Keamanan Komunikasi Pada Protokol MQTT untuk Perangkat IoT

Syaiful Andy[1], Budi Rahardjo[2]
Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung
Jalan Ganesha No 10, +62 22 2500935
e-mail: syaifulandy@gmail.com, rahard@gmail.com

Abstrak *– Protokol komunikasi yang saat ini digunakan pada perangkat IoT ada cukup banyak. Salah satu protokol yang telah memiliki standard ISO adalah protokol MQTT (sesuai standard ISO/IEC 20922:2016). Protokol MQTT ini telah digunakan oleh banyak pengembang perangkat IoT disebabkan karena protokol ini membutuhkan bandwidth yang sangat kecil dan menggunakan sedikit memory ketika beroperasi. Dalam penggunaanya, perangkat IoT terkadang mengirimkan data-data rahasia yang hanya boleh diakses oleh orang tertentu. Protokol MQTT secara default tidak mengenkripsi data-data yang dikirimkan. Protokol MQTT ini baru memiliki mekanisme keamanan berupa autentikasi, walaupun proses autentikasinya secara default belum terenkripsi. Mekanisme autentikasi yang ada juga masih belum efektif, disebabkan penggunaan pasangan user dan password yang sama untuk tiap device yang terhubung dengan broker yang sama. Selain itu, protokol ini juga belum memiliki mekanisme authorization untuk device yang terhubung ke broker, dan belum memiliki mekanisme yang menjamin integritas data yang dipublish maupun di subscribe. Oleh karena itu, makalah ini akan mereview beberapa paper terkait teknik pengamanan protokol MQTT yang telah di implementasikan pada perangkat IoT.*

Kata Kunci*: MQTT, IoT, keamanan, protokol*

1. Pendahuluan

Internet of Things (IoT) atau komunikasi antar mesin (M2M), merupakan sebuah konsep yang memungkinkan komunikasi antar perangkat melalui jaringan internet. Diprediksi jumlah perangkat IoT akan berkembang dengan cepat, jumlahnya diprediksi dapat mencapai 50 miliar di tahun 2020 [1]. Pemanfaatan teknologi IoT pada *smart city* dengan berbagai macam penyusunya seperti *smart home*, *smart transportation*, dan *smart parking* sudah menjadi hal yang biasa dilihat saat ini. Teknologi IoT menjadi salah satu komponen penyusun *smart city* demi tercapainya tujuan tertentu.

Hingga kini telah ada beberapa protokol standard yang digunakan sebagai protokol komunikasi di perangkat IoT. Beberapa pertimbangan ketika memilih protokol yang akan digunakan diantaranya adalah perangkat IoT yang memiliki keterbatasan (seperti kemampuan menghitung, memiliki daya yang terbatas, dan lainya), perangkat IoT yang digunakan bisa bermacam-macam jenisnya, dan jalur komunikasi antar perangkat yang tidak dapat diandalkan [2]. Ada beberapa protokol IoT yang digunakan saat ini, diantaranya adalah *Hypertext Transfer Protocol* (HTTP), *Extensible Messaging and Presence Protocol* (XMPP), *Constrained Application Protocol* (CoAP), *Advanced Message Queuing Protocol* (AMQP), dan *MQ Telemetry Transport* (MQTT).

Selain pertimbangan tersebut, salah satu hal yang penting untuk dipertimbangkan adalah masalah keamanan informasi. Beberapa protokol komunikasi pada perangkat IoT tersebut, belum memiliki mekanisme keamanan informasi yang menyeluruh. Menurut buku yang dikeluarkan oleh ISACA [3], objek dari keamanan informasi terdiri dari tiga komponen, yaitu kerahasiaan data, integritas data, dan ketersediaan data. Kerahasiaan data berarti melindungi informasi, sehingga informasi hanya dapat di akses oleh orang atau perangkat yang berwenang. Integritas data berarti melindungi informasi dari perubahan yang tidak diizinkan, sedangkan ketersediaan data berarti melindungi informasi agar tetap dapat diakses. Salah satu protokol yang belum memiliki mekanisme keamanan secara menyeluruh adalah protokol MQTT. Protokol ini secara default hanya memiliki mekanisme autentikasi saja. Proses autentikasinya pun masih memiliki banyak kekurangan, diantaranya adalah proses autentikasinya secara

