IMPLEMENTASI SERANGAN AKTIF PADA ALGORITME SPECK 128/128BIT BERBASIS MODUL KOMUNIKASI LoRa

Dian Astika Rini[1], Ari Kusyanti[2], Kasyful Amron[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]dianastika@student.ub.ac.id, [2]ari.kusyanti@ub.ac.id, [3] kasyful@ub.ac.id

Abstrak

LoRa (Long Range) termasuk pada konektivitas IoT nirkabel terbaru yang sedang berevolusi dan mendapatkan popularitas dalam sistem tertanam dengan dioperasikan menggunakan daya rendah. LoRa perlu mentransfer sejumlah kecil data pada interval pendek dengan jarak yang jauh. Modul komunikasi LoRa menggunakan gelombang radio untuk melakukan komunikasi. Perangkat LoRa dihubungkan dengan frekuensi yang sama. Selain itu, dikarenakan LoRa menggunakan frekuensi sebagai metode komunikasinya, maka LoRa memiliki fitur yaitu broadcast. Pada fitur ini, LoRa mentransfer data pada frekuensi yang sama. Sehingga hal tersebut menjadi sebuah masalah apabila terdapat perangkat LoRa yang melakukan transfer data penting, dikarenakan pada dasarnya LoRa sendiri tidak memiliki upaya keamanan. Solusi berdasarkan permasalahan tersebut adalah diberikannya sebuah upaya keamanan yaitu berupa enkripsi pada data atau pesan yang dikirimkan. Pada penelitian yang dilakukan oleh Semiconductor 2018 telah menerapkan algoritme AES-128 pada perangkat LoRa Namun terdapat penelitian lain yang menguji keefektifan algoritme AES-128bit dan menemukan beberapa kekurangan. Sehingga, pada penelitian ini diterapkan algoritme SPECK 128/128bit yang memiliki kelebihan sehingga dapat menutupi kekurangan dari algoritme sebelumnya untuk mengamankan pesan penting yang dikirimkan agar tidak mudah dibaca oleh pihak yang tidak sah. Untuk menguji kemananan, serangkaian serangan aktif dan pasif dilakukan yaitu dengan metode known-plaintext attack. Hasil dari penelitian ini, sistem mampu memberikan keamanan pada pesan yang dikirimkan dengan mengirimkan pesan dalam bentuk ciphertext sehingga keamanan komunikasi antar modul LoRa menjadi terjamin.

Kata kunci: LoRa, IoT, Algoritme SPECK 128/128bit, Enkripsi, Kerahasiaan, Keamanan

IMPLEMENTATION OF ACTIVE ATTACKS ON THE LORA-BASED OF SPECK 128 / 128BIT ALGORITHM

Abstract

LoRa (Long Range) is one of the latest wireless IoT connectivity that is revolutionizing and gaining popularity in low-power operated embedded systems that need to transfer small amounts of data at short intervals over long distances. LoRa's communication module uses radio waves to communicate. Between LoRa devices will be connected with the same frequency. In addition, because LoRa uses frequency as its communication method, LoRa has a broadcast feature. In this feature, LoRa will transfer data at the same frequency. So it will be a problem if there is a LoRa device that transfers important data, because basically LoRa itself has no security efforts. The solution to the problem is the granting of a security effort in the form of encryption on the data or messages sent. In research conducted by Semiconductor 2018 has applied the AES-128 algorithm to LoRa devices. But there are other studies that test the effectiveness of the AES-128bit algorithm and find some weakness. So, in this study applied 128/128bit SPECK algorithm that has the advantage so as to cover the shortcomings of the previous algorithm to secure important messages that sent so it can't to be read easily by unauthorized parties. As a result of this study, the system was able to provide security to messages sent by sending messages in the form of ciphertext so that the security of communication between LoRa modules became guaranteed..

