Yealink

Configuration Encryption Tool User Guide

Jan. 2019

SOFTWARE LICENSE AGREEMENT

SOFTWARE LICENSE AGREEMENT FOR YEALINK CONFIGURATION CONVERSION TOOL IS IMPORTANT. PLEASE READ THIS LICENSE AGREEMENT CAREFULLY BEFORE CONTINUING WITH THIS PROGRAM: YEALINK(XIAMEN) NETWORK TECHNOLOGY CO., LTD Software License Agreement (SLA) is a legal agreement between you (either an individual or a single entity) and Yealink(Xiamen) Network Technology CO., LTD. For the Yealink software product(s) identified above which may include associated software components, media, printed materials, and “online” or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this SLA. This license agreement represents the entire agreement concerning the program between you and Yealink(Xiamen) Network Technology CO., LTD., (referred to as "licenser"), and it supersedes any prior proposal, representation, or understanding between the parties. If you do not agree to the terms of this SLA, do not install or use the SOFTWARE PRODUCT.

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

COPYRIGHT

All title, including but not limited to copyrights, in and to the SOFTWARE PRODUCT and any copies thereof are owned by Yealink(Xiamen) Network Technology CO., LTD. or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by Yealink(Xiamen) Network Technology CO., LTD.

WARRANTIES

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS GUIDE ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS GUIDE ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF PRODUCTS. YEALINK(XIAMEN) NETWORK TECHNOLOGY CO., LTD. MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS GUIDE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Yealink(Xiamen) Network Technology CO., LTD. shall not be liable for errors contained herein nor for incidental or consequential damages in connection with the furnishing, performance, or use of this guide.

About This Guide

Configuration files contain sensitive information such as user accounts, login passwords, registration or information. To protect sensitive information from tampering, you must encrypt configuration files. Yealink provides tools for encrypting configuration files on Windows platform and Linux platform respectively.

You can use the encryption tool to encrypt following three types of configuration files:

- MAC-Oriented CFG file: <MAC>.cfg
- Common CFG file: y0000000000xx.cfg
- Other custom CFG file: sip.cfg, account.cfg and so on. It only applicable to the IP phones use new auto provisioning mechanism.

You can ask the distributor or the Yealink Field Application Engineer for these tools, or you can download them online: http://support.yealink.com/documentFront/forwardToDocumentFrontDisplayPage.

This guide provides detailed information on how to encrypt configuration files using Yealink-supplied encryption tools, and how to deploy Yealink IP phones using these encrypted configuration files. The information applies to the following Yealink IP phones:

- SIP-T46G, SIP-T42G, SIP-T41P IP and CP860 IP phones running firmware version 71 or later.
- SIP-T48G IP phones running firmware version 72 or later.
- W52P IP DECT phones running firmware version 73 or later.
- SIP-T58A, SIP VP-T49G, SIP-T40P, SIP-T29G, SIP-T27P, SIP-T23P/G, SIP-T21(P) E2, SIP-T19(P) E2 and W56P IP phones running firmware version 80 or later.
- SIP-T54S, SIP-T52S, SIP-T48S, SIP-T46S, SIP-T42S, SIP-T41S, SIP-T40G and SIP-T27G IP phones running firmware version 81 or later.
- VP59 IP phones running firmware version 83 or later.
- SIP-T54W/T53W/T53 IP phones running firmware version 84 or later.

The following IP phones use the new auto provisioning mechanism:

- SIP-T58A IP phones running firmware version 80 or later.
- SIP-T54S/T52S/T48G/T48S/T46G/T46S/T42G/T42S/T41P/T41S/T40P/T40G/T29G/T27P/T27G/T23P/T23G/T21(P) E2/T19(P) E2 IP phones running firmware version 81 or later.
- VP59 IP phones running firmware version 83 or later.
- SIP-T54W/T53W/T53 IP phones running firmware version 84 or later.

Other IP phones or the IP phones listed above running old firmware version use the old auto provisioning mechanism.

