

**Oracle® Communications
Network Charging and Control**

Voucher Print Shop Operations Guide

Release 12.0.0

December 2017

Copyright

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

About This Document	v
Document Conventions	vi
Chapter 1	
System Overview	1
Overview	1
Introduction	1
Managing Public/Private Key Pairs	2
Decrypting Files	12
Glossary of Terms	21
Index	23

About This Document

Scope

This document describes how Printshop:

- Generates and distributes the security key to the operator
- Decrypts the operator-provided voucher batch files

It also explains the format of the voucher batch file.

It does not include detailed design of the service.

Audience

This guide is intended for use by personnel of the print shop who will be responsible for the end-to-end voucher printing process.

Related Documents

The following documents are related to this document:

- *Charging Control Services Technical Guide*
- *Voucher Manager User's Guide*

Document Conventions

Typographical Conventions

The following terms and typographical conventions are used in the Oracle Communications Network Charging and Control (NCC) documentation.

Formatting Convention	Type of Information
Special Bold	Items you must select, such as names of tabs. Names of database tables and fields.
<i>Italics</i>	Name of a document, chapter, topic or other publication. Emphasis within text.
Button	The name of a button to click or a key to press. Example: To close the window, either click Close , or press Esc .
Key+Key	Key combinations for which the user must press and hold down one key and then press another. Example: Ctrl+P or Alt+F4 .
Monospace	Examples of code or standard output.
Monospace Bold	Text that you must enter.
<i>variable</i>	Used to indicate variables or text that should be replaced with an actual value.
menu option > menu option >	Used to indicate the cascading menu option to be selected. Example: Operator Functions > Report Functions
hypertext link	Used to indicate a hypertext link.

Specialized terms and acronyms are defined in the glossary at the end of this guide.

Terminology

This topic explains any terminology specific to this manual.

Operator

An operator is the telecommunications service provider which generates the vouchers or calling cards which need printing.

System Overview

Overview

Introduction

This chapter provides an overview of the software and formats used in preparing a voucher batch file for printing.

In this chapter

This chapter contains the following topics.

Introduction	1
Managing Public/Private Key Pairs	2
Decrypting Files	12

Introduction

Charging Control Services files and encryption

Charging Control Services (CCS) produces encrypted voucher and account batch files for printing. The encryption is used to provide security for the vouchers or subscriber accounts the files hold. Before the files are printed, the encrypted files must be decrypted using the same public private key pair that was used for the encryption.

For more information about how CCS generates vouchers and accounts, see *Charging Control Services User's Guide* and *Charging Control Services Technical Guide*.

Public and private key encryption

Public and private key encryption (also known as asymmetric encryption) involves a pair of keys:

- 1 a public key which is used to encrypt the file, and
- 2 a private key which is used to decrypt the file.

Both keys are generated by the holder of the private key. The public key is made available to others who want to send encrypted files to the private key holder. In this case, the print shop will generate the public and private keys and provide the public key to the operator.

For more information about:

- generating keys, see *Managing Public/Private Key Pairs* (on page 2).
- decrypting files, see *Decrypting Files* (on page 12).

More information about public and private key encryption is widely available in publications and on the Internet.

Recommended software

Oracle uses GnuPG to encrypt batch files. These files can be decrypted using any software which supports gnupg public private keys. This guide covers the GnuPG command line tool, and the GPG4Win software compatible with Windows.

Note: Other software such as PGP can also be used successfully for generating and exporting keys and decrypting files. Please use the software which is most suitable for your platform.

For more information about GnuPG (including downloadable software), see <http://www.gnupg.org>.

For more information about GPG4Win (including downloadable software), see <http://www.gpg4win.org>.

For more information about PGP (including purchasable software), see <http://www.pgp.com>.

Managing Public/Private Key Pairs

Generating GPG keys

A public and private GPG key can be generated from a pass-phrase. The private key is held only by the print shop and used only to decode the encrypted batch file. The public key is used to encrypt the file and must therefore be supplied to the operator who will be responsible for generating the voucher batch file.

For more information about using GPG keys with exported files, see *Print Shop Operations Guide*.

Generating keys using gpg

Follow these steps to generate a key using GnuPG.

Important: Additional documentation is available at <http://www.gnupg.org>. Always consult the recent documentation for your version of GnuPG if you are unsure of any steps in the procedure.

Step	Action
1	Log in to the machine which has the GnuPG tool installed.
2	Run the gpg binary. Example command: <code>./gpg --gen-key</code> Result: Text similar to the following appears: <pre>gpg (GnuPG) 2.0.22; Copyright (C) 2013 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. Please select what kind of key you want: (1) RSA and RSA (default) (2) DSA and Elgamal (3) DSA (sign only) (4) RSA (sign only) Your selection?</pre>
3	Enter the algorithm you have agreed with the operator you are printing for. Result: Text similar to the following will appear: <pre>RSA keys may be between 1024 and 4096 bits long. What keysize do you want? (2048)</pre>

Step	Action
4	<p>Enter the keysize you have agreed with the operator.</p> <p>Result: Text similar to the following will appear.</p> <pre>Requested keysize is 2048 bits Please specify how long the key should be valid. 0 = key does not expire <n> = key expires in n days <n>w = key expires in n weeks <n>m = key expires in n months <n>y = key expires in n years Key is valid for? (0)</pre>
5	<p>Enter the expiry period you have agreed with the operator.</p> <p>Result: Text similar to the following will appear.</p> <pre>Key does not expire at all Is this correct? (y/n) y</pre>
6	<p>If all the details entered so far are correct, type y and press Enter.</p> <p>Result: Text similar to the following will appear:</p> <pre>You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form: "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>" Real name:</pre>
7	<p>Type your real name and press Enter.</p> <p>Result: Text similar to the following will appear.</p> <pre>Email address:</pre>
8	<p>Type your email addresss and press Enter.</p> <p>Result: Text similar to the following will appear.</p> <pre>Comment:</pre>
9	<p>Type a comment and press Enter. The comment should identify who the printshop is, and may also identify the operator.</p> <p>Result: Text similar to the following will appear.</p> <pre>You selected this USER-ID: "ExampleUser (TelcoEurope-Printshop) <example.user@example.com>" Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?</pre>
10	<p>If the details which have been displayed are correct, type O and press Enter.</p> <p>Result: Text similar to the following will appear.</p> <pre>You need a Passphrase to protect your secret key. Enter a passphrase:</pre>
11	<p>Type a passphrase and press Enter.</p> <p>Important:</p> <ul style="list-style-type: none"> • This passphrase must be entered when the files are decrypted. If the passphrase is not available, the files will not be able to be decrypted and a new pair of keys and batch file will have to be generated. • The passphrase is an important contributor to the overall security of the encryption. Ensure you follow any guidelines set by the operator, and that you pick a secure password. <p>Result: Text similar to the following will appear:</p> <pre>Confirm passphrase:</pre>
12	<p>Type the passphrase from step 11 again and press Enter.</p> <p>Result: Text similar to the following will appear as gpg generates the keys.</p> <pre>We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number</pre>

Step	Action
	<pre>generator a better chance to gain enough entropy. gpg: key C57AAE57 marked as ultimately trusted public and secret key created and signed. 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/C57AAE57 2016-08-15 Key fingerprint = 1559 04E8 CFE3 523E 058B 180C 4B8D 47C2 C57A AE57 uid ExampleUser (TelcoEurope-Printshop) <example.user@example.com> sub 2048R/A7F4D59B 2016-08-15</pre>

Exporting keys using gpg

Follow these steps to export keys which have been generated by gpg.

