Low-cost and Secure Communication System for Remote Micro-grids using AES Cryptography on ESP32 with LoRa Module

Amjad Iqbal, Tariq Iqbal
Faculty of Engineering and Applied Sciences, Memorial University of Newfoundland
amjadi@mun.ca

Abstract- Modern trend, towards renewable energy sources has complicated our power systems' network with the distributed generation and its control. The major challenge of this network is to implement a low cost, secure and authentic communication system between Supervisory Control and Data Acquisition (SCADA) unit and Remote End Devices (RED). This paper addresses the issues of security and authenticity for wireless communication for SCADA system. Algorithm of Advanced Encryption Standard (AES) has been implemented on ESP32 with LoRa module to secure the wireless communication for micro-grids and authenticity has been achieved by generating unique Message Authentication Code (MAC). A point to point communication setup has been developed with range above 10km and cost less than $40 with power consumption of 5mW.

Keywords- Advanced encryption standard, Internet of things, message authentication, supervisory control and data acquisition, small microgrids.

I. Introduction

In Canada and rest of the world many communities are located remotely and isolated from the central power grid and rely on distributed renewable power generation units. To ensure the power quality and control of these distributed power generation sources, SCADA system becomes an integral part of micro-grid network. In Smart Grids (SG) and micro-grid network, major challenges of SCADA system is the lack of low cost, secure and authentic communication system with minimum power consumption [1],[2],[3]. In 2008, the Russian army took the charge of Georgian electric grid by controlling the SCADA system of their grid and made them realize the importance of the micro-grid communication security. According to the Wall Street Journal report, in 2009, spies hacked the control system of U.S electrical grid and disrupted the system [4]. Therefore, the communication security of electrical grid, specifically the setup related to its SCADA system must have strong resistance against eavesdropper and masquerader. Usually, a communication system is regarded secure if it satisfies the following four features [5].

Privacy: Message should be encoded or encrypted such that only authorized receivers can read the message.

Message Authentication: Message should be authentic and only privileged nodes can send the message and furthermore, no eavesdropper be able to masquerade the receiver by sending fake message.

Integrity: The message received at the receiver is exactly same what the sender sent.

Nonrepudiation: If there is any alteration in message, whether due to channel error or attacker interference, receiver must be able to recognize that and decline the data transfer or connection.

Although, in [6],[7],[8] few techniques have been discussed to address the communication security issue but, in their proposed methods a third party is involved to ensure the security of communication network or setup depends upon third party network to communicate with Remote End Devices (RED). Different encryption algorithms have been proposed in [9],[10] to secure communication system using cryptographic techniques eg. shift ciphers and substitution based, encryption algorithms but, they are too simple to break for a cryptanalyst, and can easily take the control of system and can modify the control messages as demonstrated in fig.1. In this way control information becomes prone to the eavesdropper and loses the authenticity and security. Specifically, in smart grid network, secure communication between domestic or commercial energy meters and distributed protection setup with SCADA system requires low cost, power efficient and secure communication setup. In this paper, we have proposed and implemented a secure and authentic communication system using Advanced Encryption Standard (AES) used for extreme confidential communication purposes usually for military applications without the involvement of any third party.

In *Section II,* implementation of different encryption algorithms' on Arduino DRF1276G/ESP32 with LoRa for communication security has been discussed, and compared based upon their security and resistance against attacks. AES algorithm implementation steps and security is explained in *Section III*. In *Section IV*, the results of AES implementation on Arduino DRF1276G-LoRa and ESP32-LoRa have been shown with the implementation of AES algorithms with unique MAC address for each message using ESP32 LoRa based secure communication system for microgrids.

