July 2022

# Multicast Implementation Over Mutual Transport Layer Security (mTLS)

Assaf Namer

# Multicast Implementation Over Mutual Transport Layer Security (mTLS)

## ABSTRACT

There is currently no straightforward way to broadcast or multicast a mutual transport layer security (mTLS) message between a server and multiple clients. If the clients are spread in different geographic regions and have varying network speeds and bandwidth, multicasting is more difficult. This disclosure describes techniques to send multicast messages to a set of endpoints over a freshly formed secure channel using temporary certificates. The task of secure channel formation is performed by functions-as-a-service (FaaS) type cloud computing, invoked in regions close to the endpoints to provide low latency and networking costs. The described techniques streamline multicast mTLS by offloading messages to serverless services rather than to the mTLS broker. The techniques can be applied to manage widely distributed endpoints.

## KEYWORDS

- Mutual transport layer security (mTLS)
- Zero trust
- Secure sockets layer (SSL)
- Internet of things (IoT)
- Multicast
- mTLS broker
- Certificate authority
- Functions as a service (FaaS)
- Serverless services

## BACKGROUND

Transport layer security (TLS) is an encryption protocol that authenticates the server in a client-server connection and encrypts communications between client and server. Mutual TLS (mTLS) is a technique for mutual (two-way) authentication that ensures that the parties at each end of a network connection are who they claim to be by verifying that they both have the correct private key. mTLS is a building block in zero-trust security architectures (no user, device, or network traffic is trusted by default) to verify users, devices, servers, application programming interfaces (APIs), etc., because mTLS ensures mutual authentications between the client and server. mTLS uses certificates to authenticate clients and servers. It can also be used in Internet of Things (IoT) applications.

Because mTLS communication between client and server is point-to-point with certificate-based authentication, there is no straightforward way to broadcast (or multicast) a message from a server to multiple clients. If the endpoints are spread across the globe and have varying network speeds and bandwidth, multicasting is more difficult. Furthermore, in some circumstances, mTLS channels to the broker are reserved for high-priority (time-sensitive) communications; out-of-band communications are therefore forced to take place on other channels while maintaining the same mTLS security control.

## DESCRIPTION

This disclosure describes techniques to send multicast messages to a set of endpoints over a freshly formed secure channel using temporary certificates. The task of secure channel formation for a set of endpoints is optimally done by functions-as-a-service (FaaS) type cloud-computing (henceforth, “cloud functions”), invoked in regions close to the endpoints to provide low latency and networking costs. The described techniques streamline multicast mTLS by offloading messages to serverless services rather than to the mTLS broker.

