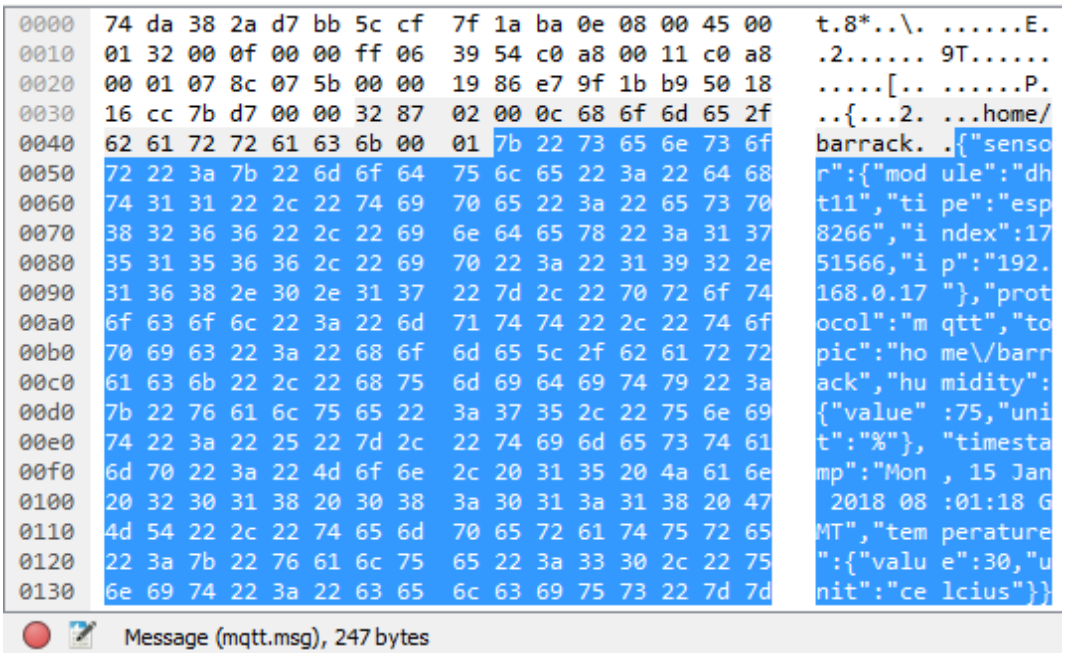
BAB 1 PEMBAHASAN

Berdasarkan pada hasil pengujian dan pengambilan data pada bab sebelumnya, dapat dibahas sebagai berikut:

1.1 Pembahasan Keamanan Data

1.1.1 Skenario 1

Pada pengujian pengiriman data skenario satu, yaitu pengiriman data dengan protokol MQTT tanpa mekanisme keamanan. Menggunakan salah satu teknik MITM yaitu *sniffing*. Dengan menggunakan *program* tcpdump untuk meng-*capture* data dari koneksi wlan0. Data hasil *capture* tcpdump kemudian diolah pada wireshark dan di-*filter* berdasar pada protokol MQTT kemudian didapatkan data *publish message* dan *publish ACK*. Dari *publish message* bisa dilihat data dari sensor nodemcu sebagai *publisher* pada topik *home/barrack* mengirimkan data seperti berikut:

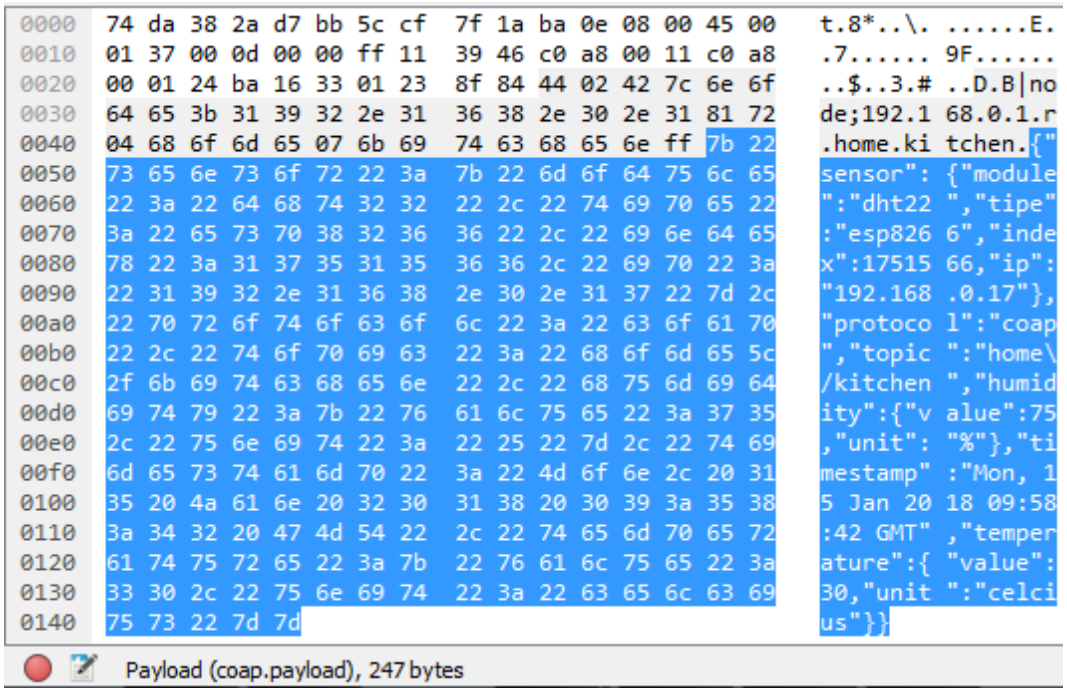


Gambar 1.1 Hasil sniffing protokol MQTT

Seperti yang terlihat pada gambar 6.1 bahwa data yang dikirim dengan protokol MQTT tanpa mekanisme keamanan mudah untuk dibaca oleh penyerang. Sehingga data bisa disalahgunakan dan merugikan korban.

1.1.2 Skenario 2

Pada pengujian pengiriman data skenario dua, yaitu pengiriman data dengan protokol CoAP tanpa mekanisme keamanan. Menggunakan salah satu teknik MITM yaitu *sniffing*. Dengan menggunakan program tcpdump untuk meng-*capture* data dari koneksi wlan0. Data hasil *capture* tcpdump kemudian diolah pada wireshark dan di-*filter* berdasar pada protokol CoAP kemudian didapatkan data CON dan ACK. Dari CON bisa dilihat data dari sensor nodemcu sebagai *publisher* pada topik *home/kitchen* mengirimkan data seperti berikut:

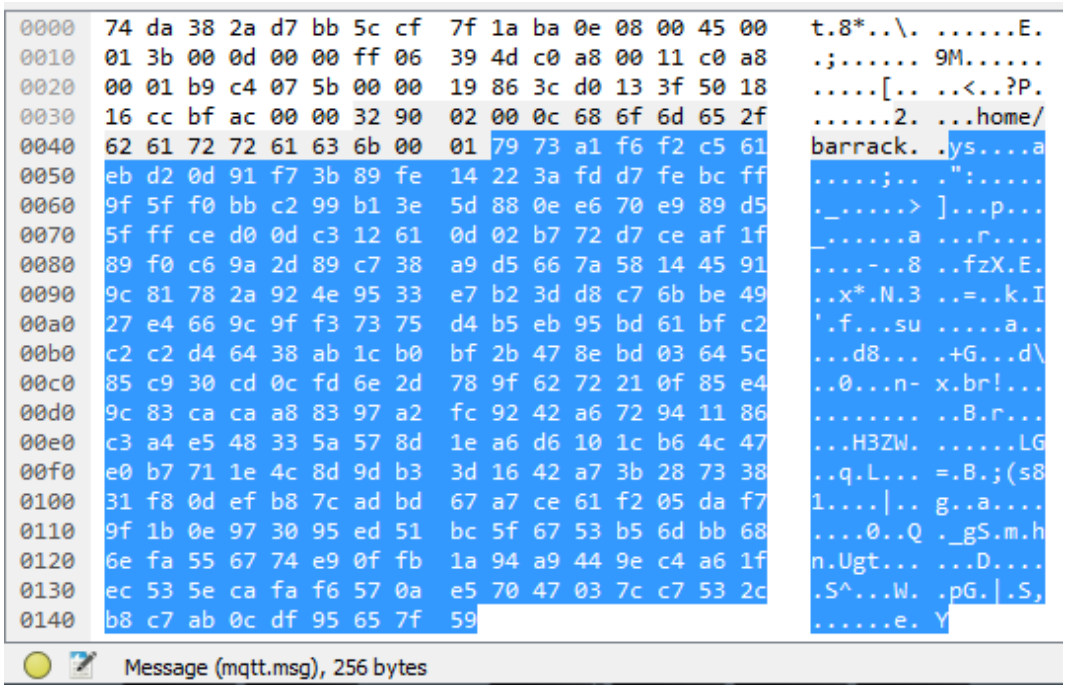


Gambar 1.2 Hasil Sniffing protokol CoAP

Seperti yang terlihat pada gambar 6.2 bahwa data yang dikirim dengan protokol CoAP tanpa mekanisme keamanan mudah untuk dibaca oleh penyerang. Sehingga data bisa disalahgunakan dan merugikan korban.

1.1.3 Skenario 3

Pada pengujian pengiriman data skenario tiga, yaitu pengiriman data dengan protokol MQTT dengan *crypto* AES-CBC 128 bits. Menggunakan salah satu teknik MITM yaitu *sniffing*. Dengan menggunakan *program* tcpdump untuk meng-*capture* data dari koneksi wlan0. Data hasil *capture* tcpdump kemudian diolah pada wireshark dan di-*filter* berdasar pada protokol MQTT kemudian didapatkan data *publish message* dan *publish ACK*. Dari *publish message* bisa dilihat data dari sensor nodemcu sebagai *publisher* pada topik *home/barrack* mengirimkan data seperti berikut:

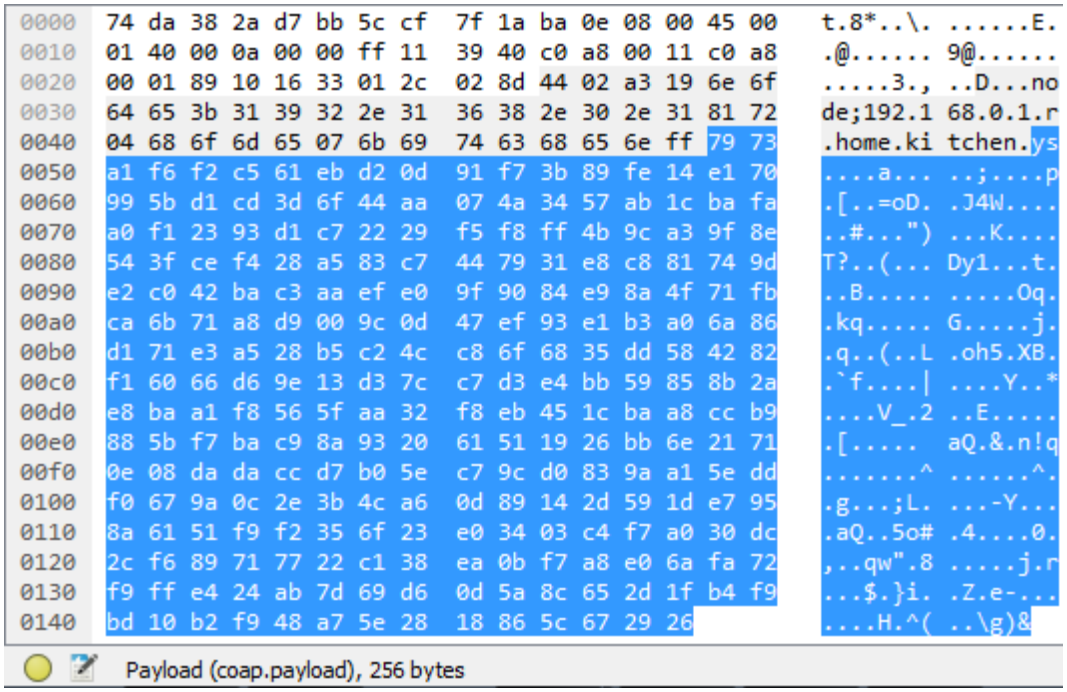


Gambar 1.3 Hasil sniffing MQTT crypto

Seperti yang terlihat pada gambar 6.3 bahwa data yang dikirim dengan protokol MQTT dengan *crypto* AES-CBC 128 bits sudah dienkripsi dan sulit dibaca oleh penyerang. Sehingga data telah aman dari segi *confidentiality*.

1.1.4 Skenario 4

Pada pengujian pengiriman data skenario empat, yaitu pengiriman data dengan protokol CoAP dengan *crypto* AES-CBC 128 bits. Menggunakan salah satu teknik MITM yaitu *sniffing*. Dengan menggunakan program tcpdump untuk meng-*capture* data dari koneksi wlan0. Data hasil *capture* tcpdump kemudian diolah pada wireshark dan di-*filter* berdasar pada protokol CoAP kemudian didapatkan data CON dan ACK. Dari CON bisa dilihat data dari sensor nodemcu sebagai publisher pada topik *home/kitchen* mengirimkan data seperti berikut:

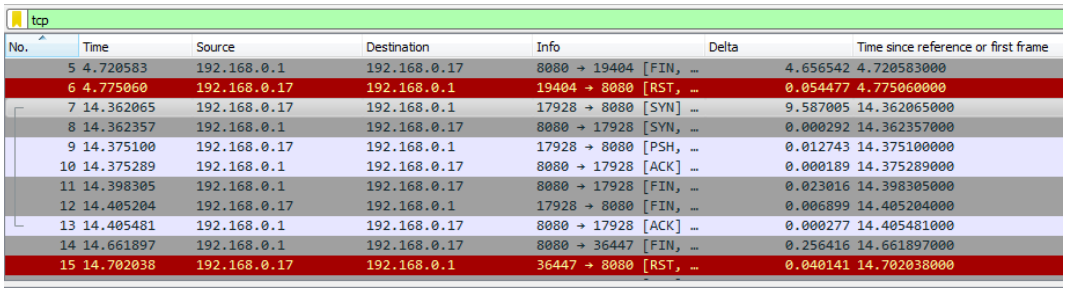


Gambar 1.4 Hasil sniffing CoAP crypto

Seperti yang terlihat pada gambar 6.4 bahwa data yang dikirim dengan protokol CoAP dengan *crypto* AES-CBC 128 bits sudah dienkripsi dan sulit dibaca oleh penyerang. Sehingga data telah aman dari segi *confidentiality*.