Step	Action
1	<p>On the machine where the gpg generated the keys, instruct gpg to export the keys.</p> <p>Example command: <code>gpg --armour --export <email_addr_from_key_creation> > <file>.key</code></p> <p>Result: gpg will export the public key to the specified file.</p> <p>If you cat the file, it will look similar to the example key shown below:</p> <pre>-----BEGIN PGP PUBLIC KEY BLOCK----- Version: GnuPG v2.0.22 (GNU/Linux) mQENBFex0L0BCADH0y/fJmv3dBYo3XEnxLg+j8A/DrvvN/Sj1BB0z7J0bsmg5l0z vXk/AfoKkFLqIBFzh5eay0bIhdWitE6FrfnuyTPRQX0eBADZTQXf2KCw0Y55OeJn ETk1sVrm/DGA3EEXnbMnjmWP+tyCCU8idzkliyxbsuzH47IpanDQiX39VvbmjQ/5 dq02hzE5x1fhxZ/xWHyZ1idL0UfsKDBpFOXeelQIHylhuTOVuSs+0kXcfSHOGG1u unCwtYxMH93faK1FhtQ8wP1muVOXAn8/8aCe7eto7PcoBXOND2UtYnple2jUX6+L 4xbDUiv6091VzF/azSbkBB/BTw31mQ4XUMhdABEBAAAG0PUV4Yw1wbGVVc2VyICChU ZWxjb0V1cm9wZS1Qcm1udHNob3ApIDxleGFtcGxlLnVzZXJAcGFTcGxlLmNvbT6J ATkEEwECACMFAlEx0L0CGwMHcwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRBL jUFcXxquV2cgB/wLFjwvPUoyfARBZ2Iw0vpec3b+lis/MktGzB3tibwLfsDtqBs5 DvvKH5DN9jtCSpnLn9vxj0AysN893yDwk3eBwBgG+TztIemsdWuvyef82qomPV/G F2KmuKkPTvA0CSRwMXqXFmPvQ6z8EG9HRO6YOgaW37J9jTy0LzI54ijvBXJH/u5S u3l04fjiulWBCPNdCb/prPvZSca3ewaS0j8dPgr/Amx6KABAnTrj13N+zwKqljqf gT2KNR3AzXfR0gWnC2FaQGKTUgp9aAaryum1ILhKdu+w6k2xFNQ18NePnb9rnA4x yyrahIqXJW6t8790xwEU9thZ2GVc6UqGgS56uQENBFex0L0BCADLwspEvxWxkZvj 58WcJ1iDoJsYPVxQzALMYHC8luqu1SHiGZ8ny/PDXclgiOZ15J5hkxtHZJ3H18VE FYld9QH6fh9wM9tTkxUELUFWN9tnnk1QsISPbdKfDWBqccqEeG4tmVDthqZtTRzP wzp8UOag70/4zF8KDVBARcLVSv5yxDa/ezGZVhntYY/XUrQEaODB8VxQyofilN6o j/lu899z2GqAioolKbx72ByO/TFv4mSlpFBXtloGDIImS/R6E50r3mGBEQtPds60 R6BB0TTG6+iOgyqFuV2NAHQavWcGBiRgyT2F14z4NQuZ5/M0+cbGMKaUVv30h1gi KNsGae2bABEBAAAGJAR8EGAECaAkFAlex0L0CGwWACgkQs41HwsV6rlcNsQf+PqUz lGhBLB/g0s8ZzJ4Hn5rz6nQeMSTbZDg4GiryWfwkIytPejdoqPYG1l1NR9ZuAuj8 LXrItgdDNQlogJtD3kjMpWawqhNkD2zFFPmcTbduWjvJFKpMN8xDaDNwoQ5Qz/J 0qGVSbHYqF+Eo2xQdjGuOe1DDQ3s00QdmUUTUVXrhNWTpciL9zQ6JBSEGXE/V5N 1xi7Az1sp5SVY/WDCYBGxBcTTXMaKmqQOgqV5jIqOZGFgh4j8RPrLI6fkuyF5SP tmnaD0L2Yt1BHvSQFAtS5WvXkdM3+FCiXzQKvTDDuJjfrXuHQeYYO8LABJ1aNPE gEeda0KeB3Et3PhneQ== =0bv7 -----END PGP PUBLIC KEY BLOCK-----</pre>
2	<p>Send the public key to the operator. The operator will import it into Charging Control Services and will use it when generating voucher and subscriber account (calling card) batches.</p>

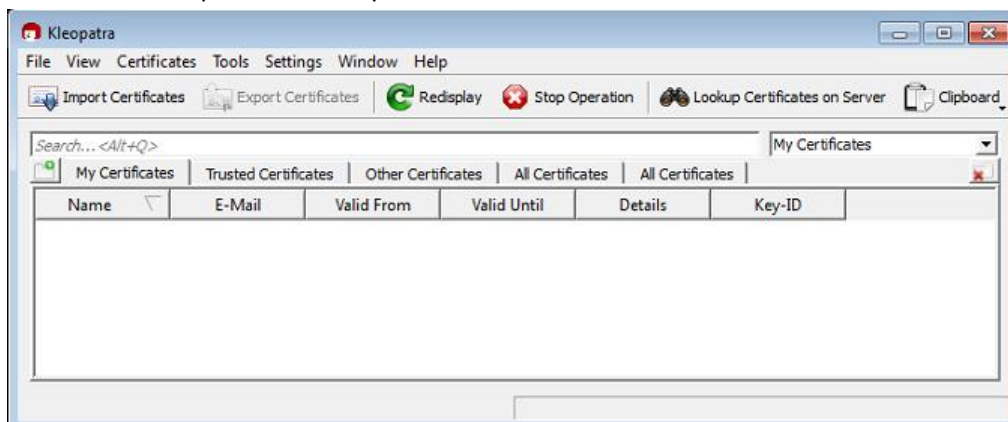
Generating keys using Gpg4Win

Follow these steps to generate a new key using Gpg4Win.