default masih belum terenkripsi, dan penggunaan user-password yang sama untuk tiap perangkat yang terhubung ke broker yang sama [4]. Mekanisme pengamanan pada protokol MQTT untuk otorisasi, penjaminan integritas data, enkripsi data, dan pencegahan serangan seperti denial of service (DoS) yang mengganggu ketersediaan sistem, masih perlu ditambahkan ke protokol ini. Dampaknya yang ditimbulkan bila tidak adanya mekanisme proteksi tambahan bisa sangat serius. Meskipun protokol ini menggunakan mekanisme autentikasi, namun bila tanpa enkripsi, seorang penyadap dapat mengetahui pasangan user dan password dengan mudah, dan dengan penggunaan user-password yang sama untuk setiap perangkat menyebabkan proses manajemen perangkat akan lebih sulit bila seorang admin menginginkan penggantian password. Selanjutnya, bila tidak adanya mekanisme otorisasi, maka tidak bisa diklasifikasikan, perangkat mana yang memiliki hak tertentu. Tanpa mekanisme penjaminan integritas data, maka data bisa saja diubah oleh pihak yang tidak berwenang. Enkripsi data menjamin bahwa data hanya dapat dibaca oleh yang berhak.

Seperti yang kita ketahui, dalam membangun sistem yang "smart" dengan menggunakan bantuan perangkat IoT untuk melakukan proses sensing, yaitu proses untuk mendapatkan data-data yang ada di lingkungan menggunakan sensor. Beberapa data yang diambil oleh sensor, sifatnya rahasia (informasi yang sifatnya personal), misalnya data-data yang ada di smart home. Oleh karena itu, pengguna harus memahami apa yang direkam oleh perangkat IoT, bagaimana informasi tersebut digunakan, ketersediaan sistem tersebut untuk siapa, dan siapa yang dapat melakukan control terhadap perangkat tersebut [5].

Selain itu, dibutuhkan mekanisme autentikasi dan otorisasi dari perangkat sensing ke broker, sehingga hanya perangkat-perangkat yang berhak saja yang dapat terhubung ke broker. Salah satu kasus terbaru terkait kerentanan perangkat IoT adalah serangan distributed denial of service (DDoS) ke Krebsonsecuirty.com yang dilakukan oleh botnet yang tertanam di perangkat IoT. Serangan lainya, diambil dari data yang dimiliki oleh tim Threat Research Akamai, pada bulan oktober 2016, dilaporkan ada jutaan perangkat IoT yang digunakan sebagai proxy untuk melakukan routing trafik korban ke situs berbahaya [6].

Selanjutnya, makalah ini akan membahas lebih dalam mengenai salah satu protokol komunikasi untuk perangkat IoT yaitu protokol MQTT. Pembahasan dimulai dari penjelasan mengenai protokol MQTT beserta keamanan komunikasinya. Selanjutnya akan dijelaskan mengenai beberapa mekanisme keamanan untuk protokol MQTT yang telah diimplementasikan pada perangakat IoT oleh peneliti lain. Keamanan yang akan dibahas pada makalah ini diantaranya adalah mekanisme untuk menjaga kerahasiaan data, autentikasi, otorisasi, integritas data, dan ketersediaan sistem.

2. Protokol MQTT dan Keamananya

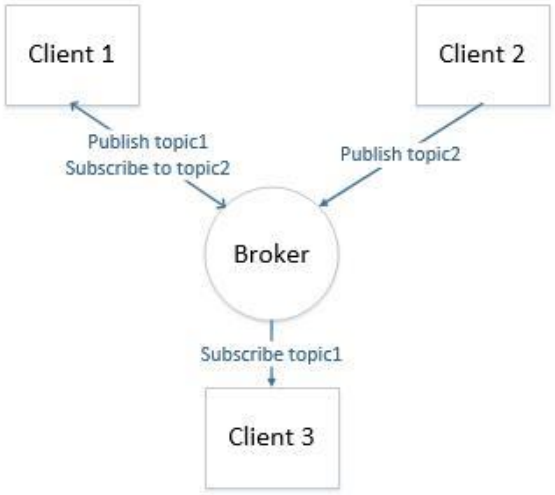
Message Queuing Telemetry Transport (MQTT) merupakan protokol pengiriman pesan dengan menggunakan mekanisme *publish/subscribe*, yang awalnya didesain oleh Andy Stanford-Clark dan Arlen Nipper, yang saat ini merupakan standard OASIS (*Organization for the Advancement of Structured Information Standards)* [7]. Saat ini, protokol MQTT juga telah memiliki standard *ISO/IEC 20922:2016* (*Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1).* Protokol ini relatif banyak digunakan untuk perangkat IoT yang memiliki keterbatasan sumber daya disebabkan protokol ini sangat ringan sehingga mampu bekerja ketika *bandwidth* terbatas, terbuka, sederhana, dan didesain semudah mungkin untuk dapat diimplementasikan [7]. Protokol ini bekerja diatas protokol TCP/IP (*port default*nya adalah 1883 dan 8883) atau protokol jaringan lainya yang dapat melakukan koneksi yang *ordered*, *lossless*, dan *bidirectional* [7,8]. Fitur-fitur yang tersedia pada protokol ini diantaranya adalah [7]:

1. Pola pesan yang dikirimkan menggunakan mekanisme *publish/subscribe* yang mampu mendistribusikan pesan dari satu ke banyak dan dapat digunakan untuk aplikasi yang berbeda.