Introduction

The encryption tool encrypts plaintext configuration files (one by one or in batch) using 16-character symmetric keys (the same or different keys for configuration files) and generates encrypted configuration files with the same file name as before. This tool also encrypts the plaintext 16-character symmetric keys using a fixed key, which is the same as the one built in the IP phone, and generates new files named as <xx_Security>.enc (xx indicates the name of the configuration file, for example, y000000000000_Security.enc for y000000000000.cfg file). This tool generates another new file named as Aeskey.txt storing the plaintext 16-character symmetric keys for each configuration file.

For the security reasons, administrator should upload encrypted configuration files, <xx_Security>.enc files to the root directory of the provisioning server. During auto provisioning, the IP phone requests to download the boot file first and then download the referenced configuration files. If the downloaded configuration file (for example, <y0000000000xx>.cfg file) is encrypted, the IP phone will request to download <y0000000000xx_Security>.enc file (if enabled) and decrypt it into the plaintext key (e.g., key2) using the built-in key (e.g., key1). Then the IP phone decrypts <y0000000000xx>.cfg file using key2. After decryption, the IP phone resolves configuration files and updates configuration settings onto the IP phone system. The way the IP phone processes the other CFG file is the same to that of the<y0000000000xx>.cfg file.

The boot file is only applicable to the IP phones use new auto provisioning mechanism.

Encrypting Configuration Files on Windows Platform

Encrypting Configuration Files via Graphical Tool

This tool supports Microsoft Windows XP and Windows 7 (both 32-bit and 64-bit) system.

To encrypt configuration files:

1. Double click "Config_Encrypt_Tool.exe" to start the application tool.

   The screenshot of the main page is shown as below:

When you start the application tool, a file folder named "Encrypted" is created automatically in the directory where the application tool is located.

2. Click **Browse** to locate configuration file(s) (e.g., y000000000000.cfg) from your local system in the **Select File(s)** field.

   To select multiple configuration files, you can select the first file and then press and hold the Ctrl key and select the next files.

3. (Optional.) Click **Browse** to locate a target directory from your local system in the **Target Directory** field.

   The tool uses the file folder "Encrypted" as the target directory by default.

4. (Optional.) Mark the desired radio box in the **AES Model** field.

   If you mark the **Manual** radio box, you can enter an AES key in the **AES KEY** field or click **Re-Generate** to generate an AES key in the **AES KEY** field. The configuration file(s) will be encrypted using the AES key in the **AES KEY** field.

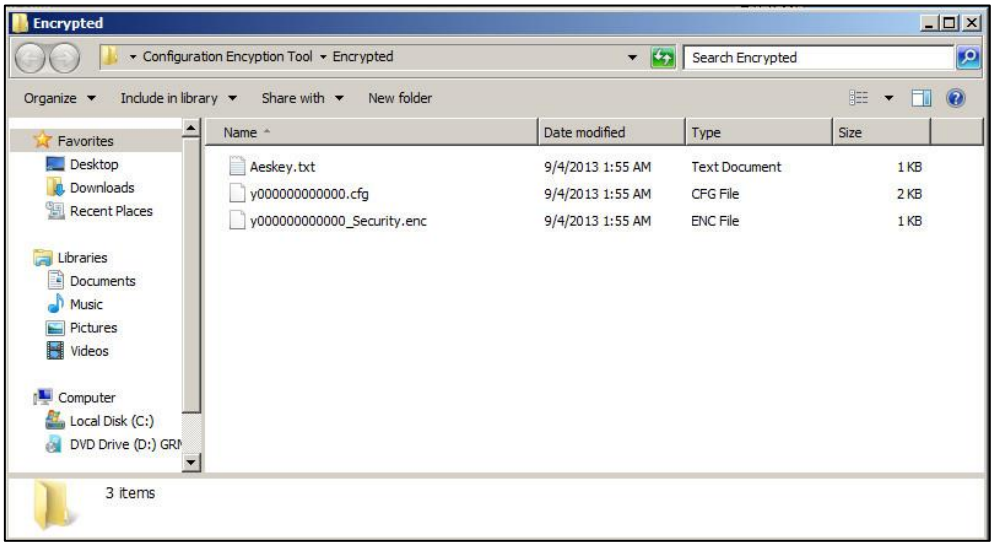
   If you mark the **Auto Generate** radio box, the configuration file(s) will be encrypted using random AES key. The AES keys of configuration files are different.

