Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 2, April 2024, pp. 377-387

DOI: https://doi.org/10.52436/1.jutif.2024.5.2.1916
p-ISSN: 2723-3863
e-ISSN: 2723-3871

COMMUNICATION SECURITY IN THE MQTT PROTOCOL FOR MONITORING INTERNET OF THINGS DEVICES USING METHODS ELLIPTIC CURVE CRYPTOGRAPHY

Axel Natanael Salim[*1], Tata Sutabri[2], Edi Surya Negara[3], M Izman Herdiansyah[4]

[1,2,3,4]Faculty of Computer Science, Masters in Informatics Engineering, Universitas Bina Darma, Indonesia
Email: [1]axelsanti610@gmail.com, [2]tata.sutabri@binadarma.ac.id, [3]e.s.negara@binadarma.ac.id, [4]m.herdiansyah@bindarma.ac.id

(Article received: March 5, 2024; Revision: March 24, 2024; Published: April 04, 2024)

Abstract

The emergence of the IoT has become one of the most significant technology trends. The application of IoT is aimed at enhancing efficiency, comfort, and facilitating various human activities. One key aspect of IoT implementation is efficient communication between devices, with one of the most commonly used protocols being MQTT protocol. MQTT enables the transmission of data in real-time or based on specific events, although there are still several challenges that need to be addressed. One of the main challenges of MQTT is information security issues, prompting this research to examine effective solutions to enhance communication security in IoT applications that utilize MQTT protocol. One method of securing communication between IoT devices can involve using lightweight cryptographic communication security methods such as ECC method. ECC method is chosen because it utilizes shorter keys while still providing high security, making it more efficient when implemented on IoT devices. The results obtained indicate that data sent to MQTT Broker cannot be read and converted manually, ensuring much safer data transmission. Based on the test results, the tool can effectively read, process, and send data to MQTT Broker. QoS measurements on the system revealed that data encrypted and sent from the subscriber to MQTT Broker had an average delay time of 54.1 ms, throughput of 410.4 bps, zero packet loss, and jitter of 0.00 ms. Looking at the research findings, it can be concluded that this ECC method could serve as a solution to data communication security issues in the MQTT protocol.

Keywords: *DHT11, ECC, IoT, Secure Communication, MQTT Broker, Wemos D1 Mini ESP8266*.

KEAMANAN KOMUNIKASI PADA PROTOKOL MQTT UNTUK MONITORING PERANGKAT INTERNET OF THINGS DENGAN METODE ELLIPTIC CURVE CRYPTOGRAPHY

Abstrak

Muculnya Internet of Things menjadi salah satu tren teknologi yang paling signifikan. Penerapan dari IoT ini untuk meningkatkan efisiensi, kenyamanan serta mempermudah manusia dalam melakukan beberapa aktifitas. Salah satu aspek kunci dalam pelaksanaan IoT adalah komunikasi yang efisien antara perangkat-perangkat tersebut, dan salah satu protokol yang paling umum digunakan dalam komunikasi antar perangkat adalah protokol Message Queuing Telemetry Transport. MQTT memungkinkan pengiriman data secara *real-time* atau berdasarkan peristiwa tertentu, namun masih terdapat beberapa tantangan yang perlu diatasi. Salah satu tantangan utama MQTT adalah masalah keamanan informasi, sehingga penelitian ini bertujuan untuk mengkaji solusi yang efektif untuk meningkatkan keamanan komunikasi dalam penggunaan IoT yang menggunakan protokol MQTT. Salah satu metode kemanan komunikasi antar perangkat IoT dapat menggunakan metode pengamanan komunikasi kriptografi yang ringan seperti metode ECC. Metode ECC digunakan karena menggunakan kunci yang lebih pendek tetapi tetap memberikan keamanan yang tinggi sehingga lebih efisien jika diimplementasikan pada perangkat IoT. Hasil yang didapat, data yang dikirimkan ke MQTT Broker tidak dapat dibaca dan dikonversi secara manual, sehingga data yang dikirim jauh lebih aman. Berdasarkan dari hasil pengujian, alat dapat bekerja dengan baik untuk membaca, memproses, dan mengirim data ke MQTT Broker. Pengukuran Qos pada sistem didapatkan bahwa data yang sudah dienkripsi dan dikirimkan dari subscriber ke MQTT Broker memiliki waktu rata-rata delay sebesar 54,1 ms, throughput 410,4 bps, packet loss sebesar 0% dan jitter sebesar 0,00 ms. Melihat dari hasil penelitian dapat disimpulkan bahwa metode ECC ini dapat menjadi solusi dari permasalah keamanan komunikasi data pada protokol MQTT.

Kata kunci: *DHT11, ECC, IoT, Keamanan Komunikasi, MQTT Broker, Wemos D1 Mini ESP8266*.

1. PENDAHULUAN

Seiring dengan kemajuan teknologi saat ini, muculnya *Internet of Things* menjadi salah satu tren teknologi yang paling signifikan dalam beberapa tahun terakhir. Penerapan dari *Internet of Things* ini bertujuan untuk meningkatkan efisiensi, kenyamanan serta mempermudah manusia dalam melakukan beberapa aktifitas [1], penerapan *Internet of Things* cocok pada digunakan pada sistem pengendalian terhadap hal-hal yang memerlukan kondisi tertentu agar tetap terjaga dan yang memerlukan pemantauan secara terus-menerus [2]. Penerapan *Internet of Things* akan membawa perubahan dalam berbagai bidang, termasuk pemantauan lingkungan, rumah pintar, manufaktur cerdas, dan layanan kesehatan berbasis *Internet of Things* [3]. Salah satu aspek kunci dalam pelaksanaan Internet *of Things* adalah komunikasi yang efisien antara perangkat-perangkat tersebut, dan salah satu protokol yang paling umum digunakan dalam komunikasi antar perangkat adalah protokol *Message Queuing Telemetry Transport.*

Protokol *Message Queuing Telemetry Transport* adalah protokol berbasis pesan yang dirancang khusus untuk lingkungan *Internet of Things* [4]. Kelebihan utama dari *Message Queuing Telemetry Transport* adalah kemampuannya untuk menyediakan komunikasi yang baik, ringan, dan efisien antara perangkat *Internet of Things* [5][6]. *Messsage Queuing Telemtry Transport* merupakan pilihan yang tepat untuk menjadi protokol *Internet of Things* karena *Message Queuing Telemetry trabsport* bersifat *Light Weighted Message* [7] dan dirancang untuk perangkat yang memiliki sumber daya terbatas [8]. Hal ini memungkinkan pengiriman data secara *real-time* atau berdasarkan peristiwa tertentu. Namun, meskipun *Message Queuing Telemetry Transport* memiliki banyak keunggulan, masih terdapat beberapa tantangan yang perlu diatasi.

