Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.201961401
Vol. 6, No. 3, Juni 2019, hlm. 335-340 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 30/E/KPT/2018 e-ISSN: 2528-6579

IMPLEMENTASI MEKANISME *END-TO-END SECURITY* PADA IoT *MIDDLEWARE*

Eko Sakti Pramukantoro [1], Fariz Andri Bakhtiar [2], Ahmad Lutfi Bayu Aji [3], Deny Hari Prasetya Dewa[4]

[1,2,3,4]Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]ekosakti@ub.ac.id, [2]fariz@ub.ac.id, [3] bayuaji732@gmail.com, [4] alternativerockers431@gmail.com

(Naskah masuk: 26 Oktober 2018, diterima untuk diterbitkan: 05 Maret 2019)

Abstrak

Pada penelitian sebelumnya telah dikembangkan sebuah *middleware* dengan pendekatan *event-driven* yang mampu mendukung interoperabilitas berbagai macam perangkat sensor pada lingkungan IoT. Namun, skema komunikasi pada *middleware* tersebut masih terdapat celah keamanan dan menimbulkan ancaman berupa *eavasdropping*. Solusi dari permasalahan ini adalah menerapakan mekanisme *end-to-end security*. Dalam penelitian ini dilakukan penerapan algoritme kriptografi AES-CBC 128 pada komunikasi node sensor ke *middleware* dan mekanisme TLS pada komunikasi *middleware* dengan aplikasi berbasis IoT. Hasil yang didapat *end-to-end security* berbasis kriptografi pada pub/sub dapat menjamin kerahasiaan data dengan enkripsi payload akan tetapi topik masih terlihat, sedangkan TLS/SSL menjamin kerahasiaan seluruh data yang dikirim. Penggunaan mekanisme ini tidak berdampak signifikan pada *delay* pengiriman data, yaitu masih dibawah 1 detik

Kata kunci: *IoT middleware, MITM, end-to-end security*

IMPLEMENTATION OF END TO END SECURITY IN IOT MIDDLEWARE

Abstract

An IoT middleware for handling interoperability is proposed in previous works. However, a vulnerability that can lead to the eavesdropping attack exist. there is no security mechanism in the communication system among middleware with other parties like node sensors and subscribers. This research implements the end to end security to the existing IoT middleware. AES-CBC 128 is used to secure communication between sensor nodes to middleware and used TLS/SLL between middleware and subscriber. The results show both mechanisms can securely communication between middleware and other parties, but AES-CBS can only secure data payload, not entire data. This mechanism has no significant impact on the delay transmission, which is still under 1 second

Keywords: *IoT middleware, MITM, end-to-end security*

1. PENDAHULUAN

Pada tahun 2009 istilah *Internet of Things* (IoT) pertama kali dikenalkan oleh Kevin Ashton (Ashton, 2009). IoT melibatkan interaksi antara beragam perangkat seperti sensor, *agregator*, actuator, dan aplikasi dalam berbagai macam domain. Pada dasarnya IoT terdiri dari dua komponen utama, yakni internet dan *things*. Internet merupakan gabungan infrastruktur jaringan dalam skala masif yang berkembang dinamis berdasarkan standar dan protokol komunikasi yang mendukung interoperabilitas. Sedangkan *things* merupakan benda atau perangkat baik konkret maupun virtual yang memiliki identitas, atribut, karakteristik, dan dapat berkomunikasi satu sama lain melalui sebuah antarmuka atau media transmisi. Salah satu tantangan dalam IoT adalah interoprabilitas antar perangkat dan solusi dari permasalahan itu dengan membangun sebuah middleware yang mampu menjawab tantangan tersebut (Desai, Sheth, and Anantharam, 2015).

Pada penelitian sebelumnya telah dikembangkan sebuah *middleware* dengan pendekatan *event-driven* untuk mendukung interoperabilitas untuk perangkat dan sensor yang beragam (Pramukantoro and Anwari, 2018). *Middleware* tersebut menyediakan *gateway* berbasis protokol MQTT dan CoAP untuk melayani komunikasi dari node sensor(*publisher*) dan gateway berbasis protokol WebSocket untuk berkomunikasi dengan aplikasi berbasis IoT(*subscriber*). Namun, skema komunikasi pada *middleware* tersebut tersebut masih terdapat celah keamanan, dimana fitur keamanan belum diterapkan pada transmisi data. Celah keamanan tersebut dapat menimbulkan ancaman berupa *eavasdropping*. Rajra dan J Deepa

menjelaskan bahwa *eavesdropping* adalah penangkapan komunikasi antara dua titik oleh pihak yang tidak berwenang (Blessy Rajra M B, 2015).