Step	Action
------	--------

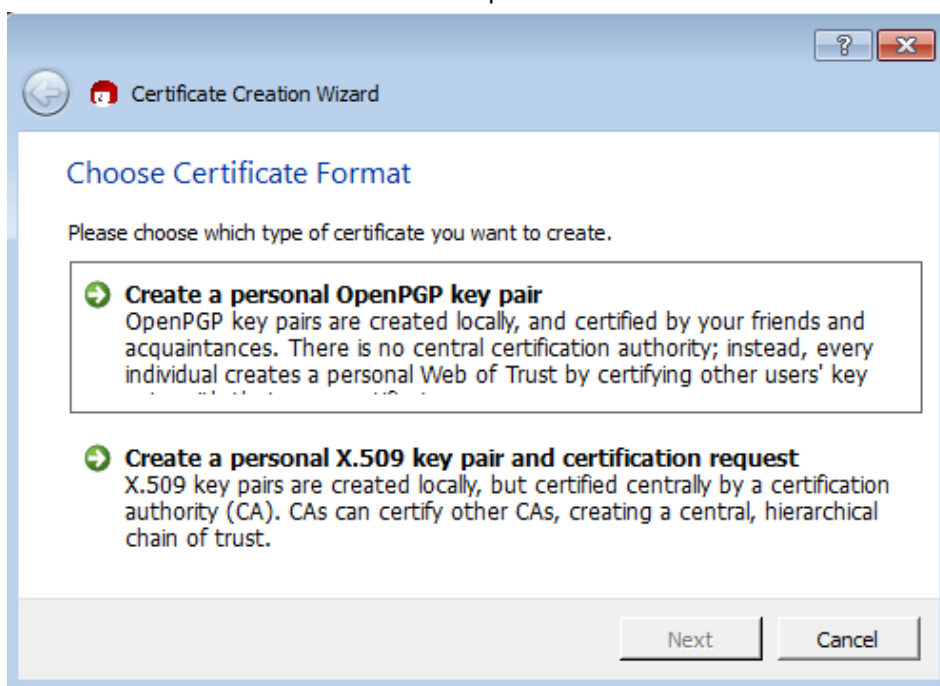
1	Start Kleopatra.
---	------------------

Result: The Kleopatra screen opens.



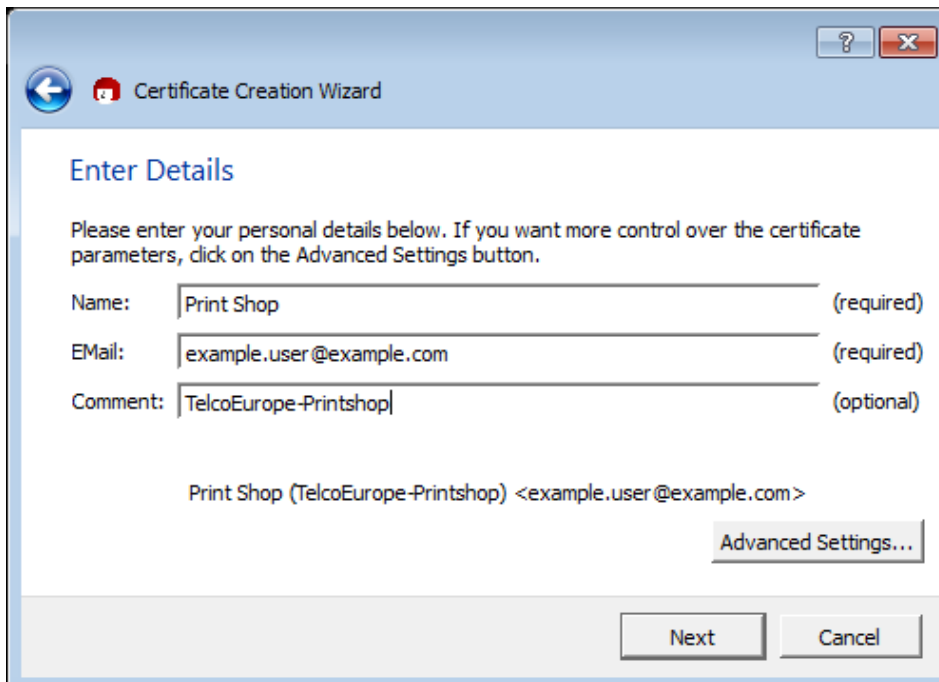
2	Select File, New Certificate
---	------------------------------

Result: The Certificate Creation Wizard opens.

