixth Edition*.

Samuddin Harsanto. (2020). *Statistik Listrik 2014-2019* (S. S. P. dan Energi (Ed.); katalog 62). Badan Pusat Statistik/Statistics Indonesia.

Sarinnapakorn, K. (2001). *IEEE 802.11b "High Rate" Wireless Local Area Networks*. *June*.

Savolainen, P. T., Kyntäjä, T., Vallant, H., Marksteiner, S., Aertgeerts, A., Van HaleWeyck, L., & Valckenaers, P. (2015). *Structured Overview of Communication Standards for Smart Grids*. 1–72.

Scala, M. La, & Bruno, S. (2017). From Smart Grids to Smart Cities: New Paradigms for Future Networks. In *From Smart Grids to Smart Cities: New Challenges in Optimizing Energy Grids*.

Schroder Filho, H. G., Pissolato Filho, J., & Moreli, V. L. (2016). The adequacy of LoRaWAN on smart grids: A comparison with RF mesh technology. *IEEE*

2nd International Smart Cities Conference: Improving the Citizens Quality of Life, ISC2 2016 - Proceedings.

Semtech. (2016). LoRa SX1276. In *Semtech Corporation* (Issue SX1276/77/78/79, pp. 1–132).

Seneviratne, P. (2019). Begining LoRa Radio Networks with Arduino. In *Beginning LoRa Radio Networks with Arduino*.

Shyam Kishore, G., & Rallapalli, H. (2019). Performance assessment of M-ary ASK, FSK, PSK, QAM and FQAM in AWGN channel. *Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019*, 273–277.

Singh, A., Agarwal, P., & Chand, M. (2017). Analysis of Development of Dynamic S-Box Generation. *Computer Science and Information Technology*, *5*(5), 154–163.

Sm, I. (1999). Recommendation ITU-R SM.328-10 Spectra and Bandwidth of Emissions. In *Assembly*.

Studies, I., & Panagi, A. (2021). *Exploring Communication Features and Security Vulnerabilities of Long-Range (LoRa) Networks* (Issue May). Cyprus University of Technology.

Sun, H., Hatziargyriou, N., Poor, H. V., Carpanini, L., & Fornié, M. A. S. (2016). Smarter Energy: From smart metering to the smart grid. In *Smarter Energy: From Smart Metering to the Smart Grid*.

To, T. H., & Duda, A. (2018). Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA. *IEEE International Conference on Communications*, *2018-May*.

Triwidyastuti, Y. (2019). Performance Analysis of Point-to-Point LoRa End Device Communication. In *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi* (p. 140). LONTAR KOMPUTER.

Vangelista, L. (2017). Frequency Shift Chirp Modulation: The LoRa Modulation. *IEEE Signal Processing Letters*, *24*(12), 1818–1821.

Verma, H., Jain, M., Goel, K., Vikram, A., & Verma, G. (2016). Smart home system based on Internet of Things. *Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016*, 2073–2075.

Wang, Y., Chen, Q., Hong, T., & Kang, C. (2019). Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. *IEEE Transactions on Smart Grid*, *10*(3), 3125–3148.

Yi, Y., Zhao, H., & Wang, Y. (2020). LoRa Signal Monitoring System of Multi-Node Software Define Radio. *2020 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2020 - Proceedings*.

Zhang, J., Hasandka, A., Alam, S. M. S., Elgindy, T., Florita, A. R., & Hodge, B. M. (2019). Analysis of Hybrid Smart Grid Communication Network Designs for Distributed Energy Resources Coordination. *2019 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2019*.

LAMPIRAN

Lampiran 1. Listing program sistem komunikasi LoRa

```
clear all
clc
SF = 7 ;
BW = 125e3 ;
fc = 915e6 ;
Power = 14 ;
message = "MODELING AND SIMULATION OF LONG RANGE (LORA) COMMUNICATION SYSTEM ON SMART GRID"
key =
'0121324354657687 98a90a1b2c3d0e0f10234213243546576879 8a0b1c2d3e4f'
%% Encryption
ciphertext =
Crypt(message,key)=========================================(1)
%% Sampling=========================================(2)
Fs = 10e6 ;
Fc = 921.5e6 ;
%%=Transmit
Signal=========================================(2)
signalIQ = LoRa_Tx(ciphertext,BW,SF,Power,Fs,Fc - fc) ;
Sxx = 10*log10(rms(signalIQ).^2) ;
disp(['Transmit Power   = ' num2str(Sxx) ' dBm'])
```

```
%% Plots
figure(1)
spectrogram(signalIQ,500,0,500,Fs,'yaxis','centered')
view(-45,65)
figure(2)
obw(signalIQ,Fs) ;
%%                                                    Received
Signal==========================================(3)
message_encrypt = LoRa_Rx(signalIQ,BW,SF,2,Fs,Fc - fc)

%% Decryption=======================================(4)
message_out = Decrypt(key,message_encrypt)
%% Message Out
disp(['Message Received = ' char(message_out)])

====================(1)============================
function [encryption] = Crypt(message,key)
% Encoding AES-256 menggunakan mode Cipher Block Chaining (CBC) dengan
ciphertext
%stealing. Panjang pesan harus lebih dari 32 karakter.
%256-bit hexadecimal string (Panjang 64 karakter). %Output terenkripsi dalam
bentuk heksadesimal.
m=double(unicode2native(message,'UTF-8'));
```

```
count=ceil(length(m)/16);
b=length(m)-16*(count-1);
intiationVector = [173,76,37,247,147,53,68,24,174,242,40,171,201,231,131,99];%
hanya memilih vektor acak
md=bitxor(m(1:16),intiationVector);
w=KeyExpansion(key);
E(1,:)=Cipher(w,md);
for k=2:count-1
   md=bitxor(m(16*(k-1)+1:16*k),E(k-1,:));
   E(k,:)=Cipher(w,md);
end
ee=E(count-1,:);
mm=zeros(1,16);
mm(1:b)=m(16*(count-1)+1:end);
md=bitxor(mm,ee);
E(count-1,:)=Cipher(w,md);
E(count,:)=ee;
E=E';
encryption=lower(dec2hex(E(1:length(m)))');
encryption=encryption(:)';
end
```

