Technical Disclosure Commons

Defensive Publications Series

October 29, 2018

FRAME COUNTER IN LOW-POWER AND LOSSY NETWORK

Yajun Xia

Feiliang Wang

Li Zhao

Chuanwei Li

FRAME COUNTER IN LOW-POWER AND LOSSY NETWORK

AUTHORS:
Yajun Xia
Feiliang Wang
Li Zhao
Chuanwei Li

ABSTRACT

Techniques are described herein for using parameters such as Group Temporal Key (GTK) rest lifetime, GTK whole lifetime, and frame counter length/max packets per second to acquire a safe, available, and non-repeated initial frame counter. A node may use this frame counter to communicate with its neighbor node after joining any Personal Area Network (PAN).

DETAILED DESCRIPTION

In mesh networks, nodes are always authenticated in the dot1x framework. The encryption keys including Pairwise Transient Key (PTK) and Group Temporal Key (GTK) are obtained with an Institute of Electrical and Electronics Engineers (IEEE) 802.11i-2004 four-way handshake. Data packets are encrypted with GTK to ensure security.

A mesh network is a narrow bandwidth network. In order to reduce the time of GTK synchronization in the whole Personal Area Network (PAN) (which costs a lot of resources), GTK lifetime is always set to very long. In some mesh networks, the default GTK lifetime is 30 days, and the maximum lifetime is 120 days.

Unlike the six-byte frame counter in the Wi-Fi® network, IEEE-Std-802.15.4TM-2015 defines a four-byte frame counter in the AUX header. To prevent the replay attack, the frame counter should increment for each data packet during the current GTK lifetime. If the packet is received from a node with a smaller frame counter than the previous one, the packet needs to be dropped by the receiver.

The problem is, during the current GTK lifetime, how to ensure that the frame counter keeps on increasing for each packet from a node. The node may be shut down at any time, and the flash is vulnerable after many erase/write operations. Saving the previous frame counter and using an appropriate frame counter is also a problem.

Published by Technical Disclosure Commons, 2018

Figure 1 below illustrates the usual method to save the frame counter into flash memory periodically in order to ensure that the frame counter increases for each packet from a sender during current GTK lifetime.

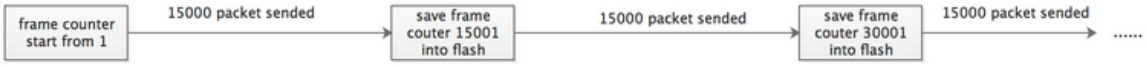

Figure 1

There are at least two issues with this mechanism. First, erasing/writing the flash memory frequently might break the flash memory. Second, since the node can be restarted at any time, the saved frame counter might be an old value.

To resolve the first issue, a usual way is to save the flash memory at a longer interval, for example 24 hours. The theoretical maximum throughput may be counted for small data packets. For example, with a bandwidth of 150kbps, the smallest encryption packet size is 81 bytes, the maximum packets per second is 150*1024/(81*8) = 237. The maximum packets sent in one day is 237*3600*24 = 20,476,800 packets. Thus, the node should save the frame counter into flash memory every 20,476,800 packets. However, the disadvantage is that the frame counter still needs to be saved into flash memory, and an individual flash sector is needed to save it to ensure the other flash sectors are not influenced.

To resolve the second issue, the node may use the frame counter of "saved frame counter value plus frame counter save interval (20,476,800)" to rejoin the PAN after it reloads. Since the frame counter length in the AUX header is 4 bytes (a maximum of 2^{32} frame counters can be used), this avoids the frame counter rollover during the GTK lifetime. The reboot times for this node may be less than 209 times (2^{32}/20476800). However, the disadvantage is that the reboot times may be larger than 209, and therefore the node may not be able to join the PAN after rebooting 209 times.

Another issue is that, in reality, there may be a lot of PANs in a small area. In this case, PAN migration may occur frequently, as illustrated in Figure 2 below.