1.1.5 Skenario 5

Pada pengujian pengiriman data skenario lima, yaitu pengiriman data dengan protokol MQTT dengan TLS. Salah satu teknik MITM adalah *sniffing*. Dengan menggunakan program tcpdump untuk meng-*capture* data dari koneksi wlan0. Data hasil *capture* tcpdump kemudian diolah pada wireshark dan di-*filter* berdasar pada TCP kemudian didapatkan data sebagai berikut:

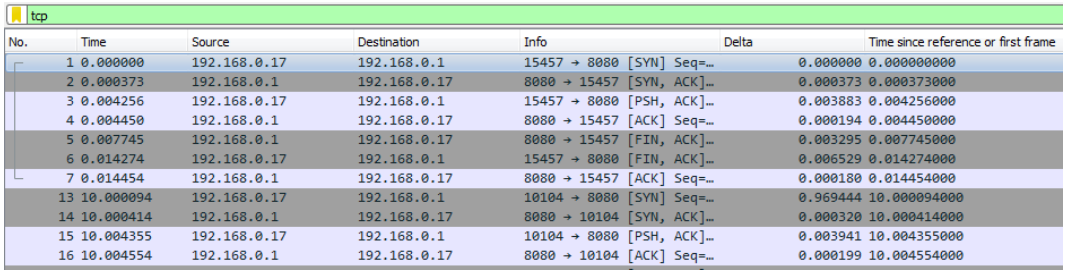


Gambar 1.5 Hasil sniffing MQTT TLS

Seperti yang terlihat pada gambar 6.5 bahwa sensor nodemcu dengan IP address 192.168.0.17 mencoba terhubung dengan server middleware dengan IP address 192.168.0.1. Tetapi tidak bisa terhubung sehingga proses pengiriman data tidak bisa terjadi sehingga implementasi TLS pada komunikasi protokol MQTT antara sensor nodemcu dengan middleware tidak berhasil.

1.1.6 Skenario 6

Pada pengujian pengiriman data skenario lima, yaitu pengiriman data dengan protocol CoAP dengan TLS. Salah satu teknik MITM adalah *sniffing*. Dengan menggunakan program tcpdump untuk meng-*capture* data dari koneksi wlan0. Data hasil *capture* tcpdump kemudian diolah pada wireshark dan di-*filter* berdasar pada TCP kemudian didapatkan data sebagai berikut:



Gambar 1.6 Hasil sniffing CoAP TLS

Seperti yang terlihat pada gambar 6.6 bahwa sensor nodemcu dengan IP address 192.168.0.17 mencoba terhubung dengan server middleware dengan IP address 192.168.0.1. Tetapi tidak bisa terhubung sehingga proses pengiriman data tidak bisa terjadi sehingga implementasi TLS pada komunikasi protokol CoAP antara sensor nodemcu dengan *middleware* tidak berhasil.

1.2 Pembahasan Pengiriman Data

1.2.1 Skenario 1

Pada pengujian skenario satu, yaitu pengiriman data dengan protokol MQTT tanpa menggunakan mekanisme keamanan didapat data hasil pengujian sebagai berikut:

Tabel 1.1 Hasil pengujian skenario 1

MQTT					
Expected	Actual	Success rate	Packet loss rate	Delay	Jitter
20	16	80 %	20 %	0.017583	0.00051
20	16	80 %	20 %	0.018451	0.0000094
20	16	80 %	20 %	0.017796	0.00077
20	16	80 %	20 %	0.01959	0.000052
20	16	80 %	20 %	0.016777	0.000035
Average (detik)				0.018039	0.000269

Pada tabel 6.1 terlihat hasil pengujian pengiriman data dengan protokol MQTT dari sensor nodeMcu ke IoT *middleware* tanpa mekanisme keamanan. Dimana rata-rata data terkirim adalah sebanyak 16 data dengan *success rate* 80 % dan *packet loss* 20% dari 4 data. Nilai rata-rata *delay* dari lima kali pengujian adalah 0.018039 detik dan dengan *jitter* adalah 0.000269 detik.