==========(2)==========

```
function [signal_mod] = LoRa_Tx(message,Bandwidth,SF,Pt,Fs,df,varargin)
% LoRa_Tx mengemulasi transmisi Lora
%   in:  message      payload message
%        Bandwidth    signal bandwidth of LoRa transmisson
%        SF           spreading factor
%        Pt           transmit power in decibels
%        Fs           sampling frequency
%        dF           frequency offset
%        varargin{1}  code rate
%        varargin{2}  symbols in preamble
%        varargin{3}  sync key
%  out:  signal       Bentuk gelombang IQ LoRa
%        packet       pesan yang dikodekan
if nargin == 6
   CR = 1 ;
   n_preamble = 8 ;
   SyncKey = 5 ;
elseif nargin == 7
   CR = varargin{1} ;
   n_preamble = 8 ;
   SyncKey = 5 ;
elseif nargin == 8
   CR = varargin{1} ;
```

```
    n_preamble = varargin{2} ;
    SyncKey = 5 ;
elseif nargin == 9
    CR = varargin{1} ;
    n_preamble = varargin{2} ;
    SyncKey = varargin{3} ;
end
packet = LoRa_Encode_Full(message,SF,CR) ; % pesan yang dikodekan
signal = LoRa_Modulate_Full(packet,SF,Bandwidth,n_preamble,SyncKey,Fs) ;
% LoRa modulate message
signal_mod = 10.^(Pt./20).*signal.*exp(-j.*2.*pi.*df/Fs.*(0:length(signal)-1))' ;
% frequency shift and convert to power
end
function [packet] = LoRa_Encode_Full(message,SF,CR)
% LoRa_Encode_Full emulates a Lora transmission
%
%   in:  message      payload message
%        SF           spreading factor
%        CR           coding rate
%
%  out:  packet       encoded lora packet

CRC_pld = 1 ;  % cyclic rate code flag
```

```
imp = 0 ;
opt = 0 ;
%% String to Decimal
message_chr = convertStringsToChars(message) ;
message_dbl = uint8(message_chr) ;
%% Packet Length Calculations
N_pld = (SF == 7).*1 + (SF == 8).*2 + (SF == 9).*3 + (SF == 10).*4 + (SF ==
11).*5 + (SF == 12).*6 ;
n_packet = 8 + max([ceil((8*(length(message_dbl) + 5) - 4.*SF + 28 +
16.*CRC_pld - 20.*imp)/(4.*(SF - 2.*opt))).*(CR + 4) 0]) ;
n_wht = SF .* floor((n_packet-8)/(4 + CR)) + N_pld - 1 ;
n_pld = ceil((n_wht + (SF == 7).*0 + (SF == 8).*1 + (SF == 9).*2 + (SF == 10).*3
+ (SF == 11).*4 + (SF == 12).*5)/2) ;
n_pad = n_pld - 5 - length(message_dbl) - CRC_pld.*2 ;
%% Create payload message
CRC_dbl = CRC_pld.*[1 1] ; % CRC is not working atm
pad_dbl = zeros(1,n_pad + N_pld - 1) ; % padding
pld_dbl = [255 255 0 0 message_dbl 0 CRC_dbl pad_dbl] ; % LoRa payload
%% Swap Nibbles
pld_swp = LoRa_encode_swap(pld_dbl) ;
%% Payload Encoding
pld_enc = LoRa_encode_hamming(pld_swp,CR) ;
pld_enc = pld_enc(1 : n_wht) ;
```

```
%% Payload Whiten
pld_wht = LoRa_encode_white(pld_enc,CR,0) ;
%% Header Encoding
packet_hdr = [(length(message_dbl)+5)
CRC_pld*16+(CR==1)*32+(CR==2).*64+(CR==3).*96+(CR==4)*128 224] ;
packet_hdr_enc_tmp = LoRa_encode_hamming(packet_hdr,4) ;
packet_hdr_enc = [packet_hdr_enc_tmp(1:5) pld_wht(1:N_pld-1)] ;
%% Packet Creation
packet_pld = pld_wht(N_pld : end) ;
packet_pld_shf = bitand(LoRa_encode_shuffle(packet_pld),2^(4+CR)-1)  ;
packet_hdr_shf = LoRa_encode_shuffle(packet_hdr_enc)  ;
% Interleaving
packet_pld_int = LoRa_encode_interleave(packet_pld_shf,SF,CR) ;
packet_hdr_int = LoRa_encode_interleave(packet_hdr_shf,SF-2,4) ;
% Graying
packet_pld_gray = LoRa_encode_gray(packet_pld_int) ;
packet_hdr_gray = LoRa_encode_gray(packet_hdr_int) ;
% Packet final
packet = [4*packet_hdr_gray packet_pld_gray] ;
end
function [symbols_swp] = LoRa_encode_swap(symbols)
% LoRa_encode_swap swaps nibbles
%
```

```
%   in:  symbols            symbol sequence
%
%  out:  symbols_swp        symbols with swapped nibbles

symbols_swp = zeros(1,length(symbols)) ;
for ctr = 1 : length(symbols)
   symbols_swp(ctr)                                                    =
bitor(bitsll(bitand(symbols(ctr),hex2dec('0F')),4),bitsra(bitand(symbols(ctr),hex2d
ec('F0')),4)) ; % swap first half of 8-bit sequencne with other half
end
end
function [encoded] = LoRa_encode_hamming(symbols,CR)
% LoRa_encode_hamming hamming encodes symbols to ensure a more accurate
decoding
%
%   in:  symbols      symmbol sequence
%        CR           hamming coding rate
%
%  out:  encoded      hamming encoded symbols

if CR > 2 && CR <= 4 % detection and correction
   n = floor(length(symbols).*(4 + 4)/4) ;
```

```
    H = [0,210,85,135,153,75,204,30,225,51,180,102,120,170,45,255] ;

    encoded = zeros(1,n) ;
    Ctr = 1 ;
    for ctr = 1 : length(symbols)
        s0 = bitand(floor(bitsra(symbols(ctr),4)),hex2dec('0F')) ;
        s1 = bitand(floor(bitsra(symbols(ctr),0)),hex2dec('0F')) ;
        encoded(Ctr+0) = H(s0+1) ;
        encoded(Ctr+1) = H(s1+1) ;
        Ctr = Ctr + 2 ;
    end
elseif CR > 0 && CR <= 2 % detection
    Ctr = 1 ;
    for ctr = 1 : length(symbols)
        s0 = bitand(floor(bitsra(symbols(ctr),4)),hex2dec('FF')) ;
        s1 = bitand(floor(bitsra(symbols(ctr),0)),hex2dec('FF')) ;
        encoded(Ctr+0) = selectbits_encode(s0) ;
        encoded(Ctr+1) = selectbits_encode(s1) ;
        Ctr = Ctr + 2 ;
    end
end
end
function [symbols_white] = LoRa_encode_white(symbols,CR,DE)
```

```
% LoRa_encode_white symbols whitening by adding a known sequence to the
payload
% bytes to reduce correlation redudancy
%
%   in:  symbols      symmbol sequence
%        CR           coding rate
%        DE           data rate optimization flag
%
%  out:  symbols_white      whitened symbols

if DE == 0
   if CR > 2 && CR <= 4
      white_sequence = [255,255,45,255,120,255,225,255,0,255,210,45,85, ...
          120,75,225,102,0,30,210,255,85,45,75,120,102,225,30,210,255, ...
          135,45,204,120,170,225,180,210,153,135,225,204,0,170,0,180,0, ...
          153,0,225,210,0,85,0,153,0,225,0,210,210,135,85,30,153,45,225, ...
          120,210,225,135,210,30,85,45,153,120,51,225,85,210,75,85,102, ...
          153,30,51,45,85,120,75,225,102,0,30,0,45,0,120,210,225,135,0, ...
          204,0,120,0,51,210,85,135,153,204,51,120,85,51,153,85,51,153, ...
          135,51,204,85,170,153,102,51,30,135,45,204,120,170,51,102,85, ...
          30,153,45,225,120,0,51,0,85,210,153,85,225,75,0,180,0,75,210, ...
          102,85,204,75,170,180,102,75,204,102,170,204,180,170,75,102, ...
          102,204,204,170,120,180,51,75,85,102,75,204,102,120,204,51, ...
```

```
        120,85,225,75,0,102,210,204,135,120,30,225,255,0,255,210,45, ...
        135,170,30,102,255,204,255,170,45,102,170,30,102,255,204,45, ...
        170,170,102,180,30,75,255,102,45,30,170,45,180,170,75,180,102, ...
        153,30,225,45,210,170,85,180,153,153,225,225,0,210,210,85,135, ...
        153,204,225,170,0,102,210,204,135,120,204,225,170,210,102,135, ...
        204,30,120,255,225,45,210,120,135,51,30,135,255,30,45,45,120, ...
        120,51,51,135,135,30,204,45,120,120,225,51,210,135,85,204,75, ...
        120,102,225,204,210,170,85,180,75,153,102,51,204,85,170,153, ...
        180,225,153,210,51,85,85,75,153,180,225,153,210,51,85,85,75, ...
        75,180,180,153,75,51,180,85,153,75,51,180,135,75,30,180,45, ...
        153,170,51,102,199,30,30,45,45,170,170,102,102,204,30,120, ...
        45,51,170,135,102,30,204,255,120,45,51,170,135,102,30,30,255, ...
        255,45,255,170,255,102,45,30,170,255,180,255,153,255,51,45,135, ...
        170,204,180,120,153,51,51,135,135,204,204,170,120,180,51,75,135, ...
        180,204,153,170,225,180,210,75,135,180,204,153,120,225,225,210,0, ...
        135,0,204,210,120,135,225,30,0,45,0,170,210,180,135,75,30,180, ...
        45,75,170,180,180,75,75,102,180,30,75,255,180,255,75,45,102,120, ...
        30,51,255,85,255,75,45,180,120,153,51,225,85,0,75,210,180,85,153, ...
        153,225,51,0,135,210,30,85,255,153,255,51,255,135,255,30,0,0,0,0, ...
        135,225,170,204] ;
elseif CR > 0 && CR <= 2
    white_sequence = [255,255,45,255,120,255,48,46,0,46,18,60,20,40,10, ...
        48,54,0,30,18,46,20,60,10,40,54,48,30,18,46,6,60,12,40,58,48,36, ...
```

```
        18,24,6,48,12,0,58,0,36,0,24,0,48,18,0,20,0,24,0,48,0,18,18,6,20, ...
        30,24,60,48,40,18,48,6,18,30,20,60,24,40,34,48,20,18,10,20,54,24, ...
        30,34,60,20,40,10,48,54,0,30,0,60,0,40,18,48,6,0,12,0,40,0,34,18, ...
        20,6,24,12,34,40,20,34,24,20,34,24,6,34,12,20,58,24,54,34,30,6,60, ...
        12,40,58,34,54,20,30,24,60,48,40,0,34,0,20,18,24,20,48,10,0,36,0, ...
        10,18,54,20,12,10,58,36,54,10,12,54,58,12,36,58,10,54,54,12,12,58, ...
        40,36,34,10,20,54,10,12,54,40,12,34,40,20,48,10,0,54,18,12,6,40,30, ...
        48,46,0,46,18,60,6,58,30,54,46,12,46,58,60,54,58,30,54,46,12,60,58, ...
        58,54,36,30,10,46,54,60,30,58,60,36,58,10,36,54,24,30,48,60,18,58, ...
        20,36,24,24,48,48,0,18,18,20,6,24,12,48,58,0,54,18,12,6,40,12,48, ...
        58,18,54,6,12,30,40,46,48,60,18,40,6,34,30,6,46,30,60,60,40,40,34, ...
        34,6,6,30,12,60,40,40,48,34,18,6,20,12,10,40,54,48,12,18,58,20,36, ...
        10,24,54,34,12,20,58,24,36,48,24,18,34,20,20,10,24,36,48,24,18,34, ...
        20,20,10,10,36,36,24,10,34,36,20,24,10,34,36,6,10,30,36,60,24,58, ...
        34,54,6,30,30,60,60,58,58,54,54,12,30,40,60,34,58,6,54,30,12,46, ...
        40,60,34,58,6,54,30,30,46,46,60,46,58,46,54,60,30,58,46,36,46,24, ...
        46,34,60,6,58,12,36,40,24,34,34,6,6,12,12,58,40,36,34,10,6,36,12, ...
        24,58,48,36,18,10,6,36,12,24,40,48,48,18,0,6,0,12,18,40,6,48,30,0, ...
        60,0,58,18,36,6,10,30,36,60,10,58,36,36,10,10,54,36,30,10,46,36,46, ...
        10,60,54,40,30,34,46,20,46,10,60,36,40,24,34,48,20,0,10,18,36,20,24, ...
        24,48,34,0,6,18,30,20,46,24,46,34,46,6,46,30,0,0,0,0,36,6] ;
    end
end
```

```
N = min([length(symbols) length(white_sequence)]) ; % cut-off to length of transmit symbols
symbols_white = bitxor(symbols(1:N),white_sequence(1:N)); % encode white
end
function [symbols_shuf] = LoRa_encode_shuffle(symbols)
% LoRa_encode_shuffle shuffles symbols by a to combine header and
% payload
%
%   in:  symbols           symbol vector
%
%  out:  symbols_shuf      shuffle symbols

for Ctr = 1 : length(symbols)
   symbols_binary = de2bi(symbols(Ctr),8) ;
   symbols_shuf_binary = [symbols_binary(2) symbols_binary(3) symbols_binary(4) ...
       symbols_binary(6) symbols_binary(5) symbols_binary(1) symbols_binary(7) ...
       symbols_binary(8)] ;
   symbols_shuf(Ctr) = bi2de(symbols_shuf_binary) ;
end
end
function [symbols_interleaved] = LoRa_encode_interleave(symbols,ppm,rdd)
```

```
% LoRa_encode_interleave imposes transposition and digit shift on the
% symbols and rotation
%
%   in:  symbols           symbol sequence
%        ppm               SF
%        rdd               CR
%  out:  symbols_interleaved       interleaved symbols


symbols_interleaved = [] ;
sym_idx_ext = 1 ;
for block_idx = 1 : floor(length(symbols)/(ppm))
   x = symbols((block_idx-1).*ppm+1:block_idx.*ppm) ;
   symbols_block_binary = de2bi(x,4+rdd) ;
   symbols_block_binary_rotated = transpose(symbols_block_binary) ; %
transposed
   symbols_block_rorated = bi2de(symbols_block_binary_rotated) ;
   mask = ppm ;
   % rotate
   for ctr = 1 : 4 + rdd
      sym_int(ctr) = rotl(symbols_block_rorated(ctr),mask,ppm) ;
      mask = mask - 1 ;
   end
   symbols_interleaved = [symbols_interleaved sym_int] ;
```

```
end
end
function [symbols] = LoRa_encode_gray(symbols)
% LoRa_encode_gray implements gray coding to reduce errors of adjacent bits.
%
%   in:  symbols            symbol sequence
%
%  out:  symbols            gray coded symbols

% XOR each symbol with a shifted mask
for ctr = 1 : length(symbols)
    symbols(ctr) = bitxor(symbols(ctr),floor(bitsra(symbols(ctr),16))) ;
    symbols(ctr) = bitxor(symbols(ctr),floor(bitsra(symbols(ctr),08))) ;
    symbols(ctr) = bitxor(symbols(ctr),floor(bitsra(symbols(ctr),04))) ;
    symbols(ctr) = bitxor(symbols(ctr),floor(bitsra(symbols(ctr),02))) ;
    symbols(ctr) = bitxor(symbols(ctr),floor(bitsra(symbols(ctr),01))) ;
end
end
function [symbol_rot] = selectbits_encode(symbol)
% selectbits_encode concat zeros (from 8-bit to 4-bit)
%   in:  symbols            symbol sequence
%  out:  symbols_rot        symbols for rotation
symbol_binary = de2bi(symbol,8) ;
```

```
symbol_binary_rot = [0 symbol_binary(1) symbol_binary(2) symbol_binary(3) 0
symbol_binary(4) 0 0] ;
symbol_rot = bi2de(symbol_binary_rot) ;
end
function [y] = rotl(bits,count,size)
% rotl modulo rotation
%   in:  bits          bit sequence
%  out:  y             symbols
len_mask = bitsll(1,size) - 1 ;
count = mod(count,size) ;
bits = bitand(bits,len_mask) ;
y = bitor(bitand(bitsll(bits,count),len_mask), floor(bitsra(bits,size - count))) ;
end
function                                [signal]                                =
LoRa_Modulate_Full(packet,SF,Bandwidth,n_preamble,SyncKey,Fs)
% LoRa_Modulate_Full constructs a lora packet (preamble + sync header +
payload)
%    in:   packet            payload 1xN symbol vector wher N=1-Inf with values
{0,1,2,...,2^(SF)-1}
%       SF           spreading factor
%       Bandwidth      signal bandwidth of LoRa transmisson
%n_preamble     number of symbols in the preamble
%SyncKey       synchronize key
```

```
%Fs            sampling frequency
%out:  signal          LoRa IQ packet
 signal_prmb = loramod((SyncKey - 1).*ones(1,n_preamble),SF,Bandwidth,Fs,1) ;
% preamble upchirps
 signal_sync_u = loramod([0 0],SF,Bandwidth,Fs,1) ; % sync upchirp

signal_sync_d1 = loramod(0,SF,Bandwidth,Fs,-1) ; % header downchirp
signal_sync_d          =          [signal_sync_d1;          signal_sync_d1;
signal_sync_d1(1:length(signal_sync_d1)/4)] ; % concatenate header

signal_mesg = loramod(mod(packet + SyncKey,2^SF),SF,Bandwidth,Fs,1) ; % add
sync key to payload messaage
signal  =  [signal_prmb;  signal_sync_u;  signal_sync_d;  signal_mesg]  ;  %
concatenate LoRa packet
end
function [y] = loramod(x,SF,BW,fs,varargin)
% loramod LoRa modulates a symbol vector specified by x
%
%   in:  x          1xN symbol vector wher N=1-Inf
%                  with values {0,1,2,...,2^(SF)-1}
%        BW         signal bandwidth of LoRa transmisson
%        SF         spreading factor
%        Fs         sampling frequency
```

```
%       varargin{1} polarity of chirp
%
%  out:  y          LoRa IQ waveform
if (nargin < 4)
   error(message('comm:pskmod:numarg1'));
end

if (nargin > 5)
   error(message('comm:pskmod:numarg2'));
end

% Check that x is a positive integer
if (~isreal(x) || any(any(ceil(x) ~= x)) || ~isnumeric(x))
   error(message('comm:pskmod:xreal1'));
end

M       = 2^SF ;

% Check that M is a positive integer
if (~isreal(M) || ~isscalar(M) || M<=0 || (ceil(M)~=M) || ~isnumeric(M))
   error(message('comm:pskmod:Mreal'));
end
```

```
% Check that x is within range
if ((min(min(x)) < 0) || (max(max(x)) > (M-1)))
   error(message('comm:pskmod:xreal2'));
end

% Polarity of Chirp
if nargin == 4
   Inv = 1 ;
elseif nargin == 5
   Inv = varargin{1} ;
end
% Symbol Constants
Ts      = 2^SF/BW ;
Ns      = fs.*M/BW ;

gamma   = x/Ts ;
beta    = BW/Ts ;

time    = (0:Ns-1)'.*1/fs ;
freq    = mod(gamma + Inv.*beta.*time,BW) - BW/2 ;

Theta   = cumtrapz(time,freq) ;
y       = reshape(exp(j.*2.*pi.*Theta),numel(Theta),1) ;
```

```
end
```

