FALCOM

Advanced Encryption Standard (AES_TCP)

PREMIUM FEATURE

in AVL firmware 2.10.0 and above

Application Note

Version: 1.0.2; Modified: Friday 15 April 2016

Version history:

This table provides a summary of the document revisions.

Version	Author	Changes	Modified
		-	
1.0.2	F. Beqiri	- Removed "," delimiter within the commands TCP.Client.RxKey=<"key"> and TCP.Client.TxKey=<"key">	15/04/2016
1.0.1	F. Beqiri	- Unnamed: PAID-FEATURES to PREMIUM-FEATURES	19/02/2015
1.0.0	F. Beqiri	- Initial version	29/10/2014

Table of contents

1 ABOUT THIS DOCUMENT.....5

1.1 Advanced Encryption Standard (AES128) within AVL devices ?.....5

1.1.1 How does AES_TCP encryption work ?.....5

1.1.2 How to define the AES_TCP encryption mode ?.....6

1.1.3 How to set up the AES_TCP encryption keys using PFAL commands ?.....7

1.2 What is required to activate a PREMIUM-FEATURE?.....8

Cautions

Information furnished herein by FALCOM is believed to be accurate and reliable. However, no responsibility is assumed for its use. Please, read carefully the safety precautions.

If you have any technical questions regarding this document or the product described in it, please contact your vendor.

General information about FALCOM and its range of products are available at the following Internet address: http://www.falcom.de/

Trademarks

Some mentioned products are registered trademarks of their respective companies.

Copyright

FALCOM GmbH.

No patent liability is assumed with respect to the use of the information contained herein.

Note

Specifications and information given in this document are subject to change by FALCOM without notice.

1 ABOUT THIS DOCUMENT

This application note provides information about how the Advanced Encryption Standard (AES_TCP) feature in the ALV firmware 2.10.x and higher developed by FALCOM works.

- AES_TCP:

1.1 Advanced Encryption Standard (AES128) within AVL devices ?

Advanced Encryption Standard is the process of transforming plain text using a cipher to make it unreadable to anyone except those possessing the key.

The Advanced Encryption Standard (AES_TCP) encryption feature developed by FALCOM secures your data by encrypting it when it is sent from the AVL device over the Internet to the destination server and decrypting it when receiving encrypted from the server using 128-bit group encryption with 128 key length. Outgoing data is encrypted immediately within the device and can be stored in encrypted format until it can be actually sent out via TCP (-> refer to FLASH TCP buffer for more information).

FALCOM AVL devices with activated AES_TCP will allow fleet managers to:

- Secure the traffic data between the AVL device and server.
- Protect data from unauthorized access.

For more information about the Advanced Encryption Standard how it works visit :

http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

1.1.1 How does AES_TCP encryption work ?

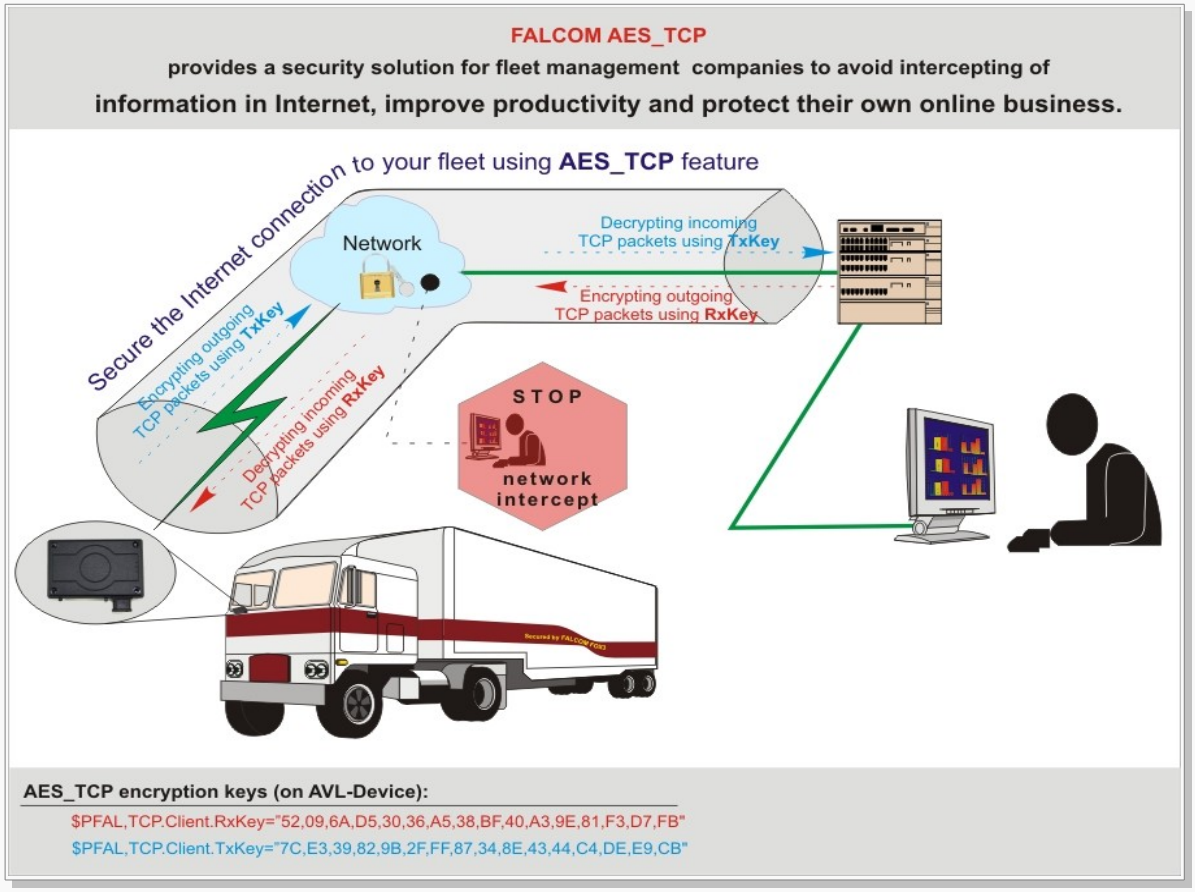
Everytime when an AVL device initiates a TCP connection to a remote server, it transfers the login data as plain text, like the example below:

```
-$<MSG.Info.ServerLogin>
$DeviceName=Unnamed AVL
$Security=1                    // indicates to the server which AES128 encryption mode the AVL device is going to use
$Software=avl_2.11.0 (BxFTVEVQUElJSSByZXY6MTEtTgEf)
$Hardware=STEPPIII rev:11-N
$LastValidPosition=$GPRMC,134418.001,A,5040.4244,N,01058.8101,E,0.33,112.70,181013,,
$IMEI=357023003010510
$PhoneNumber=017618042501
$LocalIP=10.41.54.118
$CmdVersion=2
$SUCCESS
$<end>
```