978-1-5386-5419-4/18/$31.00 ©2018 IEEE

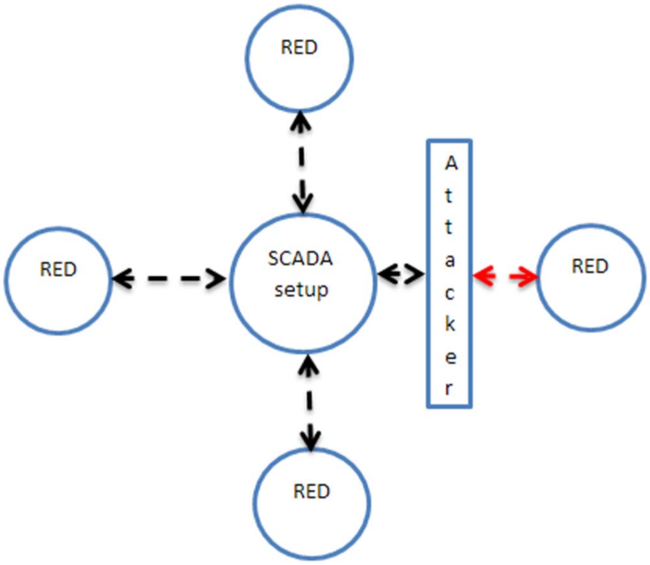


Fig. 1 Eavesdropper Masquerading the SCADA Network

II. Cryptographic Algorithms on Arduino with DRF1276G LoRa Module

To achieve the previously discussed four features of a secure and authentic communication system for microgrid, multiple encryption algorithms were implemented on arduino DRF1276G LoRa module but, all cryptographic algorithms do not provide the equal secrecy level. Fig. 2 shows a photo of Arduino LoRa module used to implement the ciphers discussed below with their security against attacks [11].

Shift Cipher: In shift cipher all characters of the message are shifted by same number. For example, if message is 'abcdef' and shift is by 3 characters, then after shift 'a' will go to 'd', 'b' to 'e' and so on as demonstrated below.

Plaintext a b c d e f
↓ ↓ ↓ ↓ ↓ ↓
Ciphertext d e f g h i

As, there are only 25 possible shifts which means its key set has only 25 elements and cipher can easily be broken within 25 attempts.

***Affine Cipher*:** Key set of affine cipher is bit larger than substitution cipher. In this, ciphertext is calculated by solving simple linear equation under modulo 26 because, there are only 26 alphabets. Let 'y' indicates ciphertext and 'x' indicates plaintext then: y=a*x+b

where a and b are constants but less than 26. Its key space is possible values for b and possible values for a which are 26 and 12 respectively. So, this cipher could be broken within 26x12=312 attempts.

***Substitution Cipher*:** Substitution cipher gives much better security than shift and affine cipher due to large key size. In implementation, it is quite similar to shift cipher but, each plaintext character is not shifted by same number eg.

Plaintext a b c d e f
Ciphertext d z h k a f

First character could be substituted by any of other 25 characters, second character by any of the rest of 24 characters and so on. In this way possible key size becomes

|K|=25x24x23……1=25!

Transposition Cipher: In this cipher, characters are not substituted but, they are shuffled with each other within the plaintext block eg.

Plaintext C A N A D A
↓
Ciphertext D N A C A A

It security depends upon the block size. If a block has n characters, then key set will have total n! possible values.

Hill Cipher: Hill cipher is based upon simple linear algebra and its feature is that it is not injective cipher. It is similar to affine cipher and the only difference is that it works on matrixes and columns of plain/ciphertext rather than individual characters. In this cipher we assign numbers to all alphabet characters eg. a=0, b=1 and similarly y=24 and z=25 and use nxn square matrix as a key matrix to get the column matrix of ciphertext from the column vector of plaintext. For example, if key matrix is $\begin{bmatrix}1 & 3\\2 & 1\end{bmatrix}$ and $\begin{bmatrix}D\\R\end{bmatrix}=\begin{bmatrix}4\\18\end{bmatrix}$ is supposed to be encrypted then our cipher will be $\begin{bmatrix}6\\0\end{bmatrix}=\begin{bmatrix}G\\A\end{bmatrix}$ as shown below.

$$\begin{array}{c}D\rightarrow\\R\rightarrow\end{array}\begin{bmatrix}1 & 3\\2 & 1\end{bmatrix}\begin{bmatrix}4\\18\end{bmatrix}=\begin{bmatrix}58\\26\end{bmatrix}=\begin{bmatrix}6\\0\end{bmatrix}\pmod{26}$$

Although, hill cipher key space is m^{n^2} where m is the modulo and n is the size of the matrix but, it is vulnerable to chosen plaintext attack.