Keywords: LoRa, IoT, Algoritme SPECK 128/128bit, Encryption, Confidentiality, Security

1. PENDAHULUAN

LoRa (*Long Range*) termasuk pada konektivitas IoT nirkabel terbaru yang yang sedang bevolusi dan mendapatkan popularitas dalam sistem tertanam yang dioperasikan dengan daya rendah yang perlu mentransfer sejumlah kecil data pada interval pendek dalam jarak jauh. Berdasarkan penelitian yang dilakukan Arijjudin (2019). Kekurangan yang dimiliki LoRa yaitu tidak adanya keamanan untuk menjamin komunikasi yang berjalan pada LoRa. Sehingga dapat disimpulkan LoRa membutuhkan upaya keamanan *end-to-end* yang kuat dikarenakan LoRa merupakan teknologi komunikasi yang berhubungan dengan banyak node. Pada penelitian tersebut masih memiliki kekurangan yaitu tidak adanya keamanan dalam pertukaran data. Sebelumnya telah dilakukan percobaan serangan pada sistem yaitu dengan melakukan *sniffing* untuk mendapatkan data yang dikirim dari node ke *gateway* menggunakan *gateway* lain yang tidak berhak dan hasilnya *gateway* yang tidak berhak tersebut mendapatkan data yang dikirimkan. Sehingga perlu dilakukan upaya yang mencangkup keamanan pada penelitian ini menggunakan metode *confidentiality.*

Pada konsep *confidentiality* ini akan dilakukan enkripsi pada data atau pesan yang akan dikirimkan dari node kemudian ke *gateway*. Berdasarkan penelitian sebelumnya yang dilakukan oleh Arrijuddin (2019) yaitu melakukan pengiriman data menggunakan modul komunikasi LoRa, peneliti sebelumnya keubelum menerapkan keamanan pada LoRa. Sehingga diperlukan keamanan pada modul komunikasi LoRa dengan menggunakan algoritma kriptografi. Salah satunya yaitu penelitian yang dilkakukan oleh Semiconductor (2018) mengenai keamanan pada LoRa yaitu peneliti menerapkan enkripsi menggunakan algoritma AES. Namun telah ditemukan celah keamanan pada AES yang dapat berpotensi untuk dilakukan serangan.

Berdasarkan kelemahan yang ditemukan, maka di perlukan slolusi keamanan terbaru yaitu dengan menggunakan algoritme SPECK. Algoritme SPECK memenuhi salah satu dari aspek keamanan informasi dalam hal *confidentiality* dengan melakukan enkripsi dan dekripsi untuk mengamankan data. Dalam penelitian ini algoritme SPECK 128/128 bit digunakan sebagai metode untuk melakukan enkripsi pada data sensor yang dikirimkan dari node ke *gateway*. Pada *gateway* akan dilakukan proses dekripsi kemudian menampilkan data berupa *plaintext*

2. LANDASAN KEPUSTAKAAN

Pada penelitian yang dilakukan oleh Arijjudin melakukan penelitian mengenai pengembangan sistem perantara pengiriman data menggunakan modul komunikasi LoRa dan protokol MQTT pada WSN (*Wireless Sensor Network).* Pada penelitian tersebut didapatkan hasil yaitu sistem pada node sensor mengambil data dari sensor kemudian dikirimkan ke *gateway* dengan modul komunikasi LoRa. Dari hasil yang didapatkan oleh Arijjudin, terdapat aspek yang belum dilengkapi dalam melakukan proses komunikasi pada LoRa. Aspek tersebut terkait dengan aspek keamanan data yang dikirimkan.

Penelitian selanjutnya dilakukan oleh Semiconductor yaitu melakukan keamanan pada perangkat IoT dengan menggunakan tekonologi LoRa. Upaya keamanan yang dilakukan menggunakan algoritme AES-128 untuk melakukan enkripsi dan dekripsi. Peneliti memilih untuk menggunakan algoritme SPECK 128/128 bit dikarenakan pada algoritme AES-128 ditemukan kerentanan yang dapat menjadi celah untuk *attacker* Referensi penelitian selanjutnya yang dilakukan oleh Beaulieu mengenai algoritme SPECK dan SIMON sebagai algoritme *block cipher* yang ringan. Tujuan dari penelitian ini yaitu algoritme SPECK dan SIMON dapat bekerja secara baik pada spectrum penuh di *platform* yang terbatas sehingga peneliti menggunakan komponen yang sederhana.

