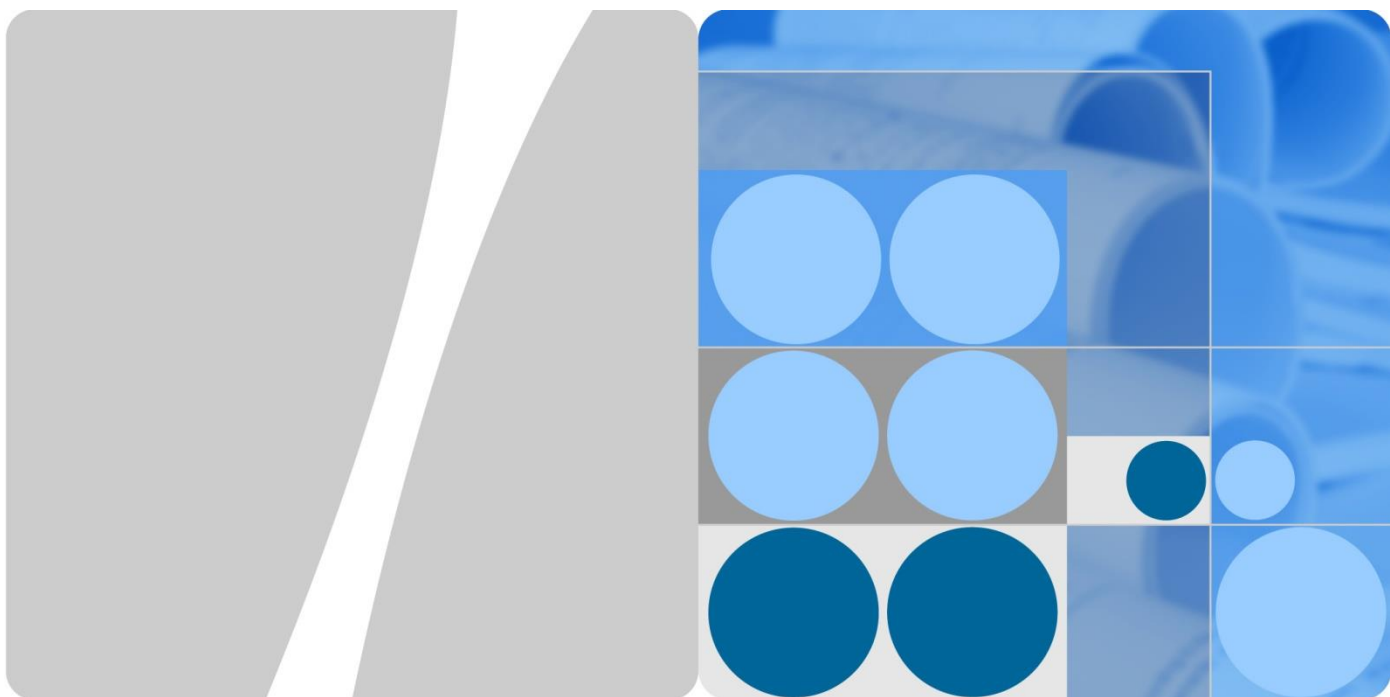




**OpenPGP
V100R001C00**

Signature Verification Guide

Issue	04
Date	2020-10-09

HUAWEI TECHNOLOGIES CO., LTD.
Network Security Competent Center.



Copyright © Huawei Technologies Co., Ltd. 2014-2020 All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.



is a trademark of Network Security Competent Center.



is a trademark of PGPVerify Tool.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <http://www.huawei.com>

Email: support@huawei.com

About This Document

Purpose

This document describes OpenPGP signature tools, and verification process.

Intended Audience

This document is intended for:

- Installation and commissioning engineers
- Technical support engineers

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
 DANGER	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
 WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
 CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.
 NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
 NOTE	Calls attention to important information, best practices and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Change History

Changes between document issues are cumulative. The latest document issue contains all the changes made in earlier issues.

Issue 01 (2014-11-20)

This issue is used for first office application (FOA).

Issue 02 (2017-09-01)

1. Change the description of the signature verification procedure.
2. Upgrade the PGPVerify Tool to V100R001C00SPC310.

Issue 03 (2017-12-12)

1. Add handle suggestion when verify failed.
2. Upgrade the PGPVerify tool to V100R001C00SPC320.

Issue 04(2020-02-20)

Added verification guidance for OpenPGP key length of 4096.

PGPVerify Tool Update History

Changes between PGPVerify tool releases are cumulative. The latest document issue contains all the change made in earlier issues.

Release V100R001C00SPC200:

1. This is the first version of PGPVerify tool, which offered verification function for PGP signature.

Release V100R001C00SPC310:

1. Upgrade library openssl to version 1.1.0f;
2. Replace icon for applicatoin with UI under Windows OS;
3. Change UI style to fit Windows 7;
4. Show a confirm dialog when closing the application window.

Release V100R001C00SPC320:

1. Add version properties for PGPVerify tool under windows;
2. Add timestamp for PGPVerify tool under windows.

Contents

About This Document	ii
1 OpenPGP Overview	7
2 Description Of The Public Key File	8
3 GnuPG (Linux)	9
3.1 Background	9
3.2 Prerequisites	9
3.2.1 Installing GnuPG	9
3.2.2 Obtaining the Public Key File	10
3.2.3 Importing the Public Key	13
3.2.4 Verifying the Public Key	14
3.3 Verifying the Signature	16
4 Gpg4Win (Windows)	19
4.1 Background	19
4.2 Prerequisites	19
4.2.1 Installing Gpg4Win	19
4.2.2 Obtaining the Public Key File	22
4.2.3 Importing the Public Key	26
4.2.4 Verifying the Public Key	27
4.3 Verifying the Signature	28
5 PGPVerify (Windows&Linux)	32
5.1 Background	32
5.2 Prerequisites	32
5.2.1 Obtaining PGPVerify	32
5.2.2 Obtaining the Public Key File	35
5.3 Verifying the Signature	36
5.3.1 Verifying Through Operations on the UI	36
5.3.2 Verification Through the CLI (Windows)	39
5.3.3 Verification Through the CLI (Linux)	40
6 FAQs	43
6.1 The Application Scope for use Verification Tool?	43

6.2 How to Obtain version of Verification Tool?	43
6.3 How to Obtain an .asc File?	45
6.4 How to Obtain the Public Key or Verification Tools?	46
6.5 How Is Signature Verification Implemented?	48
6.6 How to Switch the Language of a Web Page from Chinese to English?	49
6.7 How to Rectify the Failure in Long Path Verification Using the PGPVerify.exe Command?	49
6.8 How to Obtain and Use the PGPVerify (Solaris/Linux) Tool?	49

1 OpenPGP Overview

OpenPGP is an open security protocol standard (RFC4880) that widely applies to data encryption and signature. It has multiple commercial and non-commercial implementations, including Pretty Good Privacy (PGP) and GNU Privacy Guard (GnuPG). GnuPG is transplanted into multiple platforms, such as Linux and Windows and pre-installed in most Linux versions.

OpenPGP includes an independent digital signature standard that differentiates itself from other standards by the key storage and distribution method, message digest calculation process, signature packet format, and verification process.

Verification Tool Introduction

Select tools based on the operating systems to implement OpenPGP signature verification, as shown in the following table.

Tool Name	Operating System	Tool Description
GNU Privacy Guard (GnuPG)	Linux	GnuPG is a free open-source GNC tool that implements the OpenPGP standard defined in RFC4880. It is pre-installed in most Linux versions. Official website: http://www.gnupg.org
GNU Privacy Guard for Windows (Gpg4Win)	Windows	Gpg4Win is the official Windows version of GnuPG. The function and usage of Gpg4Win are the same as those of GnuPG. Official website: http://www.gpg4win.org/
PGPVerify.exe	Windows	PGPVerify is a PGP simplified verification tool developed by Huawei. Official website: http://support.huawei.com/carrier/digitalSignatureAction

2 Description Of The Public Key File

1.The KEYS.txt file is the public key file when the OpenPGP key length is 2048. The KEYS4096.txt file is the public key file when the OpenPGP key length is 4096.

2.Compared with KEYS.txt, KEYS4096.txt increases the key length and the length of the signature result, improving security.

Notice:

The carrier download address must be compatible with some user tools. Therefore, the public key file name of the KEYS.txt file is KEYS.

3 GnuPG (Linux)

To prevent threats to carrier networks caused by software tampering or damage in transit, verify the integrity of software packages after receiving them. Only verified software packages can be deployed.

3.1 Background

GnuPG is a free open-source GNC tool that verifies OpenPGP signatures in the SUSE Linux operating system.

Software packages and signature files are released together and stored in the same directory. A software package corresponds to a signature file.

The signature files use the same file names as those used by software packages, with file name extension **asc**. For example, if the software package name is **V100R001C04.zip**, the corresponding verification file name is **V100R001C04.zip.asc**.

3.2 Prerequisites

3.2.1 Installing GnuPG

GnuPG is pre-installed in most Linux versions. Run the **gpg --version** command in the shell. If the following command output is displayed, GnuPG is installed.

```
signsrv:~ # gpg --version
gpg (GnuPG) 2.0.9
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: ~/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA
Cipher: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH
Hash: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```