1.2.2 Skenario 2

Pada pengujian skenario dua, yaitu pengiriman data dengan protokol CoAP tanpa menggunakan mekanisme keamanan didapat data hasil pengujian sebagai berikut:

Tabel 1.2 Hasil pengujian skenario 2

CoAP					
Expected	Actual	Success rate	Packet loss rate	Delay	Jitter
20	19	95 %	5 %	0.023074	0.00042
20	19	95 %	5 %	0.019547	0.00081
20	19	95 %	5 %	0.019323	0.00064
20	19	95 %	5 %	0.016898	0.00065
20	18	90 %	10 %	0.019681	0.0008
Average (detik)				0.019705	0.000664

Pada tabel 6.2 terlihat hasil pengujian pengiriman data dengan protokol CoAP dari sensor nodeMcu ke IoT *middleware* tanpa mekanisme keamanan. Dimana rata-rata data terkirim adalah sebanyak 18,8 data dengan *success rate* 94 % dan *packet loss* 6% dari 1,2 data. Nilai rata-rata *delay* dari lima kali pengujian adalah 0.019705 detik dan dengan *jitter* adalah 0.000664 detik.

1.2.3 Skenario 3

Pada pengujian skenario tiga, yaitu pengiriman data dengan protokol MQTT dengan *crypto* AES-CBC 128 bits didapat data hasil pengujian sebagai berikut:

Tabel 1.3 Hasil pengujian skenario 3

MQTT Crypto					
Expected	Actual	Success rate	Packet loss rate	Delay	Jitter
20	16	80 %	20 %	0.017347	0.00094
20	16	80 %	20 %	0.017922	0.00081
20	12	60 %	40 %	0.019189	0.0000032
20	16	80 %	20 %	0.014171	0.000014
20	16	80 %	20 %	0.01651	0.000188
Average (detik)				0.0170278	0.0003894

Pada tabel 6.3 terlihat hasil pengujian pengiriman data dengan protokol MQTT dari sensor nodeMcu ke IoT *middleware* dengan *crypto* AES-CBC 128 bits. Dimana rata-rata data terkirim adalah sebanyak 15,2 data dengan *success rate* 76 % dan *packet loss* 24% dari 4,8 data. Nilai rata-rata *delay* dari lima kali pengujian adalah 0.0170278 detik dan dengan *jitter* adalah 0.0003894 detik.

1.2.4 Skenario 4

Pada pengujian skenario empat, yaitu pengiriman data dengan protokol MQTT dengan *crypto* AES-CBC 128 bits didapat data hasil pengujian sebagai berikut:

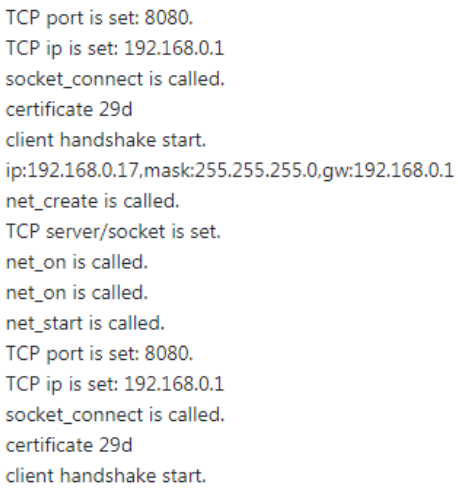
Tabel 1.4 Hasil pengujian skenario 4

CoAP Crypto					
Expected	Actual	Success rate	Packet loss rate	Delay	Jitter
20	19	95 %	5 %	0.026884	0.00061
20	19	95 %	5 %	0.020728	0.00082
20	20	100 %	0 %	0.020763	0.00083
20	19	95 %	5 %	0.023594	0.00071
20	19	95 %	5 %	0.024683	0.00074
Average (detik)				0.02333	0.000742

Pada tabel 6.4 terlihat hasil pengujian pengiriman data dengan protokol CoAP dari sensor nodeMcu ke IoT *middleware* dengan *crypto* AES-CBC 128 bits. Dimana rata-rata data terkirim adalah sebanyak 19,2 data dengan *success rate* 96 % dan *packet loss* 4% dari 0,8 data. Nilai rata-rata *delay* dari lima kali pengujian adalah 0.02333 detik dan dengan *jitter* adalah 0.000742 detik.