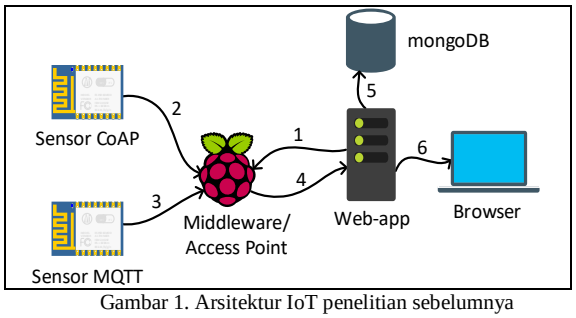
Solusi untuk mengatasi ancaman berupa *eavasdroping* adalah menerapkan mekanisme *confidentiality* atau kerahasiaan dalam proses pengiriman data, atau umum dikenal dengan *end-to-send security*. *End-to-end security* bergantung pada protokol dan mekanisme yang diimplementasikan pada koneksi *endpoints*. Keamanan pada *endpoints* memiliki beberapa kebutuhan, diantaranya; identitas, protokol, algoritme, implementasi dan operasi yang aman (Behringer, 2009). Dalam *end-to-end security* terdapat dua metode yang dapat digunakan, yaitu TLS/SSL dan Kriptografi. Kriptografi adalah seni penulisan rahasia yang digunakan sejak zaman Romawi untuk menyembunyikan informasi rahasia atau menjaga keamanan pesan. Terdapat berbagai macam algoritme kriptografi yang digunakan untuk mengamankan informasi, yaitu algoritme DES, 3DES, *Blowfish*, AES, RSA, ElGamal dan *Paillier*. Maqsood dalam penelitiannya membandingkan kinerja dari setiap algoritme berdasarkan waktu enkripsi dan dekripsi, penelitian tersebut menghasilkan data yang menunjukkan bahwa waktu enkripsi dan dekripsi algoritme AES adalah yang paling cepat (Maqsood et al., 2017).

Untuk mengatasi permasalahan keamanan pada penelitian sebelumnya, maka pada penelitian ini dilakukan (1) implementasi algoritme kriptografi AES-CBC 128 pada komunikasi node sensor dengan *middleware*. Penggunaan algoritme kriptografi AES-CBC 128 dipilih karena algoritme ini menggunakan sumberdaya yang sedikit, sehingga tepat untuk perangkat dengan sumberdata terbatas (Srivastava and Venkataraman, 2013). (2) Implementasi TLS/SSL pada komunikasi *middleware* dengan subscriber berbasis WebSocket. TLS/SSL merupakan protokol yang paling banyak digunakan untuk memastikan autentikasi, integritas dan kerahasiaan pada pertukaran informasi antara *web server* dan *web client* (Turner, 2014).

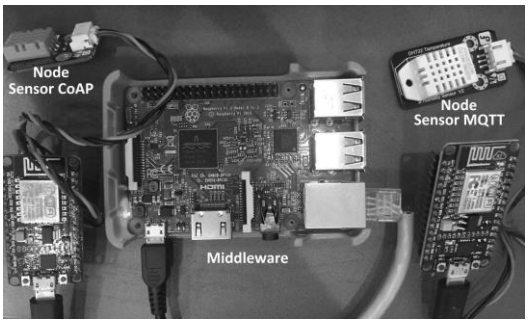
Mudassar Ahmad dalam penelitiannya membuktikan bahwa penggunaan mekanisme keamanan akan mengurangi kinerja transmisi data pada jaringan (Ahmad et al., 2012). Hal ini adalah permasalahan yang utama pada implementasi mekanisme keamanan. Oleh karena itu, selain implementasi mekanisme *end-to-end security*, juga dilakukan analisis delay pengiriman data dari node sensor ke *middleware*. Harapannya didapat sebuah sistem berbasis IoT dengan fitur yang mendukung kerahasiaan data yang ada didalamnya.