Menurut [9], Salah satu tantangan utama *Message Queuing Telemetry Transport* adalah masalah keamanan informasi, terutama ketika data yang dikirimkan melibatkan informasi yang bersifat sensitif, beberapa protokol komunikasi pada perangkat *Internet of Things* belum memiliki mekanisme keamanan informasi yang menyeluruh [9]. Keamanan dalam komunikasi *Message Queuing Telemetry Transport* merupakan aspek kritis yang harus diperhatikan dalam mengimplementasikan *Internet of Things*, karena perangkat *Internet of Things* yang tidak aman dapat menjadi target potensial bagi serangan siber, salah satu metode yang dapat digunakan dalam mengamankan komunikasi antar perangkat, sehingga diperlukan metode pengamanan komunikasi seperti metode kriptografi yang ringan seperti metode *Elliptic Curve Cryptography* [10].

Eliptic Curve Cryptography adalah suatu algoritma kriptografi kunci publik yang mengaplikasikan persamaan kurva elliptik. Pendekatan ini dikembangkan dan diusulkan oleh Neal Koblitz dan Victor S. Miller [11]. Hal ini disebabkan oleh kemampuannya menggunakan kunci yang lebih pendek [12], walaupun menggunakan kunci yang lebih pendek, tetapi masih menyediakan tingkat keamanan yang setara dengan metode asimetrik lainnya yang menggunakan kunci yang lebih panjang. Dengan menggunakan kunci yang lebih pendek dengan tingkat keamanan yang sama tinggi, implementasi *Eliptic Curve Cryptography* menjadi lebih efisien [13].

Penelitian serupa juga sudah pernah dilakukan oleh [13] dengan judul penelitian Implementasi Kriptografi dengan Metode *Ellictic Curve Cryptography* untuk Aplikasi *Chatting* Berbasis Android, penelitian ini mengimplementasikan *cryptography* dengan metode *Elliptic Curve Cryptography* untuk aplikasi *chatting* berbasis android [13], objek dari penelitian ini merupakan sebuah aplikasi *chatting* berbasis android, bebeda dengan objek penelitian yang akan dilakukan berupa sebuah perangkat *Internet of Things.* Penelitian lain yang juga melakukan pengamanan data pada protokol *Message Queuing Telemetry Transport* juga dilakukan oleh [14] dengan judul Implementasi *Transport Layer Security* Sebagai Metode Keamanan Protokol Jaringan pada *Message Queuing Telemetry Transport* Berbasis Raspberry PI, yang menggunakan *Transport Layer Security* sebagai protokol keamanan tambahan untuk mengamankan pesan atau komunikasi data didalam protokol *Message Queuing Telemetry Transport* [14][15], sedangkan penelitian yang akan dilakukan akan menggunakan metode *Elliptic Curve Cryptography* sebagai pengaman tambahan dalam mengamankan pesan atau komunikasi data dalam protokol *Message Queuing Telemetry Transport.*

Dalam penelitian ini, penulis menemukan bahwa penggunaan metode *Elliptic Curve Cryptography* untuk meningkatkan keamanan komunikasi data pada protokol *Message Queuing Telemetry Transport* pada perangkat *Internet of Things* memiliki potensi untuk mengatasai tantangan keamanan yang ditemui dalam literatur sebelumnya. Penelitian ini memberikan bukti empiris yang mendukung efektivitas metode *Elliptic Curve Cryptography* dalam mengamankan transmisi data pada perangkat *Internet of Things*, dengan menunjukkan bahwa data yang dienkripsi dengan menggunakan metode *Elliptic Curve Cryptography* tidak dapat dibaca secara manual, serta menawarkan kinerja yang memuaskan yang dapat dilihat dari

pengukuran *Quality of Service*. Penemuan ini berpotensi memberikan kontribusi pada pengembangan perangkat *Internet of Things* yang lebih aman, serta mendorong penggunaan metode enkripsi yang lebih efisien dalam konteks *Internet of Things*.

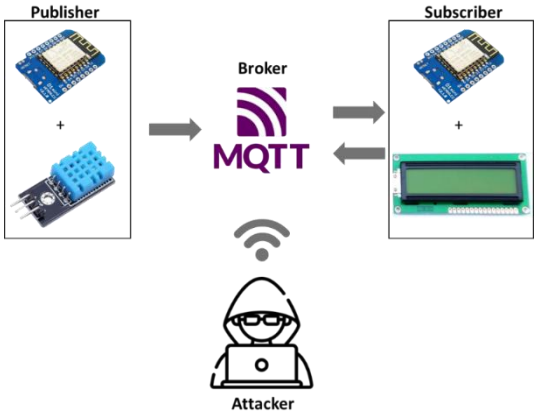
Berdasarkan latar belakang yang telah dijelaskan, maka penelitian ini bertujuan untuk mengkaji solusi yang efektif untuk meningkatkan keamanan komunikasi dalam penggunaan *Internet of Things* yang menggunakan protokol *Message Queuing Telemetry Transport.*

2. METODE PENELITIAN

Penelitian ini menggunakan metode ekperimental dengan mensimulasikan Sistem *Internet of Things* yang dapat membaca data suhu dan kelembaban ruangan. Tahapan-tahapan pada metode peneltitian ini yaitu, pertama, implementasi sistem, pada implementasi sistem dilakukan analisa butuhan seperti mikrokontroler yang digunakan, perangkat *Internet of Things* yang akan digunakan, dan library apa yang cocok sebagai penghubung antara mikrokontroler dengan perangkat *Internet of Things* agar dapat saling berkomunikasi. Kedua, proses *Elliptic Curve Cryptography*, pada proses *Elliptic Curve Cryptography* akan mengkaji dari awal pembuatan *public key* dan *private key* hingga data *plaintext* menjadi *ciphertext* dan *ciphertext* kembali menjadi *plaintext*. Ketiga, tahapan pengujian, pada tahap pengujian akan dilakukan pengujian sistem dengan beberapa pengujian meliputi pengujian performa menggunakan pengujian *Quality of Service*, dan pengujian dari sisi keamanan.

Pada penelitian ini, dibutuhkan perangkat keras untuk mengumpulkan data dan mengimplementasi algoritma *Eliptic Curve Cryptography* ini. Kebutuhan perangkat keras pada penelitian ini yaiut, sensor DHT11 untuk membaca suhu dan kelembaban, Wemos D1 mini ESP8266 sebagai mikrokontroler, kabel jumper untuk menghubungkan antara sensor dengan mikrokontroler, serta laptop sebagai *router* wifi. Kebutuhan perangkat lunak untuk pengembangan sistem *Internet of Things* menggunakan sistem operasi Windows, Ardino IDE sebagai editor untuk menuliskan baris kode perintah, *Message Queuing Telemetry Transport broker* sebagai jembatan atau penguhubung antara *publisher* dan *subscriber*, dan wireshark sebagai program untuk mengujian keamanan dengan proses *sniffing* paket data yang dikirimkan.

