Danske Bank

OpenPGP Security Implementation Guide

version 1.10

Contents

General notes and introduction

OpenPGP is offered by Danske Bank and can be used to secure the content of files sent to the bank and received from the bank via direct communication channels. OpenPGP uses public and private key encryption where the public keys are shared between the customer and the bank. This means that a customer must send copies of their public PGP keys to the bank and download copies of Danske Bank's public keys from our website www.danskebank.dk/is.

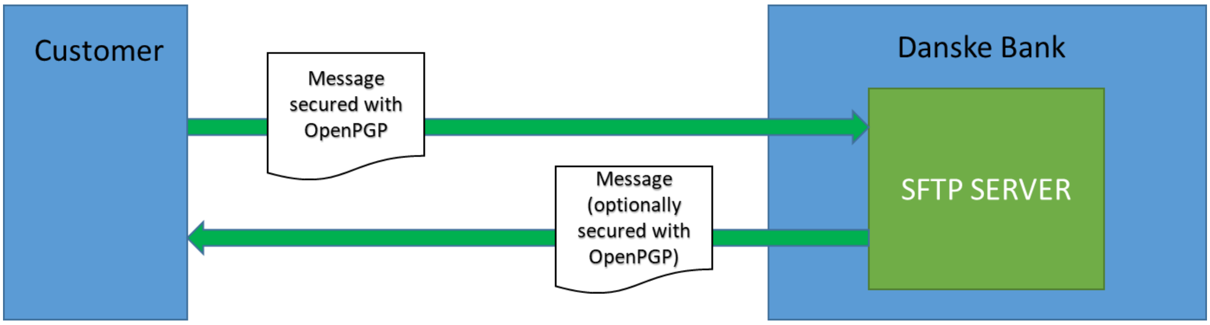
The main target group of the solution are corporate file customers who send files to the bank (i.e. payments or collections) and receive files from the bank (i.e. account statements or status files).

The OpenPGP solution works with Danske Bank's file communication channels:

- File communication via SFTP
- File communication via FTP/VPN

For file customers who use OpenPGP it is mandatory to sign and encrypt files sent to Danske Bank whereas it is optional to get files secured when they are received from the bank.

Example: Simplified file integration between a corporate file customer and Danske Bank, where the customer uses SFTP file communication and OpenPGP.


Customer
Message secured with OpenPGP
Danske Bank
SFTP SERVER
Message (optionally secured with OpenPGP)

Objectives of this guide

The objectives of this guide are to provide a step-by-step integration guide to our customers using OpenPGP security on file transfers and to provide a comprehensive understanding of the OpenPGP services for file transfers that is supported in Danske Bank.

Target group

The target group of the guide is customer's technical IT personnel who will do the technical integration of their file transfer system into Danske Bank's system - securing the file transfers with OpenPGP.

How to read the guide

The guide has a main chapter covering overall integration to the various services that Danske Bank offers in connection to OpenPGP for file transfers. This is the part that you as IT technician should use as your main guide for integrating your OpenPGP software with Danske Bank's.

The services are:

- Customer onboarding
- PGP certificate renewal
- Sign and encrypt a file to the bank
- Decrypt and verify a file from the bank
- PGP certificate revocation

Refer to appendices for detailed information.

What the guide covers

The guide covers only relevant aspects of the integration - this means it does not take into consideration which OS or platform a customer runs its OpenPGP software on. Danske Bank has tested the OpenPGP Security solution on Windows 10 and Windows 11.

The guide includes screen dumps created using GnuPG software (GnuPG is a complete and free implementation of the OpenPGP standard). The screen dumps show certificate handling and signing/encrypting and decrypting/verifying when exchanging files with Danske Bank.

In addition to screen dumps and examples using GnuPG, there are also code examples included that can be helpful when setting everything up. Note that the code examples are included as a help for anyone not familiar with how to perform operations from inside GnuPG software.

Disclaimer: Third-party software

Throughout this document, Danske Bank quotes various third-party software. However, Danske Bank does not recommend any specific third-party software over another and makes no representation, explicit or implied, as to the functionality, quality, or suitability of any third-party software referenced herein.

Before downloading, installing, or using any third-party software, your organization must make an independent assessment of the suitability of such software.