Cipher coding

```
function Out = Cipher(w,In)
%AES-256 cipher
%Sesuai standar FIBS-197, key is 256-bit hexidecimal input,
% message (In) is 128-bit hexidecimal.
state=reshape(In,4,[]);
state=AddRoundKey(state,w(:,1:4));
for k=2:14
   state=SubBytes(state);
   state=ShiftRows(state);
   state=MixColumns(state);
   state=AddRoundKey(state,w(:,4*(k-1)+1:4*k));
end
state=SubBytes(state);
state=ShiftRows(state);
state=AddRoundKey(state,w(:,57:60));
Out=state(:);
end
```

```
==========(3)==========
function [message] = LoRa_Rx(signal,Bandwidth,SF,Coherece,Fs,df,varargin)
```

```
% LoRa_Rx emulates a Lora receiver
%   in:  signal       payload message
%        Bandwidth    signal bandwidth of LoRa transmisson
%        SF           spreading factor
%        Coherence    (1) coherent or (2) non-coherent FSK Detection
%        Fs           sampling frequency
%        dF           carrier frequency shift
%        varagin{1}   SNR
%        varagin{2}   Preamble Symbol number
%  out:  message      LoRa payload message chahracters
%        symbols_Demod LoRa payload symbols vector
if nargin == 6 % number argument input
   SNR               = Inf ;
   n_preamble        = 8 ;
elseif nargin == 7
   SNR               = varargin{1} ;
   n_preamble        = 8 ;
elseif nargin == 8
   SNR               = varargin{1} ;
   n_preamble        = varargin{2} ;
end

if Fs == Bandwidth
```

```
    signal_demod        = awgn(signal,SNR,'measured') ;
else
    signal_freq_demod   = signal.*exp(j.*2.*pi.*df./Fs.*(0:length(signal)-1))' ;
    signal_filter       = lowpass(signal_freq_demod,Bandwidth,Fs) ;
    signal_demod                                                    =
awgn(resample(signal_filter,Bandwidth,Fs),SNR,'measured') ;
end

try
    symbols_message                                                 =
LoRa_Demodulate_Full(signal_demod,SF,Bandwidth,Coherece,n_preamble);
    [message_full]      = LoRa_Decode_Full(symbols_message,SF);
    message             = message_full(8:4 + message_full(1) - 2) ;
catch
    message             = NaN ;
end
end
function                    [SymbolsMessage,SymbolsDemod,NPreamb]       =
LoRa_Demodulate_Full(signal,SF,Bandwidth,Coherece,n_preamble)
% LoRa_Demodulate_Full demodulates full LoRa packet
%
%   in:  signal         IQ LoRa signal containing
%                   (preamble + sync header + payload)
```

```
%       SF            spreading factor
%       Bandwidth     signal bandwidth of LoRa transmisson
%       Coherence     type of demodulation (coherent or non-coherent)
%
%  out:  symbols_message         message symbols vector (encoded)
%       symbols_Demod          LoRa symbols vector
%       n_preamble             Number of symbols in preamble
%% Return if SF is not in the range
if SF > 12 || SF < 7
   return
end
M = 2^SF ;
%% Demodulate and Extract Preamble
Nsymbols       = floor(length(signal)/M) ;
UChirpsDemod    = loramod(zeros(1,Nsymbols),SF,Bandwidth,Bandwidth) ;

SniffSignal     = signal(1:length(UChirpsDemod)).*UChirpsDemod ;

if Coherece == 2
   fftSync     = fft(reshape(SniffSignal,M,length(SniffSignal)/M)) ;
   [~,SyncInd] = sort(max(fftSync)) ;
   sync        = sort(SyncInd(end-1:end)) ;
   sync        = sync(end) + 1 ;
```

```
    NPreamb     = sync - 5 ;
else
    NPreamb     = n_preamble ;
end
dChirpsDemod    = loramod(zeros(1,NPreamb),SF,Bandwidth,Bandwidth,-1) ;
pream_signal    = signal(1:length(dChirpsDemod)).*dChirpsDemod ;

symbols_pream   = FSKDetection(pream_signal,SF,Coherece) ;
symbol_offset   = mode(symbols_pream) + 1 ;
%% Demodulate Message
MessageStartInd = (NPreamb + 4.25)*M ;
Nmessage        = floor(length(signal)/M - MessageStartInd/M) ;
MessageEndInd   = Nmessage.*M + MessageStartInd ;

MessageSignal                                                    =
signal(MessageStartInd+1:MessageEndInd).*loramod(zeros(1,Nmessage),SF,Ban
dwidth,Bandwidth,-1) ;
SymbolsDemod    = FSKDetection(MessageSignal,SF,Coherece) ;
SymbolsMessage  = mod(SymbolsDemod - symbol_offset,2^SF) ;
end
function [y] = loramod(x,SF,BW,fs,varargin)
% loramod LoRa modulates a symbol vector specified by x
%   in:  x          1xN symbol vector wher N=1-Inf
```

```
%                 with values {0,1,2,...,2^(SF)-1}
%       BW        signal bandwidth of LoRa transmisson
%       SF        spreading factor
%       Fs        sampling frequency
%       varargin{1} set polarity of chirp
%  out:  y         LoRa IQ waveform
if (nargin < 4)
   error(message('comm:pskmod:numarg1'));
end

if (nargin > 5)
   error(message('comm:pskmod:numarg2'));
end
% Check that x is a positive integer
if (~isreal(x) || any(any(ceil(x) ~= x)) || ~isnumeric(x))
   error(message('comm:pskmod:xreal1'));
end

M      = 2^SF ;

% Check that M is a positive integer
if (~isreal(M) || ~isscalar(M) || M<=0 || (ceil(M)~=M) || ~isnumeric(M))
   error(message('comm:pskmod:Mreal'));
```

```
end

% Check that x is within range
if ((min(min(x)) < 0) || (max(max(x)) > (M-1)))
   error(message('comm:pskmod:xreal2'));
end

% Polarity of Chirp
if nargin == 4
   Inv = 1 ;
elseif nargin == 5
   Inv = varargin{1} ;
end
% Symbol Constants
Ts      = 2^SF/BW ;
Ns      = fs.*M/BW ;

gamma   = x/Ts ;
beta    = BW/Ts ;

time    = (0:Ns-1)'.*1/fs ;
freq    = mod(gamma + beta.*time,BW) - BW/2 ;
```

```
Theta   = cumtrapz(time,Inv.*freq) ;
y       = reshape(exp(j.*2.*pi.*Theta),numel(Theta),1) ;
end
function                    [message_full,CR_pld,pld_length,CRC_pld]                    =
LoRa_Decode_Full(symbols_message,SF)
% LoRa_Decode_Full decodes full payload packet
%   in:  symbols_message        LoRa payload symbol vector
%        SF                     spreading factor
%  out:  message_full           message symbols vector (decoded)
%        CR_pld                 code rate of payload
%        pld_length             length of payload
%        CRC_pld                payload cyclic rate code flag
%% Decode Header
rdd_hdr        = 4 ;
ppm_hdr        = SF - 2 ;
symbols_hdr    = mod(round(symbols_message(1:8)/4),2^ppm_hdr) ;
% Graying
symbols_hdr_gry = LoRa_decode_gray(symbols_hdr) ;
% Interleaving
symbols_hdr_int = LoRa_decode_interleave(symbols_hdr_gry,ppm_hdr,rdd_hdr) ;
% Shuffle
symbols_hdr_shf = LoRa_decode_shuffle(symbols_hdr_int,ppm_hdr) ;
% Hamming
```

```
symbols_hdr_fec = LoRa_decode_hamming(symbols_hdr_shf(1:5),rdd_hdr) ;
%% Extract info from Header
CR_pld          = floor(bitsra(symbols_hdr_fec(2),5)) ;
if CR_pld > 4 || CR_pld < 1
   return
end
CRC_pld         = mod(floor(bitsra(symbols_hdr_fec(2),4)),2) ;
pld_length      = symbols_hdr_fec(1) + CRC_pld*2 ;
%% Decode Payload
rdd_pld         = CR_pld ;
ppm_pld         = SF ;
symbols_pld     = symbols_message(9:end) ;
% Graying
symbols_pld_gry = LoRa_decode_gray(symbols_pld) ;
% Interleaving
symbols_pld_int = LoRa_decode_interleave(symbols_pld_gry,ppm_pld,rdd_pld) ;
% Shuffle
symbols_pld_shf                                                    =
LoRa_decode_shuffle(symbols_pld_int,length(symbols_pld_int)) ;
% Add part of header
symbols_pld_hdr    =    [(SF>7).*symbols_hdr_shf(end    -    SF    +    8:end)
symbols_pld_shf] ;
% White
```

```
symbols_pld_wht = LoRa_decode_white(symbols_pld_hdr,rdd_pld,0) ;
% Hamming
symbols_pld_fec = LoRa_decode_hamming(symbols_pld_wht,rdd_pld) ;
% Swaping
symbols_pld_fin = LoRa_decode_swap(symbols_pld_fec) ;
%% Final Message
message_full    = [symbols_hdr_fec symbols_pld_fin] ;
end
function [symbols_gray] = LoRa_decode_gray(symbols)
% LoRa_decode_gray degray LoRa payload
%   in:  symbols       symbols with graying
%  out:  symbols_gray  degrayed symbols
symbols_gray = bitxor(symbols,floor(bitsra(symbols,1))) ;
end
function [deocded] = LoRa_decode_hamming(symbols,CR)
% LoRa_decode_hamming LoRa payload hamming decode (4,4 + CR)
%   in:  symbols       symbols with hamming
%        CR            Code Rate
%  out:  deocded      Fully decoded payload symbols
if CR > 2 && CR <= 4 % detection and correction
   n = ceil(length(symbols).*4/(4 + 4)) ;

   H = [0,0,0,0,0,0,3,3,0,0,5,5,14,14,7,7,0,0,9,9,2,2,7,7,4,4,7,7,7,7, ...
```

```
    7,7,0,0,9,9,14,14,11,11,14,14,13,13,14,14,14,14,9,9,9,9,10,10,9, ...
    9,12,12,9,9,14,14,7,7,0,0,5,5,2,2,11,11,5,5,5,5,6,6,5,5,2,2,1,1, ...
    2,2,2,2,12,12,5,5,2,2,7,7,8,8,11,11,11,11,11,11,12,12,5,5,14,14, ...
    11,11,12,12,9,9,2,2,11,11,12,12,12,12,12,12,15,15,0,0,3,3,3,3,3, ...
    3,4,4,13,13,6,6,3,3,4,4,1,1,10,10,3,3,4,4,4,4,4,4,7,7,8,8,13,13, ...
    10,10,3,3,13,13,13,13,14,14,13,13,10,10,9,9,10,10,10,10,4,4,13, ...
    13,10,10,15,15,8,8,1,1,6,6,3,3,6,6,5,5,6,6,6,6,1,1,1,1,2,2,1,1, ...
    4,4,1,1,6,6,15,15,8,8,8,8,8,8,11,11,8,8,13,13,6,6,15,15,8,8,1,1, ...
    10,10,15,15,12,12,15,15,15,15,15,15] ;