ISBN: 978-602-60581-0-2

2. Pengiriman pesan yang tidak mempedulikan isi payload, sehingga isi pesanya bebas. Protokol ini juga menyediakan tiga jenis quality of service pada pesan yang akan dikirimkan, yaitu at most once (pesan dikirimkan dengan usaha terbaik satu kali), at least once (pesan dijamin terkirim namun dupllikasi pesan dapat terjadi), exactly once (pesan dijamin sampai ditujuan tepat satu kali.
3. Transport overhead dan protokol pertukaran data yang kecil dapat meminimalisir trafik jaringan.
4. Adanya mekanisme untuk memberitahu pihak yang berkepentingan apabila koneksi pada perangkat terputus tiba-tiba.

Arsitektur dari protokol MQTT dapat dilihat pada gambar 1. Seperti yang terlihat pada gambar 1, komunikasi dengan menggunakan MQTT memungkinkan komunikasi dari banyak perangkat (client) ke banyak perangkat lainya dan dapat memisahkan perangkat yang berperan sebagai produsen informasi (melakukan publish) dan mana yang sebagai konsumen (melakukan subscribe) [8]. Operasi publish dan subscribe sebenarnya seperti model client dan server. Server pusat dalam MQTT diberi nama broker yang berperan sebagai penerima pesan dari client (perangkat) yang pada dasarnya adalah seluruh node yang terlibat dalam proses komunikasi [8]. Pesan tersebut dapat berupa topik publish atau subscribe. Seluruh perangkat yang terhubung dalam protokol ini dapat menjadi publisher dan subscriber, biasanya dalam arsitektur MQTT terdapat beberapa buah sensor yang secara periodik mempublish hasil pengukuranya (pada payload) ke sebuah alamat topik tertentu. Setiap perangkat yang telah terdaftar sebagai subscriber dari topik tertentu akan menerima pesan dari broker setiap kali topik tersebut diperbarui [8].



Gambar 1. Arsitektur *publish* dan *subscribe* protokol MQTT [8]

Menurut buku spesifikasi tentang MQTT versi 3.1.1 yang dikeluarkan oleh OASIS [7], pada bab 5 yang membahas mengenai keamanan untuk MQTT dibahas mengenai beberapa teknik yang dapat digunakan untuk mengamankan protokol MQTT. Pembahasan implementasi keamanan untuk protokol MQTT pada buku tersebut diantaranya adalah autentikasi client oleh server, autorisasi client oleh server, autentikasi server oleh client, integritas pesan atau perintah yang dikirim, kerahasiaan dari pesan atau perintah yang dikirim, dan yang lainya sebagian besar merekomendasikan penggunaan TLS atau SSL. Penggunaan TLS atau SSL mungkin cukup baik bila perangkat yang digunakan memiliki sumber daya yang relatif besar. Namun, secara umum, perangkat IoT memiliki keterbatasan sumber daya seperti kemampuan komputasi yang terbatas, penggunaan baterai pada perangkat (bila proses untuk keamananya berat maka baterai bisa cepat

ISBN: 978-602-60581-0-2

habis), dan keterbatasan lainya. Oleh karena itu, beberapa peneliti telah mengembangkan beberapa mekanisme untuk pengamanan protokol MQTT sehingga dengan keterbatasan-keterbatasan dari perangkat yang digunakan masih tetap dapat diimplementasikan dengan baik