```
Used libraries: gcrypt(1.4.5)
signsrv:~ #
```

If GnuPG is not installed in the current system, follow the guidance on the official website (<http://www.gnupg.org/>) to install it.

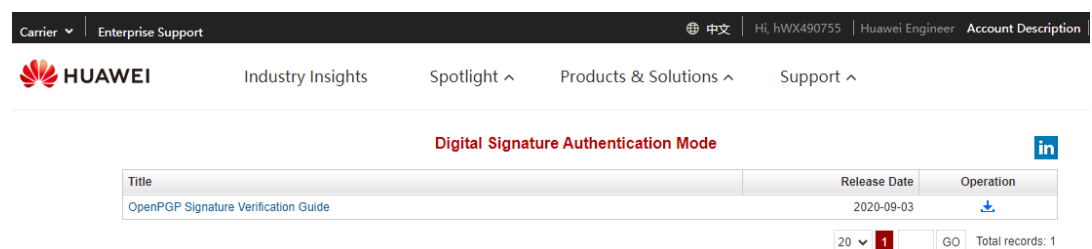
3.2.2 Obtaining the Public Key File

Downloading the File from the Support Website

- Carrier users download the public key file from the following URL at <http://support.huawei.com/carrier>:
<http://support.huawei.com/carrier/digitalSignatureAction>

The target web page may be displayed in Chinese. Click **English** on the top of the web page to switch the language to English, so that the the English document can be downloaded.

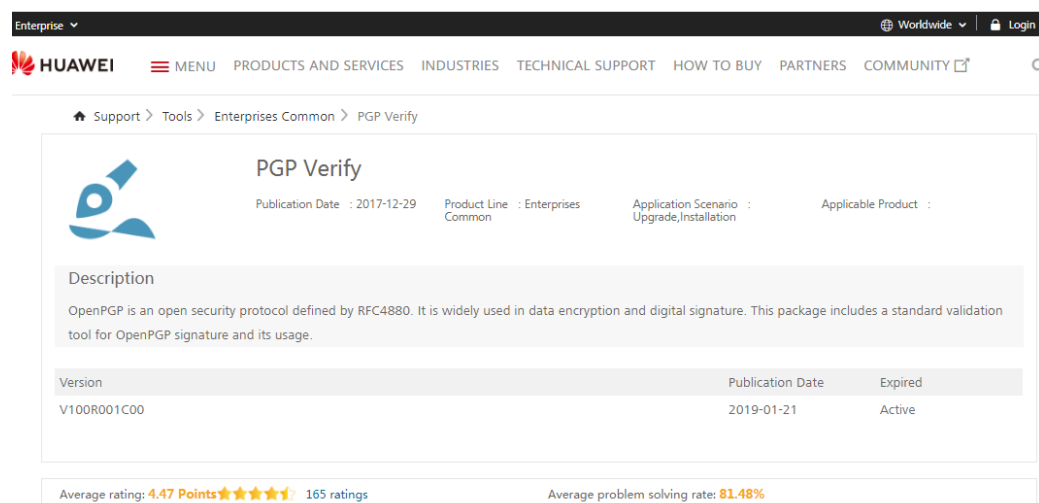
Figure 3-1 Web page for downloading the public key file from <http://support.huawei.com/carrier>



- Enterprise users download the public key file from the following URL at <http://support.huawei.com/enterprise>:
<http://support.huawei.com/enterprise/en/tool/software-digital-signature-validation-tool-%EF%BC%88pgp-verify%EF%BC%89-TL1000000054>

The target web page may be displayed in Chinese. Click **Worldwide** on the top of the web page to switch the language to English, so that the the English document can be downloaded.

Figure 3-2 Web page for downloading the public key file from <http://support.huawei.com/enterprise>



- Terminal users download the public key file from the terminal knowledge base at the following URL:

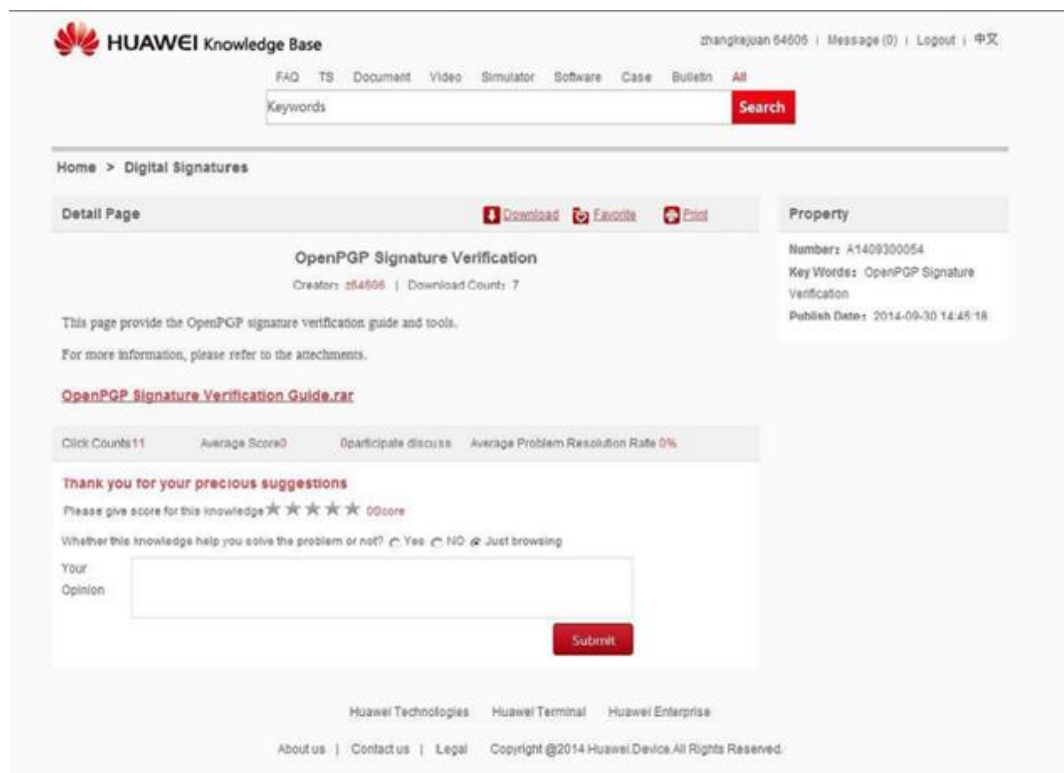
<http://app.huawei.com/tkb/#!tservice/common/base/digit.html?resourceId=146680>

To download the public key file from the terminal knowledge base, perform the following steps:

Step 1 Access <http://app.huawei.com/tkb/#!tservice/common/base/digit.html?resourceId=146680>.

Figure 3-3 shows the displayed web page.

Figure 3-3 Web page for downloading the public key file from the terminal knowledge base



Step 2 Click **Download** to download the **OpenPGP Signature Verification Guide.rar** package.

If you have relevant permissions but an error is displayed, select the correct language.

Step 3 Decompress the downloaded **OpenPGP Signature Verification Guide.rar** package.

The **KEYS.txt** or **KEYS4096.txt** file is the public key file.

----End

Obtaining from the Public Key Server

Step 1 Access the public key server at the following URL:

<https://zimmermann.mayfirst.org>

The result is displayed, as shown in Figure 3-4.

Figure 3-4 Public key server address page

Not secure | zimmermann.mayfirst.org

You can find a key by typing in some words that appear in the userid (name, email, etc.) of the key you're looking for, or by typing in the keyid in hex format ("0x...")

Search for a public key

String

Show PGP Fingerprints ☐

Show SKS full-key hashes ☐

Get regular index of matching keys ☐

Get verbose index of matching keys ☒

Retrieve ascii-armored keys ☐

Retrieve keys by full-key hash ☐

Enter " OpenPGP signature key for Huawei software "in the String text box, and click" Search for a key" to search for the public key.

Figure 3-5 Public key search result

Search results for 'software signature openpgp key huawei for'

Type	bits/keyID	cr. time	exp time	key expir
pub	4096R/6ADE4A56	2019-06-15		
uid	OpenPGP signature key for Huawei software (created on 15th Jun,2019) <support@huawei.com>			
sig	sig3 6ADE4A56	2019-06-15		[selfsig]
pub	2048R/27A74824	2013-12-30		
uid	OpenPGP signature key for Huawei software (created on 30th Dec,2013) <support@huawei.com>			
sig	sig3 27A74824	2013-12-30		[selfsig]

Step 2 Click public key ID 27A74824 to check the details, as shown in Figure 3-6. If the public key length is 4096, the corresponding ID is 6ADE4A56.

Figure 3-6 Public key details

Public Key Server -- Get "0x99ad81df27a74824 "

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: SKS 1.1.6
Comment: Hostname: zimmermann.mayfirst.org

mQENBFLBPjsBCActQyKqccsmla3GvRoPHpfrB9ITrYeNDvSF1Jel4esQQuA513ILTS9bdG
H90sluWnryVcGVKVRHpmajhuqSJycYFn/3VXtUWm27zVSHrKh0Tm3z2rIeu0dJzaKpgrtkw
RzeutDFW8GqwrQsGEOGLNcHw+FRYdjmVru6SEcC7zjhFY2/TIj4nGGVCP+ebjQGoLBMjC
2o9fJi+UV5LxHnB896TpcPKs+JTaVSKSnQ8RG23D1DofBgwr19icaNxm17bPx2WYRutslRS6
HKh17Vf4T4t3BoaKj0xE/cF761FFrDNIFxSpRqQ3As4kS3UOUFG1/5NUBkev4MI5iFAEEB
AAG0WU9wZWSQR1Age2lrbmF0dXJlIGt1eSBmb3Igc2hWd2VpIHh0ZnR3YXJlChjcmVhdGVk
IG9uIDMedGgRGVjLDIwMTMwIDxzdXBw3UQqGh1YXdlas5jb20+iQE3BEMBAgAhBQJSw747
AheD6gsJCAdAgYWCALJCgsDFgIRAh4BAheAAoJEJatgd8rp0gk0GQIAIJKdFLMiujdx1S8
INkZHejGagTch9GqKlu6HQ3Hp59OKFPINBgds1NFu0wc00WqBAxXfGILLQpBSWLa5cIOHrEi
+PqZKkdd3hZrw/G/GH1H1TjIHhNamTikalCz4B+BcsQ0UhfVXZDTkBAF9a9Md21dJsgzEaA
ygpzdIMKse4JcJ7gY75L/DSWDpEfeJQIj5RQpXbDn4sgXiZU314DCz8+Iiz30c31+Vhr
yWC+g1gMeFWKY1S1waltoqKJeb5b6VEarJKDJXmLkV3kCnB8AMq8yS0y92uBR58WNLxw37o
01grXsQm+Tl7GkBQ20xLFHvIdU9UuzVX86Xyy7A=
=0zUT
-----END PGP PUBLIC KEY BLOCK-----
```

Step 3 Copy the public key information to a TXT file and name it as KEYS.txt. If the public key length is 4096, save it as a KEYS4096.txt file.

----End

3.2.3 Importing the Public Key

Step 1 Log in to the server that contains the software package to be verified as a common user.

Step 2 Import the public key file. Set **/home/openpgp/keys** to the actual path of the public key file **KEYS.txt**.

Access the directory of public key file **KEYS.txt** and run the following command:

```
# gpg --import "/home/openpgp/keys/KEYS.txt"
```

When sign data length is 4096, please select KEYS4096.txt for the public key file.

The following output is displayed for key length of 2048:

```
gpg: key 27A74824: public key "OpenPGP signature key for Huawei software (created
on 30th Dec,2013) <support@huawei.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
```

The following output is displayed for key length of 4096:

```
gpg: key 6ADE4A56: public key "OpenPGP signature key for Huawei software (created
on 15th Jun,2019) <support@huawei.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
```



NOTICE

In this command, two hyphens precede **import**.

Step 3 Run the following command to check the public key import result.

```
# gpg --fingerprint
```

If the following output is displayed, the public key is successfully imported.

The following output is displayed for key length of 2048:

```
pub 2048R/27A74824 2013-12-30
    Key fingerprint = B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824
uid      OpenPGP signature key for Huawei software (created on 30th
Dec,2013) support@huawei.com
```

The following output is displayed for key length of 4096:

```
pub 4096R/6ADE4A56 2019-06-15
    Key fingerprint = E128 5E9D 7E7F 0DB0 A659 48AF FAAA 7A2E 6ADE 4A56
uid      OpenPGP signature key for Huawei software (created on 15th Jun,2019)
<support@huawei.com>
```



NOTICE

In this command, two hyphens precede **fingerprint**.

----End

3.2.4 Verifying the Public Key

Step 1 In normal cases, to verify the OpenPGP public key, you need to check the ID, fingerprint, and user ID of the public key with the entity that releases the public key. Information about the OpenPGP public keys released by Huawei is shown as follows:

The following output is displayed for key length of 2048:

- Public key ID: 27A74824
- Key fingerprint: B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824
- User ID (uid): OpenPGP signature key for Huawei software (created on 30th Dec,2013) support@huawei.com

The following output is displayed for key length of 4096:

- Public key ID: 6ADE4A56
- Key fingerprint: E128 5E9D 7E7F 0DB0 A659 48AF FAAA 7A2E 6ADE 4A56
- User ID (uid): OpenPGP signature key for Huawei software (created on 15th Jun,2019) <support@huawei.com>

After verifying the preceding information, set the trust level for the key.

Step 2 Run the following command to set the trust level.

```
# gpg --edit-key "OpenPGP signature key for Huawei software (created on
30th Dec,2013)" trust

set the trust level as follows, If the public key length is 4096

#gpg --edit-key "OpenPGP signature key for Huawei software (created on
15th Jun,2019)" trust
```

The output resembles the following information. You need to enter 5 behind **Your decision?** to indicate **I trust ultimately** and **y** behind **you really want to set this key to ultimate trust?** (y/N).

```
gpg (GnuPG) 2.0.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 2048R/27A74824 created: 2013-12-30 expires: never usage: SC
trust: ultimate validity: ultimate
[ultimate] (1). OpenPGP signature key for Huawei software (created on 30th
Dec,2013) <support@huawei.com>

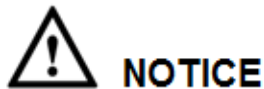
pub 2048R/27A74824 created: 2013-12-30 expires: never usage: SC
trust: ultimate validity: ultimate
[ultimate] (1). OpenPGP signature key for Huawei software (created on 30th
Dec,2013) <support@huawei.com>

Please decide how far you trust this user to correctly verify other users' keys
(by looking at passports, checking fingerprints from different sources, etc.)

 1 = I don't know or won't say
 2 = I do NOT trust
 3 = I trust marginally
 4 = I trust fully
 5 = I trust ultimately
 m = back to the main menu

Your decision? 5
Do you really want to set this key to ultimate trust? (y/N) y

pub 2048R/27A74824 created: 2013-12-30 expires: never usage: CS
trust: ultimate validity: unknown
[ unknown] (1). OpenPGP signature key for Huawei software (created on 30th
Dec,2013) <support@huawei.com>
Please note that the shown key validity is not necessarily correct
unless you restart the program.
```



In this command, two hyphens precede **edit**.