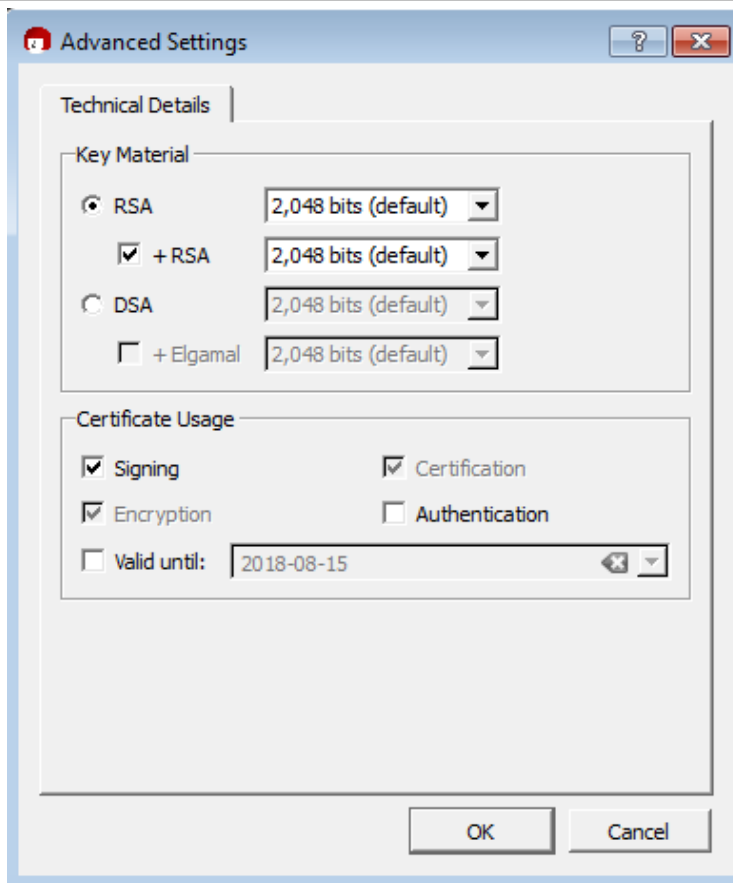


Step	Action
------	--------

3 Select Create a personal OpenPGP key pair.
Result: Enter Details dialog is shown.

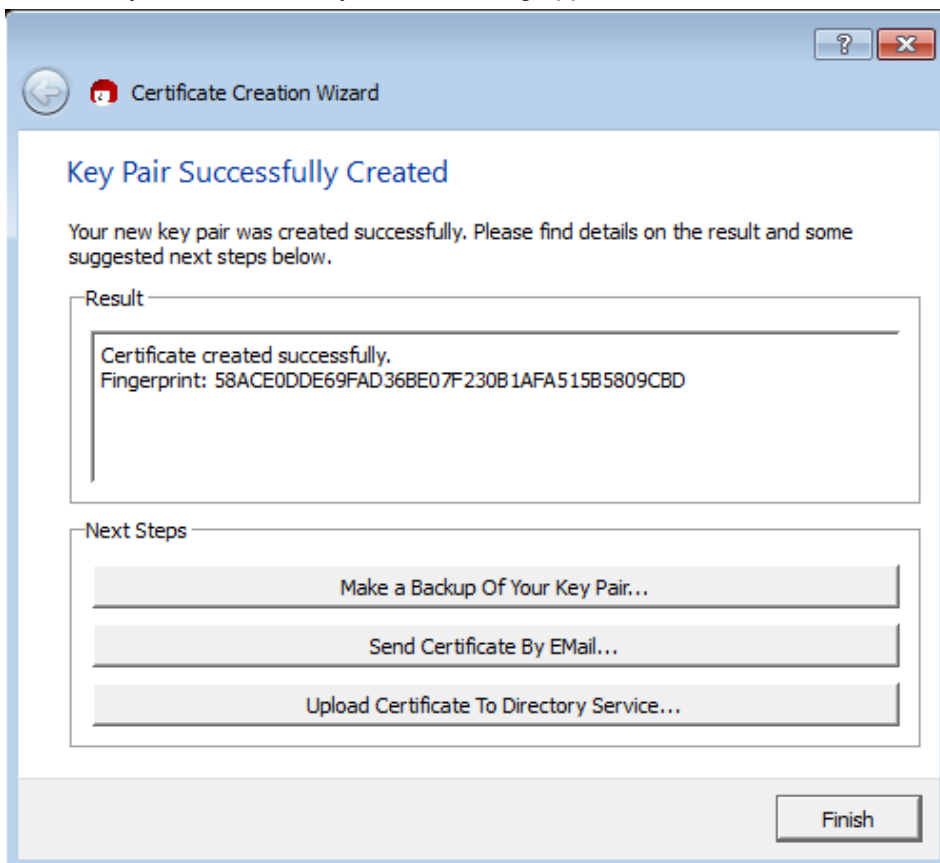


- 4 In the **Name:** field, enter your name.
- 5 In the **Email:** field, enter your email address.
- 6 In the **Comment:** field enter a description of this key. The comment should identify who the printshop is, and may also identify the operator.
- 7 Click Advanced Settings.
Result: The Advanced Settings dialog is shown.

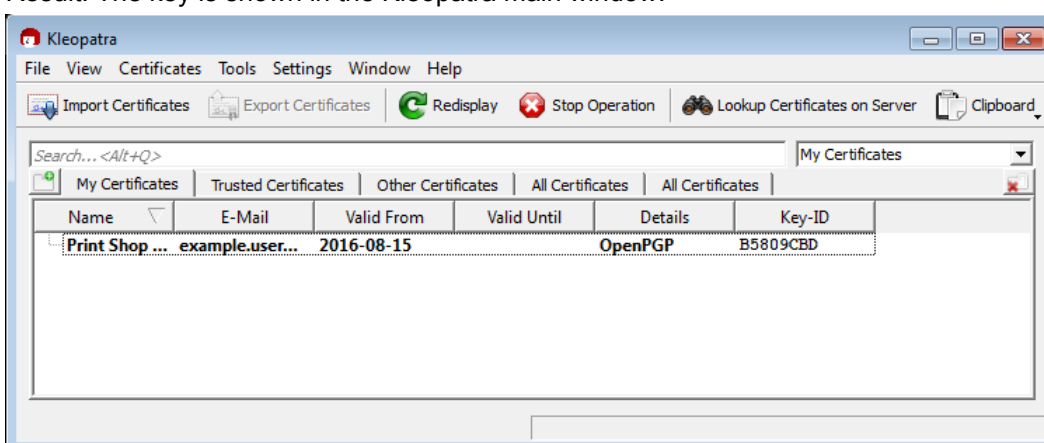
Step Action

- 8 Select the type of key and key size. The values you pick should have been agreed with the operator you are printing for.
- 9 Click **OK** and click Next on the “Enter Details” dialog.
Result: The “Review Certificate Parameters” dialog is shown.

- | Step | Action |
|------|---|
| 11 | Enter a passphrase and press OK, and then repeat for confirmation.
Result: Key Pair Successfully Created dialog appears. |



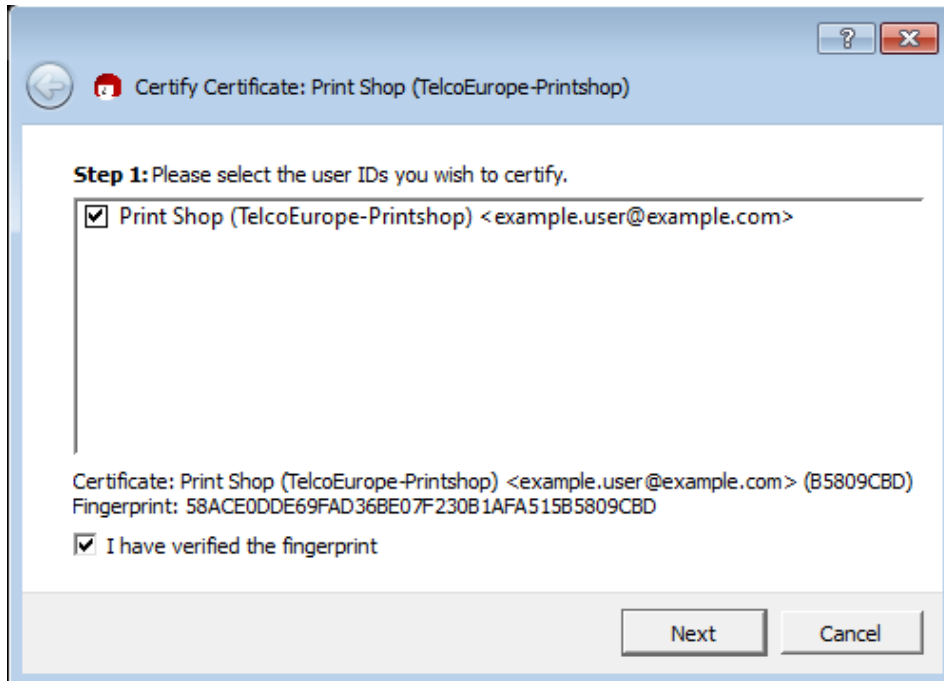
- | | |
|----|---|
| 12 | Click Finish.
Result: The key is shown in the Kleopatra main window. |
|----|---|



Step	Action
------	--------

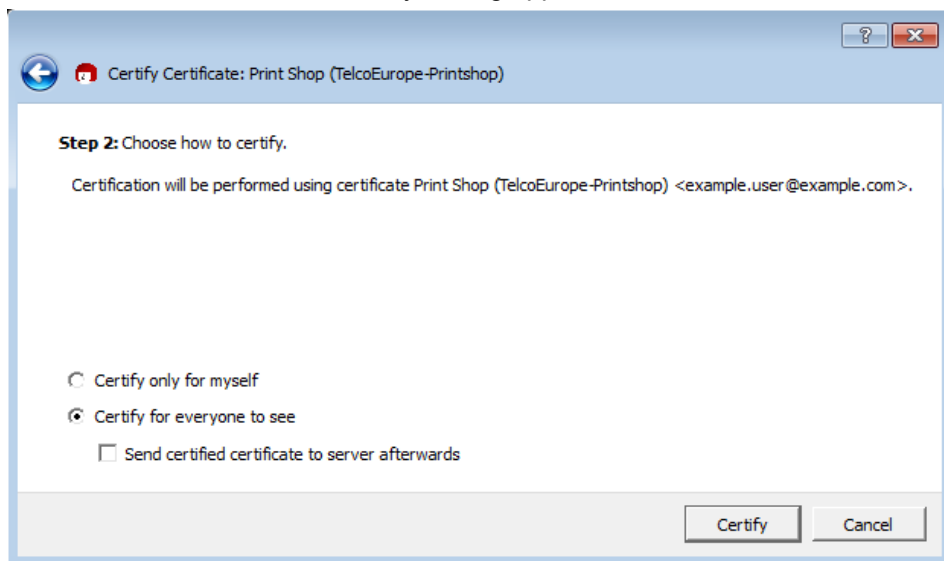
13 The certificate now requires certification. Right-click the key and select “Certify Certificate”

Result: The “Certify Certificate” dialog opens.