Danske Bank also discusses the use of third-party software in this document. Please refer to instructions from the provider of such third-party software prior to use. The use and functionality of third-party software is not controlled by Danske Bank and is subject to change without notice. You should not rely on the information provided herein regarding third-party software for any reason.

Customer’s technical setup

The customer must provide and use their own IT infrastructure to run OpenPGP routines. Danske Bank is not responsible for the customer’s IT infrastructure installations and does not support any type of software on the customer’s IT platforms (including OpenPGP software).

Please see appendices:

- A: Technical requirements
- D: Functionality not supported

Set up OpenPGP

To begin using OpenPGP, please follow the steps outlined below:

- Receive an onboarding PIN and public PGP keys from Danske Bank
- Create private/public PGP certificates
- Share your public keys with Danske Bank
- Renew your certificates and provide the bank with updated keys

Receive a PIN and public PGP keys from Danske Bank

To get started, you need to contact Danske Bank Integration Services (INTS) at ints@danskebank.com to request a one-time onboarding PIN (8 characters) and download Danske Bank's public PGP keys from Danske Bank OpenPGP Security on www.danskebank.dk/is. These keys will enable you to encrypt files sent to the bank and verify the signature on files received from the bank.

The onboarding PIN will be provided electronically. However, if you prefer, you may request to receive it via a physical letter. Please note that delivery by post may take approximately 5-10 business days.

The registered contact person will receive an email containing a link and an SMS message with a PIN to enter in the link (this is not the PGP PIN yet). After entering the PIN, the 8-character onboarding PIN code will be displayed.

To receive the PIN electronically, please ensure that:

1. You or a colleague has access to District.
2. The contact information (email and phone number) in District is accurate and up to date.

To verify or update your contact information in District, log in and navigate to Profile and Settings -> Profile.

Exchange public keys with the bank

When you have received the PIN and the public PGP keys from the bank you can start the actual onboarding to the OpenPGP solution. Remember you need to have a running file communication channel with the bank. For example, a SFTP channel.

The first action that you need to do is to send your public PGP keys to the bank. Follow this process:

Step	Action
1	Create your own OpenPGP keys (private and public) with your PGP software/hardware. See detailed key requirements in the technical requirements section in this document. Appendix A: Technical requirements.
2	Make a copy of your own public key and add this information to the file **before the PGP public key block**, see example below: • PIN; (8 characters) • Technical user ID; (Provided by the Bank. A 'Technical user' is a user under your District agreement that is used to set up integrations, for example, SFTP. It is not a personal user) Example ABCDEFGH; 123456; -----BEGIN PGP PUBLIC KEY BLOCK----- Comment: User-ID: Company XYZ Comment: Created: 14/04/2023 14:12 Comment: Expires: 14/04/2025 12:00 Comment: Type: 4096-bit RSA (secret key available) Comment: Usage: Signing, Encryption, Certifying User-IDs Comment: Fingerprint: CE71E9013E23E6D5E36CBE61BC37D78973A7B9A0 mQINBGTaGiEBEADN7XMvtp695oMQPPHgI/KsjwzXfx2sIb7csWTTQP3FTZq6PmqK t4HSXezBi0ZwaHWUmrlCSsogpCv7yFpMRqeFsGQVUObPfGfqAYvSjlUXuORJBvYe xp+jH+i1sLH3ig0k0VNTllrUTwo3z838OiKIsaV4CPs0E+DKWaegmVY19T7DvSf Ags9xX7DwmgpQ4ij5XADvhqMrAXtNhMvfCQsqOzkS4cDcSWdWxV6zI+KTxH5i4kp OIiXMKd6tRwvZ3r/WL4xoyQGcErNuMj+WDdNRiOSs0GveNwvQXXYvqwdqmtqZkz5 zIecgjNZ9K5ZhXHHrhESD1judNId6LU0ITwA+WbeYoFH7L6oOHxDxbgPVQZBkFjJ…
3	Encrypt the file with the bank's encryption key (do not sign it)
4	Add a comment in the PGP encrypted file stating who you are (technical user ID) and what your action is (OnboardKeys [case insensitive]): Comment: <UserId>123456</UserId> Comment: <Action>OnboardKeys</Action> Example of an encrypted PGP message that contains the encrypted file from step 2: -----BEGIN PGP MESSAGE----- Comment: <UserID>123456</UserID> Comment: <Action>OnboardKeys</Action> AdXPl8uyuNrUaR2OKNcNfZ31NLi7TZ5rmk8OqC8RvCQDdENNOGaRLPYU8jVfroO8 9AKBvV7N+bt8q9/VrPd+VbwAEUed5doVBZYJk77YE/QBX7BhAVRmy7D53TeHR0i6 m/0snnFlYLIhEkKd2hWHiX0FL2l2p3I9/IpHxnYV0VcTYF5YlroGdNasvDJfp+ah gYy6Aym8PzE9u3ggImmZXK4W58pFRj3ATr1qI7rmvDneUvFNZDWxKcyDjX4duHrw