deocded = zeros(1,n) ;
for ctr = 0 : n - 1
    r0 = bitand(symbols(2*ctr+1),hex2dec("FF")) ;
    if 2*ctr+2 > length(symbols)
        symbols(2*ctr+2) = 0 ;
    end
    r1 = bitand(symbols(2*ctr+2),hex2dec("FF")) ;

    s0 = H(r0+1) ;
    s1 = H(r1+1) ;

    deocded(ctr+1) = bitor(bitsll(s0,4),s1) ;
end
```

```
elseif CR > 0 && CR <= 2 % detection
   indices = [1 2 3 5] ;
   len = length(symbols) ;
   Ctr = 1 ;
   for ctr = 1 : 2 : len
      if ctr + 1 < len
         s1 = bitand(selectbits(symbols(ctr+1),indices),hex2dec("FF")) ;
      else
         s1 = 0 ;
      end
      s0 = bitand(selectbits(symbols(ctr),indices),hex2dec("FF")) ;
      deocded(Ctr) = bitor(bitsll(s0,4),s1) ;
      Ctr = Ctr + 1 ;
   end
end
end
function [symbols_interleaved] = LoRa_decode_interleave(symbols,ppm,rdd)
% LoRa_decode_interleave deinterleaves payload packet
%   in:  symbols       interleaved symbols
%        ppm
%        rdd
%  out:  symbols_interleaved  deinterleaved symbols
symbols_interleaved = [] ;
```

```
sym_idx_ext = 1 ;
for block_idx = 1 : floor(length(symbols)/(4+rdd))
   sym_int = zeros(1,ppm) ;
   for sym_idx = 1 : 4 + rdd
      sym_rot = rotl(symbols(sym_idx_ext),sym_idx-1,ppm) ;
      mask = bitsll(1,ppm-1) ;
      ctr = ppm ;
      while mask > 0
         sym_int(ctr) = sym_int(ctr) +
bitsll(double(bitand(sym_rot,mask)>0),sym_idx-1) ;
         mask = floor(bitsra(mask,1)) ;
         ctr = ctr - 1 ;
      end
      sym_idx_ext = sym_idx_ext + 1 ;
   end
   symbols_interleaved = [symbols_interleaved sym_int] ;
end
end
function [symbols_shuf] = LoRa_decode_shuffle(symbols,N)
% LoRa_decode_shuffle unshuffles payload packet
%   in:  symbols       symbol vector
%        N
%  out:  symbols_shuf  unshuffled symbols
```

```
pattern = [5 0 1 2 4 3 6 7] ;
symbols_shuf = zeros(1,N) ;
for ctr = 1 : N
   for Ctr = 1 : length(pattern)
      symbols_shuf(ctr) = symbols_shuf(ctr) +
bitsll(double(bitand(symbols(ctr),bitsll(1,pattern(Ctr)))>0),Ctr-1) ;
   end
end
end
function [symbols_swp] = LoRa_decode_swap(symbols)
% LoRa_decode_shuffle swap payload packet
%   in:  symbols       symbol vector
%  out:  symbols_swp   unswapped symbols
symbols_swp = zeros(1,length(symbols)) ;
for ctr = 1 : length(symbols)
   symbols_swp(ctr) =
bitor(bitsll(bitand(symbols(ctr),hex2dec('0F')),4),bitsra(bitand(symbols(ctr),hex2d
ec('F0')),4)) ; % swap first half of 8-bit sequencne with other half
end
end
function [symbols_white] = LoRa_decode_white(symbols,CR,DE)
% LoRa_decode_white dewhitening of payload packet
```

```
%   in:  symbols       whitened symbols
%        CR            code rate
%        DE            data rate optimization flag
%  out:  symbols_white  dewhitened symbols

if DE == 0
   if CR > 2 && CR <= 4
      white_sequence = [255,255,45,255,120,255,225,255,0,255,210,45,85, ...
         120,75,225,102,0,30,210,255,85,45,75,120,102,225,30,210,255, ...
         135,45,204,120,170,225,180,210,153,135,225,204,0,170,0,180,0, ...
         153,0,225,210,0,85,0,153,0,225,0,210,210,135,85,30,153,45,225, ...
         120,210,225,135,210,30,85,45,153,120,51,225,85,210,75,85,102, ...
         153,30,51,45,85,120,75,225,102,0,30,0,45,0,120,210,225,135,0, ...
         204,0,120,0,51,210,85,135,153,204,51,120,85,51,153,85,51,153, ...
         135,51,204,85,170,153,102,51,30,135,45,204,120,170,51,102,85, ...
         30,153,45,225,120,0,51,0,85,210,153,85,225,75,0,180,0,75,210, ...
         102,85,204,75,170,180,102,75,204,102,170,204,180,170,75,102, ...
         102,204,204,170,120,180,51,75,85,102,75,204,102,120,204,51, ...
         120,85,225,75,0,102,210,204,135,120,30,225,255,0,255,210,45, ...
         135,170,30,102,255,204,255,170,45,102,170,30,102,255,204,45, ...
         170,170,102,180,30,75,255,102,45,30,170,45,180,170,75,180,102, ...
         153,30,225,45,210,170,85,180,153,153,225,225,0,210,210,85,135, ...
         153,204,225,170,0,102,210,204,135,120,204,225,170,210,102,135, ...
```

```
        204,30,120,255,225,45,210,120,135,51,30,135,255,30,45,45,120, ...
        120,51,51,135,135,30,204,45,120,120,225,51,210,135,85,204,75, ...
        120,102,225,204,210,170,85,180,75,153,102,51,204,85,170,153, ...
        180,225,153,210,51,85,85,75,153,180,225,153,210,51,85,85,75, ...
        75,180,180,153,75,51,180,85,153,75,51,180,135,75,30,180,45, ...
        153,170,51,102,199,30,30,45,45,170,170,102,102,204,30,120, ...
        45,51,170,135,102,30,204,255,120,45,51,170,135,102,30,30,255, ...
        255,45,255,170,255,102,45,30,170,255,180,255,153,255,51,45,135, ...
        170,204,180,120,153,51,51,135,135,204,204,170,120,180,51,75,135, ...
        180,204,153,170,225,180,210,75,135,180,204,153,120,225,225,210,0, ...
        135,0,204,210,120,135,225,30,0,45,0,170,210,180,135,75,30,180, ...
        45,75,170,180,180,75,75,102,180,30,75,255,180,255,75,45,102,120, ...
        30,51,255,85,255,75,45,180,120,153,51,225,85,0,75,210,180,85,153, ...
        153,225,51,0,135,210,30,85,255,153,255,51,255,135,255,30,0,0,0,0, ...
        135,225,170,204] ;
elseif CR > 0 && CR <= 2
    white_sequence = [255,255,45,255,120,255,48,46,0,46,18,60,20,40,10, ...
        48,54,0,30,18,46,20,60,10,40,54,48,30,18,46,6,60,12,40,58,48,36, ...
        18,24,6,48,12,0,58,0,36,0,24,0,48,18,0,20,0,24,0,48,0,18,18,6,20, ...
        30,24,60,48,40,18,48,6,18,30,20,60,24,40,34,48,20,18,10,20,54,24, ...
        30,34,60,20,40,10,48,54,0,30,0,60,0,40,18,48,6,0,12,0,40,0,34,18, ...
        20,6,24,12,34,40,20,34,24,20,34,24,6,34,12,20,58,24,54,34,30,6,60, ...
        12,40,58,34,54,20,30,24,60,48,40,0,34,0,20,18,24,20,48,10,0,36,0, ...
```

```
        10,18,54,20,12,10,58,36,54,10,12,54,58,12,36,58,10,54,54,12,12,58, ...
        40,36,34,10,20,54,10,12,54,40,12,34,40,20,48,10,0,54,18,12,6,40,30, ...
        48,46,0,46,18,60,6,58,30,54,46,12,46,58,60,54,58,30,54,46,12,60,58, ...
        58,54,36,30,10,46,54,60,30,58,60,36,58,10,36,54,24,30,48,60,18,58, ...
        20,36,24,24,48,48,0,18,18,20,6,24,12,48,58,0,54,18,12,6,40,12,48, ...
        58,18,54,6,12,30,40,46,48,60,18,40,6,34,30,6,46,30,60,60,40,40,34, ...
        34,6,6,30,12,60,40,40,48,34,18,6,20,12,10,40,54,48,12,18,58,20,36, ...
        10,24,54,34,12,20,58,24,36,48,24,18,34,20,20,10,24,36,48,24,18,34, ...
        20,20,10,10,36,36,24,10,34,36,20,24,10,34,36,6,10,30,36,60,24,58, ...
        34,54,6,30,30,60,60,58,58,54,54,12,30,40,60,34,58,6,54,30,12,46, ...
        40,60,34,58,6,54,30,30,46,46,60,46,58,46,54,60,30,58,46,36,46,24, ...
        46,34,60,6,58,12,36,40,24,34,34,6,6,12,12,58,40,36,34,10,6,36,12, ...
        24,58,48,36,18,10,6,36,12,24,40,48,48,18,0,6,0,12,18,40,6,48,30,0, ...
        60,0,58,18,36,6,10,30,36,60,10,58,36,36,10,10,54,36,30,10,46,36,46, ...
        10,60,54,40,30,34,46,20,46,10,60,36,40,24,34,48,20,0,10,18,36,20,24, ...
        24,48,34,0,6,18,30,20,46,24,46,34,46,6,46,30,0,0,0,0,36,6] ;
    end
end
N = min([length(symbols) length(white_sequence)]) ; % LoRa symbol length
symbols_white = bitxor(symbols(1:N),white_sequence(1:N)) ;
end
function [y] = rotl(bits,count,size)
% rotl
```

```
% in:  bits           bit sequence
% counts
% size
%  out:  y              rotated symbols
len_mask = bitsll(1,size) - 1 ;
count = mod(count,size) ;
bits = bitand(bits,len_mask) ;
y = bitor(bitand(bitsll(bits,count),len_mask), floor(bitsra(bits,size - count))) ;
end
function [r] = selectbits(data,indices)
% selectbits concat zeros (from 4-bit to 8-bit)
% in:  data           symbol sequence
% indices        vector = [1 2 3 4 5]
%  out:  r       `       symbols
 r = 0 ;
for ctr = 0 : length(indices) - 1
   if bitand(data,bitsll(1,indices(ctr+1))) > 0
      r = r + bitsll(1,ctr) ; % shift to left
   else
      r = r + 0 ;
   end
end
end
```

```
function [symbols] = FSKDetection(signal,SF,detection)
% LoRa_Tx demodulates a Lora de-chirped signal using
% the coherence specified by the detection variable
% in:  message      payload message
% SF           spreading factor
% detection    1= coherent detection, 2= non-coherent detection
%  out:  symbols      FSK demodulated symbol vector
 if detection == 1 % coherent detection
    t = 0:1/(2^SF):0.999 ; % time vector
    for Ctr = 1 : 2^SF
        rtemp  =  conv(signal,exp(-j.*2.*pi.*(2^SF  -  Ctr  +  1).*t))  ;  % convolution
w/ideal fsk signal
        r(Ctr,:) = real(rtemp(2^SF+1:2^SF:end)) ; % save resultant array
    end
    [~,idx] = max(r) ; % take max
    symbols = idx - 1 ; % store symbol vector
elseif detection == 2 % non-coherent detection
    [~,idx] = max(fft(reshape(signal,2^SF,length(signal)/(2^SF)))) ; % take max of
fft window
    symbols = idx - 1 ; % store symbol array
end
end
```