Note AES keys must be 16 characters and the supported characters contain: 0 ~ 9, A ~ Z, a ~ z and the following special characters are also supported: # $ % * + , - . : = ? @ [ ] ^ _ { } ~.

5. Click **Encrypt** to encrypt the configuration file(s).

6. Click **OK**.

The target directory will be automatically opened. You can find the encrypted CFG file(s), encrypted key file(s) and an Aeskey.txt file storing plaintext AES key(s).



Encrypting Configuration Files via DOS Command Line

To encrypt configuration files:

1. Place the encryption tool "Config_Encrypt.exe" and configuration files in the same directory.
2. Open a Command prompt.
3. Locate the directory where the encryption tool is stored.
4. Execute one of the following commands according to your requirements:

   Config_Encrypt.exe -f file1.cfg [file2.cfg ...] [-p DESTPATH(Default as 'Encrypted')] [-k AESKEY(Default as random)]

   Example:

   Config_Encrypt.exe -f y000000000000.cfg -p D:\test -k 0123456789123456

   AES Key: 0123456789123456

   Generate Security Key File...

   Generate Encrypt Config File...

   Write file to D:\test\Aeskey.txt!

   Write file to D:\test\y000000000000_Security.enc!

   Read file y000000000000.cfg!

   Write file to D:\test\y000000000000.cfg!

This tool will encrypt the y000000000000.cfg file using the AES key 0123456789123456. You can find the encrypted y000000000000.cfg file, y000000000000_Security.enc file and an Aeskey.txt file storing the plaintext AES key 0123456789123456 in the specified directory.

Note AES keys must be 16 characters and the supported characters contain: 0 ~ 9, A ~ Z, a ~ z and the following special characters are also supported: # $ % * + , - . : = ? @ [ ] ^ _ { } ~.

Encrypting Configuration Files on Linux Platform

To encrypt configuration files:

1. Place the encryption tool "yealinkencrypt" and configuration files in the same directory.
2. Open a terminal window.
3. Execute the **cd** command to locate the directory where the encryption tool is stored. For example, execute **cd/tmp** to locate the **/tmp** directory.
4. Execute one of the following commands according to your requirements:

   - If you want to encrypt one or multiple specified configuration files, you need to execute the following command:

     ./yealinkencrypt -f file1.cfg [file2.cfg ...] [-p DESTPATH(Default as 'Encrypted')] [-k AESKEY(Default as random)]

     Example:

     [root@localhost tmp]#./yealinkencrypt -f y000000000000.cfg -p /home/test -k 0123456789123456

     AES Key: 0123456789123456

     Generate Security Key File...

     Generate Encrypt Config File...

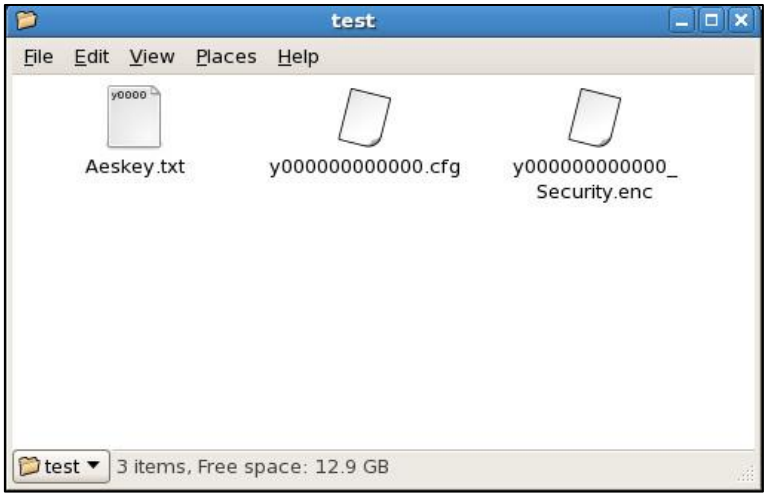
     Write file to /home/test/Aeskey.txt!

     Write file to /home/test/y000000000000_Security.enc!

     Read file y000000000000.cfg!

     Write file to /home/test/y000000000000.cfg!