2. IoT *MIDDLEWARE*

Gambar 1 merupakan arsitektur jaringan berbasis IoT yang telah dikembangkan pada penelitian sebelumnya, sedangkan Gambar 2 merupakan contoh prototipe perangkat middleware yang dibangun dari Raspberry Pi.



Gambar 1. Arsitektur IoT penelitian sebelumnya



Gambar 2. Prototipe IoT *Middleware*

```
{
  protocol = string,   // protocol using
  timestamp = string, // fetched from time server
  topic = string,
  sensor = {
    tipe = string,     // keyword type is reserved.
    index = string,    // sensor index
    ip = string,       // node's ip address
    module = string    // dht11 or dht22
  },
  humidity = {
    value = number,
    unit = string      // in percentage
  },
  temperature = {
    value = number,
    unit = string      // celsius or fahrenheit
  }
}
```

Gambar 3. Semantik data dari sensor

Dalam Gambar 2 terdapat dua buah node sensor, yang pertama mengirim data dengan protokol MQTT dan yang kedua dengan protokol CoAP. Node sensor dibangun menggunakan nodeMCU esp8266 dengan sensor DHT22. Data dikirim dengan topik yang spesifik, contohnya home/kitchen. Data yang dikirim oleh node sensor disimpan sementara dalam sebuah *broker* (dalam penelitian ini digunakan Redis), berikutnya *subscriber* akan meminta data tersebut melalui protokol WebSocket dan diteruskan ke aplikasi lain untuk proses analisis lebih lanjut.

Media transmisi yang digunakan antara node sensor dan middleware adalah wireless LAN atau nirkabel yang disediakan oleh middleware. Sedangkan antara middleware dan *subscriber* digunakan jaringan LAN, dalam hal ini jaringan kampus. Untuk data yang dikirim oleh node sensor terdiri dari informasi node sensor dan informasi sensor. Informasi node sensor berisi protokol yang digunakan, *timestamp*, *topic*, dan *payload* sensor, seperti yang terlihat di Gambar 3 semantik data yang akan dikirim oleh node sensor.

3. IMPLEMENTASI *END TO END SECURITY*

Pada bagian ini pembahasan akan dibagi menjadi tiga bagian, yaitu;

3.1. Enkripsi Payload

Seperti yang telah dibahas sebelumnya, middleware bekerja dengan pendekatan *publish/subscribe* dimana setiap data dikenali berdasarkan sebuah topik. Oleh karena itu dalam menerapkan mekanisme enkripsi tidak bisa semua data dienkripsi, hal ini akan berdampak mekanisme *publish/subscribe* tidak dapat bekerja. Selain itu terdapat kebutuhan khusus yaitu mekanisme yang sesuai dengan keterbatasan sumber daya dari node sensor.

Salah satu pendekatan yang memungkinkan adalah enkripsi payload, dalam hal ini adalah data dari sensor yaitu *humidity* dan *temperature*. Enkripsi payload dengan algoritme AES-CBC 128 dilakukan pada saat data akan dikirim oleh sensor node ke middleware. Fungsi enkripsi dapat dilihat pada Gambar 4, sedangkan Gambar 5 adalah proses dekripsi.

```
alg = "AES-CBC"
datakey = "1234567890abcdef"
iv = "2345678901abcdef"

encrypted_data = crypto.encrypt(alg, datakey, data, iv)
```

Gambar 4. Fungsi Enkripsi pada node sensor

```
var algorithm = 'aes-128-cbc';
var datakey  = '1234567890abcdef';
var iv =' 2345678901abcdef';

function decrypt (datakey, iv, data) {
        var decipher = crypto.createDecipher(algorithm, datakey
iv);
        var decrypted = decipher.update(data, 'hex', 'utf8');
        var decrypted += decipher.final('utf8');

         return decrypted;
}

let data = raw
var decryptedData = decrypt (datakey, iv, data.toString('utf-8'))
data = new Buffer.from(JSON.stringify(decryptedData))
```

Gambar 5. Fungsi dekripsi pada *subscriber*

3.2. Dekripsi Payload

Semua data yang dikirim oleh node sensor akan tersimpan dalam Redis. Selanjutnya *subscriber* akan meminta data tersebut, agar data yang terenkripsi dapat dibaca kembali perlu ditambahkan fungsi dekripsi pada *subscriber*. Berbeda dengan node sensor, *susbcriber* tidak terbatas oleh sumber daya. Dalam penelitian sebelumnya *subscriber* adalah sebuah server yang bertugas untuk melakukan analisis data sensor.