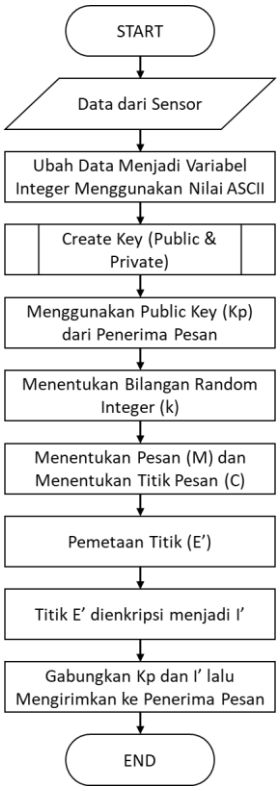
Dalam penelitian ada yang berperan sebagai publisher, broker, subscriber, dan attacker sebagai bantuan untuk menguji sistem yang dapat dilihat pada skeman perancangan sistem yang dapat dilihat pada gambar 1.



Gambar 1. Skema Perancangan Sistem.

Dari gambar 1 diatas, dapat diketahui dari sisi publisher terdapat sensor DHT11 yang terhubung dengan mikrokontroler Wemos D1 Mini ESP8266, dari sisi *broker* menggunakan mosquitto sebagai penghubung antara *publisher* dan *subscriber*, dari sisi *subscriber* juga menggunakan mikrokontroler Wemos D1 Mini ESP8266 sebagai *client* yang akan meng-*subscribe* topik dan menampilkan data melalui layar LCD I2C 16x2 sebagai media monitoring. Pada sisi *attacker*, akan dilakukan percobaan pengujian keamanan dengan proses *sniffing* paket data menggunakan *tools* dari wireshark.

Alur proses dalam mengenkripsi data akan disajikan dalam bentuk flowchart dan hitungan manual dari metode *Elliptic Curve Cryptography*. Flowchart enkripsi data dapat dilihat pada gambar 2.



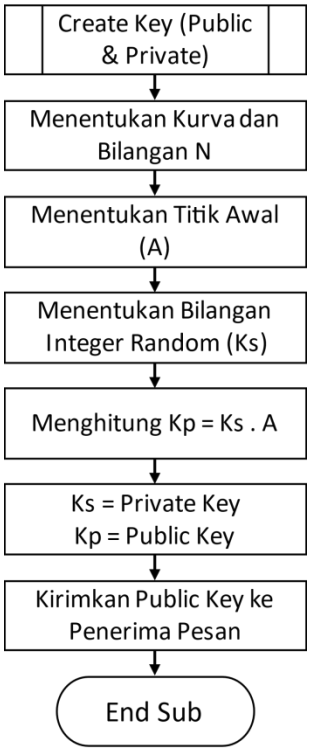
Gambar 2. *Flowchart* Enkripsi Algoritma ECC
Sumber: [13]

Gambar 2 menjelaskan mengenai alur enkripsi algoritma *Elliptic Curve Cryptography*. Proses pertama yang dilakukan adalah menginput data dari sensor yang akan dienkripsi. Data yang telah diterima dalam bentuk *plaintext* akan di ubah menjadi variabel *integer* menggunakan nilai ASCII. Kemudian lakukan proses *create key* (*public* dan *private*), dan akan mendapatkan *public key* (Kp) dari penerima pesan. Selanjutnya menentukan satu bilangan *random intiger* (k) antara 1 – (N-1). Selanjutnya menentukan pesan dan menentukan titik pesan (C) yang akan dienkripsi. Selanjutnya lakukan pemetaan titik (E'), titik E' yang telah dipetakan akan dienkripsi menjadi I'. Terakhir, masukkan nilai Kp (*Public key*) dan I' sebagai dua parameter yang akan dikirim kepada penerima pesan.

Alur proses dalam mengdekripsi data akan disajikan dalam bentuk *flowchart* dari metode *Elliptic Curve Cryptography*.*Flowchart* dekripsi data dapat dilihat pada gambar 3.

Gambar 3. *Flowchart* Dekripsi Algoritma ECC
Sumber: [13]

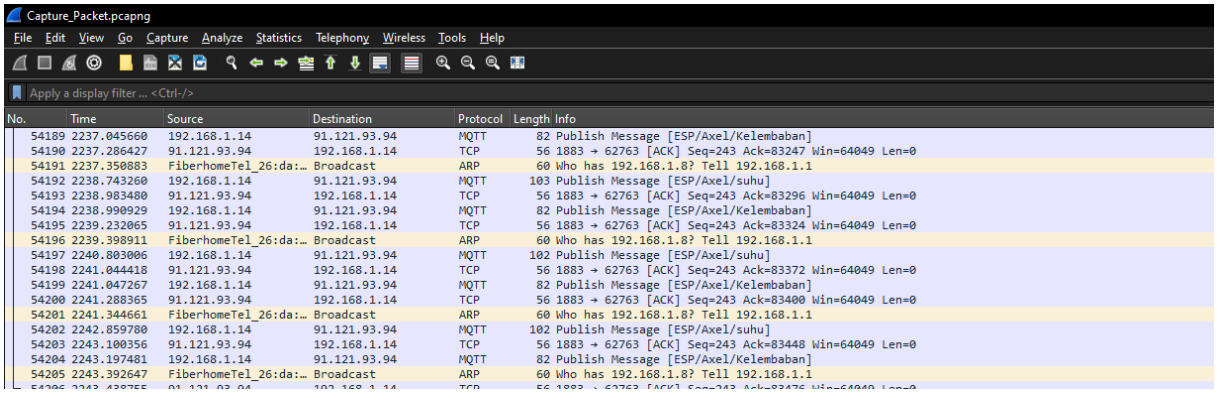
Gambar 3 menjelaskan menjelaskan proses dekripsi dalam algoritma *Eliptic Curve Cryptography*. Langkah awal adalah memasukkan hasil enkripsi. Setelah itu, proses pembuatan kunci (publik dan privat) dilakukan menggunakan kunci privat penerima pesan. Selanjutnya, nilai Z dihitung dengan mengalikan Ks (kunci privat) dengan Kp (kunci publik). Setelah itu, dilakukan proses *reverse* pemetaan untuk mengubah bilangan bulat I' menjadi titik E'. Kemudian nilai E' didekripsi untuk mendapatkan nilai ASCII yang kemudian dikonversi kembali menjadi *plaintext.*



Gambar 4. *Flowchart Create Key* Algoritma ECC
Sumber: [13]

Gambar 4 menggambarkan proses alur pembuatan kunci dalam algoritma *Eliptic Curve Cryptography*. Langkah pertama melibatkan pemilihan kurva elips dan bilangan pemodulo atau bilangan prima N. kemudian, titik awal (A) dipilih pada kurva yang telah ditentukan. Selanjutnya, satu bilangan bulat acak (Ks) ditentukan. Setelah itu, *public key* dihitung dengan mengalikan nilai bilangan bulat acak tersebut dengan titik awal (A) menghasilkan (Kp) yang juga merupakan titik pada kurva. Langkah terakhir adalah mengirimkan *public key* tersebut ke penerima pesan.