==========(5)==========

```
function message = Decrypt(key,em)
%AES-256 decoding using Cipher Block Chaining (CBC) mode with ciphertext
%stealing. Encrypted message length needs to be greater than 256-bits or 64
%hexidecimal characters. Key is a 256-bit hexidecimal string (64 characters
%long). Message output is a character string (UTF-8).
m=hex2dec(reshape(em,2,[])')';
count=ceil(length(m)/16);
b=length(m)-16*(count-1);
intiationVector =
[173,76,37,247,147,53,68,24,174,242,40,171,201,231,131,99];%just picked a
random vector
w=KeyExpansion(key);
e=InvCipher(w,m(1:16));
message(1,:)=bitxor(e,intiationVector);
for k=2:count-1
   e=InvCipher(w,m(16*(k-1)+1:16*k));
   message(k,:)=bitxor(e,m(16*(k-2)+1:16*(k-1)));
end
mm=[m(16*(count-1)+1:end),e(b+1:end)];
ee=InvCipher(w,mm);
message(count-1,:)=bitxor(ee,m(16*(count-3)+1:16*(count-2)));
message(count,:)=bitxor(e,mm);
message=message';
```

```
message=native2unicode(message(1:length(m)),'UTF-8');
end
Tambahan
Time
function a = xtime(x,c)
a=0;
if bitget(x,1)
    a=c;
end
x=bitshift(x,-1);
while x>0
    c=bitshift(c,1);
    if bitget(c,9)
        c=bitset(c,9,0);
        c=bitxor(c,27);
    end
    if bitget(x,1)
        a=bitxor(a,c);
    end
    x=bitshift(x,-1);
end
end
```

