1. Pendahuluan

Latar Belakang

Pada Internet of Things (IoT), benda sehari-hari dan mesin berada dalam peran utama dan saling berkomunikasi dengan satu sama lainnya [1]. Berdasarkan sudut pandang kemampuan dan kinerja, Perangkat IoT dibagi menjadi dua kategori yaitu Perangkat IoT *high-end* dan perangkat IoT *low-end*. Perbedaan umum antara perangkat IoT *high-end* dan *low-end* adalah perangkat IoT itu memiliki banyak sumber daya, sebagai contoh mampu menjalankan sistem operasi dikategorikan sebagai perangkat IoT kelas atas. Perbedaan umum antara perangkat IoT *high-end* dan *low-end* adalah perangkat IoT *high-end* memiliki banyak sumber daya, sebagai contoh perangkat IoT yang mampu menjalankan sebuah sistem operasi dikategorikan sebagai perangkat IoT *high-end*. Untuk perangkat IoT *low-end*, kemampuan komputasi perangkat terbatas ke sumber daya yang tersedia didalam perangkat. Dalam perangkat IoT *low-end*, terdapat tiga sub-kategori atau kelas yang berdasarkan dari ukuran memori yang digunakan [2], yaitu: *Class-0* (perangkat yang memiliki ukuran memori kurang dari 10KiB dan ukuran flash kurang dari 100KiB), *Class-1* (perangkat yang memiliki ukuran memori kurang dari sama dengan 10KiB dan ukuran flash kurang dari sama dengan 100KiB), dan *Class-2* (perangkat yang memiliki ukuran memori kurang dari sama dengan 50KiB dan ukuran flash kurang dari sama dengan 250KiB) [3]. Meskipun memiliki sumber daya terbatas, perangkat IoT *low-end* masih dapat digunakan untuk mengumpulkan dan mengirimkan data melalui Internet [4].

Pengiriman pesan dalam IoT adalah hal yang penting dikarenakan perangkat IoT harus mengirim perintah atau komunikasi yang ada pada device lain untuk mengatur sistem. Protokol push adalah salah satu jenis protokol komunikasi populer yang digunakan dalam perangkat IoT, karena protokol push dirancang untuk jaringan dengan bandwidth yang rendah pada umumnya. Protokol ini diimplementasikan dalam berbagai jenis layanan, salah satunya adalah Message Queue Telemetry Transport (MQTT). Protokol MQTT dirancang untuk bekerja dengan baik pada perangkat dengan konsumsi tenaga yang kecil [5].

Untuk menerapkan mekanisme keamanan akan membutuhkan beberapa sumber daya tambahan tergantung pada mekanisme keamanan apa akan digunakan atau dimana mekanisme keamanan tersebut akan diterapkan. Dengan menggunakan Analisis Overhead, hal ini dapat membantu untuk melacak setiap perubahan tambahan dari penggunakan sumber daya pada saat mekanisme keamanan ditambahkan [6]. Analisis Overhead juga dapat dilakukan dalam berbagai aspek seperti masalah tempat penyimpanan, konsumsi energi, perhitungan waktu pengiriman pesan atau *payload*, proses enkripsi, konsumsi memori, dan penggunaan komputasi yang terjadi setelah mekanisme keamanan diterapkan [7] [8]. Dengan Analisis Overhead juga dapat membantu menentukan solusi untuk memecahkan suatu masalah, sebagai contoh pada [9], analisis keamanan telah dilakukan pada sebuah protokol, dengan hasil terperinci yang dihasilkan dari proses analisis membantu penulis untuk menentukan solusi dari masalah yang ada. Dalam paper ini, overhead dari tanda tangan digital dalam protokol MQTT akan dianalisis untuk mencari tahu seberapa banyak overhead yang dihasilkan dalam *constrained-device*.

Topik dan Batasannya

Topik yang ingin penulis angkat adalah mengenai terbatasnya *resource* yang disediakan oleh *constrained-device* menyebabkan ukuran dari *payload* menjadi sangat terbatas, apalagi jika disertai dengan adanya program lain yang dibuat untuk melakukan proses pengamanan terhadap *payload* tersebut, sehingga *resource* yang ada harus terbagi antara *payload* dengan program yang menyediakan sistem keamanan tersebut. Selain masalah mengenai ukuran *payload*, terbatasnya *resource* pada *constrained-device* dapat menimbulkan tantangan pada proses komputasi, contohnya untuk melakukan mekanisme keamanan seperti enkripsi *payload* membutuhkan waktu yang cukup lama, sehingga dibutuhkan juga suatu algoritma yang ringan dan aman untuk meringankan proses komputasi yang ada pada *constrained-device*. Mekanisme keamanan yang penulis bahas dan lakukan adalah dengan menggunakan tanda tangan digital untuk mengamankan *payload* yang dikirim antara Publisher, Broker, dan Subscriber melalui protokol *Message Queue Telemetry Transport* (MQTT) pada tiga kelas *constrained-device*.

Tujuan

Dengan bantuan dari analisis overhead yang dilakukan melalui aspek: menganalisis waktu dekripsi, kinerja verifikasi tanda tangan, waktu pengiriman pesan, penggunaan memori dan penggunaan flash, diharapkan dapat membantu penulis untuk mengetahui kekurangan dan kelebihan mekanisme keamanan yang penulis terapkan pada tiga kelas *constrained-device* serta dapat memberikan keamanan pada *payload* yang dikirimkan hingga sampai pada tujuannya.