	eOMc3WpxFu6HB2pEtY0LMwODiYWuDFeMnt1ya82CTKk382QZw/WcaqpWvqZVFGFT R43HAy8CPXQsqDwoHHlnvX8F45F6OYlN/VJYYTfD5i3uKCtgSXEjpNQX21yu+bYr…
5	Send your public keys to Danske Bank via an already established file communication channel (FTP/SFTP)
6	You will receive a PGP notification from the bank with a status. See the chapter “PGP notifications from the bank” for more info on PGP notifications.
7	**Attention: We require you to renew your keys and send them to the bank within three weeks. If your keys are not renewed OpenPGP security will not work, and you will not be able to send encrypted files to the bank.** See the chapter ”Renew a certificate”.

Renew a certificate

As the certificate renewal is an integral part of OpenPGP, and since you will need to do a renewal at least every two years, we require that you already during the initial implementation go through with a renewal to ensure you are technically ready for future renewals. **If your keys are not renewed before the expiry date OpenPGP security will not work, and you will not be able to send encrypted files to the bank. You will have to onboard the solution again to make it work.**

Step	Action
1	Create your own new OpenPGP keys (private and public) with your PGP software/hardware. See detailed key requirements in the technical requirements section in this document. Appendix A: Technical requirements.
2	Make a copy of your own public key: Example: -----BEGIN PGP PUBLIC KEY BLOCK----- Comment: User-ID: Company ABC Comment: Created: 14-04-2023 13:30 Comment: Expires: 14-04-2025 12:00 Comment: Type: 4096-bit RSA (secret key available) Comment: Usage: Signing, Encryption, Certifying User-IDs Comment: Fingerprint: 719334DFFCIBF8546D3A46EA4E8445a31EDF90AE mQINBGUWvV4BEADJD2l3ts6odJjWNnb9ly0333KvIGO189ClhX1MPwPsfhCAFYaB gG0pLAXUU5uZZXh8E3wIrCBDKCUOkxg3/qr1Dfn60XOAPyAzVVhRgchGLrdlpJnM HAxtBr+Y4JaDU2l30AdJdN3unRhpimU0vQsItfpuGiJcIXQ1NDEw4pl66P7v3OSR gd0XIHpD/n/7fe1wS+JFsRMzcqiI50KUnz9jWm8D2vSV6AILxC8/sb0hN0t8ZBMy…
3	Sign and encrypt the file using your existing, onboarded PGP key for the signature and encrypt the file with the bank’s encryption key. This must be performed in one single action.
4	Add a comment in the PGP encrypted file stating what your action is (RenewKeys [case insensitive]): Comment: <Action>RenewKeys</Action> Example: -----BEGIN PGP MESSAGE----- Comment: <Action>RenewKeys</Action>

	rCZD34wuDdBmwfSw5Gy7hhfwW/tYkO8wy7rz3eHHjcYFc9x6N+h7yCqYvIT+kda4 xrG9rm7DNbfiYGreZkV0831kgODCxlCMVhAVyqxW8Orj+noVL/GCa0nVFrXgaAJo qsKw1TvMj/5iNCteo3+q5o3CYWus/A00JmC+zH3F+Cd7mMD2pww9VFSL2fXhVL6j ptbRf48G6IP5N2L22lyqvBpxfdEqariyjoEL+II7N14UvrIqqo7cdDFyrPJGyNLO… Note that if you are not using Danske Bank's SFTP to renew the certificate, you must also insert the following comment in the file, where 123456 represents your six character technical user ID: Comment: <UserID>123456</UserID>
5	Send your public keys to Danske Bank on the already established file communication channel.
6	You will receive a PGP notification from the bank with a status. See the chapter "PGP notifications from the bank" for more info on PGP notifications.