SubstitutionByte

```
function state = SubBytes(state)
Sbox=['637c777bf26b6fc53001672bfed7ab76';...
    'ca82c97dfa5947f0add4a2af9ca472c0';...
    'b7fd9326363ff7cc34a5e5f171d83115';...
    '04c723c31896059a071280e2eb27b275';...
    '09832c1a1b6e5aa0523bd6b329e32f84';...
    '53d100ed20fcb15b6acbbe394a4c58cf';...
    'd0efaafb434d338545f9027f503c9fa8';...
    '51a3408f929d38f5bcb6da2110fff3d2';...
    'cd0c13ec5f974417c4a77e3d645d1973';...
    '60814fdc222a908846eeb814de5e0bdb';...
    'e0323a0a4906245cc2d3ac629195e479';...
    'e7c8376d8dd54ea96c56f4ea657aae08';...
    'ba78252e1ca6b4c6e8dd741f4bbd8b8a';...
    '703eb5664803f60e613557b986c11d9e';...
    'e1f8981169d98e949b1e87e9ce5528df';...
    '8ca1890dbfe6426841992d0fb054bb16'];
Sbox=reshape(hex2dec(reshape(Sbox',2,[])'),16,16);
state=Sbox(state+1);
end
```

ShiftRows

```
function state = ShiftRows(state)
```

```
state(2,:)=circshift(state(2,:),[0 -1]);
state(3,:)=circshift(state(3,:),[0 -2]);
state(4,:)=circshift(state(4,:),[0 -3]);
end
function State = MixColumns(state)
State=state;
for a=1:4:13

State(a)=bitxor(bitxor(bitxor(xtime(state(a),2),xtime(state(a+1),3)),state(a+2)),stat
e(a+3));

State(a+1)=bitxor(bitxor(bitxor(xtime(state(a+1),2),xtime(state(a+2),3)),state(a)),s
tate(a+3));

State(a+2)=bitxor(bitxor(bitxor(xtime(state(a+2),2),xtime(state(a+3),3)),state(a)),s
tate(a+1));

State(a+3)=bitxor(bitxor(bitxor(xtime(state(a+3),2),xtime(state(a),3)),state(a+1)),s
tate(a+2));
end
end
```

AddRoundKey

```
function state = AddRoundKey(state,w)
for k=1:4
   state(:,k)=bitxor(state(:,k),w(:,k));
end
end
```

KeyExpansion

```
function w = KeyExpansion(key)
key=hex2dec(reshape(key,2,[])');
w=reshape(key,4,[]);
for i=8:59
   temp=w(:,i);
   if mod(i,8)==0
      temp=SubBytes(circshift(temp,-1));
      temp=bitxor(temp,[2^(i/8-1),0,0,0]');
   elseif mod(i,8)==4
      temp=SubBytes(temp);
   end
   w(:,i+1)=bitxor(w(:,i-7),temp);
end
end
```

InversShiftRow

```
function state = InvShiftRows(state)
state(2,:)=circshift(state(2,:),[0 1]);
state(3,:)=circshift(state(3,:),[0 2]);
state(4,:)=circshift(state(4,:),[0 3]);
end
```

InversSubByte

```
function state = InvSubBytes(state)
Sbox=['52096ad53036a538bf40a39e81f3d7fb';...
    '7ce339829b2fff87348e4344c4dee9cb';...
    '547b9432a6c2233dee4c950b42fac34e';...
    '082ea16628d924b2765ba2496d8bd125';...
    '72f8f66486689816d4a45ccc5d65b692';...
    '6c704850fdedb9da5e154657a78d9d84';...
    '90d8ab008cbcd30af7e45805b8b34506';...
    'd02c1e8fca3f0f02c1afbd0301138a6b';...
    '3a9111414f67dcea97f2cfcef0b4e673';...
    '96ac7422e7ad3585e2f937e81c75df6e';...
    '47f11a711d29c5896fb7620eaa18be1b';...
    'fc563e4bc6d279209adbc0fe78cd5af4';...
    '1fdda8338807c731b11210592780ec5f';...
    '60517fa919b54a0d2de57a9f93c99cef';...
```

```
    'a0e03b4dae2af5b0c8ebbb3c83539961';...
    '172b047eba77d626e169146355210c7d'];
Sbox=reshape(hex2dec(reshape(Sbox',2,[])'),16,16);
state=Sbox(state+1);
end
```