Although, all previously discussed cryptographic techniques could be implemented for simple communication purpose but, could not be used for SCADA system because of their vulnerability to attack by any cryptanalyst and limited number of key space. That is why we have implemented AES algorithm to ensure the security of SCADA system which not only has key space of 2^{128} possible keys but also has non-linear in nature and is flexible to change the pattern of output by changing number of cascaded encryption rounds. To implement that we were not able to use Arduino with DRF1276G LoRa module, due to its small flash size and CPU limitations. To implement AES, we have used ESP32 which has not only sufficient flash, better CPU, low cost and also has very little power consumption 3.5-5mW.

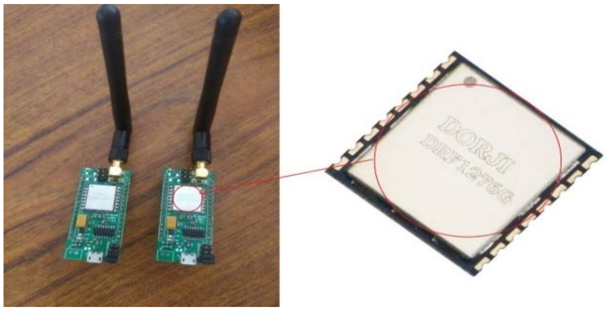


Fig 2 Arduino DRF1276G with LoRa module used for implementing cryptographic algorithms

III. Implementation of AES Algorithm using ESP32 and LoRa Module

In this implementation, MAC has been added to authenticate the communication and AES implemented to encrypt the message, which is the most secure communication algorithm to all known attacks till now [12]. Fig. 3 shows the flow chart of this implementation. Before sending any message, first that is encrypted using AES encryption algorithm and then a 64 bit unique MAC is generated from the plaintext and then cipher text and MAC, both strings are concatenated and sent. Similarly, on the receiver side, first MAC and ciphertext are separated, MAC is verified and then decryption of the message is done to proceed further. Fig. 4 shows the step by step 10 round AES encryption and decryption. In AES implementation, after getting binary value from plaintext, there are four major steps which are repeated for each round.

Adding round key: XOR sum is calculated by taking bit wise XOR of each plaintext bit with respective key bit.

Substitute Bytes: After calculating XOR sum, each byte, pair of HEX characters, is replaced with respective Rijndael's table (a standard table of 256 values) to increase the confusion.

Shift Rows: After substitution, all 16 bytes are distributed to construct a 4x4 square matrix. In resultant matrix, first row remains unchanged while the rest of the three rows are rotated left by 1, 2 and 3 bytes respectively.

Mix Column: In this, matrix left multiplication is applied using a standard 4x4 matrix on resultant of shift row operation.

This algorithm does not only has a large key set (2^{128} possible keys) but also is secure from many cryptanalysis algorithms like differential cryptanalysis, integration, linear, multiset and many others like these.