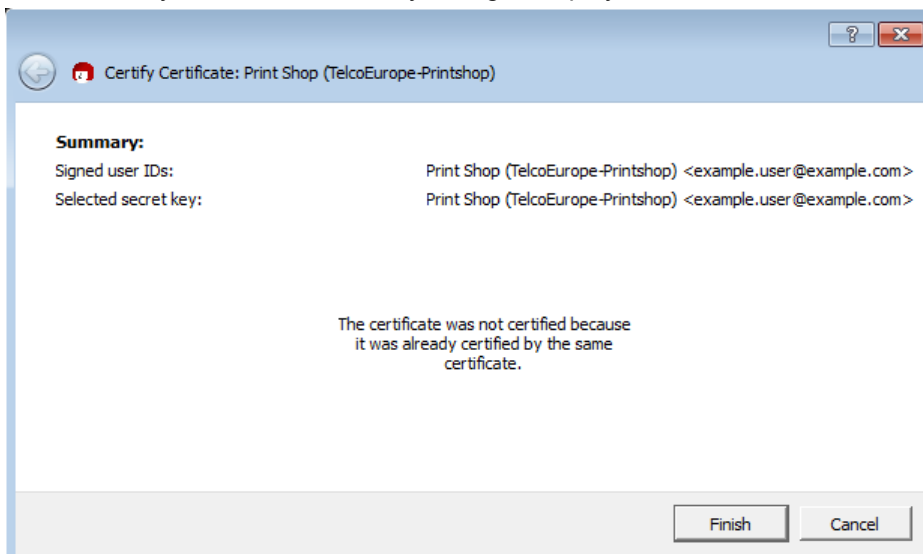


14 Select the certificate and “I have verified the fingerprint”, and click “Next”.

Result: The “Choose how to certify” dialog appears.



- | Step | Action |
|------|---|
| 15 | Select “Certify for everyone to see” and clear “Send certified certificate to server afterwards”, and then click “Certify”.
Result: Certify Certificate Summary dialog is displayed. |

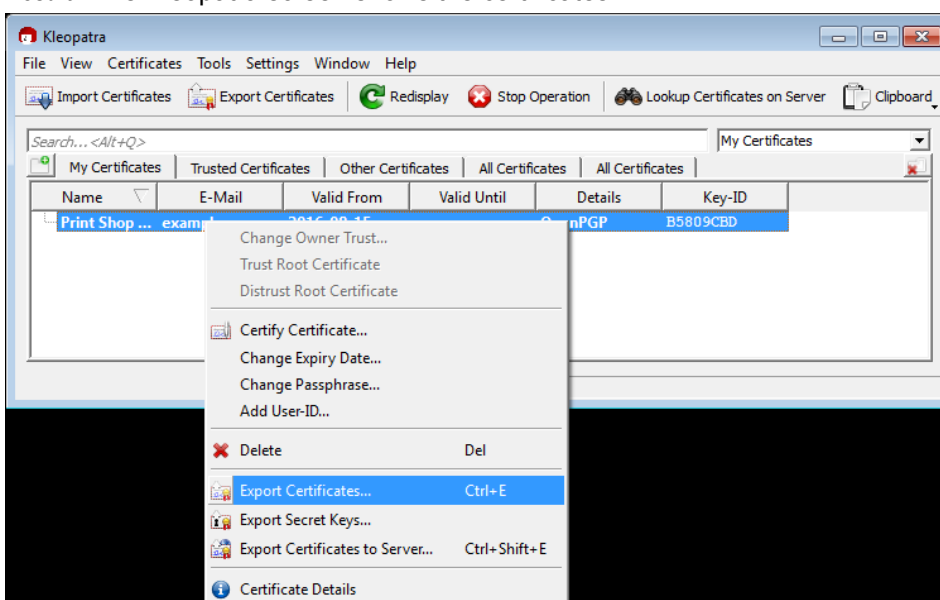


- | | |
|----|--|
| 16 | The “not certified” statement should be ignored and click “Finish”.
Result: The certificate is now certified. |
|----|--|

Exporting keys using Gpg4Win

Follow these steps to export keys which have been generated by gpg.

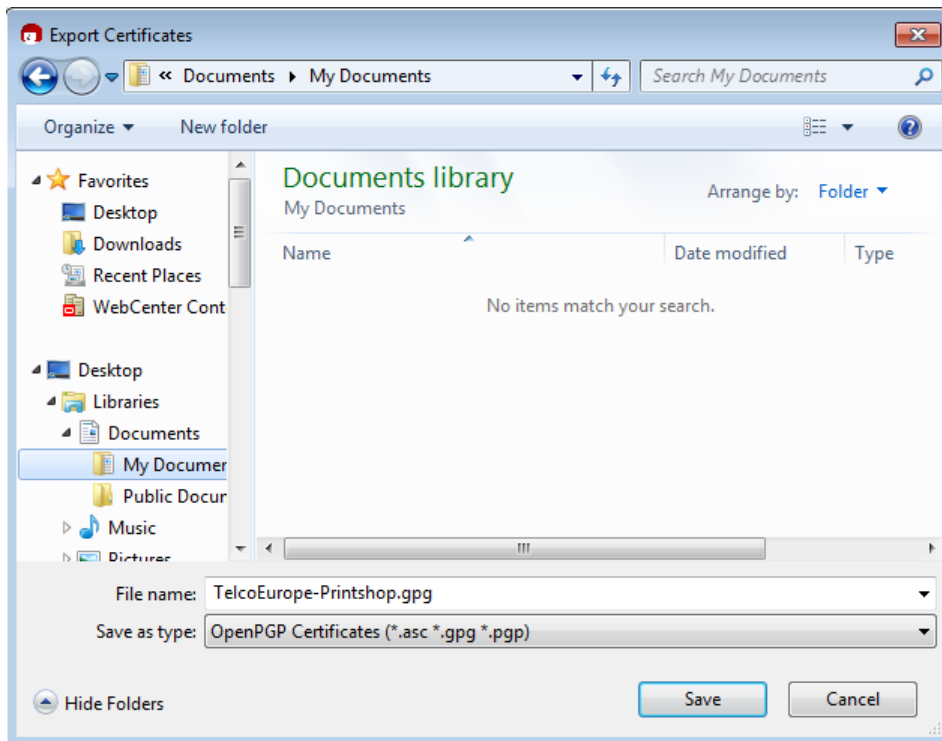
- | Step | Action |
|------|---|
| 1 | On the machine where Gpg4Win generated the keys, start Kleopatra.
Result: The Kleopatra screen shows the certificates. |



- | | |
|---|---|
| 2 | Right click the desired certificate and select “Export Certificates”. |
|---|---|

Step	Action
------	--------

Result: The Export Certificates dialog appears.