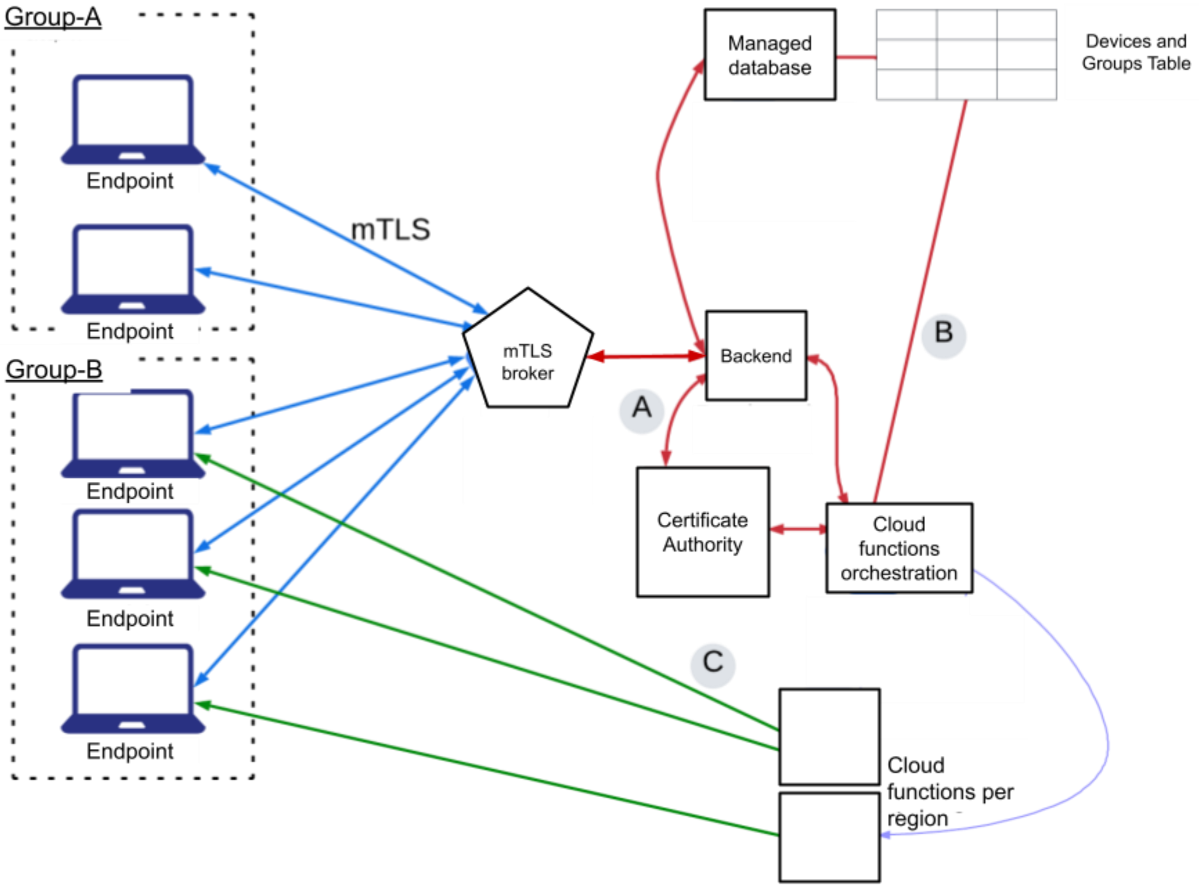

**Fig. 1: Multicast over mTLS**

Fig. 1 illustrates an example architecture for multicast over mTLS. The example illustrates two clients (endpoints), group A and group B, which can be located in different regions. The managed database holds the status of the endpoints and related metadata such as connection ID (UUID), device location, etc. When the backend (BE) initiates sending a message or data to a group of endpoints, such as group B in the example, the following procedure is set into operation:

- The BE creates a short-term certificate in the certificate authority (CA); this new certificate is used by cloud functions to establish a new channel to the endpoints. The expiry time of the certificate is dependent on the backend and may be selected to be as short as possible. The CA can be private to the customer or managed by a cloud provider. This procedure is denoted "A" in Fig. 1.
- The BE then sends the certificate (typically, a few kilobytes) and metadata (e.g., that identifies the port to use for the multicast session) to the group of endpoints. Each endpoint validates and installs the certificate, ensuring that the direct mTLS connection between the endpoint and the mTLS broker remains unsaturated.
- The endpoint opens and listens on the new TCP port as described in the metadata.
- The backend invokes cloud-function orchestration functions that consult the database to invoke cloud functions close to the endpoints ("B"). The per-region cloud functions use the same trusted CA and establish a connection to each endpoint in the region, denoted "C".
- Each cloud function uses the temporary certificate generated from the same CA to send data to each endpoint via a temporary channel. While multicast messages to groups are relayed over new sessions, this communication route preserves the always-up mTLS session free. These sessions can make use of slower and less expensive media.
- When cloud functions transmit the entire message, the temporary channels ("C") are closed, and client certificates are revoked or expired.

As described earlier, the database stores devices and their groups. To multicast a message to a group, the server writes a query to the database with the group information and the messages to send. The database emits an event upon write, which points a serverless function to the new write. The cloud function gets executed in each region and starts a new connection with each client in the group. Based on the region of the clients, the cloud function can automatically scale up to large groups. In contrast, conventional techniques to send a message to such a group (which might comprise millions of devices in hundreds of groups dispersed throughout the world) amount to the server unicasting to the mTLS connections in the group, an onerous task.

In this manner, the described techniques provide a framework and an architecture to enable numerous mTLS clients to broadcast messages to mTLS brokers. The techniques are generally applicable in situations where multicast is to be enabled over mTLS. The techniques are particularly useful when a large number of devices or endpoints (e.g., agents of various operating systems; devices of various types; etc.) linked to the server are grouped together. For example, the techniques can be applied to manage millions of endpoints (agents) on devices such as IoT, workstations, laptops, etc.

## CONCLUSION

This disclosure describes techniques to send multicast messages to a set of endpoints over a freshly formed secure channel using temporary certificates. The task of secure channel formation is performed by functions-as-a-service (FaaS) type cloud computing, invoked in regions close to the endpoints to provide low latency and networking costs. The described techniques streamline multicast mTLS by offloading messages to serverless services rather than to the mTLS broker. The techniques can be applied to manage widely distributed endpoints.