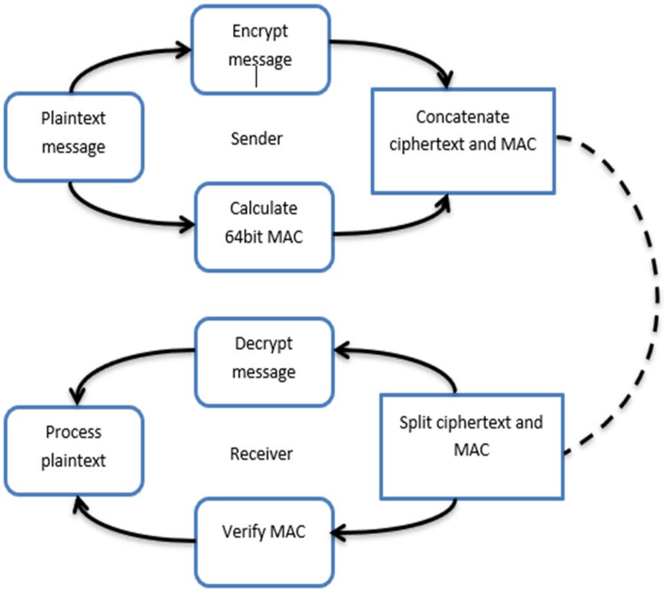


Fig 3. Flow chart of the implemented communication process

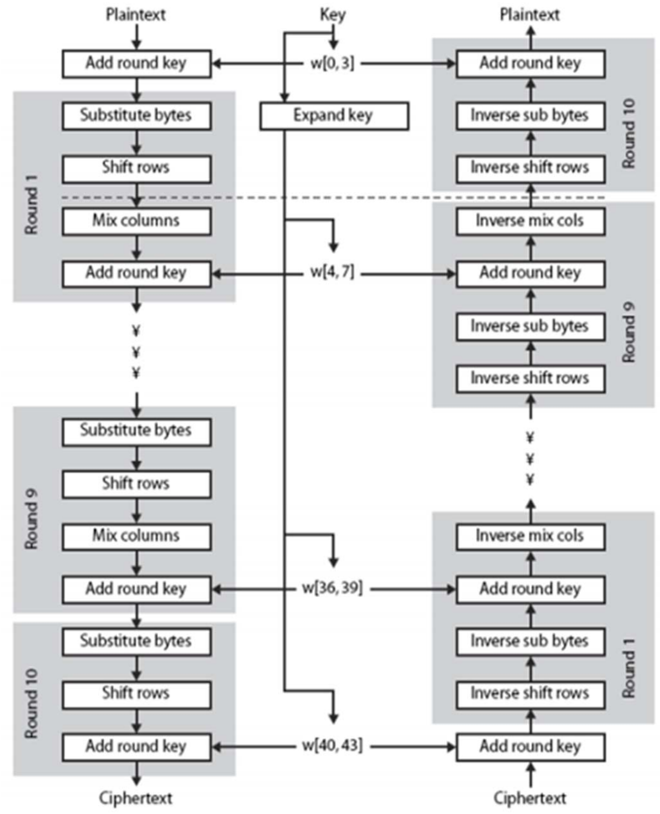


Fig 4. Step by step 10 round AES encryption and decryption [11]

In AES for each round a new key is derived from the round key and the ciphertext of the previous round. From the test results, explained in next section, it could be seen that each round cipher text is entirely different from all others which is because of the implementation of encryption at bit level. Confusion created at each round and propagation of confusion from one round to other makes it more secure.

MAC Generation: After successful implementation of AES in counter mode, a unique and of fixed size (64bit) MAC is generated using the plaintext of the message. Its implementation ensures the authenticity of the message and receiver can easily verify whether the message has been modified by any eavesdropper or due to channel error or not.

IV. Test results

We have tried all previously discussed algorithms and have selected combination of AES algorithm implemented on ESP32 with LoRa for SCADA system after comparing their security, authenticity for data, flexibility to change the key and power consumption of the controllers. After selecting AES different controllers were tried and checked their compatibility with AES. Fig. 5 shows the ESP32 with LoRa module which costs about C$40 per set and consumes power around 5mW and supports the implementation of AES algorithm and AES with MAC as well.