- Alter the filename to have a .gpg extension, select the desired directory and click “Save”.
Result: The certificate public key has been exported to the file.
- Send the public key to the operator. The operator will import it into Charging Control Services and will use it when generating voucher and subscriber account (calling card) batches.

Decrypting Files

Introduction

The batch file provided by the operator for printing will have been encrypted using the public key provided by the printshop. This file will need to be decrypted using the matching private key. There are two methods for decrypting the files.

The voucher batch file should be placed on the Printshop target PC. The PGP software on the PC should be used to decrypt the voucher batch file.

Decrypting files using gpg

Follow these steps to decrypt a file using gpg.

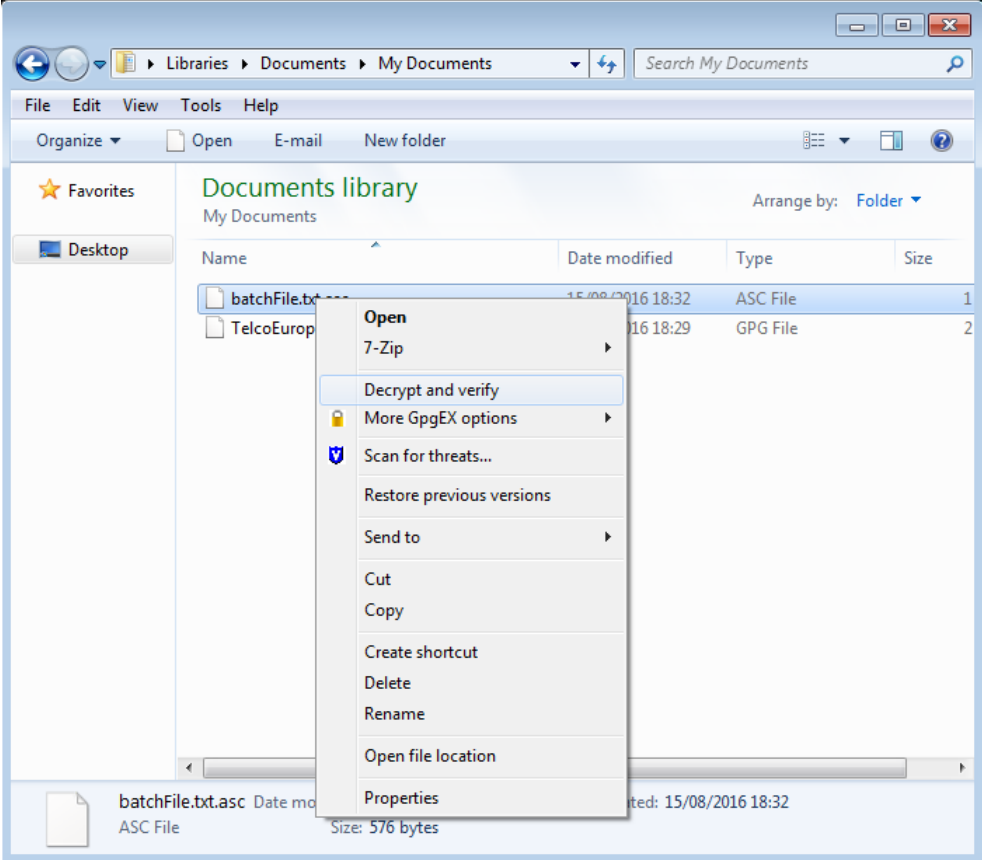
Step	Action
------	--------

- Copy the batch file to the machine where the key was exported from.

Step	Action
2	<p>Use gpg to decrypt the batch file.</p> <p>Example command: <code>gpg -output batchFile.txt --decrypt batchFile.gpg</code></p> <p>Result: Text similar to the following will appear.</p> <pre>You need a passphrase to unlock the secret key for user: "ExampleUser (TelcoEurope-Printshop) <example.user@example.com>" 2048-bit RSA key, ID A7F4D59B, created 2016-08-15 (main key ID C57AAE57) Enter passphrase:</pre>
3	<p>Type the passphrase used when the key was generated and press Enter.</p> <p>Result: gpg will use the passphrase to decrypt the file. Text similar to the following will appear.</p> <pre>gpg: encrypted with 2048-bit RSA key, ID A7F4D59B, created 2016-08-15 user: "ExampleUser (TelcoEurope-Printshop) <example.user@example.com>" The decryption should now be complete.</pre>

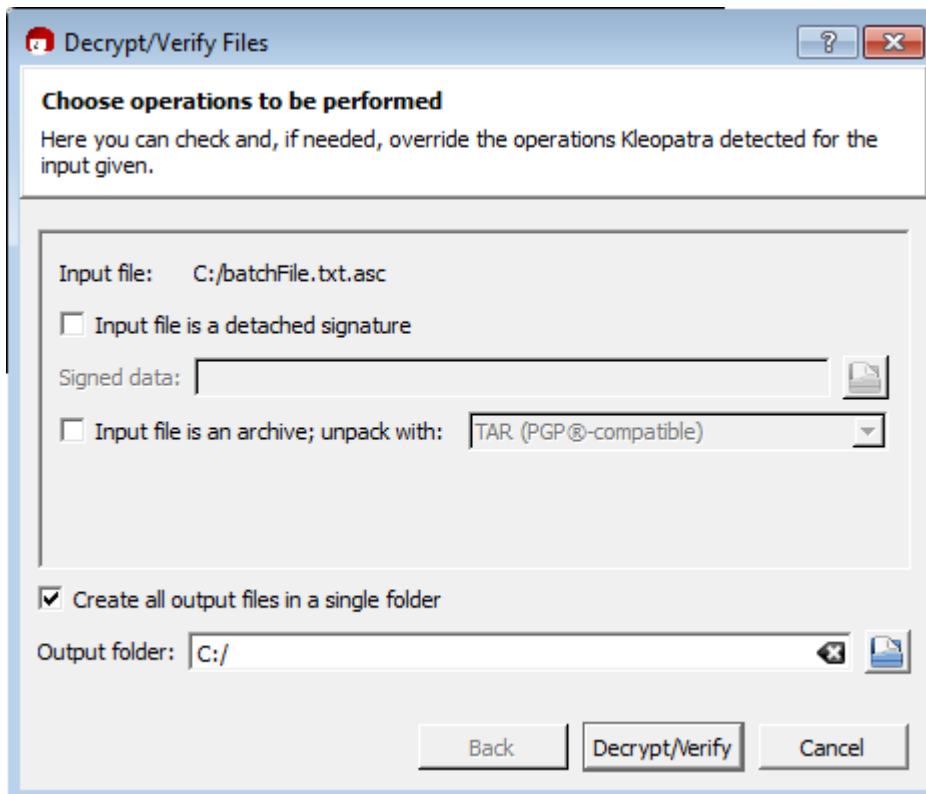
Decrypting files using Gpg4Win

Follow these steps to decrypt an encrypted file using Gpg4Win.