Send an encrypted file to the bank

After you have onboarded the OpenPGP solution and renewed your keys, you can begin to use OpenPGP on your basic day to day file communication with the bank by sending files to the bank.

We hold a copy of your public keys which enables us to both verify your PGP signature on incoming files and allows us to PGP encrypt files we send to you.

Step	Action
1	Sign and encrypt the file you want to send to the bank. This could be a payment file or a collection file. Use your own PGP key to sign the file and encrypt the file with the bank's encryption key. Example: -----BEGIN PGP MESSAGE----- hQIMA3b0kUou8gy2Ag/+NG3pBGmSJfaMAi42eFFjOcnCBMQR1q806OUvPvqEoBpx enAT6TbsnsX32jvUP5PONlsLZLUVF2kzjr4XNm6eHmWa59iKHMAnA/nEiaWyaEsZ DKpQJ5zuIWo8TuZrJaZl/M62R327bK7Fc5qcktgObh/7sZwYr+hibdsdKIz7lzv1 3eBLqZ1IKK6AmE0D29yNKy8MmaYRCngh57O3mEmlm8YesSSqMnzcHPonvhuyB/8P 1+Bv8C2MQuPUAhvs+/tBICuRAK0HZ1sMqLyO/7yH60623E4TSrIWwez56jVgG3Vn 1OCEr9/g07Nx6N6n96IBXguDGkaaaFOVeO9C+s5ax7H1egwkSwyvPXNk0g/sij3G… Note that if you are not using Danske Bank's SFTP to revoke the certificate, you must also insert the following comment in the file, where 123456 represents you six character technical user ID: Comment:<UserID>123456</UserID>
2	Send the file to Danske Bank via an already established file communication channel.
3	You will receive a PGP notification from the bank with a status if the bank *could not* decrypt the file. See the chapter "PGP notifications from the bank" for more information on PGP notifications. This must be performed in one single action.

Optional: Receive an encrypted file from the bank

When you have onboarded OpenPGP in Danske Bank we hold a copy of your public keys which enables us to both verify your PGP signature on incoming files and allows us to PGP encrypt the files we send to you.

While PGP encryption is mandatory on files you send to Danske Bank, it is optional for you if you want Danske Bank to encrypt the files we send to you.

Encryption of files sent to you is not supported for Service bureaus.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Danske Bank will sign and encrypt the file before we send it to you. This could be basically any kind of file you receive from the bank via EDI communication. For example, a payment status file, an account statement, or an OCR file.

Danske Bank will sign the file with Danske Banks sign key and encrypt it with your public PGP encryption key.

Example:<pre>
-----BEGIN PGP MESSAGE-----
Version: BouncyCastle.NET Cryptography (net6.0) v2.0.0.2

hQIMA1ZwZ696UhrjARAAmiJzx5zm2W61/GrVmOShbB7V+TclmawDfTYId24lXUh1
EabR218PCR17404Od1GLg29IXuxucjxOnlML/0FlvwEIEBteSXuHPtEcpBBbr8qz
929f0WyIA/2Zx4Gr62Qw53dcrABO4TaP7xeD5VdisbRQZLibooEVrEeDka2eHBPw
d1sPoVDcoAbDrS112trX9kTi8Q6+PJIl5eudJk7BOaMwbmlcnL0PBfohGpE0ezq9
1oIvNwhXv7Uhyy3n6y8Ht7K7qKS7i37R2Hb8XMX7cDYs8xJTOyF78roDplUiNDNf…
</pre></td></tr>
<tr><td>2</td><td>Decrypt the file and verify the signature. Then you have the clear text file from the bank.

Example:<pre>
VKSFAST 23091316452309130001202309131617EUR+   746.0900000+   0.0000000+   0.0000000
VKSFAST 23091316452309130001202309131617USD+   695.1400000+   0.0000000+   0.0000000
VKSFAST 23091316452309130001202309131617GBP+   866.6400000+   0.0000000+   0.0000000
VKSFAST 23091316452309130001202309131617SEK+   62.4500000+   0.0000000+   0.0000000
VKSFAST 23091316452309130001202309131617NOK+   65.0000000+   0.0000000+   0.0000000
VKSFAST 23091316452309130001202309131617CHF+   778.0700000+   0.0000000+   0.0000000
</pre></td></tr>
</table>

Optional: Compression of file data

When you send files to the bank compression is optional. Sending files to you we use ZIP compression.

