syn1588 ®

Application Note

PTPv2.1 Security

Version 1.0 - August 2019

Abstract

This application note describes the use and configuration of PTPv2.1 Security as specified in IEEE1588-2018.

Overview

The PTPv2.1 Security protocol deals with securing PTP traffic and prevents Man-in-the-middle attacks, Replay attacks – where old PTP messages are sent again – as well as malicious Master spoofing.

PTP messages are secured using a dedicated TLV that may be appended to all message types, depending on the configuration. The authenticity and integrity of a secure PTP message is verified with the use of a checksum and symmetric keys.

Note: Key distribution is not handled by the syn1588® PTP Stack and should be done externally. See Chapter Configuration for details.

Features

The implementation of PTPv2.1 Security in the syn1588® PTP Stack supports immediate message authentication, which means that each applicable PTP message is verified as soon as it is received by the syn1588® PTP Stack. In case a message's authenticity cannot be verified, the message is discarded and no data is processed.

The use of Transparent Clocks is not supported by PTPv2.1 Security since message integrity cannot be verified after the correction field in a message has been updated. See chapter Configuration for details.

Configuration

Base configuration via program parameters

Command line	Config file	Effect
-X	security true	Enable PTPv2.1 Security
-w allowmutable	--	Support Transparent Clocks

To enable PTPv2.1 Security, start the syn1588® PTP Stack with command line parameter **-X**. This parameter enforces message verification both at ingress and egress.

In the default configuration of PTPv2.1 Security, the use of Transparent Clocks (TCs) is not supported. In case a TC is unavoidable in the PTP network, start the syn1588® PTP Stack with the additional parameter **-w allowmutable**. When activating this setting, the correction field of a PTP message is not taken into account at verification.

Note: Use this setting at your own risk!

Further configuration via Shared Memory API

To alleviate any safety concerns, the desired configuration of PTPv2.1 Security cannot be set via program arguments and must be configured using the Shared Memory API.

The configuration of PTPv2.1 Security is split into two distinct data structures, the Security Association (SA), and the Security Policy Database (SPD). Both structures are elaborated in the following.

Security Association

The Security Association (SA) defines a set of rules needed to verify a PTP message's integrity. Multiple SAs may be configured in a single PTP instance.

Security Association Data Structure	
spp	Unique identifier for this Security Association
integrityAlgType	x Algorithm to use in checksum calculation
icvLen	Length of checksum, determined by algorithm.
keyLen	Length of the key, determined by algorithm
keys[]	One or more keys
useSeqNo	* Use the sequence number field
seqNoLen	* Length of sequence numbers in bytes, if applicable
useRes	* use reserved field
ResLen	* length of reserved field in bytes , if applicable
curKeyIdx	Index of key to use for checksum calculation at egress
useImmediateSec	Use immediate message authentication (only 'true' supported)

Table 1 - SA data fields

*** Not supported**

x Only HMAC SHA256-128 supported

A number of settings are not yet supported in the PTPv2.1 Security implementation, since the use of those settings is not specified in the base PTP2.1 standard and depend on the future publication of PTP profiles that make use of the Security feature.

Security Policy Database

The Security Policy database (SPD) provides a general guideline for secure message processing per PTP instance. This includes the minimal algorithm to be used (only HMAC SHA256-128 supported), which PTP message types should be secured, and which SA to use at egress. Exactly one SPD per PTP instance may be configured.

Security Policy Database Data Structure	
minAlgorithm	[x] Minimal algorithm to be used
tx_spp	Unique identifier of the SA to use at egress.
msgTypes	Set which PTP messages types shall be secured

Table 2 - SPD data fields

[x] **Only HMAC SHA256-128 supported**

For a thorough example on how to configure the PTPv2.1 Security feature, see the example program **sm_example_security.cpp** in the *examples* folder of the shared Memory API.

Version history

Version 0.1 - August 2019 - Ramharter

Initial version