Weave Message Ids

Revision 1
2016/11/02

All weave messages contain a 32-bit message id assigned by the sender of the message. Weave message ids serve several purposes:

Duplicate Message Detection – Receiving systems use message ids to detect messages that have been retransmitted by the sender, e.g. in response to packet loss in the network.

Message Acknowledgement – In the Weave Reliable Messaging protocol, message ids provide a way for receivers to identify messages for the purpose of acknowledging their receipt.

Encryption Nonces – When encrypted messages are sent, message ids provide an encryption nonce that ensures each message is encrypted in a unique manner.

Replay Prevention – Related to encryption, Weave message ids also provide a means for detecting and preventing the replay of encrypted messages.

In general, message ids are assigned sequentially, by incrementing a counter value maintained by the sender of the message. Each party in a two- (or many-) way conversation maintains its own message id counters, with different counters being used for different types of messages.

Global Message Id Counters

Weave nodes implement two global 32-bit counters to vend message ids for certain types of messages. An *unencrypted message id counter* is used to generate ids for unencrypted messages, while an *encrypted message id counter* is used to generate ids for messages encrypted using group keys. Some Weave nodes may choose not to use group keys, in which case they can omit the encrypted global message id counter. All nodes however must implement the unencrypted message id counter.

Typically Weave nodes will store the global unencrypted message id counter in RAM. This makes the counter subject to loss whenever the system reboots or otherwise loses its state. This is permissible because retaining the unencrypted message id counter is not essential to the confidentiality or integrity of the message. In the event that the unencrypted message id counter is lost, Weave nodes are required to randomize the initial value of the counter on startup. In doing this, implementations must use a random number generator that has been seeded from a true random number source.

In contrast to the unencrypted message id counter, Weave nodes are required to persist the global encrypted message id counter in durable storage. In particular, Weave nodes are required to ensure that the value of the encrypted message id counter never rolls back--i.e. that it is monotonic within the bounds of its range.

Session Message Id Counters

Messages that are encrypted using Weave session keys use ids generated from a per-session ephemeral counter. Session message id counters exist for as long as the associated security session is in effect. Like the global message id counters, session message id counters are 32 bits long. However they are never allowed to wrap; rather, sessions are expected to be discarded and re-established as needed before the counter overflows. Given the size of the counter, and the nature of Weave devices, this is rarely an issue in practice.

Message Ids as Encryption Nonces

In the context of encrypted Weave messages, message ids serve as nonces for the encryption algorithm, ensuring that every message is encrypted in a unique manner. The uniqueness of an encrypted message's id is vital to the confidentiality of the message, as the encryption algorithms make it trivial for an eavesdropper to decrypt messages if they can find two different messages with the same message id that were encrypted using the same key.

Weave nodes ensure the uniqueness of message ids by persisting the value of the global secure message id counter in durable storage. Additionally, nodes must rotate their encryption keys on a regular basis, to ensure that old encryption keys are retired before the global message id counter has a chance to wrap. In practice, the frequency of message transmission is such that encryption keys generally rotate at a rate that is much faster than the rate at which the global counter wraps. In the unlikely event that the global encrypted message id counter does wrap before the associated keys are rotated, Weave devices will still continue to communicate using old keys, despite the fact that messages will be sent using already used message ids. This reflects the policy that it is more important for Weave devices to continue to communicate with each other than it is to ensure the confidentiality of their communication. Despite this policy, confidentiality is important and every effort must be made to ensure that keys are rotated on a regular basis.

Replay Prevention and Duplicate Message Detection

Beyond their role as encryption nonces, message ids also serve as a means to detect repeated reception of the same message. Message duplication may occur due to network error--e.g. because a sender retransmitted a message after failing to receive an acknowledgement--or because a malicious third party attempted to replay an old message to gain some advantage. To detect duplicate messages, Weave nodes maintain a history of the messages they have received from a particular sender (see Message Reception State below). Whenever a message is received, its message id is checked against the history of messages from that sender to determine whether it is a duplicate. By default the Weave messaging layer discards duplicate messages before they reach the application layer. However, as a local configuration option, Weave applications may enable the reception of duplicate messages for particular purposes (e.g. to assess packet density in a retransmission-based trickle protocol). Whenever duplicate messages are delivered to an application they are flagged as such by the message layer to warn the application of the potential security implications.

To guard against replay attacks, Weave nodes must maintain the history of received messages for at least as long as the keys used to encrypt the messages are still active. For messages encrypted with session keys, the message history is maintained for the lifetime of the session, and is at the same time as the session keys, when the session ends. For messages encrypted with fabric or group keys the message history must be maintained until the associated keys have been rotated and are no longer in use.

Message Reception State

The state maintained by a node about the messages it has received from a particular peer is referred to as ***message reception state***. Nodes use this state information to determine whether a newly arrived message is a duplicate of a previously message. Message reception state is maintained on a per-peer or per-session basis, depending on the type of message encryption being used.

At a conceptual level, message reception state consists of a set of integers corresponding to the ids of all the messages that have been received from a particular peer. To limit the amount of memory required to store this state, Weave nodes employ a lossy compression scheme that takes advantage of the fact that message ids are generated sequentially by the sender. The scheme allows for a limited amount of out-of-order message arrivals due to network effects (e.g. flapping network routes) without inducing false detection of duplicates.

In the compressed form, message reception state is structured as a pair of values: a integer representing the maximum message id received from the peer, and an array of booleans indicating which messages immediately prior to the max message have been received. The index into the boolean array equates to the difference in the corresponding message id relative to the max message id -- i.e. the first slot in the array indicates whether the message with the id of max-message-id - 1 has been received, the second indicates whether message max-message-id - 2 has been received, and so on. As messages arrive, the message reception state is queried to determine if an arriving message is new or duplicate. If a message is new, the state is then updated to reflect the arrival of the message. When a message arrives with a logically greater message id than the current maximum message id, the maximum message id value is updated and the array of booleans shifted accordingly.

At a minimum, Weave nodes are required to be capable of tracking range of 16 message ids, which corresponds to the maximum message id plus an array of 15 booleans for previous messages.

Use of Message Reception State for Encrypted Messages

The algorithm for querying and updating message reception state varies slightly depending on whether the system is tracking reception of encrypted messages or unencrypted messages. For encrypted messages, any arriving message with an id in the range (max-message-id + 1) to (max-message-id + 2^{31} - 1) (modulo 2^{32}) is considered new, and causes the max message id value to be updated. Messages with ids from (max-message-id - 2^{31}) to (max-message-id - array-length - 1) (modulo 2^{32}) are considered duplicate. Message ids within the range of the boolean array are considered duplicate if the corresponding boolean value is set to true.

The scheme for encrypted messages effectively divides the message id space in half: those ids that are forward of the max message id, which are considered new, and those ids that are behind the max message id, which are considered duplicates unless indicated otherwise by the values in the boolean array.

Use of Message Reception State for Unencrypted Messages

For unencrypted messages, the algorithms for tracking messages and detecting duplicates are similar, but more permissive than for encrypted messages. This reflects the fact that duplicate detection of unencrypted messages is not done for security reasons, but rather to catch duplicates caused by network errors (e.g. loss of an ack), which are generally more bounded in time. The more relaxed algorithm for unencrypted duplicate detection also relaxes the durability requirement on the sender's message id counter, allowing senders to store the counter in RAM.

For unencrypted messages, any message with an id from (max-message-id + 1) to (max-message-id - array-length) (modulo 2^{32}) is considered new, as well as message ids within the range of the boolean array where the correspond boolean value is set to false. All other message ids are considered duplicates.