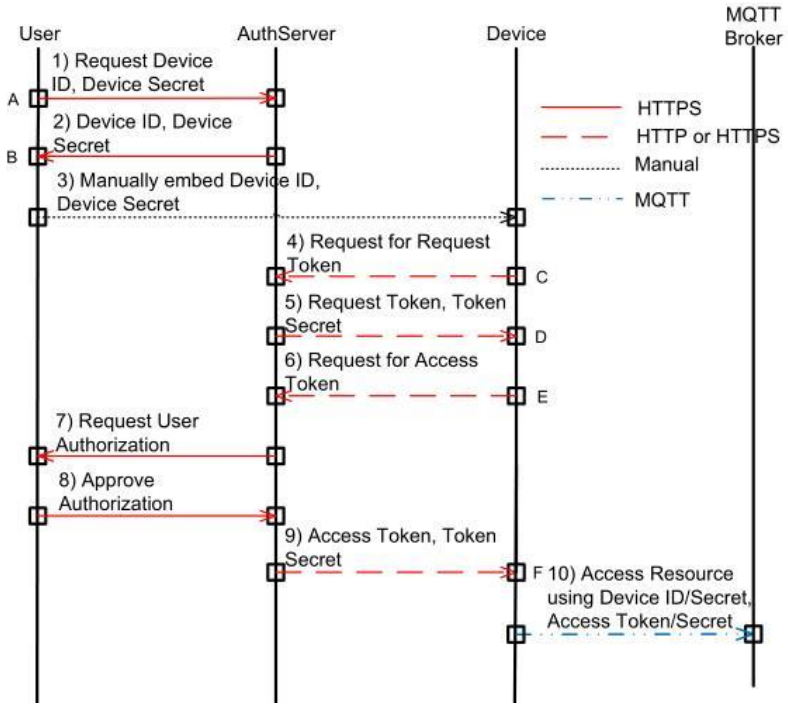
3. Mekanisme Keamanan Protokol MQTT untuk IoT

Pada bagian ini, akan dijelaskan mengenai beberapa mekanisme keamanan tambahan untuk protokol MQTT. Beberapa mekanisme keamanan yang akan dijelaskan pada sub bab ini merupakan hasil penelitian beberapa peneliti lain yang telah mengimplementasikan dan mengujinya.

3.1 Mekanisme untuk Autentikasi dan Otorisasi

Mekanisme keamanan ini memiliki fokus pada proses autentikasi dan otorisasi dari perangkat IoT. Mekanisme ini telah diimplementasikan dan diuji oleh paper yang dibawakan oleh Niruntasukrat, dkk [4]. Pada penelitian yang dilakukanya, asumsi yang dibuat diantaranya adalah sebagai berikut:

1. Perangkat IoT memiliki sumber daya yang terbatas, sehingga protokol TLS/SSL tidak dapat digunakan, dan penggunaan *memory* harus sekecil mungkin.
2. Tidak boleh ada identitas rahasia yang disimpan pada perangkat IoT, diasumsikan perangkat IoT tidak memiliki ID khusus yang melekat pada *firmware*-nya.
3. Diasumsikan perangkat IoT tidak memiliki tampilan pengguna (*user interfaces*) seperti tombol, *keypads*, atau layar sentuh, sehingga komunikasi antara perangkat dan pengguna harus melalui perantara atau *broker* yang dilengkapi dengan tampilan pengguna.



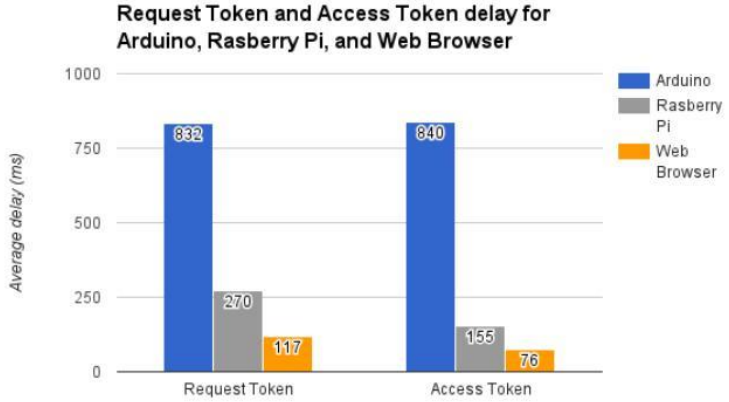
Gambar 2. Teknik otorisasi MQTT pada perangkat IoT [4]

Paper ini [4] memanfaatkan hasil modifikasi OAuth 1.0a untuk implementasinya. Mekanisme dari proses autentikasi dan otorisasinya dapat dilihat pada gambar 2 [4]. Penjelasan langkah-langkah untuk otorisasinya [4] adalah sebagai berikut.