This tool will encrypt the y000000000000.cfg file using the AES key 0123456789123456. You can find the encrypted y000000000000.cfg file, y000000000000_Security.enc file and an Aeskey.txt file storing the plaintext AES key 0123456789123456 in the specified directory.



- If you want to encrypt configuration files in batch using a random AES key, you need to execute the following command:

  ./yealinkencrypt –f *.cfg [-p DESTPATH(Default as 'Encrypted')] -m

  Example:

  [root@localhost tmp]#./yealinkencrypt -f *.cfg -p /home/test -m

  Generate AES Key...

  Write file to /home/test/Aeskey.txt!

  Write file to /home/test/0015652ac1cc_Security.enc!

  Read file 0015652ac1cc.cfg!

  Write file to /home/test/0015652ac1cc.cfg!

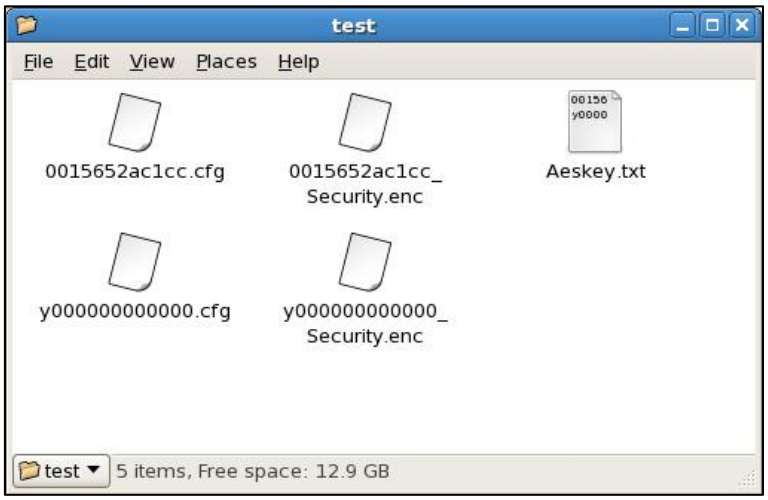
  Write file to /home/test/Aeskey.txt!

  Write file to /home/test/y000000000000_Security.enc!

  Read file y000000000000.cfg!

  Write file to /home/test/y000000000000.cfg!

This tool will encrypt all CFG files using random AES keys (each CFG file corresponds to a random AES key). You can find the encrypted CFG files, encrypted key files and an Aeskey.txt file storing the plaintext AES keys in the specified directory.



- If you want to encrypt configuration files in batch using a specified AES key, you need to execute the following command:

  ./yealinkencrypt –f *.cfg [-p DESTPATH(Default as 'Encrypted')] -k 0123456789123456

  Example:

  [root@localhost tmp]#./yealinkencrypt -f *.cfg -p /home/test -k 0123456789123456

  AES Key: 0123456789123456

  Generate Security Key File...

  Generate Encrypt Config File...

  Write file to /home/test/Aeskey.txt!

  Write file to /home/test/0015652ac1cc_Security.enc!

  Read file 0015652ac1cc.cfg!

  Write file to /home/test/0015652ac1cc.cfg!

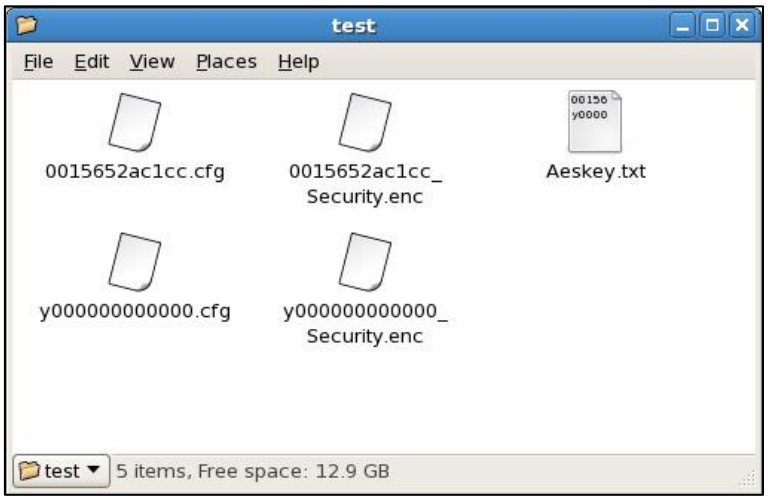
  Write file to /home/test/Aeskey.txt!