2.1 Kriptografi

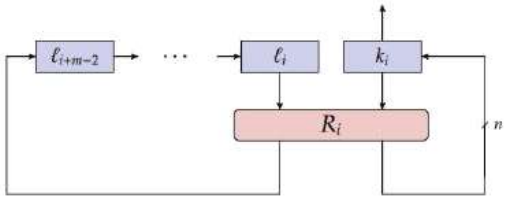
Kriptografi merupakan ilmu yang mempelajari mengenai bagaimana keamanan dan kerahasiaan suatu pesan terjaga saat dikirimkan dari tempat asal atau tempat pengirim ke tempat tujuan atau tempat penerima. Kriptografi juga dapat dikatakan sebagai sebuah cara untuk mengamankan data dengan menggunakan metode matematika (Candra, 2016)

1. *Confidentiality*, menjaga suatu informasi dari orang-orang yang tidak berhak atas informasi tersebut,
2. *Integrity*, informasi tidak boleh dirubah tanpa ijin,
3. Authentication, merupakan suatu kemampuan untuk mengonfirmasi bahwa data tersebut adalah asli, tidak palsu.

Teknik enkripsi adalah teknik yang dapat mencapai salah satu konsep yaitu *confidentiality*. Enkripsi merupakan teknik menujaga kerahasiaan pesan dengan mengubah pesan asli (*Plaintext*) menjadi pesan yang telah menjadi pesan rahasia (*Chipertext*).

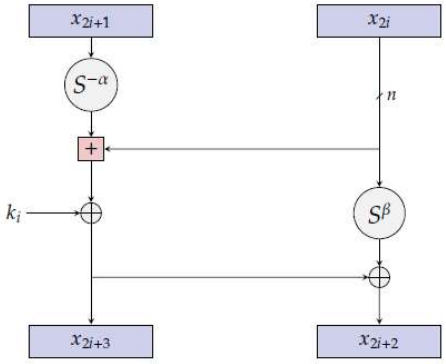
2.2 Algoritme SPECK

SPECK adalah keluarga blockcipher ringan yang di rilis oleh Badan Keamanan Nasional A.S. pada tahun 2013. Keluarga SPECK terdiri dari 10 versi, mendukung berbagai ukuran blok dan kunci (Mohanty and Mohanty, 2014). Salah satu contohnya adalah SPECK 128/128bit yang berarti versi block chiper SPECK dengan ukuran blok 128 bit dan ukuran *key* 128bit. Pada proses keamanan SPECK 128/128bit akan melalui beberapa proses yaitu *key scheduling,* enkripsi, dan dekripsi. Pertama yang dilakukan adalah proses *key scheduling.*



Gambar 1. Alur *Skematis Key Scheduling*

Pada versi 128/128bit dibutuhkan 2 *key* sepanjang 16bit masing-masing *key.* Selanjutnya akan dilakukan proses *key scheduling* sebanyak round sesuai versi yang dipilih. Pada penelitian ini proses akan diulang sebanyak 32kali. Proses pertama yang dilakukan adalah *key 1* masuk kemudian dilakukan *shifting* ke kanan sebanyak *alpha* kemudian proses selanjutnya adalah penambahan modulo dengan *key 2.* Selanjutnya melakukan XOR dengan key yang dicari. Untuk *key 2* proses selanjutnya adalah dilakukan *shifting* ke kiri sebanyak *beta*. Hasilnya akan dilakuakn XOR dengan hasil XOR *key 1* sebelumnya. Selanjutnya adalah proses enkripsi.



Gambar 2. Alur *Skematis* pada Fungsi Round

Gambar diatas merupakan gambar alur pada algoritme SPECK 128/128bit. Dibutuhkan 2 *plaintext.* Selanjutnya adalah *plaintext 1* akan dilakukan *shifting* sebanyak *alpha* ke arah kanan kemudian hasilnya akan dilakukan penambahan modulo dengan *plaintext 2.* Selanjutnya hasilnya akan di XORkan dengan *key* yang didapatkan pada round yang dicari. Selanjutnya untuk *plaintext 2* dilakukan *shifting* sebanyak beta kearah kiri. Hasilnya akan di XORkan dengan hasil proses pada *plaintext 1.* Selanjutnya proses tersebut dilakukan sebanyak round pada versi yang dipilih. Proses yang terakhir adalah dekripsi. Pada proses dekripsi prosesnya adalah kebalikan dari proses enkripsi.