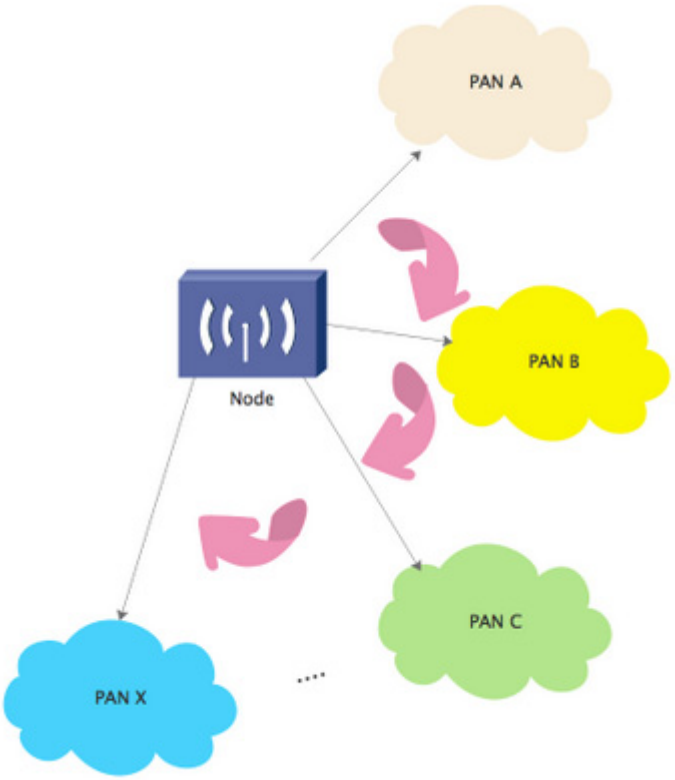

Figure 2

In this example, to ensure the node can join/rejoin all the PANs during the migration, the node should save all the PAN frame counters into the flash memory, otherwise the node may not rejoin the old PAN. For example: Node ----(Join)--->PAN A----(migrate to)----->PAN B----(migrate to)---->PAN C----(migrate to)---->PAN D----(migrate to)---->PAN E----(migrate to)---->PAN A.

In this case, the node has joined PAN A, and after some migration, it tries to rejoin PAN A. If the node has not saved its last frame counter in PAN A, and the old GTK is not overdue in PAN A, the node might not rejoin the PAN. For the neighbors that have saved the last frame counter for this node, those neighbors would drop the packets from this node with a smaller frame counter to prevent replay attacks until the old GTK becomes overdue.

In some mesh networks, PAN migration may take four hours. To make sure the node can rejoin an old PAN, the node should save context for 120 (max GTK lifetime) * 24 hours / 4 hours = 720 PANs (including at least the PAN identifier, GTK information, and frame counter information) into flash memory and Random Access Memory (RAM).

However, this is an unacceptable requirement because these resources are very limited in mesh networks.

Accordingly, techniques are described herein to resolve the aforementioned problems caused by saving frame counters into flash memory.

The frame counter in the AUX header is included in the encrypted data packet, which can only be transmitted after the node joins the PAN. A reasonable initial frame counter is counted when the node joins the PAN regardless of whether it joined the PAN previously.

In mesh networks, the rest lifetime for each GTK may be sent to the node by a router during the four-way handshake. When the node gets the whole lifetime of the current GTK, it can count an absolute available initial frame counter value for a node to communicate with its neighbor. The formula may be as follows: Initial frame counter = A * (1 - rest lifetime of current GTK / whole lifetime of current GTK).

The available value of A may be in range of the maximum number of packets sent per second for a specific bandwidth * 3600 seconds * 24 hours * GTK lifetime ~ 2^{32}.

The whole GTK lifetime may be obtained from the router during the 802.11i four-way handshake, or some other protocol message. The whole GTK lifetime may also be hard-coded in the node with the maximum value (e.g., 120 days).

With these techniques, the node does not need to save the frame counter and related PAN information into the flash memory, regardless of the number of times the node needs to be rebooted, or the number of times the node migration occurs between PANs. Thus, limited node RAM/flash resources can be saved. The node may always gets the available initial frame counter to communicate with its neighbor. Moreover, the present techniques account for frame counter roll-over within the GTK lifetime.

In summary, techniques are described herein for using parameters such as GTK rest lifetime, GTK whole lifetime, and frame counter length/max packets per second to acquire a safe, available, and non-repeated initial frame counter. A node may use this frame counter to communicate with its neighbor node after joining any PAN).