  Write file to /home/test/y000000000000_Security.enc!

  Read file y000000000000.cfg!

  Write file to /home/test/y000000000000.cfg!

This tool will encrypt all CFG files using a specified AES key. You can find the encrypted CFG files, encrypted key files and an Aeskey.txt file storing the plaintext AES key in the specified directory.



Note AES keys must be 16 characters and the supported characters contain: 0 ~ 9, A ~ Z, a ~ z and the following special characters are also supported: # $ % * + , - . : = ? @ [ ] ^ _ { } ~.

For Old Auto Provisioning Mechanism

Configuring Yealink IP Phones

You can configure the IP phones to decrypt the encrypted configuration files during auto provisioning using the encrypted AES key files or the AES keys configured on the IP phones.

Before deploying IP phones using the encrypted configuration files, you need to configure the following parameters for the IP phones using the configuration files first.

1. Add/Edit the following parameters in configuration files.

Parameter	Permitted Values	Default
auto_provision.aes_key_in_file	**0 or 1**	**0**
Description: Enables or disables the IP phone to decrypt configuration files using the encrypted AES keys. **0**-Disabled **1**-Enabled If it is set to 1 (Enabled), the IP phone will download <y0000000000xx_Security>.enc and <MAC_Security>.enc during auto provisioning, and then decrypts these files into the plaintext keys (e.g., key2, key3) respectively using the phone built-in key (e.g., key1). The IP		

Parameter	Permitted Values	Default
phone then decrypts the encrypted configuration files using corresponding key (e.g., key2, key3). If it is set to 0 (Disabled), the IP phone will decrypt the encrypted configuration files using plaintext AES keys configured on the IP phone. **Web User Interface:** None **Phone User Interface:** None		

2. Upload configuration files to the root directory of the provisioning server and trigger IP phones to perform an auto provisioning for configuration update.

   For more information on auto provisioning, refer to *Yealink_SIP-T2 Series_T19(P) E2_T4_Series_CP860_W56P_IP_Phones_Auto_Provisioning_Guide*.

Deploying Yealink IP Phones Using Encrypted Configuration Files and AES keys

This section shows a scenario on how to deploy Yealink IP phones using encrypted configuration files and AES keys.

Scenario: Encrypt configuration files and ensure no plaintext configurations and keys are transmitted across the network

Scenario Conditions:

- The administrator wants to encrypt configuration files to protect sensitive information in configuration files from tampering.
- SIP-T28 IP phone MAC: 0015651137F6.
- auto_provision.aes_key_in_file =1 (Enable the IP phone to download y000000000000_Security.enc and 0015651137f6_Security.enc files during auto provisioning)
- auto_provision.aes_key_16.com = 0123456789abcdef (The parameter value can be set to an arbitrary 16-character value, but cannot be blank)
- auto_provision.aes_key_16.mac = 0123456789abmins (The parameter value can be set to an arbitrary 16-character value, but cannot be blank)

If your IP phones are running firmware released after November 2013, parameters "auto_provision.aes_key_16.com" and "auto_provision.aes_key_16.mac" will not be needed in the above scenario.

Scenario Operations:

1. The administrator encrypts y000000000000.cfg and 0015651137f6.cfg files and then uploads y000000000000_Security.enc, 0015651137f6_Security.enc, y000000000000.cfg (encrypted) and 0015651137f6.cfg (encrypted) files to the root directory of the provisioning server.

   For more information on encrypting configuration files, refer to Encrypting Configuration Files on Windows Platform or Encrypting Configuration Files on Linux Platform.

2. Reboot the IP phone to trigger auto provisioning process. For more information, refer to *Yealink_SIP-T2 Series_T19(P) E2_T4_Series_CP860_W56P_IP_Phones_Auto_Provisioning_Guide*.