Step	Action
1	Using Windows Explorer, browse to the file you want to decrypt.
2	<p>Select the file and right-mouse-click on the document.</p> <p>Result: The right-mouse-click menu appears.</p>
	
3	Select Decrypt and Verify.

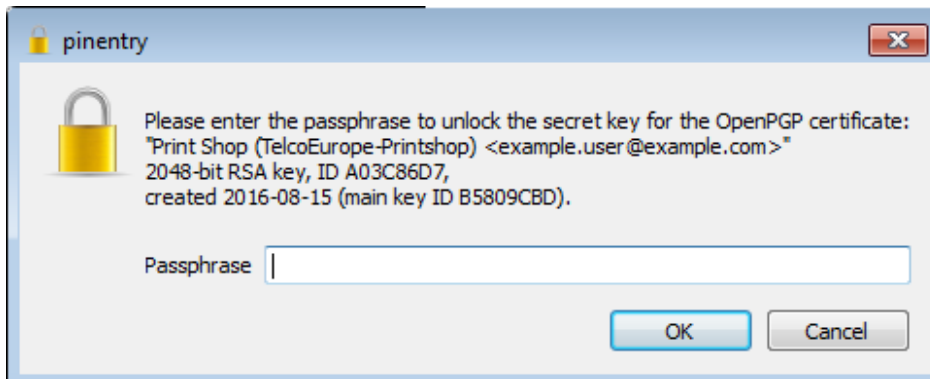
Step	Action
------	--------

Result: The Decrypt/Verify screen opens.



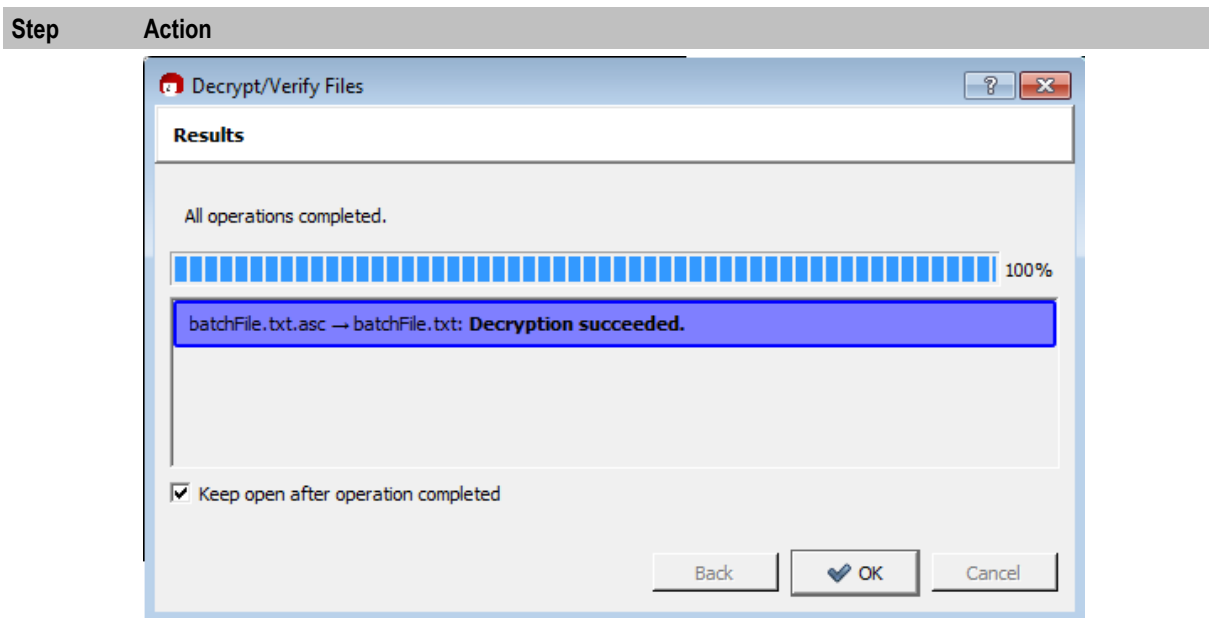
4 Press “Decrypt/Verify”.

Result: The passphrase entry dialog appears.



5 Enter the certificate passphrase and click OK.

Result: The Decrypt/Verify files Results dialog is shown.



- 6 Click OK.
Result: The decrypted file is saved in the same directory as the source file, if decryption was successful.

Exported voucher batch files

Voucher batch file format is controlled by the security library, and the voucher writer plugin used to generate the batch. Which libraries and plugins are used is defined by the Authentication Module (PAM) and the Authentication Rule specified in the New Voucher Batch screen.

Header fields are in the format "<Key field name>=<value>". Key field names always start with an alphabetic character. This makes it easy to distinguish them from voucher records (which always start with a number).

The following header fields are used in the voucher batch file header, (although downstream processors should detect any "<Key field name>=<value>" lines).

Header field	Description
BillingEngineName=<str>	The name of the Voucher and Wallet Server where the voucher resides.
VoucherTypeName=<str>	<p>The name of the voucher type as created on the NCC platform. The voucher type contains the following information:</p> <ul style="list-style-type: none"> • Pre-use expiry period (number of days and hours that this voucher is valid in a pre-use state) • Wallet expiry period (change the current wallet expiry date by this many days and hours) • Voucher number length • Voucher PIN length • A list of all the balance types, associated values and balance expiry date modifications which will be changed/updated when this voucher is redeemed <p>Note: It will be up to the operator to provide the details of the voucher type described here to the print shop so that any specific voucher</p>

Header field	Description
	details can be printed on the final vouchers.
AuthRuleName=<str>	The name of the authentication rule which was used for creating the voucher number and PIN.
AuthModName=<str>	The name of the pluggable authentication module (PAM) (NCC specific) used for creating the voucher PIN.
VoucherBatchBatch=<str>	A two character identifier (non unique) for this voucher batch.
VoucherBatchID=<int>	The system generated ID for this voucher batch.
OriginalCount=<int>	The number of vouchers created in this batch.
StartOfRange=<int>	Beginning of the range of voucher numbers.
EndOfRange=<int>	End of the range of voucher numbers.

A line consisting of a single equal sign (=) terminates the header lines. All subsequent lines are voucher detail records.

CCS3 DES voucher batch example

This text shows an example export voucher batch file generated by `ccsVoucher_CCS3` using the DES encryption library (and a bespoke voucher file writer plugin to format the non-header details), but no GnuPG key.

```
#
# Voucher file for batch 83
# Generated by ccsVoucher at Tue Nov 11 12:55:27 2008
# (key=value or
# voucherserialnumber,vouchernumber,vouchersecret,vouchercontext,voucherprivate_secret
# )
#
BillingEngineName=PCDEV
VoucherTypeName=DES
AuthRuleName= DES (VL=10 VP=4)
AuthModName=DES
VoucherBatchBatch=
VoucherBatchID=83
OriginalCount=2
StartOfRange=1000000001
EndOfRange=1000000002
=
#
# Voucher records start
#
1000000001,8986
1000000002,4887
#
# End of voucher records
#
```

CCS3 CB10 voucher batch example

This text shows an example export voucher batch file generated by `ccsVoucher_CCS3` using the 'CB10 HRN' encryption library using the 'HRNGEN' encryption algorithm, but no GnuPG key.

```
#
# Voucher file for batch 85
# Generated by ccsVoucher at Tue Nov 11 12:55:27 2008
# (key=value or voucherbatch,preuseexpiry,hrn,serialnumber)
#
BillingEngineName=PCDEV
VoucherTypeName=CB10
```

```

AuthRuleName=CB10 (S=14 R1=2 R2=2 R3=0)
AuthModName=CB10 HRN
VoucherBatchBatch=
VoucherBatchID=85
OriginalCount=2
StartOfRange=00000000000001
EndOfRange=00000000000002
=
#
# Voucher records start
#
85,20090101000000,631599527570333589,1000000138
85,20090101000000,855619036698319621,1000000139
#
# End of voucher records
#

```

CCS3 CB10 GPG voucher batch example

This text shows an example export voucher batch file generated by `ccsVoucher_CCS3` using the 'CB10 HRN' encryption library using the 'HRNGEN' encryption algorithm, and GnuPG encryption.