Step 3 Run the following command to quit.

```
quit
```

```
----End
```

3.3 Verifying the Signature

The signature file must be in the same path as that of the software package. Set **/home/openpgp/soft** to the actual path of the signature file. Run the following command to verify the signature.

```
# gpg --verify "/home/openpgp/soft/V100R001C041.zip.asc"
```

The following output is displayed for key length of 2048:

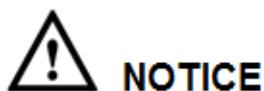
```
gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824  
gpg: Good signature from "OpenPGP signature key for Huawei software (created on  
30th Dec, 2013) <support@huawei.com>"
```

RSA key ID in bold is 27A74824, and there is no WARNING, which means this signature is published by Huawei.

The following output is displayed for key length of 4096:

```
pg: Signature made Mon Dec 30 12:16:07 2019 CST using RSA key ID 6ADE4A56  
gpg: Good signature from "OpenPGP signature key for Huawei software (created on 15th  
Jun, 2019) <support@huawei.com>"
```

RSA key ID in bold is 6ADE4A56, and there is no WARNING, which means this signature is published by Huawei.



NOTICE

- If a version has multiple signature files to be verified, the version is considered secure only when all the files pass the verification. If the verification result of a file is WARN or FAIL, the version does not pass the verification and has security risks. In this case, try to solve problem following the suggest action in table 3-1.
- In this command, two hyphens precede **verify**.

Table 3-1 Example for signature verification result judgment

Verification Result Scenario	Output Example	Verification Result	Suggest action
Signature verification succeeds, with no anomaly.	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: Good signature from "OpenPGP signature key for Huawei software (created on 30th Dec,2013)" <support@huawei.com>	PASS	NA
Signature verification fails.	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: BAD signature from "OpenPGP signature key for Huawei software (created on 30th Dec,2013)" <support@huawei.com>	FAIL	Download the software package again, or turn to product support
The public key is not found.	gpg: Signature made Thu Jan 9 15:20:01 2014 CST using RSA key ID 27A74824 gpg: Can't check signature: public key not found	FAIL	Download public key, see: Obtaining the Public Key File
Signature verification succeeds, but the public key is not set to be completely trusted.	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: Good signature from "OpenPGP signature key for Huawei software (created on 30th Dec,2013)" <support@huawei.com> gpg: WARNING: This key is not certified with a trusted signature! gpg: There is no indication that the signature belongs to the owner. Primary key fingerprint: B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824	WARNING	First check if key ID is 27A74824, and set the public key to be fully trusted, see: Verifying the Public Key

Verification Result Scenario	Output Example	Verification Result	Suggest action
The corresponding source file cannot be found.	gpg: no signed data gpg: can't hash datafile: No data	FAIL	Download the software package again, or turn to product support
The signature has expired.	gpg: Signature made 04/24/13 10:50:29 CST using RSA key ID 133B64E5 gpg: Expired signature from " OpenPGP signature test key <support@huawei.com>" gpg: Signature expired 04/25/13 10:50:29 CST	FAIL	Download the software package with updated signature, or turn to product support
Signature verification succeeds, but the public key is revoked.	gpg: Signature made 06/13/13 11:14:49 CST using RSA key ID 133B64E5 gpg: Good signature from " OpenPGP signature test key <support@huawei.com>" gpg: WARNING: This key has been revoked by its owner! gpg: This could mean that the signature is forged. gpg: reason for revocation: Key is no longer used gpg: revocation comment:	WARNING	Download the newest public key and software package with updated signature, or turn to product support
The signature file corresponding to the source file cannot be found.	N/A	WARNING	Download signature file of software package, or turn to product support

4 Gpg4Win (Windows)

To prevent threats to carrier networks caused by software tampering or damage in transit, verify the integrity of software packages after receiving them. Only verified software packages can be deployed.

4.1 Background

Gpg4Win is a free open-source GNU tool that verifies OpenPGP signatures in Windows. The function and usage of Gpg4Win are the same as those of GnuPG. For details, visit its official website <http://www.gpg4win.org/>.

A software package and its corresponding signature file are stored in the same directory.

The signature files use the same file names as those used by software packages, with the file name extension being **asc**. For example, if the software package name is **V100R001C04.zip**, the corresponding verification file name is **V100R001C04.zip.asc**.

4.2 Prerequisites

4.2.1 Installing Gpg4Win

First download the install package as follows:

- Step 1** Visit <https://www.gpg4win.org/download.html>.
- Step 2** Click the download link in the red box shown in the preceding figure. (Maybe the latest version is not 3.1.11 as in this document, but the download link does not change. In the following steps, you can ignore the version of the installation package.)

Figure 4-1 Gpg4win download step 1

Download

Gpg4win 3.1.11 (Released: 2019-12-17)

You can download the full version (including the Gpg4win compendium) of Gpg4win 3.1.11 here:



[OpenPGP signature](#) (for gpg4win-3.1.11.exe)

SHA256: 156de9f3f50bb5a42b207af67ae4ebcb2d10a7aaf732149e9c468eaf74ce7ffc

[Changelog](#)

More Gpg4win-3.1.11 downloads

- Gpg4win source code package:
[gpg4win-3.1.11.tar.bz2](#) (Size: 5.3 MByte)
[OpenPGP signature](#)
SHA256 checksum:
15927a175d5ab802d36a257adeae7af989868a5dfbbbfd49593b6300ce486a52
- All versions and OpenPGP signatures:
[files.gpg4win.org](#).

Step 3 Select **Bank transfer**.

Figure 4-2 Gpg4win download step 2

Please donate for Gpg4win to support maintenance and development!
Pay what you want! – Thank you!

Donate with

☐ PayPal

☐ Bitcoin

☒ Bank transfer

Bank transfer details:

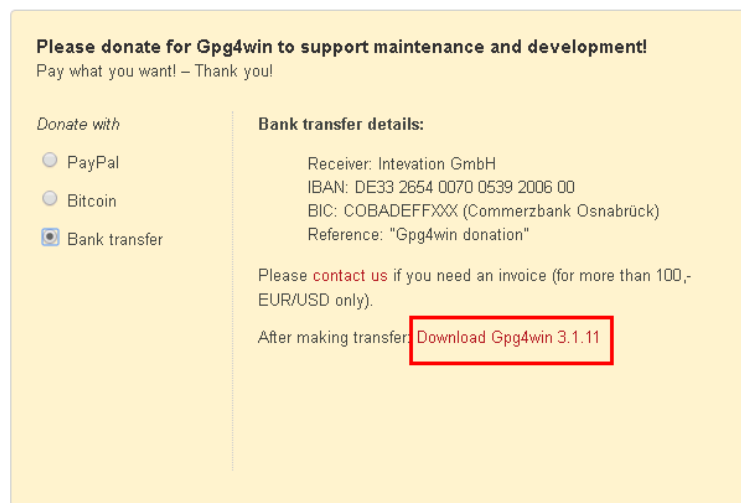
Receiver: Intevation GmbH
IBAN: DE33 2654 0070 0539 2006 00
BIC: COBADEFFXXX (Commerzbank Osnabrück)
Reference: "Gpg4win donation"

Please [contact us](#) if you need an invoice (for more than 100,- EUR/USD only).

After making transfer: [Download Gpg4win 3.1.11](#)

Step 4 Click the download link in the red box to download the installation package.

Figure 4-3 Gpg4win download step 3



Then install the package as follows:

- Step 5** Double-click **gpg4win-3.11.1.exe** and follow the installation wizard to install it . You can keep the default settings.

Figure 4-4 Gpg4win install step



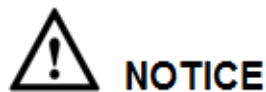
- Step 6** Check whether it is successfully installed.

After installation, select the default installation path **C:\Program Files (x86)\GNU\GnuPG>** (default installation path on the x86_64 Windows platform. The default installation path on the x86 Windows platform is **C:\Program Files\GNU\GnuPG>**) and run the **gpg.exe – version** command in the CLI. If the following output is displayed (the package version may differ from the following one), Gpg4Win is successfully installed.

Figure 4-5 Gpg4win introduction

```
D:\Program Files (x86)\GnuPG\bin>gpg --version
gpg (GnuPG) 2.2.19
libgcrypt 1.8.5
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: C:/Users/hwx490755/AppData/Roaming/gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2
```



NOTICE

In this command, two hyphens precede **verify**.

----End

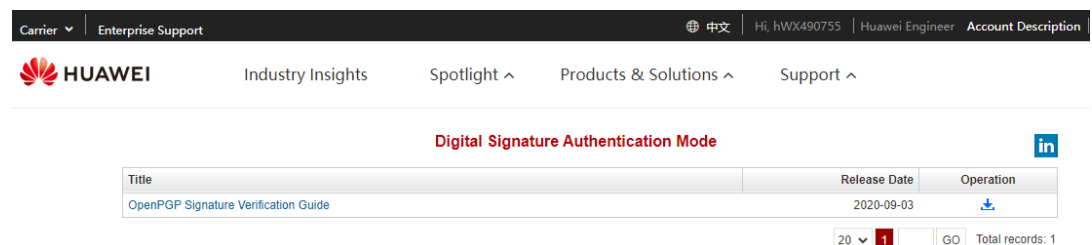
4.2.2 Obtaining the Public Key File

Downloading the File from the Support Website

- Carrier users download the public key file from the following URL at
<http://support.huawei.com/carrier>:
<http://support.huawei.com/carrier/digitalSignatureAction>

The target web page may be displayed in Chinese. Click **English** on the top of the web page to switch the language to English. Figure 4-6 shows the web page in English. Download and decompress the **OpenPGP Signature Verification Guide** package to obtain the **KEYS.txt** or **KEYS4096.txt** file (which is the public key file).

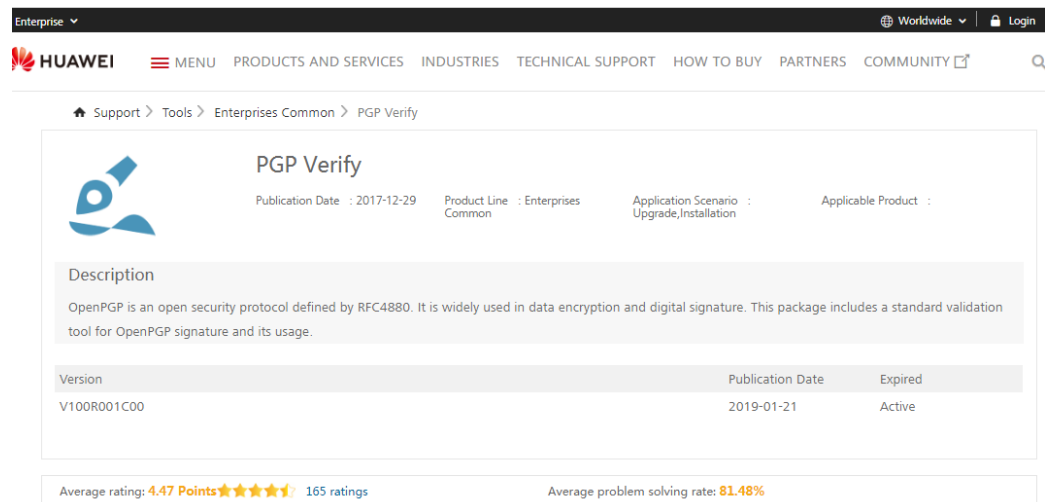
Figure 4-6 Web page for downloading the public key file from <http://support.huawei.com/carrier>



- Enterprise users download the public key file from the following URL at
<http://support.huawei.com/enterprise>:
<http://support.huawei.com/enterprise/en/tool/software-digital-signature-validation-tool-%EF%BC%88pgp-verify%EF%BC%89-TL1000000054>

The target web page may be displayed in Chinese. Click **Worldwide** on the top of the web page to switch the language to English. Figure 4-7 shows the web page in English. Click the version number in the **Version** list. Then click  corresponding to the public key file **KEYS.txt** or **KEYS4096.txt** to download the file.