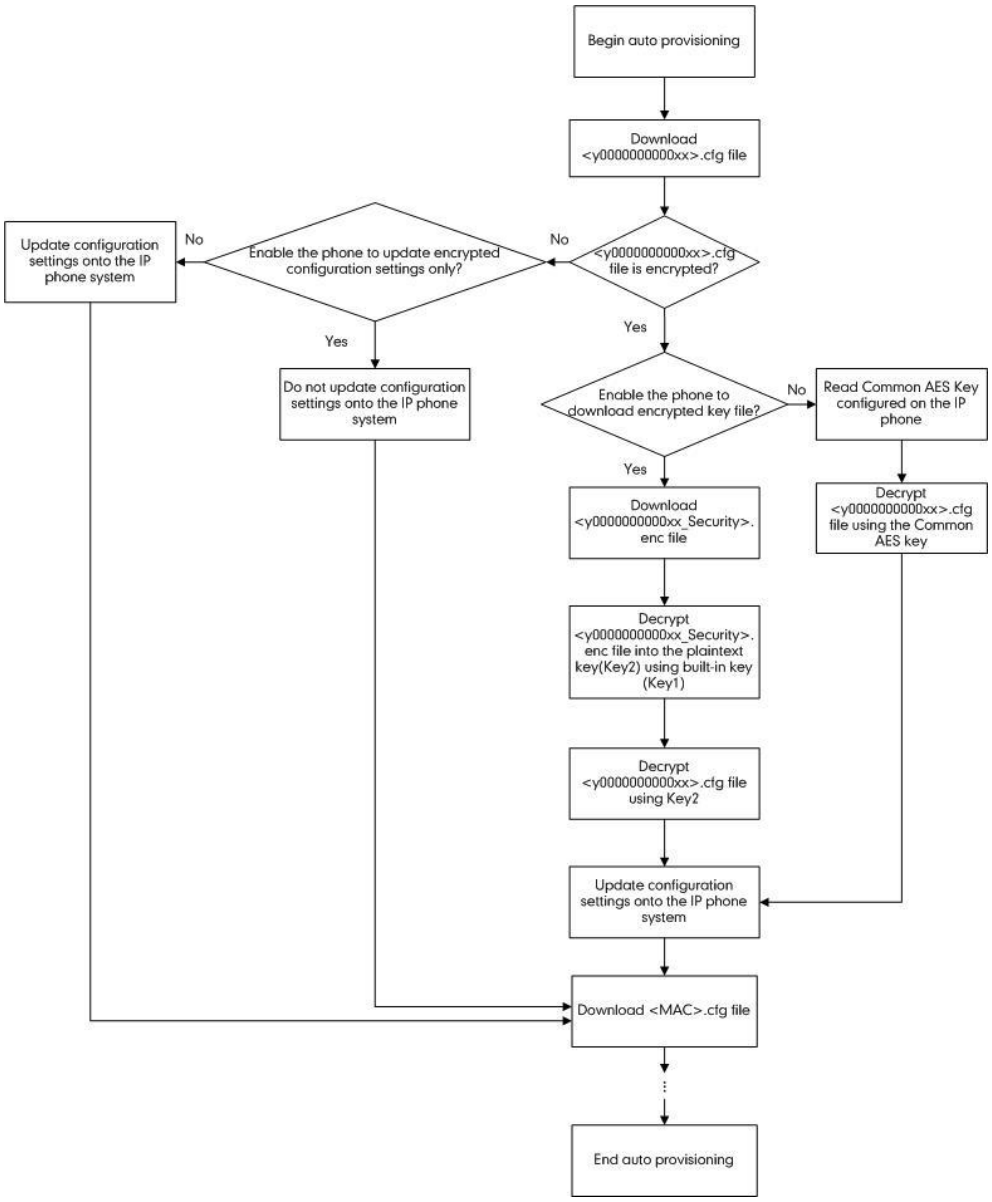
During auto provisioning, the IP phone requests to download y000000000000.cfg file first. Because the downloaded configuration file is encrypted, the IP phone requests to download y000000000000_Security.enc file and then decrypts it into the plaintext key (e.g., key2) using the built-in key (e.g., key1). The IP phone then decrypts the configuration file y000000000000.cfg using the key2. After decryption, the IP phone resolves configuration files and updates configuration settings onto the IP phone system.

The way the IP phones process the <MAC>.cfg file is the same as the <y000000000000>.cfg file.