ISBN: 978-602-60581-0-2

1. Pengguna mengautentikasi dirinya ke AuthServer menggunakan identitas dirinya (ID dan password rahasia dari aplikasi), kemudian pengguna mengirimkan permintaan dengan HTTPS untuk mendapatkan data rahasia terkait perangkat IoT (ID dan password dari perangkat).
2. AuthServer akan memeriksa identitas dari pengguna dan memberikan pengguna pasangan ID dan password perangkat yang unik pada body dari HTTPS.
3. Pengguna secara manual akan menanamkan identitas rahasia yang diterima ini ke firmware dari perangkat.
4. Kemudian perangkat IoT akan mengirimkan permintaan token sementara yang disebut "Request Token" kepada AuthServer dengan cara mengautentikasi dirinya sendiri menggunakan ID dan password milik perangkat tersebut. Pengirimanya dapat menggunakan protokol HTTP dan HTTPS. Pemilihan protokol biasanya menyesuaikan dengan kemampuan dari perangkat IoT yang digunakan.
5. AuthServer memvalidasi identitas rahasia dari perangkat dan mengeluarkan kumpulan Request Token dan Request Token Secret ke perangkat IoT, pada respon body HTTP atau HTTPS (tergantung protokol yang digunakan).
6. Perangkat IoT kemudian mengirimkan permintaan ke AuthServer dengan HTTP atau HTTPS untuk mendapatkan token permanen yang dikenal dengan Access Token.
7. AuthServer kemudian akan memberikan peringatan kepada pengguna melalui berbagai cara, misalnya email atau SMS. Mekanisme ini digunakan untuk memenuhi permintaan otorisasi. Pengguna dapat login ke AuthServer melalui web browser untuk melihat permintaan dan memverifikasi ID dari perangkat dan lingkup dari akses yang diperbolehkan.
8. Pengguna dapat memutuskan apakah akan menerima atau menolak permintaan otorisasi in. Browser kemudian akan mengirim pemberitahuan ke AuthServer persetujuan atau penolakan dari permintaan otorisasi.
9. Setelah otorisasi di terima, AuthServer kemudian akan mengirimkan Access Token dan Access Token Secret ke perangkat IoT.
10. Perangkat IoT akan mendapatkan credential untuk mengautentikasi dirinya ke broker MQTT. Credential tersebut diantaranya adalah ID dan password dari perangkat, Access Token, dan Access Token Secret.

Pengujian yang dilakukan oleh paper ini [4] dilakukan terhadap tiga perangkat yang berbeda, yaitu Arduino Mega 2560 (Bahasa C), Raspberry Pi 2 Model B (node.js), dan *browser* Chrome (HTML5) di laptop. Ketiga perangkat tersebut terhubung ke *AuthServer* melalui HTTP. Rata-rata *delay* yang dihasilkan untuk tahap mendapatkan ID dan password dari perangkat adalah sekitar 121 ms. Sedangkan rata-rata waktu *delay* yang dibutuhkan untuk mendapatkan *Request Token* dan *Access Token Secret* berbeda-beda tergantung perangkat yang digunakan (lihat gambar 3).

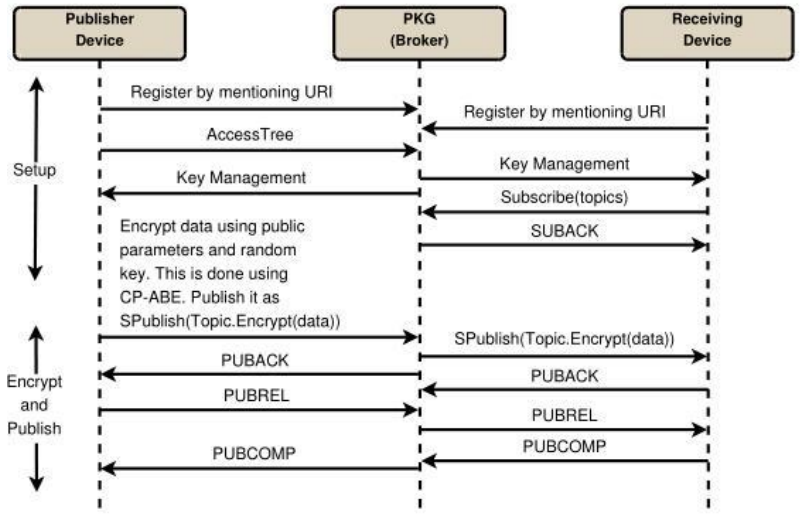


Gambar 3. *Token Request delay* dari berbagai perangkat [4]

ISBN: 978-602-60581-0-2

3.2 Mekanisme untuk Menjaga Kerahasiaan Data

Mekanisme keamanan ini memiliki fokus pada proses enkripsi data dari proses komunikasi pada perangkat IoT. Proses enkripsi tersebut dapat menjamin kerahasiaan data. Mekanisme ini telah diimplementasikan dan diuji oleh paper yang dibawakan oleh Singh, dkk [9]. Pada penelitian yang dilakukanya, asumsi yang dibuat diantaranya adalah broker (yang juga berperan sebagai Public Key Generator / PKG) mendistribusikan kunci ke perangkat yang melakukan publish dan subscribe dengan cara yang aman.