Figure 4-7 Web page for downloading the public key file from <http://support.huawei.com/enterprise>



- Terminal users download the public key file from the terminal knowledge base at the following URL:

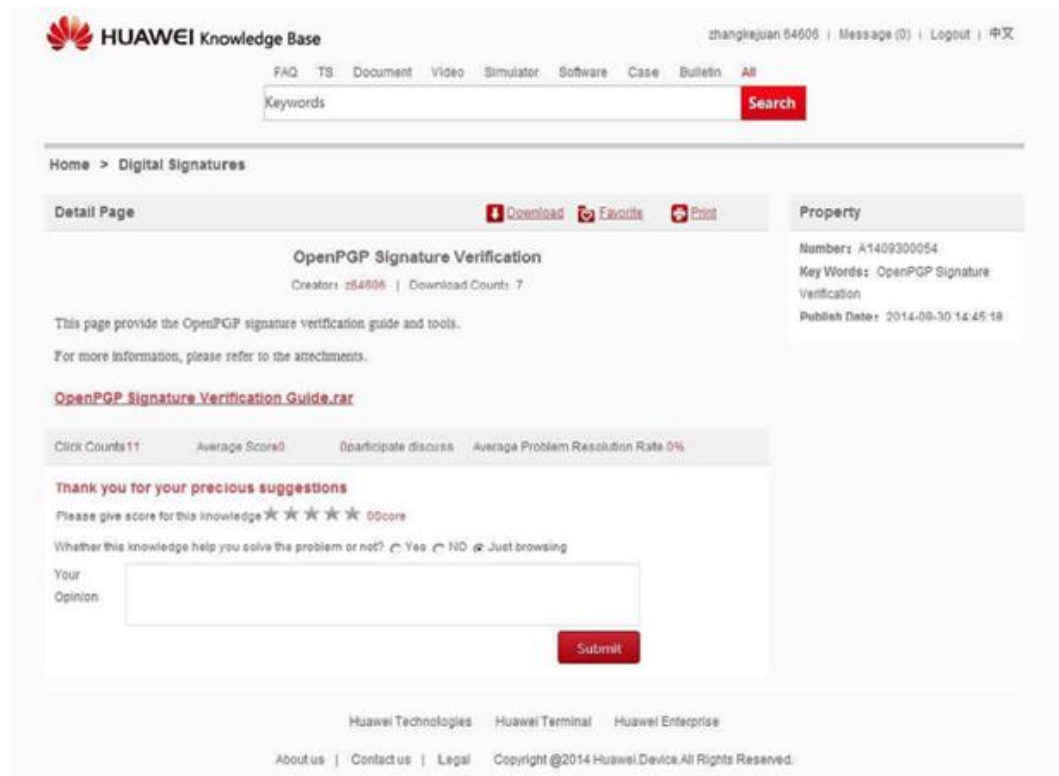
<http://app.huawei.com/tkb/#!/tservice/common/base/digit.html?resourceId=146680>

To download the public key file from the terminal knowledge base, perform the following steps:

Step 1 Access <http://app.huawei.com/tkb/#!/tservice/common/base/digit.html?resourceId=146680>.

Figure 4-8 shows the displayed web page.

Figure 4-8 Web page for downloading the public key file from the terminal knowledge base



Step 2 Click **Download** to download the **OpenPGP Signature Verification Guide.rar** package.

If you have relevant permissions but an error is displayed, select the correct language.

Step 3 Decompress the downloaded **OpenPGP Signature Verification Guide.rar** package.

The **KEYS.txt** or **KEYS4096.txt** file is the public key file.

----End

Obtaining from the Public Key Server

Step 1 Access the public key server at the following URL:

<https://zimmermann.mayfirst.org>

The result is displayed, as shown in Figure 4-9.

Figure 4-9 Public key server address page

Not secure | zimmermann.mayfirst.org

You can find a key by typing in some words that appear in the userid (name, email, etc.) of the key you're looking for, or by typing in the keyid in hex format ("0x...")

Search for a public key

String

Show PGP Fingerprints ☐

Show SKS full-key hashes ☐

Get regular index of matching keys ☐

Get verbose index of matching keys ☒

Retrieve ascii-armored keys ☐

Retrieve keys by full-key hash ☐

Reset

Enter " OpenPGP signature key for Huawei software "in the String text box, and click" Search for a key" to search for the public key.

Figure 4-10 Public key search result

Search results for 'software signature openpgp key huawei for'

Type	bits/keyID	cr. time	exp time	key expir
pub	4096R/ 6ADE4A56	2019-06-15		
uid	OpenPGP signature key for Huawei software (created on 15th Jun,2019) <support@huawei.com>			
sig	sig3 6ADE4A56	2019-06-15		[selfsig]
pub	2048R/ 27A74824	2013-12-30		
uid	OpenPGP signature key for Huawei software (created on 30th Dec,2013) <support@huawei.com>			
sig	sig3 27A74824	2013-12-30		[selfsig]

Step 2 Click public key ID 27A74824 to check the details, as shown in Figure 4-11. If the public key length is 4096, the corresponding ID is 6ADE4A56.

Figure 4-11 Public key details

Public Key Server -- Get "0x99ad81df27a74824 "

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: SKS 1.1.6
Comment: Hostname: zimmermann.mayfirst.org

mQENBEPjsDCACtQyXqecsnla3GvRoPHpfrB9ITrYeNDvfSF1Jel4asOgQuA513ILTS9bdG
H90sluWnyVocVRVHrpajhuqSJycTPn/3VXtUWm27zVSHrKhOTmm3z2rIeu0dJzaKpgwtkw
RzeutDFW8GqprRQsGEOGLNcHv+FRYdjatVru6SKC7zjhY2/TIj4nGGWCP+ebjQGoLBMjC
2o9fJi+UV5LxHrB896YpcFKs+JTavSKSnQ8fG23D1DofBgwr19icaXNml7bPxZWYRutalURS6
HKh1TVfvo4T4t3BoaKj0xE/cF761FPzDNIFxSpRqQ3As4kS3UOUFg1/5NUbkeV4MI5ifaEEB
AAG0WU9wZW5QR1Agc2lrbmF0dXJlIGt1eSBmb3JlSHVhd2VpIHhvZnR3YXJlIChjcmVhdGVk
IG9uIDMwZGZzRGVjLDIwMTMwIDxzdXBw3U0QCh1YXdlas5jb20+igQE3BEMBAgAhBQJSwT47
AheDBgsJCACdAgYVCAIJCgsDFgIRAh4BAheAAAJEJatgd8rp0gk0QQAIAIJEdFLMiVJdx1S8
INxZHejGagTch9GqK1u6HQ3Hp590KRPINBgdsINFuOwc00WqBAzXtGILLQpBSWLas5c10HrEi
+Pq2Xkdbd3hZhrw/G/GH1HU1jIHNamTikalCz4B+BcsQ0UhfVXZDTbBAF9a9Md21dJsgzEaA
yqozd1MKse4JcJ7gY75L/DSWDcPHeJQJ1j5RQpxbDn4sgK1ZU314DCz8+Iiz30c31+WHr
yYC+g1gMeFWY1S1waltoqKJeb5b6VearJkDjXmLkV3kCnB8AMq8y50y92uBR58WNLxw37o
01gXaQmVtL7GkBQ20xLFHvIdU9UuzVX86Xyy7A=
=0zUT
-----END PGP PUBLIC KEY BLOCK-----
```

- Step 3** Copy the public key information to a TXT file and name it as KEYS.txt. If the public key length is 4096, save it as a KEYS4096.txt file.

----End

4.2.3 Importing the Public Key

- Step 1** Log in to the server that contains the software package to be verified as administrator and enter the CLI.
- Step 2** Run the following command to import the public key file. Set **C:\Users** to the actual path of public key file **KEYS.txt**.
- Step 3** Access the directory of the public key file **KEYS.txt** and run the following command.

```
gpg --import "C:\Users\KEYS.txt"
```

When sign data length is 4096, please select KEYS4096.txt for the public key file.

The following output is displayed for key length of 2048:

```
gpg: key 27A74824: public key "OpenPGP signature key for Huawei software (created
on 30th Dec,2013) <support@huawei.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
```

The following output is displayed for key length of 4096:

```
gpg: key 6ADE4A56: public key "OpenPGP signature key for Huawei software (created
on 15th Jun,2019) <support@huawei.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
```



NOTICE

In this command, two hyphens precede **import**.

- Step 4** Run the following command to check the public key import result.

```
gpg --fingerprint
```

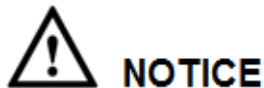
If the following output is displayed, the public key is successfully imported.

The following output is displayed for key length of 2048:

```
pub 2048R/27A74824 2013-12-30
    Key fingerprint = B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824
uid      OpenPGP signature key for Huawei software (created on 30th
Dec,2013) support@huawei.com
```

The following output is displayed for key length of 4096:

```
pub 4096R/6ADE4A56 2019-06-15
    Key fingerprint = E128 5E9D 7E7F 0DB0 A659 48AF FAAA 7A2E 6ADE 4A56
uid OpenPGP signature key for Huawei software (created on 15th Jun,2019)
<support@huawei.com>
```



In this command, two hyphens precede **fingerprint**.

----End

4.2.4 Verifying the Public Key

Step 1 In normal cases, to verify the OpenPGP public key, you need to check the ID, fingerprint, and user ID of the public key with the entity that releases the public key. Information about the OpenPGP public keys released by Huawei is shown as follows:

The following output is displayed for key length of 2048:

- Public key ID: 27A74824
- Key fingerprint: B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824
- User ID (uid): OpenPGP signature key for Huawei software (created on 30th Dec,2013)
support@huawei.com

The following output is displayed for key length of 4096:

- Public key ID: 6ADE4A56
- Key fingerprint: E128 5E9D 7E7F 0DB0 A659 48AF FAAA 7A2E 6ADE 4A56
- User ID (uid): OpenPGP signature key for Huawei software (created on 15th Jun,2019)
<support@huawei.com>

After verifying the preceding information, set the trust level for the key.

Step 2 Run the following command to set the trust level.

```
#gpg --edit-key "OpenPGP signature key for Huawei software (created on
30th Dec,2013)" trust
set the trust level as follows, If the public key length is 4096
#gpg --edit-key "OpenPGP signature key for Huawei software (created on
15th Jun,2019)" trust
```

The output resembles the following information. You need to enter 5 behind **Your decision?** to indicate **I trust ultimately** and y behind **you really want to set this key to ultimate trust? (y/N)**.

```
gpg (GnuPG) 2.0.9; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

```
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
pub 2048R/27A74824 created: 2013-12-30 expires: never usage: SC
trust: ultimate validity: ultimate
[ultimate] (1). OpenPGP signature key for Huawei software (created on 30th
Dec,2013) <support@huawei.com>

pub 2048R/27A74824 created: 2013-12-30 expires: never usage: SC
trust: ultimate validity: ultimate
[ultimate] (1). OpenPGP signature key for Huawei software (created on 30th
Dec,2013) <support@huawei.com>

Please decide how far you trust this user to correctly verify other users' keys
(by looking at passports, checking fingerprints from different sources, etc.)

1 = I don't know or won't say
2 = I do NOT trust
3 = I trust marginally
4 = I trust fully
5 = I trust ultimately
m = back to the main menu

Your decision? 5
Do you really want to set this key to ultimate trust? (y/N) y

pub 2048R/27A74824 created: 2013-12-30 expires: never usage: CS
trust: ultimate validity: unknown
[ unknown] (1). OpenPGP signature key for Huawei software (created on 30th
Dec,2013) <support@huawei.com>
Please note that the shown key validity is not necessarily correct
unless you restart the program.
```



NOTICE

In this command, two hyphens precede **edit**.

Step 3 Run the following command to quit.

```
quit
```

----End

4.3 Verifying the Signature

The signature file must be in the same path as that of the software package. In this example, the signature file path is **C:\Users**. Run the following command to verify the signature.

```
gpg --verify "C:\Users\V100R001C041.zip.asc"
```

The following output is displayed for key length of 2048:

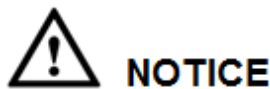
```
gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824
gpg: Good signature from "OpenPGP signature key for Huawei software (created on
30th Dec,2013) <support@huawei.com>"
```

RSA key ID in bold is the same as the public key ID. If no WARNING, public key expiry, signature expiry, or public key revocation information is display, the signature is valid.