3.3. TLS/SSL

Mekanisme TLS/SSL diterapkan pada komunikasi middleware dengan *subscriber* yang menggunakan protokol WebSocket. Pada sisi middleware diperlukan sebuah modul https, sertifikat server, dan kunci private untuk membangun komunikasi berbasis TLS/SSL. Pada Gambar 6 adalah potongan kode yang berfungsi untuk membangun komunikasi dengan TLS/SSL

```
https = require('https')

var options = {
    key:fs.readFileSync('server-key-deny.pem'),
    cert:fs.readFileSync('server-cert-deny.pem'),
    ca:fs.readFileSync('server-csr-deny.pem'),
    rejectUnauthorized: true,
  }

server = https.createServer(options, app)
```

Gambar 6. Pembentukan mekanisme TLS/SSL

Pada sisi *subscriber* seperti yang terlihat pada Gambar 7, terdapat sebuah fungsi untuk membangun koneksi TLS/SSL dengan middleware berbasis WebSocket. Komunikasi TLS/SSL diinisiasi oleh client ke alamat middleware 10.34.8.6 dengan port 3000.

```
const client = io.connect('https://10.34.8.6:3000/',
{agent: https.globalAgent});
```

Gambar 7. Fungsi TLS/SSL pada sisi *subscriber*

4. PENGUJIAN DAN PEMBAHASAN

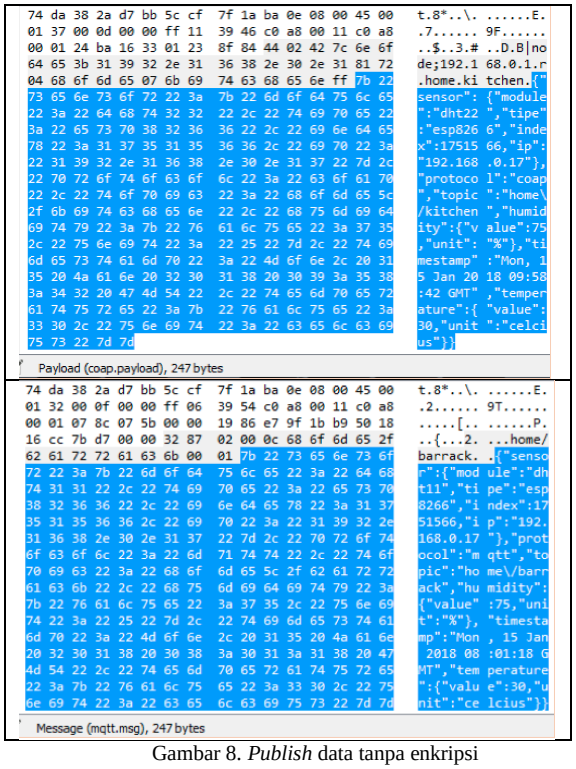
Setelah Implementasi mekanisme enkripsi dan dekripsi dilakukan, tahapan berikutnya adalah melakukan pengujian untuk mengetahui apakah mekanisme yang diterapkan, yaitu TLS/SSL dan AES-CBC-128 berhasil mengamankan pengiriman data antara (1) node sensor ke *middleware*, (2) *middleware* dengan *subscriber*.

Pengujian pertama untuk mengetahui bagaimana kerahasian data yang sedang dikirim antara node sensor dengan middleware. Pengujian dilakukan dengan metode *sniffing* untuk melihat data yang dikirim dengan menkanisme enkripsi dan tanpa mekanisme eknripsi. *Sniffing* dilakukan dengan perangkat lunak Wireshark yang terpasang antara node sensor dan middleware.