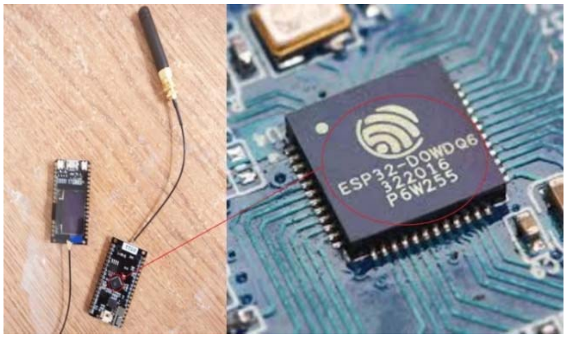
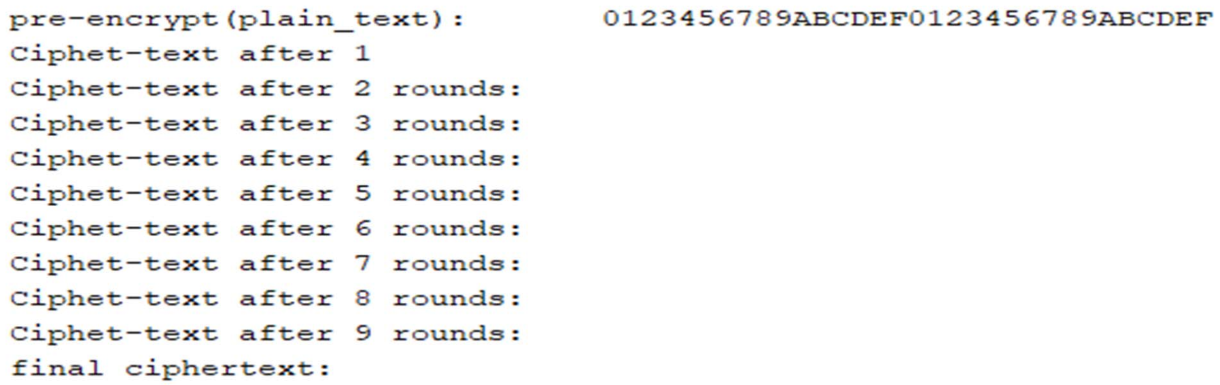


Fig 5. ESP32 with LoRa module used for AES implementation

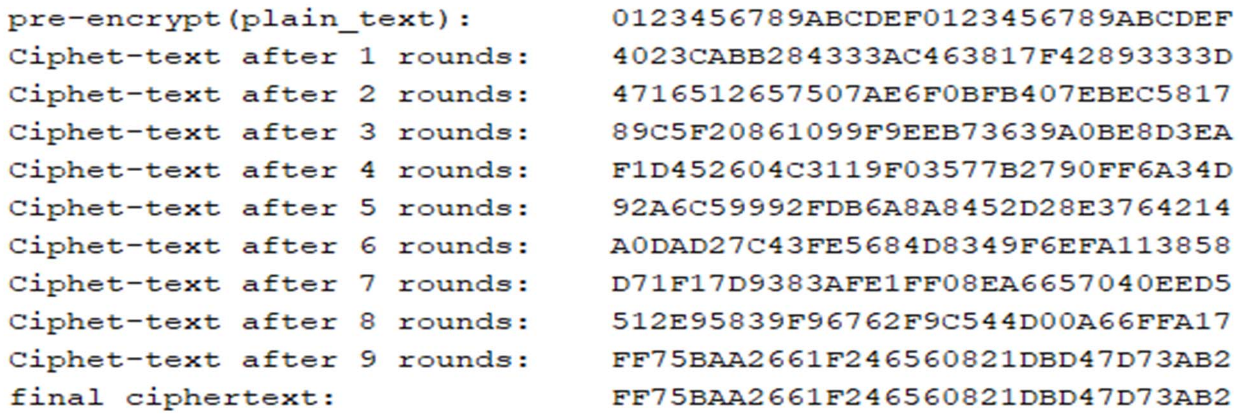
Fig. 6 shows the results of AES implementation on Arduino DRF1276G with LoRa module. The results show that this controller can not support to even a single round of AES implementation due to small flash size and many other limitations of the board.