For more information, refer to Appendix Auto Provisioning Flowchart.

Appendix Auto Provisioning Flowchart

The following shows the flowchart for provisioning Yealink IP phones using the encrypted configuration file (taking the Common CFG file as an example). The way the IP phone processes the MAC-Oriented CFG file is the same as that of the Common CFG file.



For New Auto Provisioning Mechanism

Configuring Yealink IP Phones

You can configure the IP phones to decrypt the encrypted configuration files during auto provisioning using the encrypted AES key files or the AES keys configured on the IP phones.

Before deploying IP phones using the encrypted configuration files, you need to configure the following parameters for the IP phones using the configuration files first.

1. Add/Edit the following parameters in the configuration file (e.g., sip.cfg).

Parameters	Permitted Values	Default
static.auto_provision.aes_key_in_file	**0 or 1**	**0**
Description: Enables or disables the IP phone to decrypt configuration files using the encrypted AES keys. **0**-Disabled **1**-Enabled If it is set to 0 (Disabled), the IP phone will decrypt the encrypted configuration files using plaintext AES keys configured on the IP phone. If it is set to 1 (Enabled), the IP phone will download <xx_Security>.enc files (e.g., <sip_Security>.enc, <account_Security>.enc) during auto provisioning, and then decrypts these files into the plaintext keys (e.g., key2, key3) respectively using the phone built-in key (e.g., key1). The IP phone then decrypts the encrypted configuration files using corresponding key (e.g., key2, key3). **Web User Interface:** None **Phone User Interface:** None		

2. Reference the configuration file in the boot file (e.g., y000000000000.boot).

   Example:

   include:config "http://10.2.1.158/sip.cfg"

   include:config "http://10.2.1.158/account.cfg"

3. Upload the boot file and configuration file to the root directory of the provisioning server.
4. Trigger IP phones to perform an auto provisioning for configuration update.

   For more information on auto provisioning, refer to the latest Auto Provisioning Guide on Yealink Technical Support.

Deploying Yealink IP Phones Using Encrypted Configuration Files and AES keys

Scenario: Encrypt configuration files and ensure no plaintext configurations and keys are transmitted across the network

Scenario Conditions:

- The administrator wants to encrypt configuration files to protect sensitive information in configuration files from tampering.
- SIP-T23G IP phone MAC: 001565918998.
- static.auto_provision.aes_key_in_file = 1 (Enable the IP phone to download xx_Security.enc files during auto provisioning)

Scenario Operations:

1. The administrator encrypts y000000000044.cfg and sip.cfg files.

   For more information on encrypting configuration files, refer to Encrypting Configuration Files on Windows Platform or Encrypting Configuration Files on Linux Platform.

2. Uploads 001565918998.boot, y000000000044_Security.enc, sip_Security.enc, y000000000044.cfg (encrypted) and sip.cfg (encrypted) files to the root directory of the provisioning server.

3. Edit the boot file (001565918998.boot file) as following:

```
#!version:1.0.0.1
## The header above must appear as-is in the first line

   include:config <sip.cfg>
   include:config "y000000000044.cfg"

   overwrite_mode = 1
```

4. Reboot the IP phone to trigger auto provisioning process.

   For more information on auto provisioning, refer to the latest Auto Provisioning Guide on Yealink Technical Support.

During auto provisioning, the IP phone will try to download 001565918998.boot file firstly and then download the configuration files referenced in the boot file in sequence from the provisioning server.

At this scenario, the IP phone requests to download sip.cfg file first. Because the downloaded configuration file is encrypted, the IP phone requests to download sip_Security.enc file and then decrypts it into the plaintext key (e.g., key2) using the built-in key (e.g., key1). The IP phone then decrypts the configuration file sip.cfg using the key2. After decryption, the IP phone resolves configuration files and updates configuration settings onto the IP phone system.

The way the IP phones process the y000000000044.cfg file is the same as the sip.cfg file.

For more information, refer to Appendix Auto Provisioning Flowchart.

Appendix Auto Provisioning Flowchart

The following shows the flowchart for provisioning Yealink IP phones using the encrypted

configuration file (taking the sip.cfg as an example). The way the IP phone processes other CFG file is the same as that of the Common CFG file.