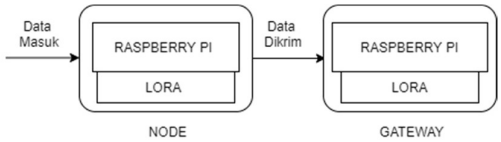
2.3 LoRa (Long Range)

LoRa adalah teknik modulasi yang memungkinkan transfer informasi jarak jauh dengan kecepatan transfer rendah. Modulasi LoRa telah dipatenkan oleh Semtech Corporation. LoRa adalah jenis modulasi Spektrum SS-Spread, dan teknik ini terdiri dari penggunaan sinyal yang bervariasi secara konstan dengan frekuensi. Keuntungan menggunakan metode ini adalah bahwa waktu dan frekuensi offset

3. PERANCANGAN

3.1 Perancangan Sistem

Dalam tahap perancangan, peneliti melakukan perancangan yang meliputi alur sistem yang dibuat serta *library* yang digunakan. Perancangan penelitian bersifat implementatif, data yang digunakan menggunakan data yang berasal dari sensor. Perancangan yang akan dilakukan berupa perancangan umum, perancangan node dan *gateway* secara umum sebelum ditambahkan algoritme. Setelah memasang perangkat yang dibutuhkan, semua peramgkat LoRa akan diatur mememiliki frekuensi yang sama agar dapat berkomunikasi.

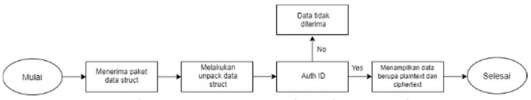


Gambar 3.1 Rancang Alur Sistem

Pada gambar 3.1 merupakan rancang alur sistem. Pada rancangan alur sistem diatas yaitu alur sistem secara umum. Alur sistem yang akan dijelaskan belum disisipi proses keamanan pada data yang dikirimkan antara node dan *gateway.* Proses pertama adalah node menerima sebuah data. Data yang masuk tersebut berupa data suhu dan kelembapan ruang pada tempat sistem dijalankan. Setelah itu akan melakukan *pack* data yang berisi data yang diterima oleh node dan ID. Sebelumnya antara node dan *gateway* akan diberikan kode ID yang sama sebagai penanda bahwa node berkomunikai dengan *gateway* yang sah. Selanjutnya pesan tersebut akan dikirimkan ke *gateway.* Selanjutnya *gateway* menerima pesan yang dikirimkan oleh node. Pesan yang sampai akan di lakukan *unpack* oleh *gateway* dan kemudian tampilkan pada *gateway.*

Gambar 3.2 Rancangan Alur Sistem Pada Node

Gambar 3.2 merupakan gambar rancang alur sistem pada node. Pertama node akan menerima data sensor suhu dan kelembapan. Pada node memiliki ID untuk mencegah node mengirimkan informasi ke *gateway* yang tidak sah. Kemudian data akan ditampilkan pada node. Selanjutnya node akan melakukan *pack* yang berisi ID dan pesan dalam bentuk *struct* dan dikirimkan ke *gateway.* Selanjutnya akan dilakukan proses pada *gateway,*

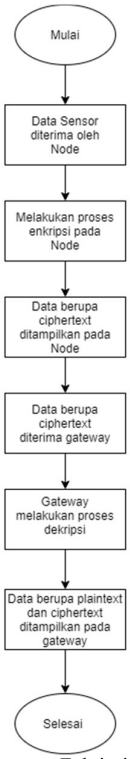


Gambar 3. 3 Rancang Alur Sistem Pada *Gateway*

Proses pertama yaitu *gateway* akan menerima paket data berupa struct yang dikirimkan oleh node. Proses selanjutnya adalah *gateway* melakukan *unpack* data berupa struct tersebut untuk nantinya ditampilkan pada *gateway*. Setelah *unpack* data, sistem akan melakukan pengecekan pada ID atau melakukan proses autentikasi ID. Apabila ID yang diterima berbeda atau salah, maka *gateway* tersebut bukanlah *gateway* yang sah. Gateway yang tidak sah ini merupakan *gateway* yang ditambahkan sebagai penyerang. Sehingga, data tidak akan diterima oleh *gateway* tersebut. Pada *gateway* yang tidak sah, akan dilakukan pengecekan ID yang digunakan oleh node dan *gateway* yang sah hal tersebut akan dijelaskan lebih lanjut pada perancangan pengujian. *Gateway* yang tidak sah, akan menampilkan data yang dikirimkan oleh node.