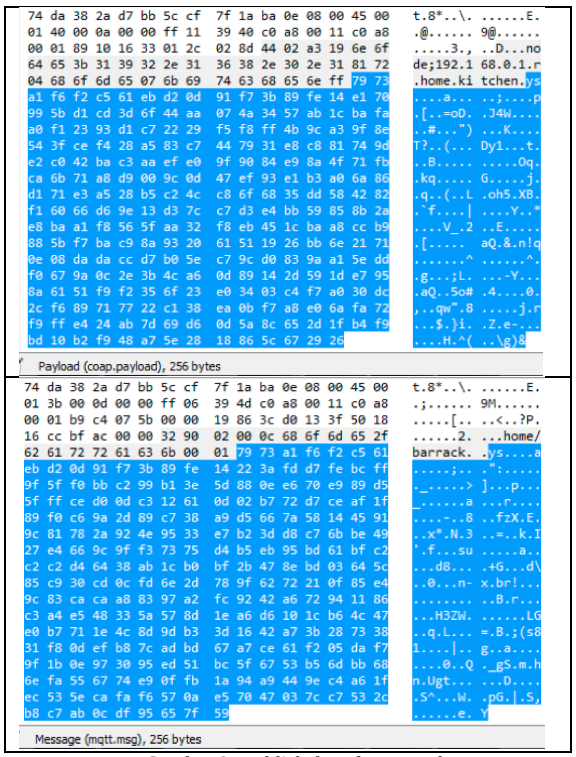
Pada Gambar 8. adalah hasil *sniffing* pengiriman data baik dengan protokol CoAP dan MQTT. Terlihat jelas dari semua informasi yang dikirim baik informasi node senor dan payload sensor dengan ukuran 247 bytes.

Pada Gambar 9, adalah hasil sniffing pengiriman data dari node sensor ke middleware dengan mekanisme enkripsi payload. Informasi yang dapat dilihat dengan jelas adalah topik yang digunakan ”home/barrack”, sedangkan informasi dari sensor tidak dapat dibaca atau berupa chipertext. Hal ini dapat diartikan mekanisme enkripsi payload berhasil untuk menjaga kerahasiaan data yang dikirim dari sensor node ke middleware. Data yang berisi suhu dan temperatur tidak dapat dilihat oleh orang ketiga dalam komunikasi antara middleware dan node sensor.

Pengujian berikutnya adalah untuk mengetahui dampak penerapan enkripsi payload dalam pengiriman data. Parameter yang digunakan adalah delay. Pengujian dilakukan dengan membandingkan delay pengiriman data dengan enkripsi payload dan tanpa enkripsi payload.

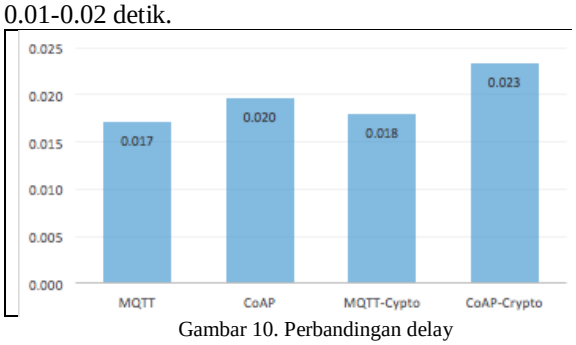


Gambar 8. *Publish* data tanpa enkripsi



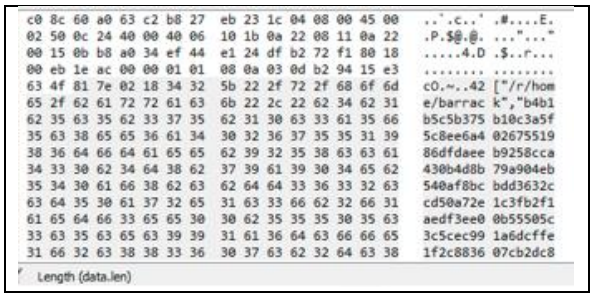
Gambar 9. *Publish* data dengan enkripsi

Pada Gambar 10, adalah hasil ekperimen untuk membandingkan delay pengiriman. Tampak pada gambar delay pengiriman dengan mekanisme enkripsi payload lebih tinggi dari pada pengiriman tanpa eknripsi. Hal ini didukung dengan Gambar 8 dan 9, ukuran payload bertambah dengan adanya mekanisme enkripsi yaitu dari 247 bytes menjadi 256 bytes. Akan tetapi delay pengiriman masih dikatakan baik, karena masih dibawah 0.025 detik dan penambahan delay tidak terlalu signifikan bertambah 0.01-0.02 detik.