Setelah sistem selesai di bangun, tahap selanjutnya akan dilakukan pengujian terhadap sistem yang telah dibangun, beberapa pengujian yang akan dilakukan berupa, pengujian dengan parameter *Latency* atau *Delay*, *Jitter*, *Throughput*, *Packet Loss*, dan akan di lakukan pengujian keamanan. Pengujian akan dilakukan pada *software* wireshark yang terinstal pada laptop yang di jadikan *gateway* untuk *traffic* data Wemos D1 Mini ESP8266 agar memudahkan dalam monitoring *traffic* data. Gambar 5 merupakan gambaran proses *capture traffic* menggunakan wireshark.



Gambar 5. Proses Capture Traffic Menggunakan Wireshark

Data yang diperoleh kemudian diolah menggunakan *microsoft excel* untuk menyaring protokol yang akan digunakan, sehingga data dapat digunakan untuk menghitung masing-masing nilai dari keempat parameter dengan dua skenario *transport* data. Berdasarkan TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*) yang merupakan standar penilaian parameter *Quality of Service* yang dikeluarkan oleh badan standar *European Telecommunications Standards Institute* [16][17], beberapa parameter tersebut antara lain. Throughput, Packet Loss, Delay, dan Jitter. Throughput merupakan bandwidth aktual yang diukur pada waktu tertentu, throughput adalah jumlah data yang berhasil masuk ke jaringan pada interval waktu tertentu, standarisasi dan cara perhitungan nilai Throughput dapat dilihat pada tabel 1 dan persamaan 1.

Tabel 1. Standarisasi Throughput (*bps*)

Kategori Throughput	Throughput (bps)	Index
Excellent	100 bps	4
Good	75 bps	3
Fair	50 bps	2
Poor	25 bps	1

$$Throughput = \frac{Paket\ data\ yang\ diterima}{Lama\ Pengamatan} \quad (1)$$

Packet Loss adalah persentase paket yang hilang selama transmisi data[18], standarisasi dan perhitungan dapat dilihat pada tabel 2 dan persamaan 2.

Tabel 2. Standarisasi Packet Loss (%)

Kategori Packet Loss	Packet Loss (%)	Index
Excellent	0%	4
Good	3%	3
Fair	15%	2
Poor	25%	1

$$Packet\ Loss = \frac{Paket\ data\ yang\ dikirim - Paket\ data\ yang\ diterima}{Packet\ data\ yang\ dikirm}\ x\ 100\% \quad (2)$$

Delay adalah *latency* adalah waktu yang diperlukan untuk mengirim suati paket dari komputer ke komputer tujuan[18]. Standarisasi dan perhitungan dapat dilihat pada tabel 3 dan persamaan 3.

Tabel 3. Standarisasi Delay (*ms*)

Kategori Packet Loss	Packet Loss (%)	Index
Excellent	<150 ms	4
Good	150 mb – 300 ms	3
Fair	300 ms – 450 ms	2
Poor	>450 ms	1

$$Rata - Rata\ Delay = \frac{Total\ Delay}{Total\ Paket\ yang\ diterima} \quad (3)$$

Jitter adalah sebuah variasi delay yang disebabkan oleh panjangnya antrian dalam pengelolahan data[18]. Strandarisasi dan perhitungan dapat dilihat pada tabel 4 dan persamaan 4.

Tabel 4. Standarisasi Jitter (ms)

Kategori of Latency	Delay (ms)	Index
Excellent	<150 ms	4
Good	150 mb – 300 ms	3
Fair	300 ms – 450 ms	2
Poor	>450 ms	1

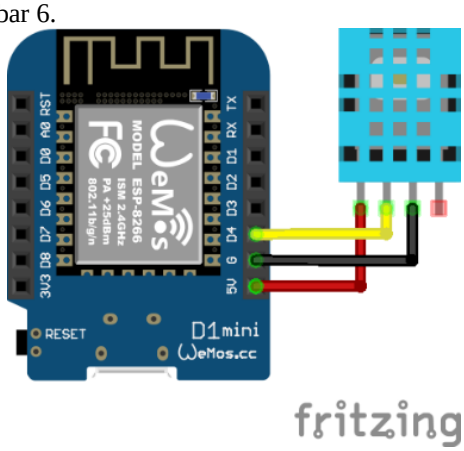
$$Jitter = \frac{Total\ variasi\ delay}{Total\ Paket\ yang\ diterima - 1} \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1. Implementasi Sistem

Sensor yang digunakan pada penelitian ini menggunakan sensor DHT11 yang merupakan sensor dengan kalibrasi sinyal digital yang mampu memberikan informasi suhu dan kelembaban yang cukup akurat. Penulis menggunakan library <DallasTemperature.h> untuk mengakses sensor tersebut. Sensor DHT11 memiliki beberapa pin umum, gambaran pin pada DHT11 yang terhubung

pada Wemos D1 Mini ESP8266 dapat dilihat pada gambar 6.

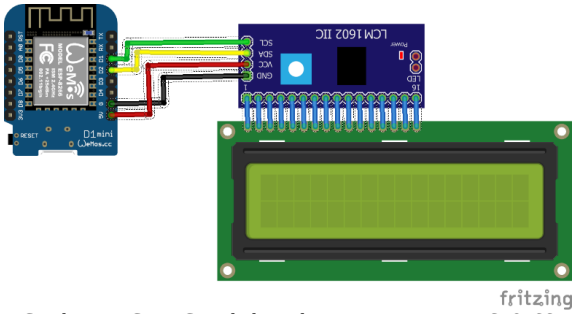


Gambar 6. DHT11 Terhubung ke Wemos D1 Mini ESP8266

Sensor DHT11 menggunakan 3 pin yang terdiri dari VCC (+), GND (-), dan OUT. Pin VCC pada sensor DHT11 terhubung dengan pin 5V pada Wemos D1 Mini ESP8266 berfungsi untuk menyalurkan arus positif sebesar 5V. Pin GND (-) pada DHT11 terhubung ke G pada Wemos D1 Mini ESP8266 berfungsi untuk menyalurkan arus negatif. Pin OUT pada DHT11 terhubung dengan pin digital D4 pada Wemos D1 Mini ESP8266 yang digunakan sebagai pengirim data dari nilai sensor DHT11.