3.1 Perancangan Keamanan

Perancangan keamanan dilakukan dengan proses enkripsi pada data yang berasal dari node yaitu suhu dan kelembapan. Enkripsi dilakukan dengan algoritme SPECK 128/128bit dengan menggunakan metode padding untuk melakukan enkripsi data untuk mencukupi kebutuhan panjang data pada algoritme SPECK 128/128bit. Pada proses enkripsi, dibutuhkan kunci untuk mengenerate *plaintext* atau data asli. Peneliti menggunakan versi SPECK dengan spesifikasi *block size* 128 dan *key size* 128. Pada algoritme SPECK 128/128bit memiliki sebanyak 32 round dan membutuhkan *key* word sebanyak 2.



Gambar 3. 5 Perancangan Enkripsi Algoritme SPECK 128/128bit

Pada gambar menunjukkan perancangan enkripsi pada algoritme SPECK 128/128bit. Tahapan dimulai dengan data suhu dan kelembapan masuk kemudian data tersebut dilakukan proses enkripsi hingga menghasilkan *ciphertext*. Data berupa *ciphertext* kemudian dilanjutkan dengan proses dekripsi untuk menhasilkan data aslinya atau *plaintext.*

3.2 Perancangan Lingkungan Uji

Rancangan ini juga digunakan untuk merepresentasikan bagaimana Algoritme SPECK 128/128bit dapat melakukan enkripsi pada modul komunikasi LoRa. Pertama yang harus diketahui adalah bagaimana modul komunikasi LoRa sendiri dapat berkomunikasi. Terdapat dua perangkat yang memiliki peran berbeda. Untuk perangkat pertama berperan sebagai node dan perangkat yang kedua sebagai gateway. Node berperan untuk mengirimkan data pada gateway dan gateway menerima data dari node. Kedua perangkat memiliki fungsinya sendiri dalam pengimplementasian algoritme SPECK 128/128bit ini.

Perangkat pertama adalah node yang juga memiliki fungsi untuk menerima data berupa suhu dan kelembapan. Data berupa suhu dan kelembapan tersebut kemudian akan dilakukan proses enkripsi menggunakan algoritme SPECK 128/128bit. Pada sistem ini, proses enkripsi akan dijalankan di node. Sehingga ketika data masuk node akan melakukan proses key scheduling kemudian data hasil enkripsi

dari node yaitu data dengan bentuk ciphertext, akan di kirimkan ke gateway. Selanjutnya adalah perangkat gateway yang disisipi proses dekripsi. Proses ini dilakukan untuk membaca data yang di kirimkan oleh node. Setelah node mengirimkan pesan berupa struct, pesan tersebut kemudian dilakukan porses unpack pesan tersebut. Pesan yang diterima oleh gateway berisi ID dan data dengan bentuk ciphertext. Selanjutnya data tersebut dilakuakn proses dekripsi untuk mendapatkan data aslinya. Data yang telah dilakukan proses dekripsi akan kembali menjadi bentuk aslinya atau kembali menjadi plaintext. Apabila algoritme SPECK 128/128bit berhasil diterapkan maka penyerang hanya akan mendapatkan data berupa ciphertext tanpa mengetahui bentuk asli atau plaintextnya.

3.3 Perancangan Pengujian

Pada penelitian ini terdapat beberapa pengujian diantaranya meliputi test vector dan pengujian keamanan yang meliputi pengujian aktif dan pasif. Pengujian kinerja dilakukan dengan membandingkan algoritme yang digunakan yaitu algoritme SPECK 128/128bit dengan AES-128 yang merupakan algoritme yang digunakan pada penelitian sebelumnya. Waktu yang akan dibandingkan adalah saat proses enkripsi dan proses dekripsi. Selanjutnya adalah pengujian *Test vector*. *Test vector* merupakan pengujian yang dilakukan untuk memastikan apakah algoritme yang dijalankan sesuai dengan paper rujukan dalam hal ini yaitu paper algoritme SPECK.

Selanjutnya adalah pengujian pasif yaitu serangan yang tidak menyisipkan data pada aliran data, tetapi hanya mengamati atau memonitor pengiriman informasi ke tujuan. Pada penelitian ini, sistem yang telah diberi keamanan akan dilakukan penyerangan berupa serangan *sniffing*. Pengujian yang terakhir dilakukan adalah pengujian aktif. Pada pengujian aktif yaitu penyerang mencoba untuk melakukan modifikasi data yang dikirim. Pada penelitian ini, pengujian aktif dilakukan dengan melakukan modifikasi terhadap *plaintext*.