Gambar 4. Protokol *Secure* MQTT pada perangkat IoT yang diajukan oleh Singh [9]

Paper ini [9] memanfaatkan Attribute Based Encryption yang ringan berbasis teknik elliptic curve cryptography (ECC). Hal ini menjamin kerahasiaan data dan access policy, karena hanya perangkat dengan atribut yang sesuai saja yang dapat mendekripsi pesan terenkripsi yang dikirimkan oleh publisher. Protokol secure MQTT yang diajukan oleh paper tersebut dapat dilihat pada gambar 4. Dalam protokol ini, ada tiga buah entitas yang berperan. Yang pertama adalah perangkat yang berperan sebagai publisher, yang akan mengirimkan data dengan topik tertentu. Yang kedua adalah perangkat yang berperan sebagai subscriber, yang akan menerima data dengan topik tertentu melalui broker. Yang ketiga adalah broker (PKG) sebagai pihak terpercaya yang menghubungkan antara publisher dengan subscriber. Penjelasan langkah-langkah [9] yang ada di gambar 4 adalah sebagai berikut.

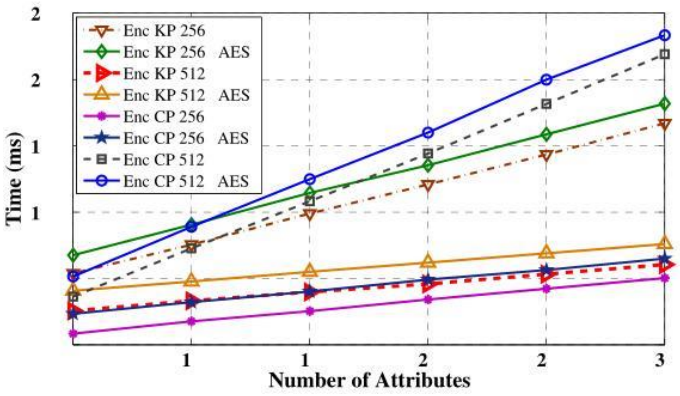
1. *Setup Phase*
   Perangkat IoT yang berperan sebagai publisher dan subscriber mendaftarkan dirinya ke PKG dengan mengirimkan identitas unik seperti Universal Resource Identifier (URI) bersama dengan atributnya. Atribut yang dikirimkan merupakan subset dari himpunan atribut universal U = {A1, A2, …, An}.
   Kemudian PKG akan membuat himpunan Master Secret Key dan parameter publik berdasarkan skema CP/KP (Ciphertext Policy / Key Policy)-ABE dan mengeluarkan parameter publik bersamaan dengan himpunan atribut universal U. Khusus untuk skema CP-ABE, semua perangkat yang memiliki atribut akan menerima himpunan kunci dari PKG.
2. *Encrypt Phase*
   Perangkat yang berperan sebagai publisher akan merancang access policy berdasarkan access tree dari himpunan atribut A dan logical connector. Untuk yang menggunakan skema KP-ABE, publisher akan mengirimkan access tree ke PKG dan kemudian PKG

ISBN: 978-602-60581-0-2

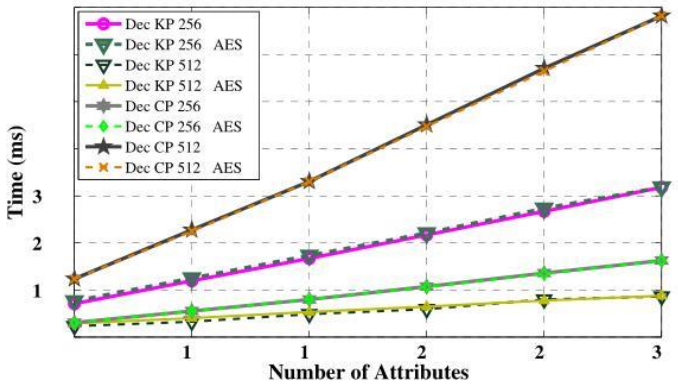
akan membuat key policy dan kunci yang sesuai. Sedangkan untuk skema CP-ABE, publisher akan membuat access tree dan access policy. Pada skema CP-ABE, publisher akan mengenkripsi payload dan menyediakan informasi tambahan untuk memfasilitasi proses dekripsi bersamaan dengan policy. Publisher akan mengenkripsi data menggunakan parameter publik dan membuat ciphertext sesuai dengan skema yang digunakan (CP/KP-ABE). Setelah itu, broker akan mengirimkan SUBACK pada perangkat yang menerima.