Dalam penelitian ini, Wemos D1 Mini ESP8266 publisher yang terhubung dengan sensor DHT11 berperan sebagai publisher yang akan mengirim data suhu ke *Message Queuing Telemetry Transport Broker* dalam bentuk *ciphertext*, untuk mengubah data suhu yang masih dalam bentuk plaintext ke dalam bentuk ciphertext akan digunakan metode kriptografi *Eliptic Curve Cryptography*. Pada penelitian ini menggunakan library <TinyECC.h> yang memiliki fitur berupa, Arduino-Based mikrokontroler, memiliki fitur *Elliptic Curve Digital Signature Algorithm*, dan memiliki tiga *mapping table* yang dapat digunakan sesuai dengan kebutuhan, jika data sudah dalam bentuk *ciphertext*, data tersebut akan dikirim oleh Wemos D1 Mini ESP8266 yang sudah terhubung ke jaringan internet dan terhubung ke *Message Queuing Telemetry Transport broker* sesuai dengan topik yang sudah ditentukan.

Untuk melakukan monitoring, pada penelitian ini menggunakan LCD I2C 16x2 yang terhubung ke Wemos D1 Mini ESP8266. LCD I2C 16x2 dengan bantuan modul LCM1602 IIC yang dapat digunakan untuk monitoring data yang sudah diolah oleh Wemos D1 Mini ESP8266. LCD I2C 16x2 ini menggunakan 16 pin, tetapi dengan bantuan dari modul LCM1602 IIC ini sehingga hanya memerlukan 4 pin yaitu, GND yang terhubung ke G pada Wemos D1 Mini ESP8266, VCC yang terhubung ke 5V pada Wemos D1 Mini ESP8266, SDA atau Serial Data yang terhubung ke pin D2 pada Wemos D1 Mini ESP8266 yang digunakan untuk mengirim dan menerima data, dan SCL atau Serial Clock yang terhubung ke pin D1 pada Wemos D1 Mini ESP8266 yang digunakan untuk sinyal *clock*. Gambar 7 merupakan gambar gambaran pin LCD I2C 16x2 dengan bantuan modul LCM1602 IIC yang terhung ke Wemos D1 Mini ESP8266.



Gambar 7. LCD I2C Terhubung ke Wemos D1 Mini ESP8266

Setelah LCD I2C dengan modul LCM1602 IIC sudah terhubung ke Wemos D1 Mini ESP8266, selanjutnya mengkonfigurasi Wemos D1 Mini ESP8266 menjadi *subscriber* untuk meng-*subscribe* topik yang sudah ditentukan oleh publisher, pada penelitian ini akan mencoba meng-*subscribe* topik [ESP/Axel/suhu] yang berisi ciphertext, ciphertext yang diterima akan di dekripsi kembali menggunakan metode *Eliptic Curve Cryptography* dan juga *menggunakan* bantuan dari *library* <TinyECC.h>, ciphertext yang sudah di dekripsi akan ditampilkan pada LCD I2C IIC

3.2. Proses Elliptic Curve Cryptography

Elliptic Curve Cryptography adalah pendekatan algoritma kriptografi kunci publik berdasarkan pada struktur aljabar dari kurva ellips pada daerah finite [19]. Berikut ini akan menjelaskan proses data dari sensor DHT11 menjadi *ciphertext* yang akan dikirim ke MQTT Broker.

Adapun proses pembentukan kurva dan pembentukan kunci pada kriptografi kurva eliptik adalah sebagai berikut:

- Menentukan bilangam prima (k), bilangan prima yang akan digunakan pada penelitian ini adalah 193.
- Menentukan titik pada kurva yang akan disimbolkan dengan (P) dengan memilih secara acak pada himpunan penyelesaian dengan batas atas bilangan prima 193. Pada penelitian ini menggunakan (133, 78) sebagai titik awal pada kurva.
- Menghitung kunci publik yang akan disimbolkan dengan kP, kP dapat dihitung dengan cara sesuai dengan persamaan 5.

$$\begin{aligned} kP &= k * P \\ &= 4 * (133, 78) \\ &= [133,78] + [133,78] = [112, 192] \\ &= [112, 192] + [112, 192] \\ &= [163, 143] \end{aligned} \quad (5)$$

Proses enkripsi ini dilakukan oleh pengguna 1 yang akan mengirimkan pesan kepada pengguna 2, adapun cara enkripsi adalah sebagai berikut:

- Pengguna memilih sebuah angka acak yang akan dijadikan private key yang akan disimbolkan dengan k. Nilai k yang akan digunakan adalah 4.
- Menghitung private key yang akan disimbolkan dengan kkP dan dapat dihitung dengan cara;

$$kkP = k * kP$$
$$= 4 * (163, 143) \quad (6)$$
$$= (128, 170)$$

- Selanjutnya mengambil nilai absis dari kkP diatas, (128, 170) maka nilai absisnya adalah 128 → 10000000
- Jika semua langkah sudah selesai, maka sudah dapat melakukan proses enkripsi, pesan yang akan dienkripsi di XOR kan dengan nilai absis dari kkP, pesan yang dienkripsi adalah "Binadarma123" yang dapat dilihat pada tabel 5.

Tabel 5. Konversi Karakter ke Kode ASCII

CHAR	ASCII (dec)	Biner		Absis *kkP* (biner)	Hasil
B	66	01000010	XOR	10000000	194
i	105	01101001	XOR	10000000	233
n	110	01101110	XOR	10000000	238
a	97	01100001	XOR	10000000	225
d	100	01101000	XOR	10000000	228
a	97	01100001	XOR	10000000	225
r	114	01110010	XOR	10000000	242
m	109	01101101	XOR	10000000	237
a	97	01100001	XOR	10000000	225
1	49	00110001	XOR	10000000	177
2	50	00110010	XOR	10000000	178
3	51	00110011	XOR	10000000	179

3.3. Pengujian

Pada penelitin ini akan dilakukan dua pengujian pada sistem yang telah dibangun, yaitu pengujian *Quality of Service* dan pengujian keamanan pada komunikasi data.

Pengamatan dilakukan selama 2244 detik atau sekitar 37.4 menit, data diambil pada malam hari antara jam 20:13:12 hingga 20:50:36. Terdapat 54,206 packet data yang terdiri dari berbagai protokol, setelah dilakukan penyaringan, protokol MQTT terdapat 2301 packet data, yang terdiri dari 1092 packet data untuk topik [ESP/Axel/suhu], 1091 packet data untuk topik [ESP/Axel/Kelembaban], dan 118 packet data request dan response, untuk lebih jelas, data yang berhasil di capture traffic dapat dilihat pada tabel 6.