The line [$Security=X] in the login data tells the destination server that future messages from the AVL device will be encrypted or not. Where **X** can be : **0**, **1** or **2** (**0** = plain text [no encryption]; **1** = AES encryption with ECB cipher mode; **2** = AES encryption with CBC cipher mode).

After the remote server knows what kind of encryption the AVL device is going to start, both (server and AVL device) use the encrypt/decrypt keys to encrypt and decrypt the data, they send to each other.

It is not recommended to use the same AES key for all AVL devices with activated AES_TCP feature.



How to define the AES_TCP encryption mode, refer to the chapter 1.1.2 below. How to set the encrypt/decrypt keys is explained in chapter 1.1.3 below.

1.1.2 How to define the AES_TCP encryption mode ?

Define the mode for AES_TCP encryption

Syntax	TCP.CLIENT.LOGIN=1,<security_mode>
Examples	$PFAL,Cnf.Set,TCP.CLIENT.LOGIN=1,1

To enable and use this setting, the PREMIUM Feature "AES_TCP" will be required.

<security_mode> It defines how the information between the AVL device and the server will be transferred. For more details about the authenticate encryption mode of operation, visit: http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation.

It can be set to:

Value	Meaning
0	Plain text as default mode (no AES encryption)
1	AES encryption, Electronic Codebook (ECB mode)
2	AES encryption, Cipher Block Chaining (CBC mode)

1.1.3 How to set up the AES_TCP encryption keys using PFAL commands ?

Decrypting incoming TCP packets

Command syntax	TCP.Client.RxKey=<”decrypting_key">
Examples	$PFAL,TCP.Client.RxKey=”17A1BB67A8A9012396F2DAB63C603A5F"

This command sets a 16 hexadecimal key used for decrypting incoming TCP packets from an encrypted transmission. The (PREMIUM) feature AES_TCP must be enabled to support this function. It is recommended to reset firmware after setting up the key to safely restart TCP connection with the new key.

<"decrypting_key">

Separated by commas or another "non-hexadecimal" sign, it defines a 16 hexadecimal key for decrypting incoming packets between the device and destination TCP server.

Encrypting outgoing TCP packets

Command syntax	TCP.Client.TxKey=<”encrypting_key">
Examples	$PFAL,TCP.Client.TxKey=”07AABA8908AB1122AAF9CCBB336AAAFF"

This command sets a 16 hexadecimal key used for encrypting outgoing TCP packets to the destination server. The (PREMIUM) feature AES_TCP must be enabled to support this function. It is recommended to reset firmware after setting up the key to safely restart TCP connection with the new key.

<"encrypting_key">

Separated by commas or another "non-hexadecimal" sign, it defines a 16 hexadecimal key for encrypting outgoing packets to the destination TCP server

1.2 What is required to activate a PREMIUM-FEATURE?

Please refer to the application note "**AppNotes_HowToActivatePremiumFeatures.pdf**".

After activation, set the 16 hexadecimal keys for encrypting and decrypting TCP packets transmitted between AVL device and destination server and start a AES_TCP connection. The encryption/decryption keys should be setup in both AVL device and your TCP server (see the schematic in chapter 1.1.1).

Examples:

Encrypting key	$PFAL,TCP.Client.TxKey="07AABA8908AB1122AAF9CCBB336AAAFF"
Decrypting key	$PFAL,TCP.Client.RxKey="17A1BB67A8A9012396F2DAB63C603A5F"
Start AES_TCP	$PFAL,Cnf.Set,TCP.CLIENT.LOGIN=1,1 *// ECB encryption mode of operation* $PFAL,Cnf.Set,TCP.CLIENT.LOGIN=1,2 *// CBC encryption mode of operation*

Encryption example: AVL device sends a GPRMC protocol to the server (AES encryption ECB mode)

TxKey	$PFAL,TCP.Client.TxKey="07AABA8908AB1122AAF9CCBB336AAAFF"
Plaintext	$GPRMC,133725.569,A,5040.4365,N,01058.5650,E,0.05,302.98,251004,
Ciphertext in Hex	4f3d18bb24a66bc8500f18f4d900a284d894f48ec9b89ac0f632ce5069232d70aaa0 d0429367a17bb85c0ca52487b6bf10dbaadd4b936d4c1b9a39ed752ea4b6

At the end reset the AVL device to safely restart the TCP connection with the new encryption/decryption keys.