Revoke a certificate

If you need to revoke your certificate, it can be done in two ways:

1. Contact Danske Bank Integration Services during opening hours (see the chapter Contact information), or
2. Send an electronic OpenPGP revoke message to the bank. If successful, the certificate will be revoked immediately.

By having the certificate revoked in Danske Bank means that both sub keys will be revoked and we will reject PGP encrypted messages where the certificate is used.

Step	Action
1	Create a clear text file with the 40 characters fingerprint of the certificate you want to revoke, no spaces in fingerprint. Example: `89ED0D4C8EA06D301A35C272048D18710D4EED4D`
2	Sign and encrypt the file. Use your own PGP key to sign the file and encrypt the file with the bank's encryption key. Important: The file must be signed with the same key that you want to revoke.

3	Add a comment in the PGP encrypted file stating what your action is (RevokeKeys [key insensitive]):
	
	Comment: <Action>RevokeKeys</Action>
	
	Example:
	
	`-----BEGIN PGP MESSAGE-----`
	`Comment: <Action>RevokeKeys</Action>`
	
	`mS2Yj8ru0IMupX6iOoCRLTZ6Ax7WxkMAGgLExN0j6kzDoBv3RSDwMOv5lKwuuiqV`
	`ShBROUI3lmLZFnTeCLEji8oMvRhn2juGPLICCAiJEp1M9EMx6nOB40AthGGGXpg8`
	`Rlgb921Hf2oVkyADvwZ6+y/ErBNvhkxOzzvfOTthy6g6ld5vRNWmbbCPiPS98bZ`
	
	`NJaEtyYS4LGo/p1hxAAqDqN/NlGw5s6VtISVUnggc5Tv4KW8Yr2fZHcyNot1U1dF`
	`DV1ZcqaRn0euyuToWwHQKpdQx5bqOXvteNfVZPV86d3RVew08QcmKHd8K26dybOJ`
	`GUN0/ApOdBt5M/gWBruDXDnzq5aTja5UXXxOcuVgc0VmSChB5DF1xndRw43+mqu7…`
	
	Note that if you are not using Danske Bank's SFTP to revoke the certificate, you must also insert the following comment in the file, where 123456 represents your six character technical user ID:
	`Comment: <UserID>123456</UserID>`

4	Send your revoke request to Danske Bank via an already established file communication channel.
5	You will receive a PGP notification from the bank with a status. See the chapter "PGP notifications from the bank" for more info on PGP notifications.

How to apply a new bank certificate

Danske Bank public OpenPGP certificate expires every two years.

The new public certificate will be available on our homepage approximately 2-3 months before the expiration date of the current certificate: https://danskeci.com/ci/transaction-banking/instructions/integration-services/channels-and-security…

After downloading the new certificate:

1) Import the new certificate into your system.
2) Update your code as necessary to accommodate the new certificate.

From this point forward, you must encrypt your files to the bank using the new public key.

If you receive signed and encrypted files from the bank, we will automatically begin using the new key to sign the files once you start using the new key to encrypt files sent to the bank.

Test scenarios

If you would like to test your OpenPGP setup, Danske Bank supports the following scenarios:

- Exchange public keys with the bank
- Test mark a file to the bank

Scenario: Exchange public keys with the bank

Before sending your public keys to the bank it is possible to do a test onboarding, allowing you to verify that your public keys are valid and that we are able to read and process your onboarding request.

You do this by replacing the action OnboardKeys in the PGP encrypted file with the action OnboardKeysTest. Example:

```
Comment: <UserId>123456</UserId>
Comment: <Action>OnboardKeysTest</Action>
```

You do this in combination with the following eight-character PIN code, which can be used as many times as needed for test purposes: **ABCDEFGH**

You will receive notifications from the bank informing you of the result of the test, please see the chapter with chapter PGP notifications from the bank. Everything else remains as in the chapter Exchange public keys with the bank. **Important:** Note that after you have done a test onboarding, you need to contact the bank to request a new, valid PIN code and then do an actual onboarding before you can start using OpenPGP.

Scenario: Test mark a file to the bank

It is possible to test mark an OpenPGP secured file and send it to the bank via an established communication channel. The payment will appear in District but will not be executed. Test marking a file is done by using the Comment field in the OpenPGP message with this content:

```
Comment: <Test>1</Test>
```

Please note:

- Testing must be coordinated with Integration Services.
- It is not possible to receive a test marked OpenPGP secured file from the bank.