Tabel 6. Data Hasil Capture Traffic Protokol MQTT

Transport	[ESP/Axel/suhu]	[ESP/Axel/Kelembaban]
Sumber	Wemos D1 Mini (Publisher)	Wemos D1 Mini (Publisher)
Tujuan	MQTT Broker	MQTT Broker
IP Source	192.168.1.14	192.168.1.14
IP Destination	91.121.93.94	91.121.93.94
Jumlah Packet Data yang dikirim	1092	1091
Jumlah Packet Data yang diterima	1092	1091
Total Delay (s)	59,05	61,11
Total Variasi Delay (s)	0.00	0.15
Packet Data yang diterima (bytes)	115125	95583
Lama Pengamatan (s)	2244	2244

Tabel 6 akan digunakan untuk perhitungan *delay*, *jitter*, *throughput*, dan *packet loss* menggunakan transport Wemos D1 Mini (*publisher*) ke MQTT Broker dengan topik [ESP/Axel/suhu]. Untuk mendapatkan rata-rata delay per packet dapat dihitung seperti dibawah ini.

$$Rata - Rata\ Delay = \frac{59{,}05\ s}{1092}$$
$$= 0{,}0541\ s \quad (7)$$
$$= 54{,}1\ ms$$

Sedangkan nilai jitter maka didapat nilai jitter selama proses *capture traffic* adalah

$$Jitter = \frac{0.00}{1092 - 1} \quad (8)$$
$$= 0.00\ ms$$

Nilai *packet loss* dihitung menggunakan rumus, maka didapat nilai packet selama proses *capture traffic* adalah

$$Packet\ Loss = \frac{1092-1092}{1092} \text{ x } 100\% \quad (9)$$
$$= 0\%$$

Nilai throughput dihitung menggunakan rumus, maka didapat nilai throughput selama proses capture traffic adalah

$$Throughput = \frac{115125\ bytes}{2244\ s} \quad (10)$$

= 51.303475 *Bps*
= 410.42780 *bps*

Data hasil pengukuran keseluruhan traffic data dari Wemos D1 Mini (Puublisher) ke MQTT Broker dengan topik [ESP/Axel/suhu] dan [ESP/Axel/Kelembaban] dimasukkan kedalam tabel 7.

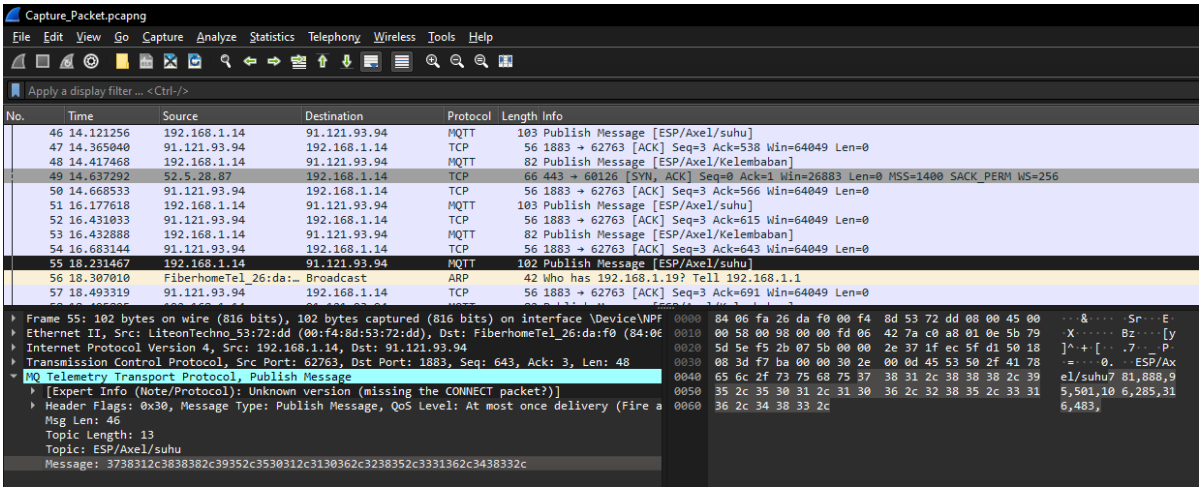
Tabel 7. Hasil Pengukuran Traffic Data

Transport	[ESP/Axel/suhu]	[ESP/Axel/Kelembaban]
Rata-rata Delay (ms)	54,1	55,96
Jitter (ms)	0.00	0.1374
Throughput (bps)	410.42780	340.75935
Packet Loss (%)	0	0

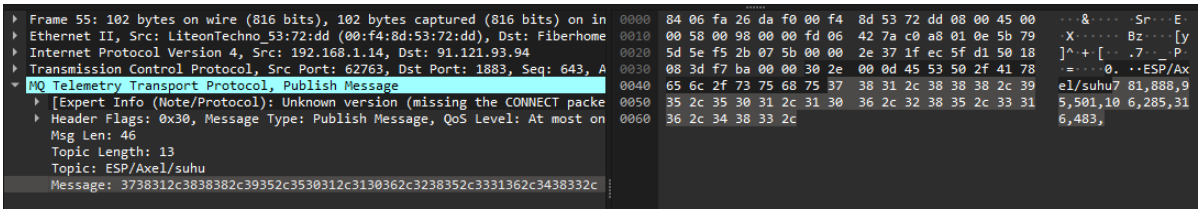
Dari data terlihat pengukuran parameter dalam melakukan pengiriman data dari Wemos D1 Mini (*Publisher*) ke MQTT Broker. Perbedaan antara dua topik tersebut terletak pada, topik [ESP/Axel/suhu] data yang berasa dari sensor dienkripsi menggunakan algoritma *Elliptic Curve Cryptography* terlebih dahulu lalu dikirim ke MQTT Broker, sedangkan topik [ESP/Axel/Kelembaban] dari sensor langsung dikirim ke MQTT Broker tanpa melewati proses enkripsi.

Dari tabel 7 dapat dilihat, rata-rata delay mendapati data yang dikirim dengan proses enkripsi *lebih* minim delay dari pada data yang dikirm tanpa proses enkripsi, sehingga pesan yang dikirim ke topik [ESP/Axel/suhu] lebih cepat di terima oleh MQTT Broker dari pada pesan yang dikirim ke topik [ESP/Axel/Kelembaban]. Jitter didapatkan pada pesan yang dikirim ke topik [ESP/Axel/suhu] mendapatkan nilai 0 ms, sehingga tidak terjadi gangguan pada komunikasi yang disebabkan oleh perubahan sinyal. Data juga dikirm lebih cepat pada data yang dienkripsi menggunakan algoritma *Eliptic Cve Cryptography* dari pada data yang tidak dienkripsi, hal ini dapat dilihat dari nilai Throughput pada topik [ESP/Axel/suhu] mendapatkan nilai kecepatan *pengiriman* data sebesar 410,4 bps. Data yang dikirm baik yang dienkripsi maupun tidak dienkripsi tidak ada packet data yang hilang, hal ini dapat dilihat dari packet loss yang didapat sebesar 0%.