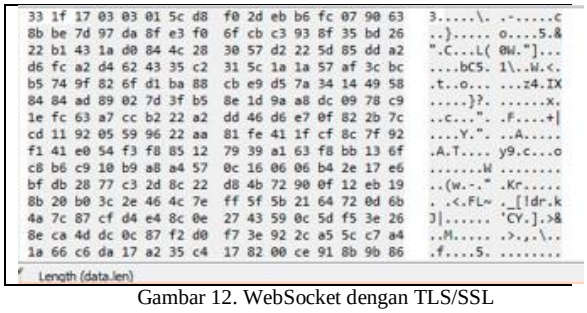


Gambar 10. Perbandingan delay

Pengujian berikutnya adalah komunikasi antara middleware dengan *subscriber*. Pengujian juga dilakukan dengan metode *sniffing* untuk membandingkan komunikasi dengan TLS/SSL dan tanpa TLS/SSL.



Gambar 11. WebSocket tanpa TLS/SSL



Gambar 12. WebSocket dengan TLS/SSL

Hasil yang didapat dari pengujian, data yang dikirim dari middlware ke *subscriber* masih bisa dibaca sebagian yaitu informasi node sensor tampak pada Gambar 11. Setelah komunikasi dibungkus dengan dengan TLS/SSL semua informasi dari baik informasi mengenai sensor node dan payload tidak bisa dibaca dengan jelas, tampak pada Gambar 12.

5. KESIMPULAN

Dari penelitian ini dapat disimpulkan baik mekanisme enkripsi payload dengan AES-CBC dan TLS/SSL dapat memberikan jaminan kerahasiaan pada mekanisme publish/subscribe. Untuk e*nd-to-end security* berbasis kriptografi hanya bisa menjaga kerahasiaan data sensor, tidak keseluruhan data yang dikirim. Berikutnya, algoritme ini tepat jika digunakan pada node sensor dengan sumber daya yang terbatas. Dari hasil pengujian pengiriman dengan parameter delay, tidak ada peningkatan delay

yang signifikan dan masih jauh dibawah 1 detik yaitu 0.025 detik. Sedangkan untuk e*nd-to-end security* berbasis TLS/SSL menawarkan jaminan kerahasiaan yang lebih tinggi. Dari pengujian data yang dikirim tidak dapat dilihat dengan jelas baik informasi node sensor atau data dari sensor. Akan tetapi mekanisme TLS/SSL terbatas dalam penerapan, terlebih pada perangkat dengan keterbatasan sumberdaya.

DAFTAR PUSTAKA

AHMAD, MUDASSAR, SUMAIRA TAJ, TASLEEM MUSTAFA, AND MD ASRI. 2012. 'Performance Analysis of Wireless Network with the Impact of Security Mechanisms'. In *International Conference on Emerging Technologies (ICET)*,.

ASHTON, KEVIN. 2009. 'That "Internet of Things" Thing, in the Real World Things Matter More than Ideas'. *RFID Journal*.

BEHRINGER, MICHAEL H. 2009. 'End-to-End Security'. *The Internet Protocol Journal* 12(3). https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-45/123-security.html.

BLESSY RAJRA M B, 2A J DEEPA. 2015. 'A Survey on Network Security Attacks and Prevention Mechanism'. *Journal of Current Computer Science and Technology* 5(2): 1–5.

DESAI, PRATIKKUMAR, AMIT SHETH, AND PRAMOD ANANTHARAM. 2015. 'Semantic Gateway as a Service Architecture for IoT Interoperability'. In *Proceedings - 2015 IEEE 3rd International Conference on Mobile Services, MS 2015*, , 313–19.

MAQSOOD, FAIQA, MUHAMMAD MUMTAZ ALI, MUHAMMAD AHMED, AND MUNAM ALI SHAH. 2017. 'Cryptography : A Comparative Analysis for Modern Techniques'. *International Journal of Advanced Computer Science and Applications* 8(6): 442–48.

PRAMUKANTORO, EKO SAKTI, AND HUSNUL ANWARI. 2018. 'An Event-Based Middleware For Syntactical Interoperability And Enabling Web-Oriented In Internet Of Things'. *International Journal of Electrical and Computer Engineering* 8(5).

SRIVASTAVA, ANKIT, AND N REVATHI VENKATARAMAN. 2013. 'AES-128 Performance in Tinyos with CBC Algorithm'. 7(5): 40–49.

TURNER, SEAN. 2014. 'Transport Layer Security'. *IEEE Internet Computing* 18(6): 60–63.

Halaman ini sengaja dikongkan