3.5 Pengujian

Pengujian yang pertama dilakukan adalah *test vector* yaitu pengujian pada validasi Algoritme SPECK 128/128bit itu sendiri apakah sudah valid dan benar atau masih menampilkan hasil yang berbeda dengan referensi asli. Pengujian selanjutnya yaitu pengujian dengan serangan pasif. Pengujian pasif dilakukan dengan melakukan serangan berupa *sniffing* yang dilakukan dengan menyerang *gateway* untuk mendapatkan data berupa *plaintext*. Pengujian yang terakhir adalah pengujian serangan aktif menggunakan metode Known *Plaintext* Attack. Pada serangan aktif penyerang akan mencoba untuk melakukan modifikasi informasi atau data yang dikirimkan

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Pengujian

4.1.1 Pengujian Test Vector

Mekanisme dalam melakukan *test vector* yaitu dengan menjalankan algoritme SPECK 128/128bit dengan masukan *plaintext* dan *key* sesuai dengan referensi jurnal original algoritme SPECK. Sehingga keluaran dari algoritme haruslah memiliki nilai yang sama dengan nilai pada *test vector* jurnal rujukan. Keluaran yang ditampilkan harus memiliki nilai yang sama sebagai validasi bahwa tidak adanya kesalahan dalam pengimplementasian algoritme. Berikut adalah tabel *test vector* algoritme SPECK 128/128bit.

Key	0f0e0d0c0b0a0908	0706050403020100
Plaintext	6c61766975716520	7469206564616d20
Ciphertext	a65d985179783265	7860fedf5c570d18

Tabel 4. Test Vector pada Algoritme SPECK 128/128bit

4.1.2 Pengujian Pasif

Pengujian pasif dilakukan untuk melakukan pengujian pada efisiensi algoritme SPECK 128/128bit dalam melakukan keamanan pada sistem yang telah dibangun. Pengujian pasif pada penelitian ini menggunakan metode serangan *sniffing.* Pada pengujian pasif, akan diberikan skenario berupa penambahan *client* sebagai *gateway* yang tidak sah atau pada gambar dinamakan *gateway* serang. Seperti pada penjelasan diatas bahwa modul LoRa akan mengirimkan atau melakukan *broadcast* ke modul LoRa dengan frekuensi yang sama. Serangan *sniffing* yaitu mendapatkan data yang dikirimkan oleh node ke *gatway* sah. Pengimplementasian algoritme SPECK 128/128bit disini memiliki fungsi agar penyerang tidak mendapatkan pesan asli yang dikirimkan

4.1.3 Pengujian Aktif

Serangan aktif merupakan serangan dengan memiliki tujuan untuk melakukan modifikasi pada data yang diserang. Pada penelitian ini, serangan aktif yang digunakan yaitu menggunakan serangan *XOR Known-Plaintext-Attack.* Pada penelitian ini, melakukan serangan aktif dengan menggunakan *ciphertext* hasil penyerangan pasif. Dengan beberapa informasi yang dimiliki oleh penyerang, penyerang melakukan proses *XOR Known-Plaintext-Attack* untuk menemukan beberapa probabilitas *key* yang mungkin digunakan oleh sistem yang diserang. Dengan *key* yang didapatkan oleh penyerang, maka akan dilakukan kembali proses untuk mengubah *ciphertext* yang didapatkan menjadi *plaintext.*

5. HASIL DAN PEMBAHASAN

5.1 Hasil Pengujian *Test Vector*

Pengujian test vector dilakukan sebagai validasi bahwa Algoritme SPECK 128/128bit telah berjalan dengan benar sesuai dengan fungsi yang digunakan pada algoritme SPECK. Masukkan dan *key* yang digunakan merujuk pada jurnal resmi peneliti algoritme yang digunakan. Hasil yang diharapkan setelah melakukan *test vector* yaitu hasil yang ditampilkan akan sama dengan *test vector* yanga ada pada jurnal rujukan. Apabila hasil yang ditampilkan tidak sesuai dengan *test vector* yang terdapat pada jurnal resmi, maka dapat dipastikan algoritme berjalan tidak semestinya dan menghasilkan data yang tidak valid.

```
Enkripsi      :
Dalam hex     : a65d985179783265        7860fedf5c570d18

Dalam decimal : 11987905258827821669    8674213117595946264

Dekripsi      :
Dalam hex     : 6c61766975716520        7469206564616d20

Dalam decimal : 7809653424151160096     8388271400802151712

---------------------------------------
```