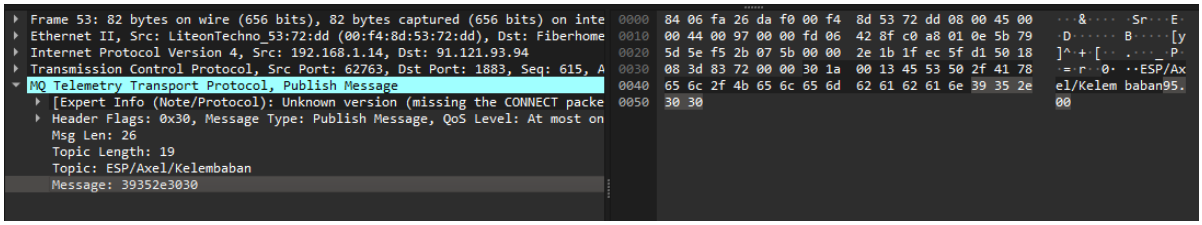
Setelah pengukuran keempat parameter sudah dilakukan, selanjutnya dilakukan peengujian keamanan yang akan *dilakukan* juga pada *software* wireshark, skenario akan dilakukan *capture packet* data yang dikirim dari Wemos D1 Mini (*Publisher*) ke MQTT Broker. Gambar 8 merupakan *capture packet* data topik [ESP/Axel/suhu] dan [ESP/Axel/Kelembaban]. Gambar 9 dan 10 merupakan gambar *packet detail* dan *bytes* data menggunakan algoritma *Eliptic Curve Cryptography* dan tanpa menggunakan algoritma *Eliptic Curve Cryptography*.



Gambar 8. *Capture Packet* Data Menggunakan ECC



Gambar 9. *Packet Detail* dan *Bytes* Data Menggunakan ECC



Gambar 10. Packet Detail dan Bytes Data Tanpa Menggunakan ECC

Dapat dilihat dari gambar 8, packet data yang dikirim dari IP 192.168.1.14 yang merupakan alamat IP mikrokontroler Wemos D1 Mini ESP8266 mengirimkan data ke alamat IP 91.121.93.94 yang merupakan alamat IP dari Broker MQTT, pada gambar 9 dapat dilihat data yang telah dikirim dari mikrokontroler sudah berhasil meng-enkripsi data suhu dan dikirim ke topik [ESP/Axel/suhu] sehingga data yang akan ditampilkan oleh MQTT merupakan sekumpulan angka yang tidak bisa di mengerti, sedangkan gambar 10 merupakan data yang dikirim langsung oleh mikrokontroler tanpa melalui metode *Eliptic Curve Cryptography*, data yang di tampilkan merupakan data asli dari sensor DHT11.

4. DISKUSI

Setelah melihat hasil dari penelitian, menurut penulis, walaupun data yang dikirim ke MQTT Broker sudah dalam bentuk *ciphertext*, hal ini dapat membantu meminimalisir pencurian data dari karena dari pengalaman penulis saat menggunakan *software* MQTT Explorer 0.4.0 sebagai *subscriber*, dari *software* tersebut dapat mem-*publish* data dengan mengirimkan secara manual, data yang dikirimkan oleh *subscriber* atau *publisher* masuk ke MQTT Broker dan dapat diterima oleh *subscriber* lainnya, jika hal ini terjadi pada topik [ESP/Axel/Kelembaban] yang menampung nilai asli dari sensor, maka *subscriber* akan menerima pesan palsu dari si penyerang, tetapi jika hal ini terjadi pada topik [ESP/Axel/suhu] yang menampung data ciphertext, *subscriber* atau *publisher* juga dapat mengirimkan *text* secara manual ke topik ini tetapi hal ini tidak ada berpengaruh karena di sisi *subscriber*, data yang diterima dari MQTT Broker harus berbentuk *ciphertext*, sehingga data yang masuk ke sisi *subscriber* harus selalu di dekripsi terlebih dahulu menggunakan algoritma *Eliptic Curve Cryptography*. Sehingga hal ini lebih aman dari penyerang.

Walaupun algoritma *Eliptic Curve Cryptography* menggunakan konversi nilai ASCII, tetapi seseorang yang memperoleh *ciphertext* tidak mudah untuk mengkonversi secara manual untuk mendapatkan nilai aslinya, karena pemilihan titik pada kurva dipilih secara acak, titik kurva sendiri digunakan sebagai titik awal untuk menghasilkan kunci *public* dan kunci *private*, sehingga *ciphertext* yang dihasilkan akan berbeda-beda juga walaupun *plaintext* nya sama.

Hasil penelitian diketahui bahwa proses enkripsi *Elliptic Curve Cryptography* melibatkan operasi matematika pada titik-titik pada kurva elips. Pesan yang akan dienkripsi diubah menjadi titik pada kurva elips, dan operasi pengalian titik dilakukan antara titik tersebut dengan kunci publik. Hasil operasi ini adalah titik lain pada kurva elips yang kemudian diubah menjadi bentuk biner dan dijadikan sebagai bagian dari pesan terenkripsi. Jika dibandingkan dengan metode enkripsi lain seperti penelitian yang dilakukan oleh [20] yang membahas mengenai proses enkripsi metode *Advanced Encryption Standard* melibatkan penggantian dan permutasi bit dalam blok data menggunakan serangkaian tahan subsitusi, permutasi, dan campuran linier. Data yang akan dienkripsi dipecah menjadi blok-blok yang sama panjang, dan setiap blok dienkripsi secara independen dengan kunci yang sama. Melihat dari perbandingan proses enkripsi kedua metode, menurut penulis, metode *Elliptic Curve Cryptography* menggunakan ukuran kunci yang lebih pendek dalam rentang puluhn hingga ratusan bit, sedangkan algoritma *Advanced Encryption Standard* menggunakan ukuran kunci tetap yaitu 128 bit, 192 bit, atau 265 bit. Sehingga metode *Elliptic Curve Cryptography* lebih efisien dalam penggunaan memori karena proses enkripsi hanya melibatkan perhitungan matematika pada kurva elips sedangkan metode *Advanced Encryption Standard* melibatkan banyak operasi perhitungan matematika. Dalam segi keamanan, baik metode *Elliptic Curve Cryptography* dan *Advanced Encryption Standard* sama sama memiliki tingkat keamanan yang tinggi.