3. *Publish Phase*
   Publisher akan menyematkan data terenkripsi sebagai payload pada perintah Spublish. Kemudian, publisher akan mengatur nama topik yang sesuai di dalam variable header Kemudian paket Spublish ini dikirim ke broker. Setelah itu, broker akan merespon balik dengan paket PUBACK. Publisher kemudian akan mengirimkan paket PUBREL ke broker. Broker akan meneruskan pesan ke seluruh subscriber dari topik yang sesuai. Kemudian, broker akan menghapus data tersebut, dan mengirimkan paket PUBCOMP pada pengirim.
4. *Decrypt Phase*
   Perangkat yang berperan sebagai subscriber akan mendekripsi ciphertext menggunakan private attribute keys miliknya. Proses ini hanya dapat dilakukan bila access policy dari subscriber terpenuhi. Bila skema yang digunakan adalah KP-ABE, perangkat subscriber akan memeriksa apakah perangkat ini memenuhi access policy. Bila memenuhi, maka perangkat ini akan meminta PKG untuk mengirimkan kunci yang sesuai dan PKG akan memeriksa permintaan ini dan mengirimkan kunci ke subscriber. Sedangkan bila skema CP-ABE yang digunakan, subscriber yang memenuhi access policy akan mendekripsi ciphertext menggunakan kunci privat miliknya dan informasi yang disediakan dalam policy. Dengan demikian, skema CP-ABE memungkinkan proses dekripsi ciphertext tanpa interaksi dengan PKG.Ketika proses setup selesai, bila menggunakan skema CP-ABE, PKG sudah tidak dibutuhkan lagi. Dengan demikian, skema CP-ABE terlihat lebih umum dan cocok untuk melakukan scalling pada IoT, namun kompleksitas dari skema ini relatif tinggi terutama terkait storage dan proses komputasi yang dibutuhkan dibandingkan skema KP-ABE.

Pengujian performa yang dituliskan oleh *paper* ini [9], merujuk ke *paper* yang dibawakan oleh Wang, dkk [10] dapat dilihat pada gambar 5 (performa enkripsi ABE) dan gambar 6 (performa dekripsi ABE). Dari gambar, dapat dilihat bahwa secara umum, waktu yang dibutuhkan untuk proses enkripsi dan dekripsi pada skema CP-ABE lebih banyak dibandingkan bila menggunakan skema KP-ABE, disebabkan pada CP-ABE diperlukan informasi tambahan yang harus dihitung dan diberikan ke *access policy*.



Gambar 5. Performa Enkripsi ABE [9]



Gambar 6. Performa Dekripsi ABE [9]

3.3 Mekanisme untuk Menjamin Integritas Data

Mekanisme keamanan ini memiliki fokus untuk menjamin bahwa data yang diterima (oleh *broker* atau *subscriber* misalnya) merupakan data asli yang belum diubah. Penulis belum menemukan *paper* khusus yang fokus pada mekanisme ini. Namun sebenarnya pada *paper* [4] yang menggunakan OAuth telah menggunakan *signature* dengan *Hash-based Message Authentication Code* (HMAC)-SHA1. Selain itu, di *paper* yang dibawakan oleh Ntuli, dkk [11], dijelaskan pada bagian *secure communication*, untuk mengamankan komunikasi salah satunya menggunakan fungsi kriptografi hash untuk menghasilkan nilai *hash*. Nilai *hash* ini selanjutnya dikirimkan bersamaan dengan data lainya. Ketika perangkat penerima menerima pesan, maka penerima dapat memeriksa nilai *hash* dari data yang diterima dan mencocokkanya dengan nilai *hash* yang tadi telah diterima bersamaan dengan data lainya

3.4 Mekanisme untuk Menjamin Ketersediaan Sistem

Mekanisme keamanan ini memiliki fokus untuk menjamin ketersediaan sistem. Penulis belum menemukan *paper* khusus yang fokus pada mekanisme ini. Sistem IoT dengan menggunakan protokol MQTT biasanya terdiri dari tiga komponen, yaitu *publisher*, *subscriber*, dan *broker*. *Publisher* dan *subscriber* ini bergantung pada *broker* untuk saling berkomunikasi. Oleh Karena itu, *broker* harus terus tersedia dalam sistem ini. Beberapa teknik yang dapat digunakan untuk menjamin ketersediaan *broker* diantaranya adalah *broker* harus mampu mendeteksi keadaan tidak normal dari *client*. Berdasarkan referensi [7], beberapa keadaan tidak normal dari *client* ini diantaranya adalah :

1. Percobaan koneksi berulang kali.
2. Percobaan proses autentikasi berulang kali.
3. Pemutusan koneksi yang tidak normal.
4. *Topic scanning* (mencoba mengirim atau *subscribe* ke banyak topik).
5. Mengirimkan pesan yang tak dapat dikirim (tidak ada *subscriber* topik tersebut).
6. *Client* yang telah terhubung namun tidak mengirimkan data apapun.