Only pain.001.001.03 (example: pain,-d-,001,-d-,001,-d-,03-examples.pdf) and Danske Bank comma message types are supported. Should you wish to test a PAYMUL file, please instead use the test mark inside the file, according to the format description. Other formats are not supported.

Step	**Activity**	**Description**
1	Create your message to the bank	You must have a file to send to the bank. We recommend using a payment initiation file, examples can be found here: https://danskeci.com/ci/transaction-banking/instructions/integration-services/formats-by-standard
2	Secure the file with OpenPGP security	- PGP comment field must be filled in with information that this is a test. - Digitally sign the message using your private OpenPGP certificate and encrypt the message with Danske Bank's public encryption OpenPGP certificate.
3	Send the file to the bank	Send the file to the bank via an established communication channel.

PGP notifications from the bank

When you use OpenPGP encryption to secure your files we will sometimes send PGP notifications to you. The notifications are primarily used to inform you about events that are related to your PGP certificate and keys. We will always inform you when these events occur.

- Onboard certificate: Success or failure
- Renew certificate: Success or failure
- Revoke certificate: Success or failure
- Send an encrypted message to the bank: Only when failure to decrypt the message
  - We do not send notifications to you when we successfully received a file from you (if you want a reply to your messages, you should use standard return files from the bank like Payment Status Reports (ISO20022 XML). Likewise, we do not send PGP messages when we successfully encrypted a message sent to you (here you know the encryption went well because you received the encrypted file).
- Send an encrypted message to a customer (from the bank): Only when failure to encrypt the message
- Expiry of your public keys: Due to the importance of renewing your keys in a timely manner, we will notify you frequently and at a rate that will increase until actual expiration. More specifically, you will receive a notification when there are 90, 60, 30, 14, 7, 6, 5, 4, 3, 2 and 1 day(s) until expiration. We will stop sending notifications, when you used the certificate to sign a new certificate.

PGP notifications are delivered on District agreement level. This means that we cannot deliver the notifications to individual users on a District agreement (in case you have more than one PGP user on your District agreement). Notifications can only be sent to one fixed receiver channel on a District agreement: one email receiver address or via SFTP/FTP.

Response codes

You can receive PGP notifications from the bank in one of two different ways:

- an email (here we need your email address, preferably to a department mailbox), or
- an XML file which will be sent in the same channel as you use for sending files to the bank. We use two different kinds of messages:
  - Positive response: admi.004.001.02
  - Negative response: admi.002.001.01

The following table shows an overview of the messages.

Scenario	Code	Response type	Message
Onboard keys	0100	Success	PGP certificate for user {UserId} with fingerprint {Fingerprint} was received and approved
	0101	Failure	PGP certificate for user {UserId} was rejected
	0102	Failure	PGP comment <Action>XXXXXXXXXXX</Action> is not valid
Decrypt customer file	0201	Failure	Decryption of file failed for user: {UserId}
	0202	Failure	The decrypted file was empty and will not be processed for user: {UserId}
	0203	Failure	The file does not contain a signature for user: {UserId}
	0204	Failure	The file does not contain a valid signature for user: {UserId}

	0205	Failure	Signature verification failed for user: {UserId}. Public key not found
Encrypt customer file	0301	Failure	Sign and encrypt file failed for user: {UserId}
	0302	Failure	Customer certificate not found for user: {UserId}
Renew keys	0400	Success	User {UserId} cert with fingerprint {Fingerprint} was renewed and approved
	0401	Failure	Decrypt and verify renew certificate failed for user: {UserId}
	0402	Failure	The certificate cannot be imported due to invalid data for user: {UserId}
Revoke keys	0500	Success	PGP certificate for user {UserId} with fingerprint {Fingerprint} was revoked
	0501	Failure	Revocation of PGP certificate for user {UserId} failed
Key expiry	0600	Success	PGP certificate for User {UserId} with fingerprint {Fingerprint} will expire in {NumberOfDays} day(s)
Technical error in the bank	0700	Failure	PGP file processing failed due a technical error. Please contact the bank or retry sending the message
Test of onboarding	0800	Success	OnboardKeysTest: PGP Certificate for User {UserId} with fingerprint {Fingerprint} was successfully validated. In order to complete registration, change comment to OnboardKeys and resend.
	0801	Failure	OnboardKeysTest: PGP certificate validation failed. Please refer to our OpenPGP Guide and retry.