InversMixColumn

```
function State = InvMixColumns(state)
State=state;
for a=1:4:13

State(a)=bitxor(bitxor(bitxor(xtime(state(a),14),xtime(state(a+1),11)),xtime(state(
a+2),13)),xtime(state(a+3),9));

State(a+1)=bitxor(bitxor(bitxor(xtime(state(a),9),xtime(state(a+1),14)),xtime(state
(a+2),11)),xtime(state(a+3),13));

State(a+2)=bitxor(bitxor(bitxor(xtime(state(a),13),xtime(state(a+1),9)),xtime(state
(a+2),14)),xtime(state(a+3),11));

State(a+3)=bitxor(bitxor(bitxor(xtime(state(a),11),xtime(state(a+1),13)),xtime(stat
e(a+2),9)),xtime(state(a+3),14));
end
```

end

Standar LoRa

Output

message =

"MODELING AND SIMULATION OF LONG RANGE (LORA) COMMUNICATION SYSTEM ON SMART GRID"

key =

'012132435465768798a90a1b2c3d0e0f10234213243546576879 8a0b1c2d3e4f'

ciphertext =

'aa33decbb022ca90b73b7d74ec82c7112e5e4024bbb0f6529c85e91104213c8ae36a
7c39699e5fb2da0775447545b51d0ad48a0e78f1c7eaa60236f292b22832f4076e1b
ca430d2c480868898bb3f3'

Transmit Power = 14 dBm

message_encrypt =

Columns 1 through 17

97 97 51 51 100 101 99 98 98 48 50 50 99 97 57 48 98

Columns 18 through 34

55 51 98 55 100 55 52 101 99 56 50 99 55 49 49 50 101

Columns 35 through 51

53 101 52 48 50 52 98 98 98 48 102 54 53 50 57 99 56

Columns 52 through 68

53 101 57 49 49 48 52 50 49 51 99 56 97 101 51 54 97

Columns 69 through 85

55 99 51 57 54 57 57 101 53 102 98 50 100 97 48 55 55

Columns 86 through 102

53 52 52 55 53 52 53 98 53 49 100 48 97 100 52 56 97

Columns 103 through 119

48 101 55 56 102 49 99 55 101 97 97 54 48 50 51 54 102

Columns 120 through 136

50 57 50 98 50 50 56 51 50 102 52 48 55 54 101 49 98

Columns 137 through 153

99 97 52 51 48 100 50 99 52 56 48 56 54 56 56 57 56

Columns 154 through 158

98 98 51 102 51

message_out =

'MODELING AND SIMULATION OF LONG RANGE (LORA) COMMUNICATION SYSTEM ON SMART GRID'

Message Received = MODELING AND SIMULATION OF LONG RANGE (LORA) COMMUNICATION SYSTEM ON SMART GRID

>>

Lampiran 2. Karya ilmiah/artikel yang telah dipublikasikan selama Pendidikan

1. Judul Publikasi : *Improved Data Security using Advanced Encryption Standard Algorithm on Long-Range Communication System at Smart Grid*
   Nama Jurnal : *ICIC Express Letters, Part B : Applications –An International Journal of Research and Surveys*
   URL : www.ijicic.org/icicelb.htm
   Volume : 14, *Number* 5
   ISSN : 2185-2766
   Penerbit : ICIC *Express Letters Office*
   Tahun : *May* 202
   Status ketercapaian : *Accepted*

ICIC Express Letters, Part B: Applications
An International Journal of Research and Surveys

November 29, 2022

ACCEPTANCE LETTER (Paper ID: ICICIC2022-075)

Dr. Amil Ahmad Ilham
Universitas Hasanuddin
Indonesia
E-mail: amil@unhas.ac.id

Dear Dr. Amil Ahmad Ilham,

We are happy to inform you that after peer review, your following paper,

Reference No.: ICICIC2022-075
Title: Improved Data Security Using Advanced Encryption Standard Algorithm on Long-Range Communication System at Smart Grid
Author(s): Isminarti, Amil Ahmad Ilham, Ardiaty Arief and Syafaruddin

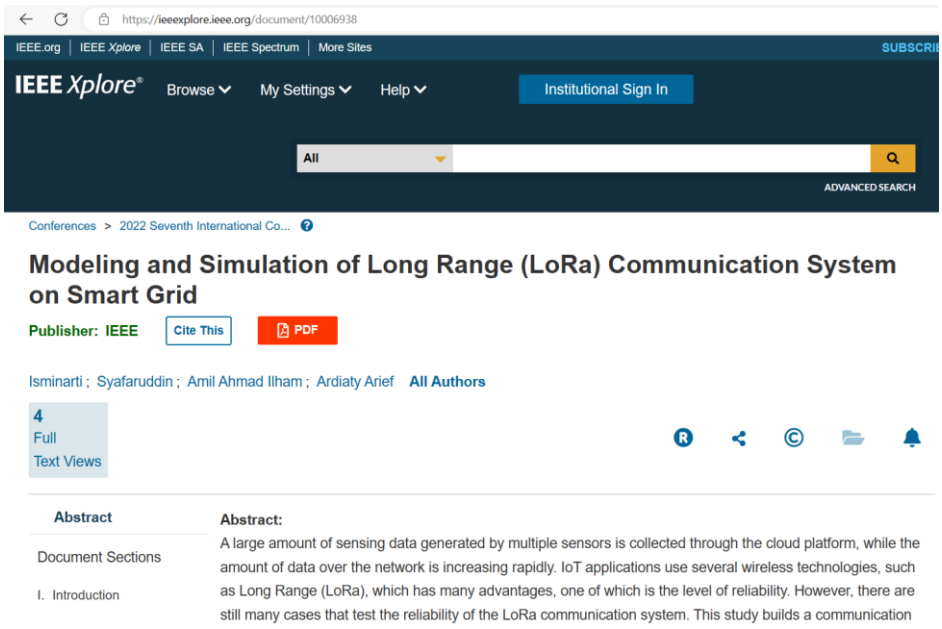
has been formally accepted for publication in ICIC Express Letters, Part B: Applications – An International Journal of Research and Surveys (ISSN 2185-2766).

We are planning tentatively to publish your paper in Volume 14, Number 5, May 2023.

Thank you for submitting your work to ICIC-ELB.

ICIC-ELB Editorial Office
Tokai University, Kumamoto Campus
9-1-1, Toroku, Kumamoto 862-8652, Japan
Tel: +81-96-386-2666
E-mail: office@icicelb.org
URL: http://www.icicelb.org

2. Judul Publikasi : *Modeling and Simulation of Long Range (LoRa) Communication System on Smart Grid*
   Nama Prosiding : *International Conference on Informatics and Computing (ICIC) 2022*
   URL : https://icic-aptikom.org/2022/
   Penerbit : IEEE Xplore
   Tahun : 2023
   Status ketercapaian : *Published*



3. Judul Publikasi : *Security Performance of Advanced Encryption Standard 128 for Long-Range Communication System*
   Nama Prosiding : *International Conference on Computer Science and Engineering Technology (ICCSET) 2022*
   URL : https://iccset.umk.ac.id/2022
   Penerbit : AIP Publishing
   Tahun : 2022
   Status ketercapaian : *Accepted*


CERTIFICATE

ICCSET 2022

Kampus Merdeka INDONESIA JAYA

NO.138/ICCSET/FT/UMK/XI/2022

This to certify that

ISMINARTI

who has attended the conference as

Presenter

International Conference on Computer Science and Engineering Technology 2022

"The Role of Industry 5.0, Innovation and Infrastructure for SDGs."

Faculty of Engineering - Universitas Muria Kudus, 26 November 2022

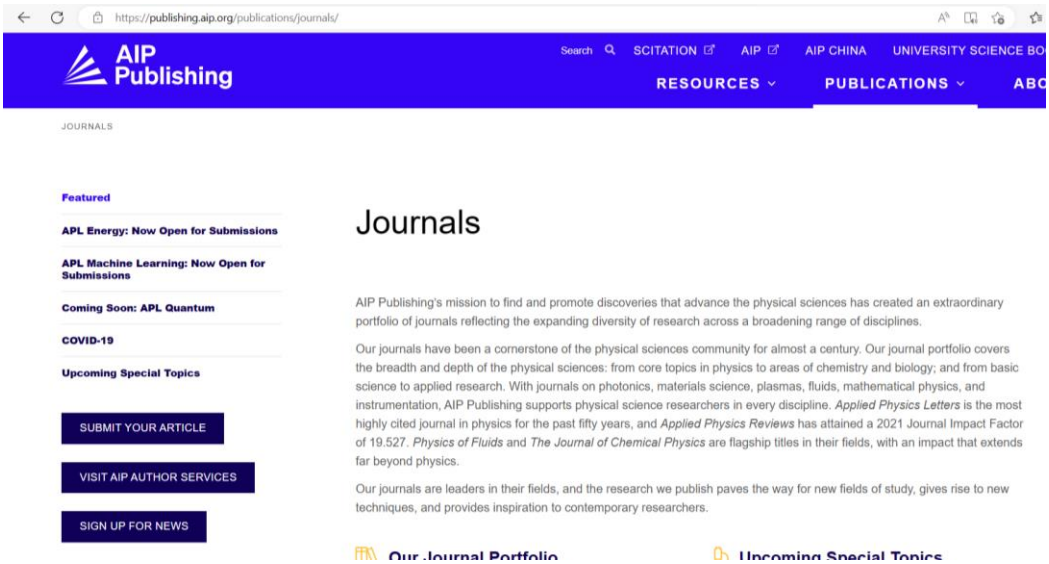
Dean Faculty of Engineering

Universitas Muria Kudus

Muhammad Dahlan, S.T., M.T.