The following output is displayed for key length of 4096:

```
gpg: Signature made Mon Dec 30 12:16:07 2019 CST using RSA key ID 6ADE4A56
gpg: Good signature from "OpenPGP signature key for Huawei software (created on 15th
Jun,2019) <support@huawei.com>"
```

RSA key ID in bold is 6ADE4A56, and there is no WARNING, which means this signature is published by Huawei.



- If a version has multiple signature files to be verified, the version is considered secure only when all the files pass the verification. If the verification result of a file is WARN or FAIL, the version does not pass the verification and has security risks. In this case, try to solve problem following the suggest action in table 4-1.
- In this command, two hyphens precede **verify**.

Table 4-1 Example for signature verification result judgment

Verification Result Scenario	Output Example	Verificati on Result	Suggest action
Signature verification succeeds, with no anomaly.	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: Good signature from "OpenPGP signature key for Huawei software (created on 30th Dec,2013) <support@huawei.com>"	PASS	NA
Signature verification fails.	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: BAD signature from "OpenPGP signature key for Huawei software (created on 30th Dec,2013) <support@huawei.com>"	FAIL	Download the software package again, or turn to product support
The public key is not found.	gpg: Signature made Thu Jan 9 15:20:01 2014 CST using RSA key ID 27A74824 gpg: Can't check signature: public key not found	FAIL	Download public key, see: Obtaining the Public Key File

Verification Result Scenario	Output Example	Verification Result	Suggest action
Signature verification succeeds, but the public key is not set to be completely trusted.	gpg: Signature made Thu Jan 9 15:29:06 2014 CST using RSA key ID 27A74824 gpg: Good signature from "OpenPGP signature key for Huawei software (created on 30th Dec,2013) <support@huawei.com>" gpg: WARNING: This key is not certified with a trusted signature! gpg: There is no indication that the signature belongs to the owner. Primary key fingerprint: B100 0AC3 8C41 525A 19BD C087 99AD 81DF 27A7 4824	WARNING	First check if key ID is 27A74824, and set the public key to be fully trusted, see: Verifying the Public Key
The corresponding source file cannot be found.	gpg: no signed data gpg: can't hash datafile: No data	FAIL	Download the software package again, or turn to product support
The signature has expired.	gpg: Signature made 04/24/13 10:50:29 CST using RSA key ID 133B64E5 gpg: Expired signature from " OpenPGP signature test key <support@huawei.com>" gpg: Signature expired 04/25/13 10:50:29 CST	FAIL	Download the software package with updated signature, or turn to product support
Signature verification succeeds, but the public key is revoked.	gpg: Signature made 06/13/13 11:14:49 CST using RSA key ID 133B64E5 gpg: Good signature from " OpenPGP signature test key <support@huawei.com>" gpg: WARNING: This key has been revoked by its owner! gpg: This could mean that the signature is forged. gpg: reason for revocation: Key is no longer used gpg: revocation comment:	WARNING	Download the newest public key and software package with updated signature, or turn to product support

Verification Result Scenario	Output Example	Verification Result	Suggest action
The signature file corresponding to the source file cannot be found.	N/A	WARNING	Download signature file of software package, or turn to product support

5 PGPVerify (Windows&Linux)

To prevent threats to carrier networks caused by software tampering or damage in transit, verify the integrity of software packages after receiving them. Only verified software packages can be deployed.

5.1 Background

PGPVerify is a PGP simplified verification tool developed by Huawei. It runs on Windows 7, Windows Server 2008, Windows 8, and Windows 10 platforms.

Software packages and signature files are stored in the same directory. A software package corresponds to a verification file.

The signature files use the same file names as those used by software packages, with the file name extension being **.asc**. For example, if the software package name is **V100R001C04.zip**, the corresponding verification file name is **V100R001C04.zip.asc**.

5.2 Prerequisites

5.2.1 Obtaining PGPVerify

PGPVerify requires no installation. You can download it from the following websites.

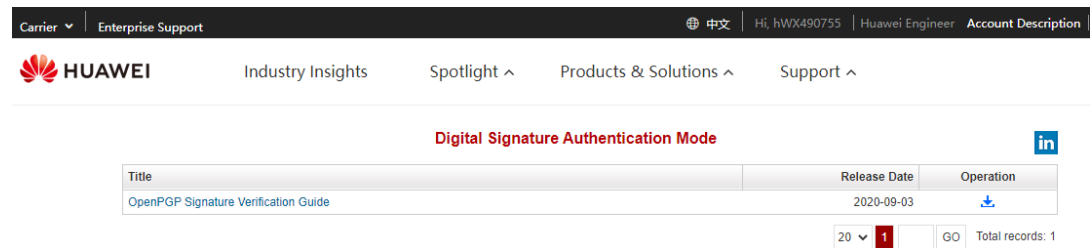
<http://support.huawei.com/carrier>

Carrier users download PGPVerify from the following uniform resource locator (URL):

<http://support.huawei.com/carrier/digitalSignatureAction>

The target web page may be displayed in Chinese. Click **English** on the top of the web page to switch the language to English.

Figure 5-1 Web page for downloading PGPVerify from <http://support.huawei.com>



To download PGPVerify, perform the following steps:

- Step 1** Click **Download** to download the **OpenPGP Signature Verification Guide** ZIP package. Then decompress it.
 - Step 2** Further decompress the **VerificationTools.zip** package.
 - Step 3** Open the decompressed folder **VerificationTools** and obtain the **PGPVerify** verification tool.
- End

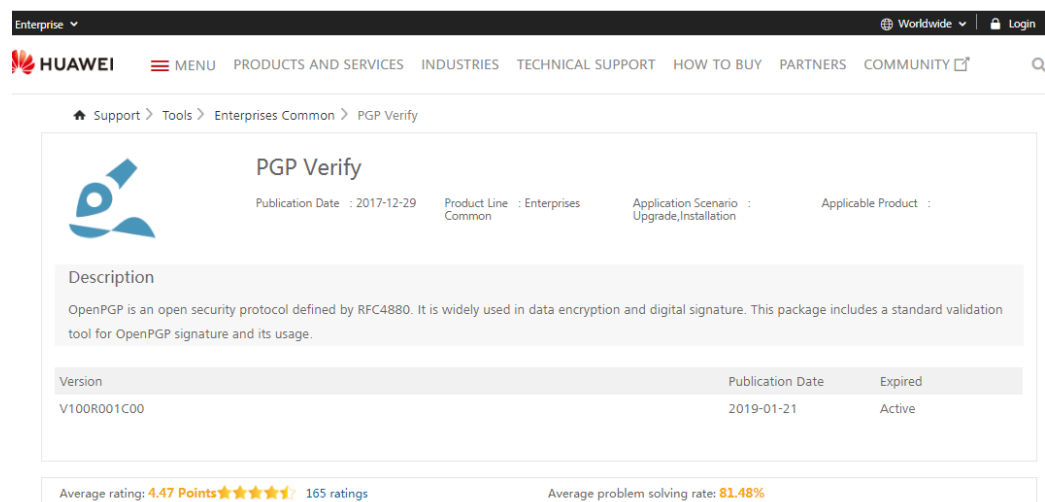
<http://support.huawei.com/enterprise>

Enterprise users download PGPVerify from the following uniform resource locator (URL):

<http://support.huawei.com/enterprise/en/tool/software-digital-signature-validation-tool-%EF%BC%88pgp-verify%EF%BC%89-TL1000000054>

The target web page may be displayed in Chinese. First click the earth mark on the top of the page, and then choose English to change the page's language, finally you can download the English document.

Figure 5-2 Web page for downloading PGPVerify from <http://support.huawei.com/enterprise>



To download PGPVerify, perform the following steps:

- Step 1** Click the version number in the **Version** list.

- Step 2** In the displayed page, click  corresponding to **VerificationTools.zip** to download this package.
- Step 3** Decompress the **VerificationTools.zip** package and obtain the **PGPVerify** verification tool.
- End

Terminal Knowledge Base

Terminal users download PGPVerify from the terminal knowledge base at the following URL:

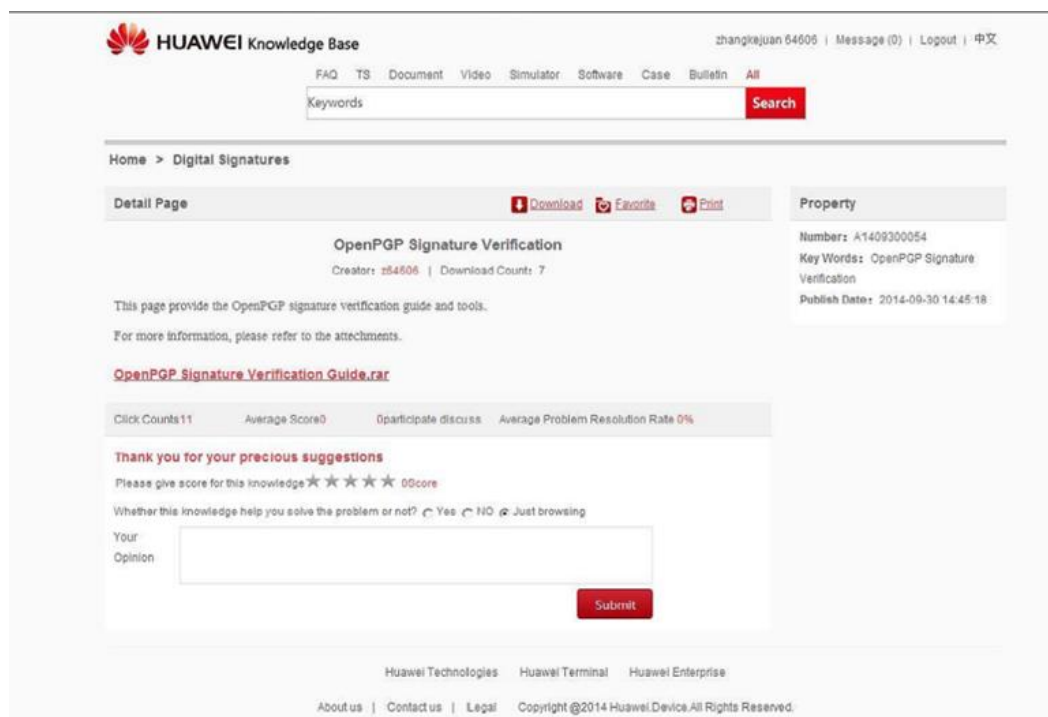
<http://app.huawei.com/tkb/#!tservice/common/base/digit.html?resourceId=146680>

To download PGPVerify, perform the following steps:

- Step 1** Access <http://app.huawei.com/tkb/#!tservice/common/base/digit.html?resourceId=146680>.

Figure 5-3 shows the displayed web page.

Figure 5-3 Web page for downloading PGPVerify from the terminal knowledge base



- Step 2** Click **Download** to download the **OpenPGP Signature Verification Guide.rar** package.
- If you have relevant permissions but an error is displayed, select the correct language.
- Step 3** Decompress the downloaded **OpenPGP Signature Verification Guide.rar** package and obtain the **PGPVerify** verification tool.
- End

5.2.2 Obtaining the Public Key File

Downloading the File from the Support Website

- URL at <http://support.huawei.com/carrier>
<http://support.huawei.com/carrier/digitalSignatureAction>
- URL at <http://support.huawei.com/enterprise/>
<http://support.huawei.com/enterprise/en/tool/software-digital-signature-validation-tool-%EF%BC%88pgp-verify%EF%BC%89-TL1000000054>
- URL for downloading the terminal knowledge base
<http://app.huawei.com/tkb/#!/tservice/common/base/digit.html?resourceId=146680>



NOTE

The public key file is compressed in the same package as the verification tool. Therefore, the URL for downloading the public key file is the same as that for downloading the verification tool. The public key file is named **KEYS.txt** or **KEYS4096.txt**.

Obtaining from the Public Key Server

Step 1 Access the public key server.

<https://zimmermann.mayfirst.org>

The result is displayed, as shown in Figure 5-4.