Notification by email or XML

Notifications can either be received by email or in an XML file, where you will be able to see the specific error message.

Example XML notifications

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:admi.004.001.02"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <SysEvtNtfctn>
                <EvtInf>
                        <EvtCd>0100</EvtCd>
                        <EvtDesc>PGP certificate for user 123456 with fingerprint
2D484185CE04EAEB13AA2E84B0CA862350218049 was received and approved.</EvtDesc>
                        <EvtTm>2023-05-11T10:47:53.909</EvtTm>
                </EvtInf>
        </SysEvtNtfctn>
</Document>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:admi.002.001.01"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <admi.002.001.01>
                <RltdRef>
                        <Ref>Agreement=654321,FileId=230511I064</Ref>
                </RltdRef>
                <Rsn>
                        <RjctgPtyRsn>0101</RjctgPtyRsn>
                        <RjctnDtTm>2023-05-11T10:44:05.728</RjctnDtTm>
                        <RsnDesc>PGP certificate for user 123456 was rejected.</RsnDesc>
        </Rsn>
    </admi.002.001.01>
</Document>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="urn:iso:std:iso:20022:tech:xsd:admi.002.001.01" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <admi.002.001.01>
        <RltdRef>
            <Ref>Agreement=654321,FileId=230511I068</Ref>
        </RltdRef>
        <Rsn>
            <RjctgPtyRsn>0205</RjctgPtyRsn>
            <RjctnDtTm>2023-05-11T10:56:37.259</RjctnDtTm>
            <RsnDesc>Signature verification failed for user: 123456. Public key not found.</RsnDesc>
        </Rsn>
    </admi.002.001.01>
</Document>
```

Contact information

Contact Integration Services for support on OpenPGP.

Telephone

Denmark	70 114 115
International	+45 70 152 151 (at local charge)

E-mail

ints@danskebank.com

Appendices

Appendix A: Technical requirements

To be able to implement and use OpenPGP successfully we recommend you check the following technical prerequisites and requirements:

Concept	Notes
Message format	• PGP certificates, including their keys, must be provided in the OpenPGP format. • All OpenPGP output should be ASCII-armored and Base64 encoded to ensure compatibility and secure transmission, and the message format must follow the specifications defined in RFC 4880. • RSA asymmetric cryptography must be used for encryption, with keys having a minimum length of 4096 bits to maintain strong security standards. • The OpenPGP software configuration must have data compression enabled to optimize message size and performance.
Customer keys	• Key length: 4K • Only one public key with two subkeys are accepted. Subkeys are for verify signature and for encryption. • Your keys can have a maximum life span of two years in Danske Bank. This means that if we receive a set of keys that are valid more than two years, we will set an end date in our system that runs two years from the day we received the keys (only exception from this is when you onboard the solution, here we set the end date to three weeks from the day you onboarded). • Note: It is only allowed to onboard with a unique certificate (meaning it will be rejected if you try to onboard the same certificate on another user)
Bank keys	• Key length: 4K • Only one public key with two subkeys are provided. Subkeys are for verify signature and for encryption. • The bank's keys will expire every two years with an overlapping period of three months where you need to add the new bank keys to your PGP routine.
Supported algorithms	• cipher-algo AES256 • digest-algo SHA256 • digest-algo SHA384 • digest-algo SHA512 The main difference between a cipher algorithm and a digest algorithm is that with a cipher algorithm the encrypted data will have the same size as the unencrypted data, while a digest algorithm will use hash functions so that the encrypted data is usually shorter than the unencrypted data. RSA asymmetric keys with length of at least 4096 bit must be used
Character encoding	• ASCII Armor
File communication channel with Danske Bank	You must already have a running file communication channel with Danske Bank.