Chief Committee of ICCSET 2022

Tri Listyorini, S.Kom., M.Kom



4. Judul Publikasi : *Increased AES 128 Security to AES 256 for Long-Range Communication System Reliability*

   Nama Prosiding : *The 17th ICAST 2022 International Student Conference on Advanced Science and Technology*

   Penerbit : Kumamoto University, Jepang

   Tahun : 2022

   Status ketercapaian : *Accepted*

Kumamoto University

Certificate

This is to certify that

Isminarti

Universitas Hasanuddin

gave a presentation in Research Session of

the 17th International Student Conference on Advanced Science and Technology (ICAST) Online

December 1 - 2, 2022

Prof. Hitoshi FUJIMOTO
Dean, Graduate School of Science and Technology
Kumamoto University, Japan

5. Judul Publikasi : *Reducing the Error Rate using Additive White Gaussian Noise and Rayleigh Channels in the Communication System.*

   Nama Jurnal : *International Journal on Informatics Visualization*

   Status ketercapaian : *Submit*

[JOIV] Submission Acknowledgement
3 pesan

Rahmat Hidayat <joiv@pnp.ac.id> 3 Januari 2023 pukul 15.33
Kepada: Mrs Isminarti isminarti <isminarti@politeknikbosowa.ac.id>

Mrs Isminarti isminarti:

Thank you for submitting the manuscript, "Reducing the Error Rate using Additive White Gaussian Noise and Rayleigh Channels in the Communication System" to JOIV : International Journal on Informatics Visualization. With the online journal management system that we are using, you will be able to track its progress through the editorial process by logging in to the journal web site:

Manuscript URL: https://joiv.org/index.php/joiv/author/submission/1540
Username: isminarti

If you have any questions, please contact me. Thank you for considering this journal as a venue for your work.

In addition, due to the rising costs of academic publications, starting 2021, publication fees shall be implemented to all accepted papers. For more details, please email to joiv [at] pnp.ac.id. This journal charges the following author fees (Article Publication Fee):
- Indonesian authors: 3.500.000 IDR per article
- International authors: 280 USD per article

This fee includes:
- DOI registration for each paper
- Checking the article similarity by turnitin
- English proofreading

For hardcopy, the fee (excluding postage and handling cost) is 500.000 IDR per issue.

If the manuscript is accepted for publication, you will be asked to pay an Article Publication Fee to cover publications costs. This charge is for the first 10 pages, and if any published manuscript over 10 pages will incur extra charges 50 USD per page.

Best regards,

Rahmat Hidayat
JOIV : International Journal on Informatics Visualization

http://joiv.org/index.php/joiv

Isminarti ` <isminarti@politeknikbosowa.ac.id> 3 Januari 2023 pukul 15.36
Kepada: Rahmat Hidayat <joiv@pnp.ac.id>

Yes, I accept.
[Kutipan teks disembunyikan]

Lampiran 3. Biodata

BIODATA

A. IDENTITAS DIRI

Nama	:	Isminarti
Tempat/Tanggal Lahir	:	Ujung Pandang, 30 Januari 1979
Jenis Kelamin	:	Perempuan
NIDN	:	0930017903
NIK	:	2013004281
Pangkat/Jabatan	:	Lektor / IIId
Alamat rumah	:	Citra Garden Cluster Bronze Astelia Blok H3/9, Gowa
Nomor HP	:	081355080221
Alamat Email	:	isminarti@politeknikbosowa.ac.id
Nama Orang Tua		
Ayah	:	Drs. H. Lukman (Alm)
Ibu	:	Dra. Hj. Rusmi Dahlan
Nama Suami	:	Muhammad Taslim, S.T.
Nama anak	:	1. Afiqah Faiza Anindita 2. Afifah Faiqa Anindya 3. Muhammad Farzan Alghifari

B. RIWAYAT PENDIDIKAN

No	Tahun Lulus	Jenjang	Lokasi (Dlm/Luar Negeri)	Perguruan Tinggi	Program Studi
1	2012	Magister	Dalam Negeri	Universitas Hasanuddin	Teknik Elektro
2	2004	Sarjana	Dalam Negeri	Universitas Hasanuddin	Teknik Elektro

No	Tahun Lulus	Jenjang	Lokasi (Dlm/Luar Negeri)	Perguruan Tinggi	Program Studi
3	2000	Diploma	Dalam Negeri	Politeknik Negeri Ujung Pandang	Teknik Telekomunikasi

C. RIWAYAT PEKERJAAN

1. Dosen Tetap Yayasan Program Studi Teknik Mekatronika Politeknik Bosowa tahun 2013 - sampai sekarang,
2. Dosen di Sekolah Tinggi Managemen Informatika dan Komputer (STMIK) Dipanegara Makassar tahun 2011 - 2012.
3. Dosen di Universitas Indonesia Timur tahun 2012 - 2013.
4. Sebagai Asisten Manager Keuangan di PT. Singa Langit tahun 2005 – 2013.
5. Sebagai General Manager di PT. Dhelmara Kurnia Raya Jakarta Selatan tahun 2000 - 2002.

D. KARYA ILMIAH/ARTIKEL YANG TELAH DIPUBLIKASIKAN DALAM 5 TAHUN TERAKHIR

Judul Artikel Ilmiah	Nama Jurnal
Improved Data Security Using Advanced Encryption Standard Algorithm on Long-Range Communication System at Smart Grid	ICIC Express Letters, Part B:Applications - An International Journal of Research and Surveys Volume :14, Number 5 ISSN:2185-2766 Penerbit : ICIC Express Letters Office - Jepang
Modeling and Simulation of Long Range (LoRa) Communication System on Smart Grid	International Conference on Informatics and Computing (ICIC) 2022 Penerbit: IEEE Xplore
Security Performance of Advanced Encryption Standard 128 for Long-Range Communication System	International Conference on Computer Science and Engineering Technology (ICCSET) 2022 Penerbit : AIP Publishing

Judul Artikel Ilmiah	Nama Jurnal
Reducing the Error Rate using Additive White Gaussian Noise and Rayleigh Channels in the Communication System	JOIV : International Journal on Informatics Visualization , ISSN : 2549-9610 (print) 2549-9904 (online) Penerbit : Scopus Tahun : 2023
Increased AES 128 Security to AES 256 for Long-Range Communication System Reliability	Prosiding International Student Conference on Advanced Science and Technology The 17th ICAST 2022 – Kumamoto Jepang
Rekognisi Karyawan Berdasarkan Kinerja Menggunakan Alogaritma K-MEANS Pada Aplikasi RapidMiner	Ramatekno, ejournal.pei.ac.id/index.php/ LPPM Politeknik Enjinering Indorama
BER, PER dan SER Sistem Komunikasi Long Range (LoRa) pada Smart Grid	Proseding SNETI (Seminar Nasional Elektroteknik dan Teknologi Informasi)
Rancang Bangun Media Pembelajaran Elektronika Analog Untuk Memahami Fungsi dan Karakteristik Op-Amp LM741	Jurnal Rekayasa Teknologi Informasi (JURTI) Vol. 4 No. 2 (sinta 5) http://e-journals.unmul.ac.id/index.php/INF/article/view/4923
Rancang Bangun Pembelajaran Piranti Elektronika untuk Memahami Karakteristik Transistor Bipolar Menggunakan Curve Tracer Arduino	Prosiding Seminar Nasional Teknologi Industri VII 2020 ISBN : 978-602-60451-5-7
Rancang Bangun Media Pembelajaran Praktikum Elektronika Analog II untuk memahami Fungsi dan Karakteristik Multivibrator Monostabil, Bistabil dan Astabil Menggunakan Op-Amp 741	Prosiding Seminar Nasional Teknologi Industri VII 2020 ISBN : 978-602-60451-5-7
Monitoring dan Pengaturan suhu dan kelembaban mesin penetas telur berbasis android	Prosiding Seminar Nasional Teknologi Industri VI 2018
Rancang Bangun Media Pembelajaran Praktikum Elektronika Analog untuk Meningkatkan Pemahaman Mahasiswa dalam Mengetahui Fungsi dan Karakteristik Operational Amplifier	Jurnal Sains Terapan Vol. 4 No. 1 (sinta 4)