Comentario técnico: CTC-103
Componente: **Conexión a Google Cloud Platform desde SIMCom y otros**
Autor: Sergio R. Caprile, Senior R&D Engineer

Revisiones	Fecha	Comentarios
0	16/07/20	

En el CTC-099 analizamos la Google Cloud Platform (GCP) y desarrollamos la utilización de Cloud IoT Core, el servicio de conectividad, utilizando MQTT. Comentamos allí brevemente que la autenticación emplea JSON Web Tokens (JWT), por lo que desarrollaremos aquí algunas de las alternativas a la hora de implementarlo. Si bien el enfoque principal es desde el empleo de módulos SIMCom, las alternativas son válidas para cualquier desarrollo con conectividad a esta plataforma.

Módulos SIMCom

Los módulos SIMCom con cliente MQTT como el SIM7600, no poseen una estructura de comandos que permita soportar la generación de JWT. Si bien el módulo posee manejo de SSL y hasta comandos AT para recibir claves, el fabricante no tiene pensado introducir los comandos necesarios para esto.

Alternativas

Las alternativas disponibles se dividen fundamentalmente en dos categorías, con una tercera obvia y tal vez sugerida como mejor y más interesante: utilizar Amazon en vez de Google.

Delegar la seguridad

Dentro de este esquema podemos sugerir la recomendación de Google, que consiste en utilizar un *gateway*.[1] Dicho *gateway* se registra como tal en la *device registry* y puede recibir la delegación de la seguridad en la conexión. Su función es recibir las conexiones de otros dispositivos y entregar la información a Cloud IoT Core.
Este *gateway* debe residir, obviamente, en un sitio accesible para los dispositivos que dependen de él, lo cual, fuera de la obviedad de un *gateway* físico diseñado para que dispositivos sin conexión a la Internet puedan formar parte de la red (una red ZigBee, por ejemplo), generalmente se traduce en un servidor en la nube, con toda la logística e infraestructura asociadas. El desarrollador debe ocuparse de la seguridad en la conexión, el desarrollo del código (aunque Google provee ejemplos), la infraestructura de soporte del *gateway*, y la logística de seguridad que desee proveer a los dispositivos que utilicen este *gateway*.

Implementar la generación del JWT

Como hemos visto en el CTC-100, Mongoose-OS genera los JWT. El ESP32, además, tiene un acelerador para operaciones criptográficas basadas en curva elíptica, por lo que considerar realizar el desarrollo sobre esta plataforma es una alternativa válida.[2]
Si el desarrollo involucra módulos SIMCom, debemos tener presente que la longitud máxima del password (donde se envía el JWT) en el cliente MQTT de un SIM7600 es de 256 bytes, lo que significa que la longitud del token no debe superar este valor. Esto se logra limitando la longitud de la firma; hemos observado que Mongoose-OS, por ejemplo, utiliza 204 bytes en total.[3] Aunque en estos casos cabe preguntarse si es

1 https://cloud.google.com/iot/docs/how-tos/gateways/mqtt-bridge
2 La conexión a un módulo de telefonía celular puede hacerse por PPP y aprovechar todo lo que provee Mongoose-OS
3 mientras que en el ejemplo en la página de Google, sólo la firma es de casi 350 bytes

conveniente emplear el cliente MQTT del módulo o mejor utilizar uno open source como Paho[4]; o si directamente hacer el desarrollo en una arquitectura con mayor soporte y conectarse al módulo por PPP.

Entre las alternativas existentes a la hora de generar el JWT, disponemos de varios ejemplos para lenguajes de alto nivel,[5] soporte de Google[6], y libraries para sistemas dedicados[7] que emplean MbedTLS[8], por ejemplo. La implementación de Mongoose-OS está basada en MbedTLS, por lo que es buena idea tomarla como ejemplo.[9]

En el caso de sistemas muy limitados, retomando el enfoque analítico, cabe nuevamente preguntarse si no es conveniente utilizar una arquitectura con mayor soporte.

Finalmente, algunos módulos tienen la posibilidad de ejecutar código del usuario, lo cual permitiría incluir allí la generación del JWT.

4 https://www.eclipse.org/paho/
5 https://jwt.io/introduction/
6 https://cloud.google.com/iot/docs/how-tos/credentials/jwts
7 https://github.com/GlitchedPolygons/l8w8jwt
8 https://github.com/ARMmbed/mbedtls
9 https://github.com/mongoose-os-libs/gcp/blob/5783969990211e5a8c0810d3e93efd8a557a4120/src/mgos_gcp.c#L90