Figure 5-4 Public key server address page

Not secure | zimmermann.mayfirst.org

You can find a key by typing in some words that appear in the userid (name, email, etc.) of the key you're looking for, or by typing in the keyid in hex format ("0x...")

Search for a public key

String

Show PGP Fingerprints ☐

Show SKS full-key hashes ☐

Get regular index of matching keys ☐

Get verbose index of matching keys ☒

Retrieve ascii-armored keys ☐

Retrieve keys by full-key hash ☐

Enter " OpenPGP signature key for Huawei software "in the String text box, and click" Search for a key" to search for the public key.

Figure 5-5 Public key search result

Search results for 'software signature openpgp key huawei for'

Type	bits/keyID	cr. time	exp time	key expir
pub	4096R/6ADE4A56	2019-06-15		
uid OpenPGP signature key for Huawei software (created on 15th Jun,2019) <support@huawei.com>				
sig	sig3 6ADE4A56	2019-06-15		[selfsig]
pub	2048R/27A74824	2013-12-30		
uid OpenPGP signature key for Huawei software (created on 30th Dec,2013) <support@huawei.com>				
sig	sig3 27A74824	2013-12-30		[selfsig]

Step 2 Click public key ID 27A74824 to check the details, as shown in Figure 5-6. If the public key length is 4096, the corresponding ID is 6ADE4A56.

Figure 5-6 Public key details

Public Key Server -- Get "0x99ad81df27a74824 "

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: SKS 1.1.6
Comment: Hostname: zimmermann.safefirst.org

mQENBFLPjsBCActQyKqccsmIa3GvRoPHpfzB9ITrYeN0vfSF1Jal4esQgQuA513ILTS9bdG
H90sLufuryVogVRHrpajhuqSJycTPn/3VXtUWm27zVSHrKh0Tm3z2rIeu0dJzaKpgrtkw
RzeutDFW8GpwrQsGEOGLNclm+FRYdjatVru6SKcC7zjhfY2/TIj4nGGWCP+ebjQGoLBMjC
2o9fJi+UV5LxHrB896YpcKKs+JTaVSKSnQ8fG23D1DofBgwr19icaXm17bPx2WYRutdUKS6
HKh17Vf4v4T4t3BoaKj0xE/cf761FFrDNIFxSpRqQ3A4kS3UOUFgl/SNUBkevdMI5afAEEB
AAG0WU9wZW5QR1Agc2lrbaf0dXJlIGtleSBmb3JlSHVhd2VpIHhvZnR3YXJlChjcmVhdGVk
IG9uIDMwdGgRGVjLDIwMTMwIDxzdBw3J0QGH1YXdlas5jb20+iQE3BEMBAgABQJSwT47
AheD6gsJCACdAgYWCALJCgsDFgIBAh4BAheAAoJEJatgd8rp0gk0GQIAIJkFLMvJdx1S8
INxZHejGagTch9GqKlu8HQ3Hp59OKKPINBgd6INFu0wc00WqBAzXfGLLQpBSWLa5cIOHrEi
+Pq2XkddL3hZrw/G/GHJHJtjIHNaTikalCz4B+BcsQ0UhfVXZDTkBAF9a9Md2ldJsgzEaA
yqpozdlMKse4Jcjt7Y75L/DSWDcPHeJQ1j5RQpxbDn4sgxiZU314DCz8+Iiz30c31+VHr
yYC+g1gMeFWKY1S1waltoqKJeb5b6VEarJKDJXmLkV3kCnB8AMq8yS0y92uBR58WNLxw3To
01grXaQMvTl7GkbQ20xLFH+IdU9UUzVX86Xyy7A=
=0zUT
-----END PGP PUBLIC KEY BLOCK-----
```

Step 3 Copy the public key information to a TXT file and name it as KEYS.txt. If the public key length is 4096, save it as a KEYS4096.txt file.

----End

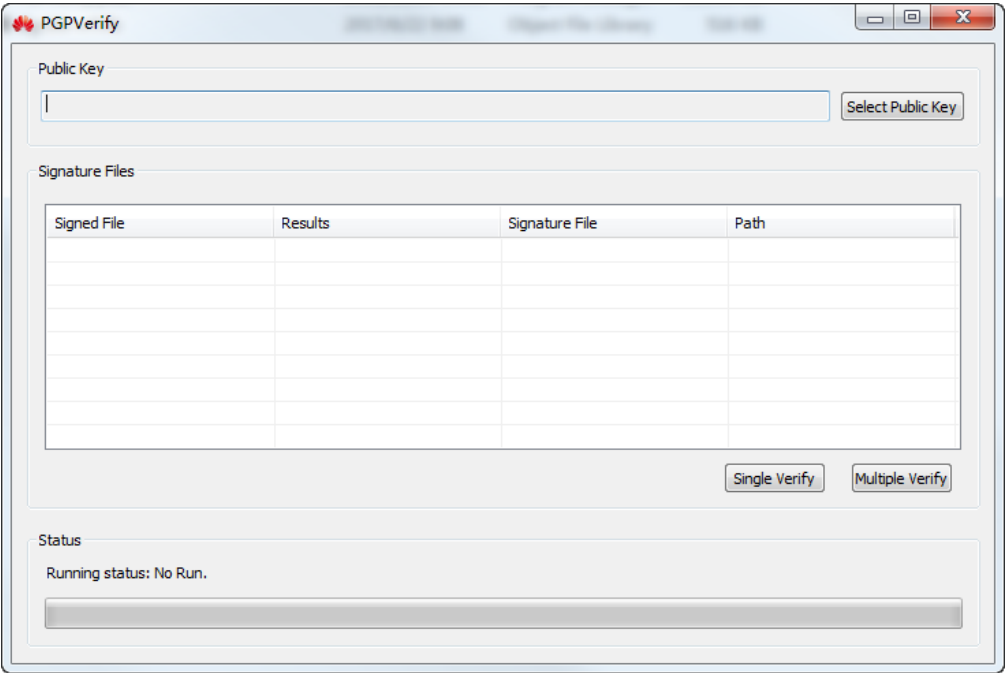
5.3 Verifying the Signature

The signature file must be in the same path as the software package. For example, if the signature file is in C:\PGP, but the software package is in C:\, you must move the signature file to C:\.

5.3.1 Verifying Through Operations on the UI

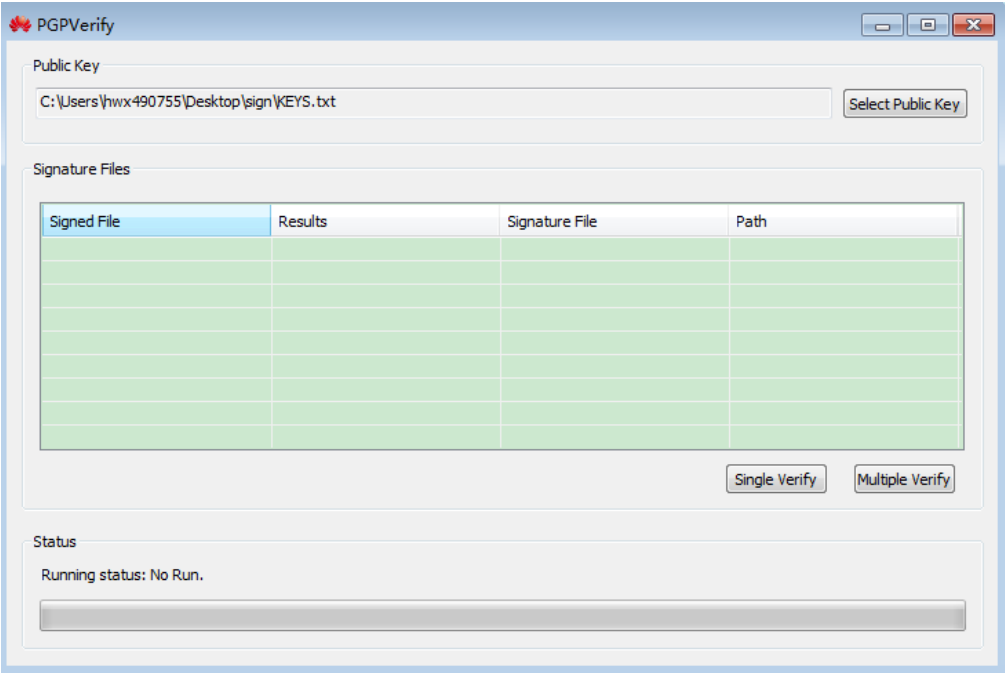
Step 1 Double-click **PGPVerify.exe** to start it, as shown in Figure 5-7.

Figure 5-7 PGPVerify UI



Step 2 Load the public key file.
Click **Select Public Key** to select the KEYS.txt file downloaded in section 5.2.2.

Figure 5-8 Loading the public key



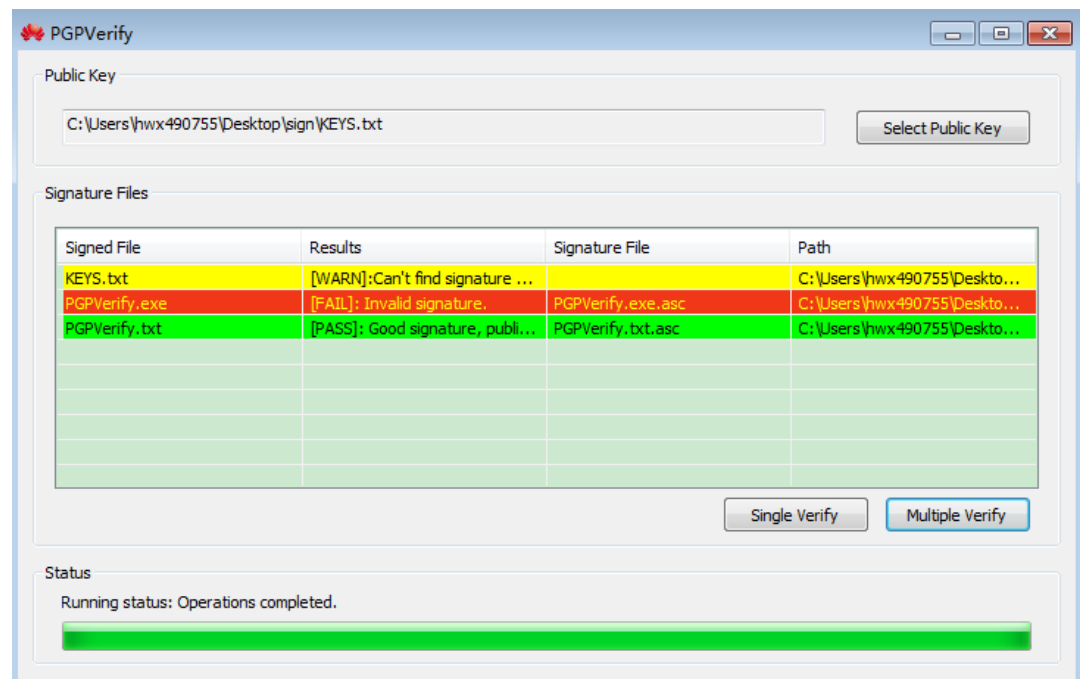
when sign data lenth is 4096, please select KEYS4096.txt for the public key file.

If you have used this verification tool on the same computer before, the last key you selected will be automatically reloaded when you use this tool once again.

Step 3 Verify files.

- Verifying a single file
Click **Single Verify** and select the .asc signature verification file.
- Verifying all files in the directory
Click **Multiple Verify** and select the **C:\PGP** directory. Figure 5-9 shows the verification result.

Figure 5-9 PGPVerify verification result



Step 4 Check the result.

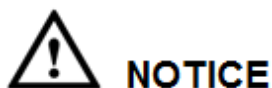
If an item is highlighted in yellow and the **Results** column is **WARN**, it indicates that the signature cannot be verified.

If an item is highlighted in red and the **Results** column is **FAIL**, it indicates that the verification failed.

If an item is highlighted in green and the **Results** column is **PASS**, it indicates that the verification using the specified public key succeeded.

If an item is highlighted in green and the public key fingerprint in the **Results** column is **B1000AC3 8C41525A 19BDC087 99AD81DF 27A74824**, it indicates that this signature is a valid one issued by Huawei OpenPGP key length for 2048.

If an item is highlighted in green and the public key fingerprint in the **Results** column is **E128 5E9D 7E7F 0DB0 A659 48AF FAAA 7A2E 6ADE 4A56**, it indicates that this signature is a valid one issued by Huawei OpenPGP key length for 4096.



If a version has multiple signature files to be verified, the version is considered secure only when all files are highlighted in green ([PASS]) and all public key fingerprints are **B1000AC3 8C41525A 19BDC087 99AD81DF 27A74824** (when OpenPGP key length is 4096, fingerprints are **E1285E9D 7E7F0DB0 A65948AF FAAA7A2E 6ADE4A56**), which means the signature is issued by Huawei. Otherwise, please obtain a new software package.

----End

5.3.2 Verification Through the CLI (Windows)

The signature file must be in the same path as the software package. For example, if the signature file is in C:\PGP, but the software package is in C:\, you must move the signature file to C:\.

Step 1 Verify files.

- Verifying a single file

Enter the CLI and run the following command to verify the signature:

```
"C:\PGPVerify.exe" -k "C:\KEYS.txt" -f "C:\PGP\Tecal CH224.zip.asc"
```

C:\KEYS.txt is the public key, and **C:\PGP\Tecal CH224.zip.asc** is the signature file. If sign data length is 4096, please select **KEYS4096.txt** for the public key file.

The following output is displayed for key length of 2048:

```
[PASS]:Good Signature. File path: C:\PGP\Tecal CH224.zip.asc, Public key  
fingerprint: B1000AC3 8C41525A 19BDC087 99AD81DF 27A74824  
[INFO]: Verify Complete.
```

The following output is displayed for key length of 4096:

```
[PASS]:Good Signature. File path: C:\PGP\Tecal CH224.zip.asc, Public key  
fingerprint: E1285E9D 7E7F0DB0 A65948AF FAAA7A2E 6ADE4A56  
[INFO]: Verify Complete.
```

- Verifying all files in a directory

Enter the CLI and run the following command to verify the signatures:

```
"C:\PGPVerify.exe" -k "C:\KEYS.txt" -d "C:\PGP"
```

C:\KEYS.txt is the public key, and **C:\PGP** is the directory in which signature files will be verified. If sign data length is 4096, please select **KEYS4096.txt** for the public key file.

The following output is displayed:

```
[INFO]:Filter file in directory, please wait...  
[WARN]:Can't find signature file, signed file position: C:\PGP\Tecal CH221.zip.  
[WARN]:Can't find signed file, signature file position: C:\PGP\Tecal  
CH222.zip.asc.  
[FAIL]:Invalid Signature. File path: C:\PGP\Tecal CH223.zip.  
[PASS]:Good Signature. File path: C:\PGP\Tecal CH224.zip, Public key fingerprint:  
B1000AC3 8C41525A 19BDC087 99AD81DF 27A74824
```

```
[INFO]: Verify Complete.
```

Step 2 Check the result.

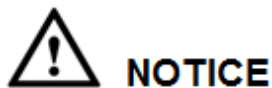
If an item's verification result is **WARN**, it indicates that the signature cannot be verified.

If an item's verification result is **FAIL**, it indicates that the verification failed.

If an item's verification result is **PASS**, it indicates that the verification using the specified public key succeeded.

If an item's verification result is **PASS** and the public key fingerprint is **B1000AC3 8C41525A 19BDC087 99AD81DF 27A74824**, it indicates that this signature is a valid one issued by Huawei OpenPGP key length for 2048.

If an item's verification result is **PASS** and the public key fingerprint is **E128 5E9D 7E7F 0DB0 A659 48AF FAAA 7A2E 6ADE 4A56**, it indicates that this signature is a valid one issued by Huawei OpenPGP key length for 4096.



If a version has multiple signature files to be verified, the version is considered secure only when all items' verification results are all **PASS** and all public key fingerprints are **B1000AC3 8C41525A 19BDC087 99AD81DF 27A74824** (when OpenPGP key length is 4096, fingerprints are **E1285E9D7E7F0DB0A65948AFFAAA7A2E6ADE4A56**), which means the signature is issued by Huawei. Otherwise, please obtain a new software package.

5.3.3 Verification Through the CLI (Linux)

Signature files must be stored in the same directory (**/usr1** in this example) as the software package. The verification tools and public key file can be stored in another directory or in the same directory as the signature files. In this example, the verification tools and public key file are also stored in **usr1**.

Step 1 Verify files.

- Verifying a single file

```
./PGPVerify -k KEYS.txt -f scw.cab.asc
```

KEYS.txt is the public key, and **scw.cab.asc** is the signature file. If sign data length is 4096, please select **KEYS4096.txt** for the public key file.

The following output is displayed for key length of 2048:

```
[PASS]:Good Signature. File path: scw.cab.asc, Public key fingerprint: 97399A82  
CD5D7160 13D181FC 0D7AC54D F0B00048.  
[INFO]: Verify Complete.
```

The following output is displayed for key length of 4096:

```
[PASS]:Good Signature. File path: scw.cab.asc, Public key fingerprint: E1285E9D  
7E7F0DB0 A65948AF FAAA7A2E 6ADE4A56  
[INFO]: Verify Complete.
```

- Verifying all files in a directory

In this example, all signature files and the software package are stored in the **openpgp** directory. To verify all files in this directory, run the following command:

```
./PGPVerify -k KEYS.txt -d openpgp
```

The following output is displayed:

```
[INFO]:Filter file in directory, please wait...
[PASS]:Good Signature. File path: openpgp/plugins-cloudtask-C01.zip.asc, Public
key fingerprint: FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/twain.dll.asc, Public key fingerprint:
FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/buildcloud-proxy.zip.asc, Public key
fingerprint: FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/buildcloud_pvmtrans.zip.asc, Public key
fingerprint: FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/plugins-cicloud-C01.zip.asc, Public key
fingerprint: FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/ConfigCenter.war.asc, Public key
fingerprint: FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/watcher-wrapper.zip.asc, Public key
fingerprint: FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/watcher.zip.asc, Public key fingerprint:
FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/rpm.war.asc, Public key fingerprint:
FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[PASS]:Good Signature. File path: openpgp/buildcloud-rpm.zip.asc, Public key
fingerprint: FA60975B 1160DF6D 0059662D 2689E7E3 393905AC.
[INFO]: Verify Complete.
```

Step 2 Check the result.

If an item's verification result is **WARN**, it indicates that the signature cannot be verified.

If an item's verification result is **FAIL**, it indicates that the verification failed.

If an item's verification result is **PASS**, it indicates that the specified public key passed the verification.

If an item's verification result is **PASS** and the public key fingerprint is **B1000AC3 8C41525A 19BDC087 99AD81DF 27A74824**, it indicates that this signature is a valid one issued by Huawei OpenPGP key length for 2048.

If an item's verification result is **PASS** and the public key fingerprint is **E128 5E9D 7E7F 0DB0 A659 48AF FAAA 7A2E 6ADE 4A56**, it indicates that this signature is a valid one issued by Huawei OpenPGP key length for 4096.



NOTICE

1. If a version has multiple signature files to be verified, the version is considered secure only when all items' verification results are **PASS** and all public key fingerprints are **B1000AC38C41525A 19BDC087 99AD81DF 27A74824** (when OpenPGP key length is 4096, fingerprints are **E1285E9D7E7F0DB0A65948AFFAAA7A2E6ADE4A56**), which means the signature is issued by Huawei. Otherwise, please obtain a new software package.
 2. If the verify result is "Permission denied" in step 1, please add execution attribute to file by executing "chmod u+x PGPVerify" first.
-

--End

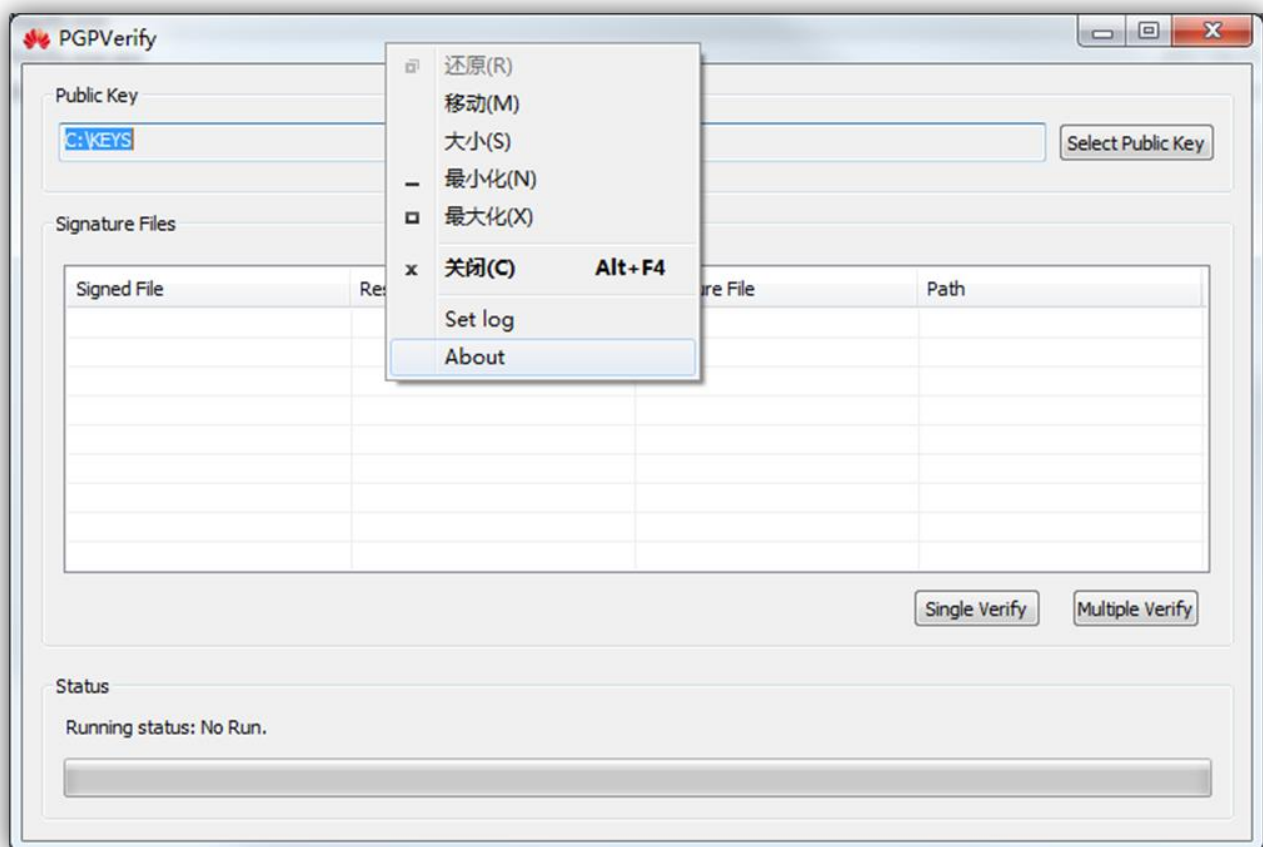
6 FAQs

6.1 The Application Scope for use Verification Tool?

PGPVerify tool is a simple signature verify tool published by Huawei Technologies Co.,Ltd. It's can only be used to verify integrity of binary packages, which published by Huawei, by hand. It SHOULD NOT be repacked or republished with any commercial products; Besides, according to positioning of PGPVerify tool, it will not accessing to users' network or data.

6.2 How to Obtain version of Verification Tool?

- **With UI mode(suitable for Windows®)**
 1. Right click title bar of tool window, and click "About" menu.



2. Check version of application:



- **Console mode (suitable for Windows® and Linux)**

1. Input command line:

```
PGPVerify -h
```

2. Take “Windows console” for example, show result as below(for version V100R001C00SPC310):

```
PGPVerify library version V100R001C00SPC310  
Copyright (c) Huawei Technologies Co., Ltd. 2017. All rights reserved.
```

Command:

```
-k: public key.  
-d: The directory which to be verified.  
-f: The file which to be verified.  
-l: Set log file.
```

Example:

```
PGPVerify -k KEYS -d file-directory  
PGPVerify -k KEYS -f signed-file
```

The message in first line contains the version info.

- **Obtain the version by PowerShell (For Windows® only)**

1. Input command line:

```
((.\PGPVerify.exe -h | findstr " V100" | Out-String).split(" ") | findstr  
V100).remove(0,1).replace("SPC", ".").replace("C", ".").replace("R", ".")
```

Attention: The string “V100” is fixed for the version, and it won’t change for releases in feature, so just use it as a version keyword here.

2. Show result as below (task V100R001C00SPC310 for example):

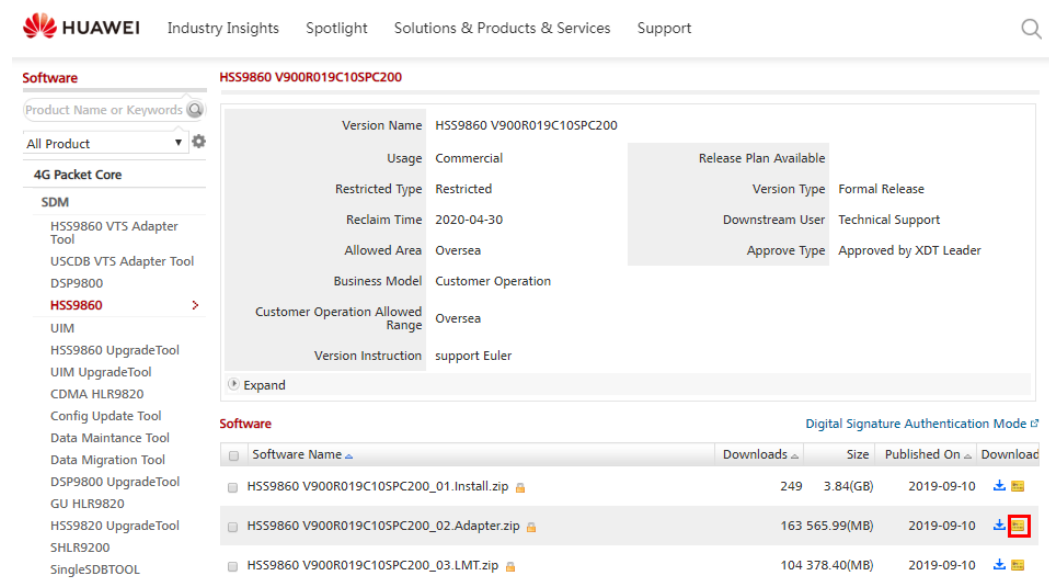
```
100.001.00.310
```

6.3 How to Obtain an .asc File?

This section uses <http://support.huawei.com/carrier> as an example to describe how to obtain the .asc file for the specific product software.

As shown in Figure 6-1 the **Software Name** column lists the software. Click yellow envelope button corresponding to a software package to download the .asc signature file for the software package.

Figure 6-1 Web page for downloading the .asc file from <http://support.huawei.com/carrier>



6.4 How to Obtain the Public Key or Verification Tools?

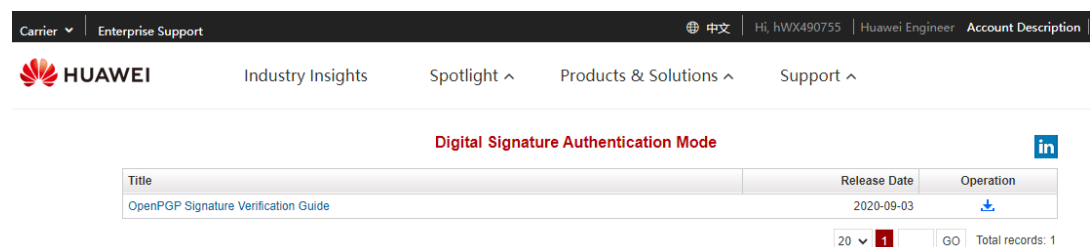
- Carrier users download the public key file or verification tools from the following URL at <http://support.huawei.com/carrier>:

<http://support.huawei.com/carrier/digitalSignatureAction>

The target web page may be displayed in Chinese. Click **English** on the top of the web page to switch the language to English, as shown in Figure 6-2.

Download and decompress the **OpenPGP Signature Verification Guide** package. The **KEYS.txt** or **KEYS4096.txt** file is the public key file. The **VerificationTools.zip** package contains two signature verification tools: PGPVerify and Gpg4Win. You can decompress **VerificationTools.zip** to obtain either tool.

Figure 6-2 Web page (in Chinese) for downloading the public key file or verification tools from <http://support.huawei.com>



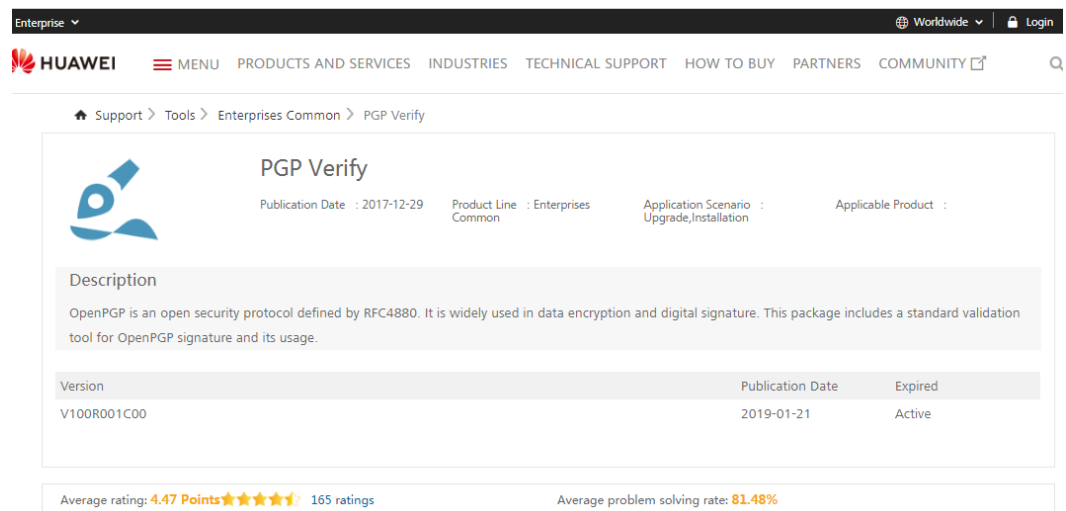
- Enterprise users download the public key file or verification tools from the following URL at <http://support.huawei.com/enterprise>:

<http://support.huawei.com/enterprise/en/tool/software-digital-signature-validation-tool-%EF%BC%88pgp-verify%EF%BC%89-TL1000000054>

The target web page may be displayed in Chinese. Click **Worldwide** on the top of the web page to switch the language to English, as shown in Figure 6-3.

Click the version number in the **Version** list. Then click  corresponding to the public key file **KEYS.txt** or **KEYS4096.txt** to download the file. Download and decompress **VerificationTools.zip** to obtain either of the signature verification tools PGPVerify and Gpg4Win.

Figure 6-3 Web page (in Chinese) for downloading the public key file or verification tools from <http://support.huawei.com/enterprise>



- Terminal users download the public key file or verification tools from the terminal knowledge base at the following URL:

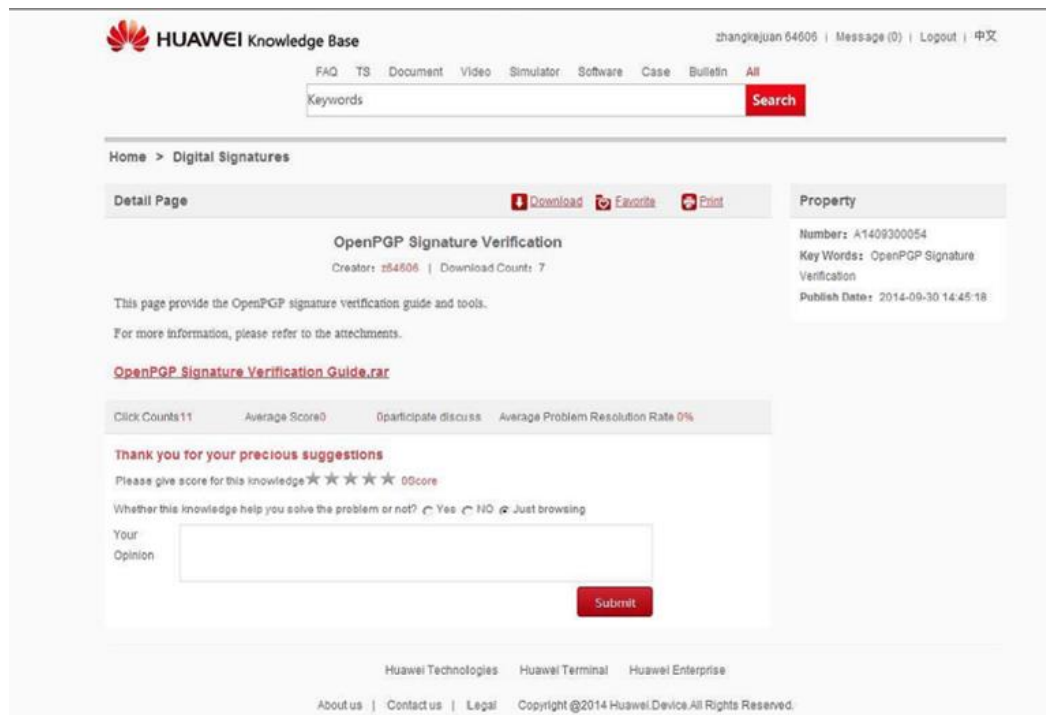
<http://app.huawei.com/tkb/#!/tservice/common/base/digit.html?resourceId=146680>

To download the public key file or verification tools from the terminal knowledge base, perform the following steps:

Step 1 Access <http://app.huawei.com/tkb/#!/tservice/common/base/digit.html?resourceId=146680>.

Figure 6-4 shows the displayed web page.

Figure 6-4 Web page for downloading the public key file or verification tools from the terminal knowledge base



Step 2 Click **Download** to download the **OpenPGP Signature Verification Guide.rar** package.

If you have relevant permissions but an error is displayed, select the correct language.

Step 3 Decompress the downloaded **OpenPGP Signature Verification Guide.rar** package.

- The **KEYS.txt** or **KEYS4096.txt** file is the public key file.
- **VerificationTools.zip** contains the verification tools. You can decompress the package to obtain the Gpg4Win and PGPVerify verification tools.

----End

6.5 How Is Signature Verification Implemented?

Signature verification is implemented as follows:

- This document provides three OpenPGP signature verification methods. In all these methods, the files to be verified must have been signed using OpenPGP signatures, and verification is performed on the basis of an .asc file.
- The public key file and signature verification tools are stored in the same path.

6.6 How to Switch the Language of a Web Page from Chinese to English?

If any of the web pages at the given URLs is displayed in Chinese, click **English** on the top of the web page to switch the language to English.

6.7 How to Rectify the Failure in Long Path Verification Using the PGPVerify.exe Command?

- If the failed path is a local path, for example, **D:\testfile.txt**:
Add \\?\ before the path name as follows:
\\?\D:\testfile.txt
Original command:
`PGPVerify.exe -k D:\KEYS.txt -f D:\testfile.txt`
New command:
`PGPVerify.exe -k \\?\D:\KEYS.txt -f \\?\D:\testfile.txt`
- If the failed path is a network path, for example, **\\10.172.12.12\sharedir\testfile.txt**:
Add \\?\UNC\ before the path name as follows:
\\?\UNC\10.172.12.12\sharedir\testfile.txt
Original command:
`PGPVerify -k \\10.172.12.12\sharedir\KEYS.txt -f
\\10.172.12.12\sharedir\testfile.txt`
New command:
`PGPVerify -k \\?\UNC\10.172.12.12\sharedir\KEYS.txt -f
\\?\UNC\10.172.12.12\sharedir\testfile.txt`

6.8 How to Obtain and Use the PGPVerify (Solaris/Linux) Tool?

- The path for downloading the PGPVerify (Solaris/Linux) tool and the method for obtaining the public key are the same as those for the PGPVerify (Windows) tool.
- The PGPVerify (Solaris/Linux) tool supports only signature verification using commands but does not support interface-based signature verification. The signature verification using commands is the same as that of the PGPVerify (Windows) tool.