Broker/server harus mampu mendeteksi tingkah laku *client* yang berpotensi mengganggu ketersediaan sistem. Menurut *paper* [4], aktivitas yang mencurigakan seperti serangan *brute force*, *flooding*, dan sebagainya, harus dapat terpantau sehingga sistem dapat memblokir akses dengan cara menonaktifkan atau mencabut ID perangkat dan *access token.* Selain pendeteksian dan pencegahan serangan tersebut, teknik lain yang dapat digunakan untuk dapat meningkatkan ketersediaan dari *broker* adalah dengan menerapkan *high availability* dari *broker.* Misalnya saja dengan menggunakan dua buah *broker* redundan. Kedua *broker* ini dihubungkan dengan *load balancer* yang dapat membagi beban ke masing-masing *broker*. Bila ada satu *broker* yang mati, maka secara otomatis *broker* yang satunya tetap dapat membuat sistem ini berjalan.

ISBN: 978-602-60581-0-2

4. Kesimpulan

Salah satu protokol komunikasi untuk perangkat IoT adalah MQTT. Secara default protokol ini tidak memberikan mekanisme keamanan secara menyeluruh. Mekanisme keamanan yang disediakan hanya berupa autentikasi tanpa adanya enkripsi, sehingga sangat mudah untuk diserang. Oleh karena itu, dibutuhkan mekanisme tambahan agar perangkat IoT yang kita gunakan dapat lebih aman. Mekanisme keamanan yang akan diterapkan haruslah dapat dijangkau oleh keterbatasan sumber daya yang ada di perangkat IoT. Pada makalah ini telah dibahas beberapa mekanisme yang telah diuji oleh peneliti lain, yaitu mekanisme keamanan untuk proses otorisasi, autentikasi, dan enkripsi data. Selain itu, pada makalah ini juga telah dijelaskan tentang mekanisme untuk menjamin integritas data dan mekanisme untuk menjamin ketersediaan sistem, terutama ketersediaan dari broker. Kedepanya, diharapkan ada peneliti yang dapat menggabungkan seluruh mekanisme keamanan yang telah dijelaskan dalam makalah ini agar proses komunikasi di perangkat IoT terutama komunikasi yang menggunakan protokol MQTT dapat lebih aman dari sebelumnya.

Daftar Pustaka

[1] D. Evans, "The Internet of things: how the next evolution of the Internet is changing everything," *Cisco Internet Business Solution Group White Paper*, April 2011.

[2] M. A. Iqbal and M. Bayoumi, "Secure End-to-End key establishment protocol for resource-constrained healthcare sensors in the context of IoT," *2016 International Conference on High Performance Computing & Simulation (HPCS)*, Innsbruck, 2016, pp. 523-530.

[3] ISACA Volunteer Member, "Cybersecurity Fundamentals Study Guide", 2015.

[4] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul and A. Panya, "Authorization mechanism for MQTT-based Internet of Things," *2016 IEEE International Conference on Communications Workshops (ICC)*, Kuala Lumpur, 2016, pp. 290-295.

[5] E. Borgia, D. Gomes, B. Lagesse, R. Lea, D. Puccinelli, "Special issue on "Internet of Things: Research challenges and Solutions", *Computer Communications, Volumes 89–90*, 1 September 2016, Pages 1-4, ISSN 0140-3664.

[6] E. Caltum, O. Segal, "Exploitation of IoT devices for Launching Mass-Scale Attack Campaigns," *Akamai Threat Research*, 2016.

[7] A. Banks and R. Gupta, "MQTT version 3.1.1", *OASIS Standard*, 2014.

[8] A. Prada, P. Reguera, S. Alonso, A. Morán, J. Fuertes, M. Domínguez, "Communication with resource-constrained devices through MQTT for control education," *11th IFAC Symposium on Advances in Control Education ACE 2016, Bratislava, Slovakia*, 2016, pp. 150-155.

[9] M. Singh, M. A. Rajan, V. L. Shivraj and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on, Gwalior, 2015, pp. 746-751.

[10] X. Wang, J. Zhang, E. Schooler, and M. Ion, "Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT," in *Communications (ICC)*, 2014 IEEE International Conference on, June 2014, pp. 725–730.

[11] N. Ntuli, A. Abu-Mahfouz, "A Simple Security Architecture for Smart Water Management System," *Procedia Computer Science, Volume 83*, 2016, pp. 1164-1169.

ISBN: 978-602-60581-0-2