1.2.5 Skenario 5

Pada pengujian skenario lima, yaitu pengiriman data dengan protokol MQTT dengan TLS. Data tidak berhasil dikirimkan karena client handshake gagal terjadi seperti pada gambar berikut:



Gambar 1.7 Pengiriman data MQTT TLS pada ESPlorer

Terlihat pada gambar 6.7 sertifikat untuk verifikasi TLS sudah tersedia, namun koneksi antara sensor nodemcu ESP8226 dengan server middleware tidak berhasil. Ini dibuktikan dengan client handshake start yang terus berulang. Yang artinya server tidak menerima handshake dari klien atau sensor dan berhenti pada tahap client hello.

1.2.6 Skenario 6

Pada pengujian skenario enam, yaitu pengiriman data dengan protokol CoAP dengan TLS. Data tidak berhasil dikirimkan karena client handshake gagal terjadi seperti pada gambar berikut:

```
Connected to CoAP
URL: coap://192.168.0.1:5683/r/home/kitchen
Client ID: 1751566
pm open,type:2 0
client handshake start.
espconn_mbedtls.c 652, type[certificate],length[669]
please start sntp first !
client handshake failed!
Reason:[-0x7880]
```

Gambar 1.8 Pengiriman data CoAP TLS pada ESPlorer

Terlihat pada gambar 6.8 sertifikat untuk verifikasi TLS sudah tersedia, namun koneksi antara sensor nodemcu ESP8226 dengan server middleware tidak berhasil. Ini dibuktikan dengan client handshake failed yang terus berulang. Yang artinya server tidak menerima handshake dari klien atau sensor dan berhenti pada tahap client hello. Dan alasan dari *client handshake failed* adalah sebagai berikut:

```
...  77  #define MBEDTLS_ERR_SSL_PEER_CLOSE_NOTIFY
-0x7880  /**< The peer notified us that the connection is going to be closed. */
```

Gambar 1.9 mbedTLS error code

Pada gambar 6.9 dijelaskan bahwa peer SSL memberi pesan bahwa koneksi akan segera ditutup.

1.3 Perbandingan Hasil Pengujian

Tabel 1.5 Perbandingan hasil pengujian

Average			
Mekanisme	Packet loss	Delay	Jitter
MQTT	20 %	0.018039	0.000269
CoAP	6 %	0.019705	0.000664
MQTT-Cypto	24 %	0.0170278	0.0003894
CoAP-Crypto	4 %	0.02333	0.000742

Pada tabel 6.5 terlihat rata-rata packet loss antara tiap protokol sebelum dan sesudah ditambahkan crypto tidak berubah drastis. Pada protokol MQTT adalah 20% sebelum dan 24%

setelah ditambahkan crypto. Sedangkan pada protokol CoAP adalah 6% sebelum dan 4%. Packet loss bisa disebabkan oleh beberapa faktor, yaitu:

1. Sensor nodemcu ESP8266 terkadang berhenti berfungsi dan bisa mengirimkan data kembali setelah interval waktu sekitar 1-2 menit
2. Paket tidak bisa terkirim karena kondisi jaringan.

Nilai delay pada setiap protokol juga hampir tidak berubah. Pada protokol MQTT adalah 0.018039 detik dan 0.0170278 detik setelah ditambahkan crypto. Sedangkan pada protokol CoAP adalah 0.019705 detik sebelum dan 0.02333 detik.

Nilai jitter juga menunjukkan hasil yang serupa. Pada protokol MQTT adalah 0.000269 detik sebelum dan 0.0003894 detik setelah ditambahkan crypto. Dan protokol CoAP adalah 0.000664 detik sebelum dan 0.000742 detik sesudah.

Hal ini terjadi karena data payload yang berisi data temperature dan humindy dienkripsi terlebih pada payload baru kemudian dikirim ke middleware.