Note: This file has been decrypted using the gpg key.

```

#
# Voucher file for batch 86
# Generated by ccsVoucher at Tue Nov 11 12:55:27 2008
# (key=value or voucherserialnumber,hrnserialnumberseed,hrn,nrnlength,hrnc)
#
BillingEngineName=PCDEV
VoucherTypeName=CB10 HRN
AuthRuleName= CB10 (S=14 R1=2 R2=2 R3=0)
AuthModuleName=CB10 HRN
VoucherBatchBatch=
VoucherBatchID=86
OriginalCount=2
StartOfRange=00000000000003
EndOfRange=00000000000004
=
#
# Voucher records start
#
86,20090101000000,057195727842702414,1000000138
86,20090101000000,363323157948027866,1000000139
#
# End of voucher records
#

```

Exported card/account batch files

Subscriber account/calling card batch file format is controlled by the account writer plug-in used to generate the batch. Which libraries are used is defined by the authentication name specified in the New Subscriber Batch screen.

Header fields are in the format "*Key_field_name=value*". Key field names always start with an alphabetic character. This makes it easy to distinguish them from voucher records (which always start with a number).

The following header fields are used in the voucher batch file header, (although downstream processors should detect any "*Key_field_name=value*" lines).

Header field	Description
AccountBatchID= <i>int</i>	The ID of the subscriber account batch.
ServiceProviderID= <i>int</i>	The ID number of the service provider the subscriber batch belongs to. When ccsAccount is started by the screens the value of this field is populated by the id of the service provider which is selected in the Service Provider field of the Subscriber Management screen when the New button is clicked.
AccountTypeID= <i>int</i>	The product type the subscriber batch has. When ccsAccount is started by the screens the value of this field is populated by the Product Type field on the New Subscriber Batch screen.
maxConcurrent= <i>int</i>	The maximum number of concurrent connections wallets generated with this subscriber batch can have. When ccsAccount is started by the screens the value of this field is populated by the Maximum Concurrent Accesses field on the New Subscriber Batch screen.
BatchSize= <i>int</i>	The number of subscriber accounts in this batch. When ccsAccount is started by the screens the value of this field is populated by the Batch Size field on the New Subscriber Batch screen.
RangeStart= <i>int</i>	Beginning of the range of subscriber account numbers. When ccsAccount is started by the screens the value of this field is populated by the Card Number Start Range field on the New Subscriber Batch screen.
RangeEnd= <i>int</i>	End of the range of subscriber account numbers. When ccsAccount is started by the screens the value of this field is populated by the Card Number End Range field on the New Subscriber Batch screen.
AuthenticationModuleID= <i>int</i>	The ID of the authentication module used for: <ul style="list-style-type: none"> • Encryption and/or random generation of PINs for this batch • (optionally) sends the output file for encryption by gpg. When ccsAccount is started by the screens the value of this field is populated by the PAM Name field on the New Subscriber Batch screen.
BillingEngineID= <i>int</i>	The ID number of the Voucher and Wallet Servers .
CurrencyID= <i>int</i>	The ID of the currency the wallets generated with this subscriber batch will use. When ccsAccount is started by the screens the value of this field is populated by the Wallet Currency field on the New Subscriber Batch screen.
LimitType= <i>str</i>	The type of limit the wallets generated with this subscriber batch will use.
BalanceType= <i>int</i>	The balance type ID of the balance type this wallet will have any initial value stored in.

A line consisting of a single equal sign (=) terminates the header lines. All subsequent lines are voucher detail records.

Card/account output file

This text shows an example export subscriber account/calling card output file.

```
# Account Batch Output File
# Generated Wed Dec 31 01:24:29 2008
```



```
#
AccountBatchID=59
ServiceProviderID=1
AccountTypeID=7
maxConcurrent=1
BatchSize=10
RangeStart=8815000000
RangeEnd=8819990000
AuthenticationModuleID=4
BillingEngineID=2
CurrencyID=2
LimitType=DEBT
BalanceType=1
=
Dec 31 01:24:29.861203 ccsAccount(15179) NOTICE: Beginning account generation.
16309877,3415992,7,G8.H3zCjoKzbY,8800127
19052821,0363266,7,G8fRbQy015unk,8800128
18627603,5447142,7,G82efn9Gh2gSY,8800129
16635167,9003194,7,G8nkF67MOzS9g,8800130
19498256,8441931,7,G8tfZtbQvbOIg,8800131
18758105,8744644,7,G8CSYLULMZtw,8800132
17349265,3517347,7,G8GH/BM14HHzs,8800133
16223817,0064708,7,G8MbgIe4gPO.U,8800134
16089674,7771756,7,G81Xd7ySSzsVw,8800135
16405822,1207166,7,G8JugOSguxjqg,8800136
Dec 31 01:24:35.514685 ccsAccount(15179) NOTICE: Progress 10/10 (100.0%) Complete
Dec 31 01:24:35.515578 ccsAccount(15179) NOTICE: Account generation complete.
```


Glossary of Terms

CCS

- 1) Charging Control Services component.
- 2) Common Channel Signalling. A signalling system used in telephone networks that separates signalling information from user data.

HRN

Hidden Reference Number or Human Readable Number

PC

Point Code. The Point Code is the address of a switching point.

PIN

Personal Identification Number

Service Provider

See Telco.

Telco

Telecommunications Provider. This is the company that provides the telephone service to customers.

Telecommunications Provider

See Telco.

Index

A

About This Document • v
Audience • v

C

Card/account output file • 18
CCS • 21
CCS3 CB10 GPG voucher batch example • 17
CCS3 CB10 voucher batch example • 16
CCS3 DES voucher batch example • 16
Charging Control Services files and encryption •
1
Copyright • ii

D

Decrypting Files • 1, 12
Decrypting files using gpg • 12
Decrypting files using Gpg4Win • 13
Document Conventions • vi

E

Exported card/account batch files • 17
Exported voucher batch files • 15
Exporting keys using gpg • 4
Exporting keys using Gpg4Win • 11

G

Generating GPG keys • 2
Generating keys using gpg • 2
Generating keys using Gpg4Win • 5

H

HRN • 21

I

Introduction • 1, 12

M

Managing Public/Private Key Pairs • 1, 2

O

Operator • vi
Overview • 1

P

PC • 21
PIN • 21
Public and private key encryption • 1

R

Recommended software • 2
Related Documents • v

S

Scope • v
Service Provider • 21
System Overview • 1

T

Telco • 21
Telecommunications Provider • 21
Terminology • vi
Typographical Conventions • vi