5.3 Hasil Test Vector Algoritme SPECK 128/128bit

Dengan masukkan atau *plaintext* dan *key* yang sama persis dengan jurnal resmi, menghasilkan *ciphertext* yang panjangnya sesuai. Sehingga dapat dipastikan bahwa algoritme SPECK 128/128bit telah melakukan fungsinya dengan baik dan benar

5.2 Hasil Pengujian Serangan Pasif

Pengujian pasif dilakukan dengan menambahkan satu *gateway* baru yang berperan sebagai *gateway* yang tidak sah dengan skenario *gateway* yang tidak sah tersebut memiliki frekuensi yang sama. Selanjutnya node akan mengirimkan data yang diterima ke pada semua *gateway* baik *gateway* yang sah dan *gateway* yang tidak sah. Pada skenario pengujian pasif ini, *gateway* sistem akan dipasangkan metode autentikasi sederhana yaitu dengan menambahkan ID sederhana pada masing-masing node dan *gateway. Gateway* yang akan melakukan penyerangan, *gateway* tidak sah akan mencoba untuk melakukan *brute force* untuk melakukan pencarian ID yang telah diatur pada node dan *gateway* yang sah. Setelah *gateway* yang tidak sah mendapatkan ID yang sesuai maka *gateway* akan melakukan serangan *sniffing*

```
pi@raspberrypi:~/dianserang $ sudo python gwserang.py
RF95 LoRa mode ok, Let's Go!!
Subscribing topic ...
-------------ATTACKER-------------
---------loop and trial !!!------
0
-------------ATTACKER-------------
---------loop and trial !!!------
1
-------------ATTACKER-------------
---------loop and trial !!!------
2
------------!! ID DETECTED !!-------------
-------ID Detected | data ke | RSSI-------
(2, 3, '135902741466535781666580526671526358906')
-------------ATTACKER-------------
---------loop and trial !!!------
2
------------!! ID DETECTED !!-------------
```

Gambar 5.4 Hasil *Sniffing Gateway* Tidak Sah

Dapat dilihat pada gambar diatas, *gateway* yang tidak sah menampilkan ID yang ditemukan, nomor urut data yang masuk dan nilai *ciphertext* saja. Penyerang hanya akan mendapatkan hasil berupa data hasil enkripsi yaitu data berupa *ciphertext* dengan format data berupa bilangan desimal dengan panjang 128bit. Sehingga dapat dipastikan bahwa penyerang pada skenario penyerangan pasif ini adalah *gateway* yang tidak sah tidak berhasil mendapatkan informasi penting

5.3 Hasil Pengujian Serangan Aktif

Pengujian aktif pada penelitian ini menggunakan *Known Plaintext Attack*. Pada *Known Plaintext Attack* penyerang akan berusaha untuk memodifikasi *plaintext* yang akan dikirimkan ke tujuan yang sah. Syarat penyerang agar dapat mengimplementasikan serangan *Known Plaintext Attack* adalah penyerang harus memiliki *ciphertext* data yang akan dilakukan penyerangan, beberapa nilai *plaintext* dan asumsi *key* yang mungkin digunakan. Penyerang menerapkan serangan *Known Plaintext Attack* dengan melakukan pencarian *key* terlebih dahulu. Dengan memasukkan data yang didapatkan oleh penyerang berupa data *ciphertextnya, plaintext* dan asumsi panjang *key*. Sehingga dari *ciphertext* yang didapatkan itulah dilakukan percobaan untuk serangan *Known Plaintext Attack*

```
C:\Users\P A V I L I O N\Desktop\KPA>python find_key.py -i a65d985179783265 -o 7469206
564616d20 64
[+] I find the key ---> 0x3 0x11 0x69 0x1c 0x5c 0x5e 0x69 0x49 0xf 0x5c 0x6b 0x40 0x51
 0x56 0x6a 0x4d
```