5. KESIMPULAN

Berdasarkan dari hasil pengujian, alat dapat bekerja dengan baik untuk membaca, memproses, dan mengirim data ke MQTT *Broker*. Pengukuran *Quality of Service* pada sistem didapatkan bahwa data yang sudah dienkripsi dan dikirimkan dari subscriber ke MQTT *Broker* memiliki waktu delay yang lebih rendah yang dapat dilihat dari nilai rata-rata delay sebesar 54,1 ms, memiliki kecepatan pengiriman data yang lebih cepat yang dapat dilihat dari throughput 410,4 bps, tidak terjadi kehilangan paket data pada saat pengiriman yang dapat dilihat dari peroleh packet loss sebesar 0% dan tidak terjadi gangguan pada saat pengiriman packet data yang dapat dilihat dari jitter sebesar 0,00 ms, dari pada

data yang tidak dienkripsi. Hasil percobaan pencurian data didapati, data yang telah diolah menggunakan algoritma *Eliptic Curve Cryptography* berhasil mengirimkan *ciphertext* ke MQTT *Broker* dan tidak mudah untuk diketahui apa plaintext yang ingin disampaikan oleh *publisher* ke *subscriber*.

DAFTAR PUSTAKA

[1] M. S. Audita, A. G. Putrada, and N. A. Suwastika, "Implementasi dan Analisis Pengurasan Otomatis Aquascape Berdasarkan Kualitas Air Menggunakan Fuzzy Logic," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 2091–2099, 2019.

[2] T. Sutabri, Y. B. Widodo, S. Sibuea, I. Rajiani, and Y. Hasan, "Tankmate Design For Settings Filter, Temperature, and Light on Aquascape," *J. Southwest Jiaotong Univ.*, vol. 54, no. 5, 2019.

[3] A. Rouf and W. Agustiono, "Literature Review : Pemanfaatan Sistem Informasi Cerdas Pertanian Berbasis Internet of Things (IoT)," *J. Teknol. dan Inform.*, vol. 9, no. 1, pp. 45–54, 2021.

[4] D. Iskandar, A. Febbiansyah, and L. L. Firanda, "Monitoring Suhu dan Kelembaban Udara pada Ruangan Tertutup Berbasis IoT pada PT. Thembuzz Berkat Alam," *Incomtech*, vol. 10, no. 2, pp. 8–15, 2021.

[5] M. A. Hananto, A. Kusyanti, and R. Primananda, "Implementasi Algoritme Acorn untuk Pengamanan Data pada Protokol MQTT menggunakan Perangkat Wemos ESP8266," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.* , vol. 3, no. 8, pp. 2548–964, 2019, [Online]. Available: http://j-ptiik.ub.ac.id.

[6] F. D. Silalahi, J. Dian, and N. D. Setiawan, "Implementasi Internet Of Things (Iot) Dalam Monitoring Suhu Dan Kelembaban Ruang Produksi Obat Non Steril Menggunakan Arduino Berbasis Web," *J. JUPITER*, vol. 13, no. 2, pp. 62–68, 2021.

[7] R. W. Febrianto and A. Zulianto, "Kriptografi Ringan di Internet of Things: Tinjauan Literatur Sistematis," vol. 2, no. 1, 2021.

[8] Y. B. Widodo, A. M. Ichsan, and T. Sutabri, "Perancangan Sistem Smart Home Dengan Konsep Internet Of Things Hybrid Berbasis Protokol Message Queuing Telemetry Transport," *J. Teknol. Inform. dan Komput.*, vol. 6, no. 2, pp. 123–136, 2020, doi: 10.37012/jtik.v6i2.302.

[9] S. Andy and B. Rahardjo, "Keamanan Komunikasi Pada Protokol MQTT untuk Perangkat IoT," *Semin. Nas. Tek. Elektro 2016*, no. 10, pp. 176–184, 2016.

[10] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "Lightweight Elliptic Curve Cryptography Accelerator for Internet of Things Applications," *Ad Hoc Networks*, vol. 103, p. 102159, 2020, doi: 10.1016/j.adhoc.2020.102159.

[11] D. Perdana, P. Purwiko, F. Dewanta, and F. Afianti, "Analisa Penggunaan Elliptic Curve Cryptography pada Sistem Autentikasi pada Internet of Things," vol. 8, no. 1, pp. 42–49, 2022.

[12] S. L. Nita and M. I. Mihailescu, "Elliptic Curve-Based Query Authentication Protocol for IoT Devices Aided by Blockchain," *Sensors*, vol. 23, no. 3, 2023, doi: 10.3390/s23031371.

[13] A. R. Taqwa and D. H. Sulaksono, "Implementasi Kriptografi Dengan Metode Elliptic Curve Cryptography (ECC) Untuk Aplikasi Chatting Berbasis Android," *J. Ris. Inov. Bid. Inform. dan Pendidik. Inform.*, vol. 1, no. 1, pp. 42–48, 2020.

[14] M. Fauzan, U. B. Hanafi, and T. Irfan, "Implementasi TLS Sebagai Metode Keamanan Protokol Jaringan Pada MQTT Berbasis Raspberry PI," *13th Ind. Res. Work. Natl. Semin.*, pp. 13–14, 2022, [Online]. Available: https://jurnal.polban.ac.id/ojs-3.1.2/proceeding/article/view/4332%0Ahttps://jurnal.polban.ac.id/ojs-3.1.2/proceeding/article/download/4332/2862.

[15] T. Sutabri and D. Napitupulu, *Sistem Informasi Bisnis*. Penerbit ANDI, 2019.

[16] ETSI, *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)*, vol. 2.1.1. 1999.

[17] F. Palaha, E. Ermawati, M. Machdalena, and E. H. Arya, "Analisa Traffic Data Esp8266 Pada Kontrol Dan Monitoring Daya Lisrik Menggunakan Aplikasi Blynk Berbasis Arduino Nano," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 4, no. 6, pp. 480–489, 2021, doi: 10.32672/jnkti.v4i6.3646.

[18] H. Apriyanto, R. A. Laksono, and A. K. Ramadhani, "Quality Of Service (QoS) Analysis on The Internet Network (Case Study: Purwodadi Botanical Garden – BRIN)," *SMARTICS J.*, vol. 8, no. 1, pp. 8–13, 2022, [Online]. Available: https://doi.org/10.21067/smartics.v8i1.6503.

[19] S. E. A. P. Damanik, "Implementasi Algoritma Elliptic Curve Cryptography (ECC) UntukPenyandian Pesan Pada Aplikasi Chatting Client Server Berbasis Desktop," *J. Ris. Komput.*, vol. 6, no. 4, pp. 395–400, 2019.

[20] S. P. Wahyuni, M. A. Murti, and G. B. Satrya, "Sistem Pengamanan Data IoT Menggunakan Enkripsi AES," *E-Proceeding Eng.*, vol. 10, no. 4, pp. 3678–3682, 2023.