The results of AES implementation are shown in fig. 7 in which a nine round AES has been implemented on ESP32 with Lora module. Flexibility in changing key has been achieved by externally connected button on pressing which number of encryption rounds and HEX value of key is changed.



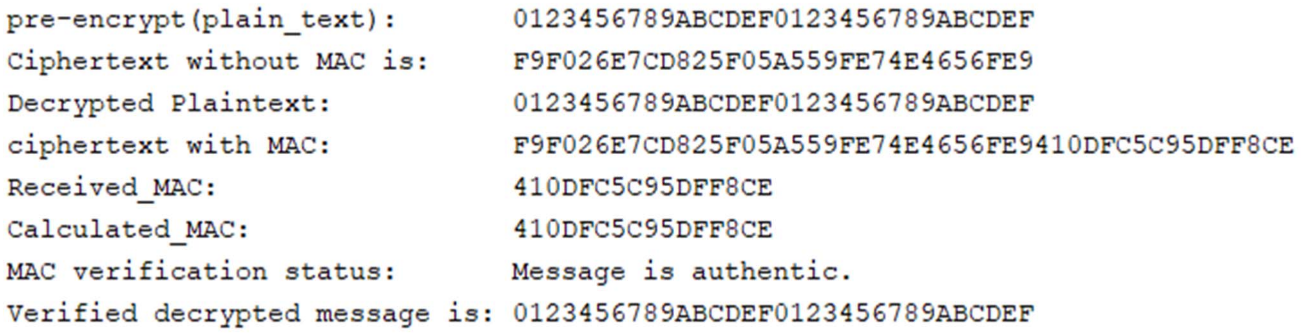
```
pre-encrypt(plain_text):        0123456789ABCDEF0123456789ABCDEF
Ciphet-text after 1
Ciphet-text after 2 rounds:
Ciphet-text after 3 rounds:
Ciphet-text after 4 rounds:
Ciphet-text after 5 rounds:
Ciphet-text after 6 rounds:
Ciphet-text after 7 rounds:
Ciphet-text after 8 rounds:
Ciphet-text after 9 rounds:
final ciphertext:
```

Fig 6. AES implementation on arduino DRF1276G with LoRa module



```
pre-encrypt(plain_text):        0123456789ABCDEF0123456789ABCDEF
Ciphet-text after 1 rounds:     4023CABB284333AC463817F42893333D
Ciphet-text after 2 rounds:     4716512657507AE6F0BFB407EBEC5817
Ciphet-text after 3 rounds:     89C5F20861099F9EEB73639A0BE8D3EA
Ciphet-text after 4 rounds:     F1D452604C3119F03577B2790FF6A34D
Ciphet-text after 5 rounds:     92A6C59992FDB6A8A8452D28E3764214
Ciphet-text after 6 rounds:     A0DAD27C43FE5684D8349F6EFA113858
Ciphet-text after 7 rounds:     D71F17D9383AFE1FF08EA6657040EED5
Ciphet-text after 8 rounds:     512E95839F96762F9C544D00A66FFA17
Ciphet-text after 9 rounds:     FF75BAA2661F246560821DBD47D73AB2
final ciphertext:               FF75BAA2661F246560821DBD47D73AB2
```

Fig 7. AES implementation results on ESP32 with LoRa module



```
pre-encrypt(plain_text):        0123456789ABCDEF0123456789ABCDEF
Ciphertext without MAC is:      F9F026E7CD825F05A559FE74E4656FE9
Decrypted Plaintext:            0123456789ABCDEF0123456789ABCDEF
ciphertext with MAC:            F9F026E7CD825F05A559FE74E4656FE9410DFC5C95DFF8CE
Received_MAC:                   410DFC5C95DFF8CE
Calculated_MAC:                 410DFC5C95DFF8CE
MAC verification status:        Message is authentic.
Verified decrypted message is: 0123456789ABCDEF0123456789ABCDEF
```