Gambar 5.5 Hasil Serangan Known Plaintext Attack

Berdasarkan hasil yang ditampilkan setelah melakukan upaya untuk menemukan *key* yang digunakan dapat disimpulkan bahwa penyerang gagal melakukan serangan terhadap sistem. Hal ini dapat dilihat dari *key* yang ditemukan oleh penyerang memiliki nilai yang berbeda dengan *key* yang dipakai oleh sistem yang diserang. Pada sistem menggunakan *key* yaitu 0f0e0d0c0b0a0908 0706050403020100 nilai tersebut berbeda dengan nilai yang didapatkan dari serangan *known plaintext attack* yang telah dilakukan. Sehingga berdasarkan informasi yang didapatkan, data yang berupa *cipheretxtnya* tidak dapat dikembalikan ke bentuk aslinya atau bentuk *plaintextnya*

6. KESIMPULAN

Berdasarkan hasil dan analisa dari tahap perancangan, implementasi, dan pengujian yang telah dilakukan, maka untuk menjawab pertanyaan pada rumusan masalah, dapat disimpulkan bahwa :

1. Implementasi algoritme SPECK 128/128bit berhasil diterapkan dalam sistem komunikasi pada modul komunikasi LoRa

dengan menerapkan enkripsi dan dekripsi pada node dan gateway.

2. Setelah menerapkan keamanan dengan menggunakan algoritme SPECK 128/128bit, dapat dipastikan bahwa algoritme SPECK 128/128bit berhasil mengamankan komunikasi pada modul kominikasi LoRa. Hal tersebut dapat dibuktikan dengan pengujian yang dilakukan terhadap sistem dengan menggunakan metode serangan aktif dan serangan pasif. Serangan pasif berupa *sniffing* dan serangan aktif berupa serangan *Known Plaintext Attack*. Berdasarkan hasil yang didapatkan baik pengujian aktif maupun pasif, kedua serangan mendapatkan hasil gagal atau penyerang tidak berhasil mendapatkan informasi yang asli.

7. DAFTAR PUSTAKA

APRILIA, S., 2006. Crytographic Protocol. [online] pp.1–11. Available at: <http://en.wikipedia.org/wiki/Cryptographic_protocol>.

ARIJUDDIN, H., BHAWIYUGA, A. AND AMRON, K., 2019. Pengembangan Sistem Perantara Pengiriman Data Menggunakan Modul Komunikasi LoRa dan Protokol MQTT Pada Wireless Sensor Network. 3(2), pp.1655–1659.
Asriyanik, 2017. Studi Terhadap Advanced Encryption Standard (Aes) Dan.

BEAULIEU, R., SHORS, D., SMITH, J., TREATMAN-CLARK, S., WEEKS, B. AND WINGERS, L., 2015. The SIMON and SPECK lightweight block ciphers. *Proceedings of the 52nd Annual Design Automation Conference on - DAC '15*, [online] pp.1–6. Available at: <http://dl.acm.org/citation.cfm?doid=2744769.2747946>.

BISWAS, A.R. AND GIAFFREDA, R., 2014. IoT and cloud convergence: Opportunities and challenges. *2014 IEEE World Forum on Internet of Things (WF-IoT)*, [online] pp.375–376. Available at: <http://ieeexplore.ieee.org/document/6803194/>.

CANDRA, 2016. Keamanan Data Dengan Metode Kriptografi Kunci Publik. *Jurnal TIMES*, 2(2), pp.11–15.

DEVALAL, S. AND KARTHIKEYAN, A., 2018. LoRa Technology - An Overview. *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, [online] (Iceca), pp.284–290. Available at: <https://ieeexplore.ieee.org/document/8474715/>.

LAVRIC, A. AND POPA, V., 2017. Internet of Things and LoRa™ Low-Power Wide-Area Networks: A survey. *ISSCS 2017 - International Symposium on Signals, Circuits and Systems*.

LORA-ALLIANCE, 2015. A technical overview of LoRa ® and LoRaWAN ™ What is it? [online] (November). Available at: <https://lora-alliance.org/resource-hub/what-lorawantm>.

PURBA, Y., 2017. *Enkripsi*.

SEMICONDUCTORS, N.X.P., 2018. IoT Device Secure Connection with LoRa.

SONG, L., HUANG, Z. AND YANG, Q., 2016. Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9723, pp.379–394.