Fig 8 Implementation of AES with MAC on ESP32 with LoRa module

Fig. 8 shows the results of implementation of AES with MAC on ESP32 with LoRa module. In which the 192-bit received message is split into 128bit ciphertext and 64bit MAC. Decryption is applied on ciphertext and plaintext is extracted from that after n decryption rounds. From that plaintext again, n+1 round ciphertext is calculated. XOR sum is calculated between alternate bits of n+1 round ciphertext and respective plaintext bits. To check the authenticity of the message, calculated MAC is compared with the received MAC at bit level and even a single bit change, in received message is also detected in this comparator.

V. Conclusions

Implementation of AES cryptography with MAC for SCADA system using ESP32 with LoRa is the best method of secured, authentic and flexible communication. A point to point secure and authentic communication has been achieved for which setup costs less than C$40 and consuming less than 5mW. AES is the most resistant cryptographic algorithm to every attack known till now. It is only vulnerable to brute force attack which requires 2^{128} different keys to be tested to ensure the successful decryption. So, by implementing this for SCADA network we can protect our grids from spies and enemies and can send wireless data over many kilometers with low cost and negligible power consumption.

VI. References

[1] C. Mavrokefalidis, D. Ampeliotis and K.Berberidis, "A study of the communication needs in micro-grid systems," in *General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS), 2017 XXXIInd*, 2017, pp. 1–4.

[2] A. García-Domínguez, "Enabling SCADA cluster and cloud for smart grid using hierarchical multicast; The PTMF framework," in *Proceedings of the IEEE International Conference on Industrial Technology*, 2015, vol. 2015–June, no. June, pp. 218–225.

[3] H. H. and D. M. S. and M. G. and A. K. Safa, "Cyber security of smart grid and SCADA systems, threats and risks," in *CIRED Workshop 2016*, 2016, pp. 1–4.

[4] E. Bou-Harb, C. Fachkha, M. Pourzandi, M. Debbabi, and C. Assi, "Communication security for smart grid distribution networks," *IEEE Commun. Mag.*, vol. 51, no. 1, pp. 42–49, 2013.

[5] A. Tanenbaum, "Network Security," in *Computer Networks*, 5th ed., PEARSON, 2011, pp. 767–790.

[6] D. NamdeoHire, "Secured Wireless Data Communication," *Int. J. Comput. Appl.*, vol. 54, pp. 27–30, 2012.

[7] A. A. P. R. and R. F. S. Amiruddin, "A testbed implementation of secure and lightweight privacy preservation mechanism using scrambled Fibonacci and XOR for ZigBee," in *Region 10 Conference, TENCON 2017 - 2017 IEEE*, 2017, pp. 863–868.

[8] Y.-S. Tsai, C.-Y. Chu, M.-C. Li, Y.-H. Lin, and P. Chen, "Intelligent DC power monitoring system and sensor network based on ZigBee-equipped smart sockets," in *2016 5th International Symposium on Next-Generation Electronics, ISNE 2016*, 2016.

[9] Aa. Shahzad, Y. G. Kim, and A. Elgamoudi, "Secure IoT Platform for Industrial Control Systems," in *2017 International Conference on Platform Technology and Service, PlatCon 2017 - Proceedings*, 2017.

[10] A. V. D. M. Kayem, H. Strauss, S. D. Wolthusen, and C. Meinel, "Key management for secure demand data communication in constrained micro-grids," in *Proceedings - IEEE 30th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2016*, 2016, pp. 585–590.

[11] W. Stallings, *Cryptography and Network Security*, vol. 139, no. 3. 2011, pp. 312-328.

[12] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish," in *Procedia Computer Science*, 2016, vol. 78, pp. 617–624.