Further requirements and recommendations	• Danske Bank requires that OpenPGP customers use separate keys for signing and encryption; signing and encryption must not be done with the same key. • We recommend using one PGP certificate that contains both the signing key and the encryption key. • If you use one public PGP certificate for the signing key and another public PGP certificate for the encryption key, and you are in an onboarding scenario, the PIN must be used together with the signing certificate, which must be sent to the bank as the first file. The signing certificate can then be used to sign the encryption certificate when sending it to the bank.

Appendix B: Compatible systems

There are several OpenPGP compatible software solutions available, and many of them are free of charge.

OpenPGP in Danske Bank has been tested with Kleopatra/ gpg4win version 4.4.1, but this does not mean there aren’t other certificate management systems available that work as well.

Before choosing a solution, we recommend that you check the technical requirements in Appendix A to ensure that the solution is compatible with our requirements.

Appendix C: Supported features of the input files

This list contains file conditions that we support in scenarios where customers send PGP encrypted files to the bank via the FTP/VPN and SFTP channels used for Cash Management and District related services.

File condition	Example
Scandinavian letters in file name	Fileæøåäö.txt
Scandinavian and Polish letters in file content	ÅÆØæåø äö ć ń ó ś ź
Irish fada signs/accents	áÁ óÓ úÚ íÍ éÉ
Code page support for xml files	UTF-8, ISO-8859-1 and ISO-8859-15
Code page support for other file types	WIN1252

No empty line after
-----BEGIN PGP MESSAGE-----
	`-----BEGIN PGP MESSAGE-----`
	`hQEMA2y2nTzeZBPDAQf+JknFYursSMfvw7NhedjKNh+Tip`
	`TNsqvvU7Hnf11Oj89M`
	`Ddy5JqgWsRrB46FcZOOD1kCtZP2tqro0H8gpGegrH9N9Lg`
	`JIB1vBEOFCGiCa4/Ps…`

No empty line after PGP comment	`-----BEGIN PGP MESSAGE-----`
	`Comment: <Action>RenewKeys</Action>`
	`hQEMA2y2nTzeZBPDAQf+JknFYursSMfvw7NhedjKNh+Tip`
	`TNsqvvU7Hnf11Oj89M…`

One or two empty lines after PGP comment
-----BEGIN PGP MESSAGE-----
	`Comment: <Action>RenewKeys</Action>`
	
	`hQEMA2y2nTzeZBPDAQf+JknFYursSMfvw7NhedjKNh+Tip`
	`TNsqvvU7Hnf11Oj89M…`

GPG commands	`--se` (sign and encrypt must be done in one action)

Appendix D: Functionality not supported

The following functionality is not supported in Danske Bank's OpenPGP implementation:

- Sending and receiving e-mails from Danske Bank.
- Sending/receiving files on behalf of more than one District agreement.
- Web of trust concept

Appendix E: List of terms

Term	**Definition**
Asymmetric cryptography	Public-key cryptography, also known as asymmetric cryptography, is a class of cryptographic algorithms which require two separate keys, one of which is secret (or private) and the other is public. Although different, the two parts of this key pair are mathematically linked.
Certificate	See 'PGP certificate'
Clear text/cipher text	Clear text: readable text Cipher text: encrypted text
Communication channel	The method used for file exchange between customer and Danske Bank, example: Making file transfers using a host-to-host channel such as SFTP.
Digital signature	A digital signature or sign key is a mathematical scheme for demonstrating the authenticity of a digital file. Digital signature gives a recipient reason to believe that the file was created by a known sender. You sign a file with your private sign key, and others verify with the corresponding public sign key.
District user	Your technical user ID (Provided by the Bank. A 'Technical User' is a user under your District agreement that is used to set up integrations, for example, SFTP. It is not a personal user).
Expiry date	A specific date used in a PGP certificate. The date from which you can no longer use OpenPGP Security on file transfers to the bank or to renew your PGP certificate. In Danske Bank we only allow certificates to run for 2 years, and then it needs to be renewed. Therefore, we will set an expiry date in our system for two years, if the certificate we receive has an expiry date that is beyond 2 years.
GnuPG – Gnu Privacy Guard	GnuPG is the GNU project's complete and free implementation of the OpenPGP standard.
Host-to-host channel	See 'Communication channel'

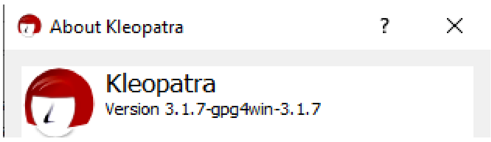
Key	See 'PGP certificate'
Middleware providers	Software vendors delivering B2B integration solutions that handle both file communication and security between ERP systems and e.g. EDI systems.
OpenPGP	OpenPGP is the most widely used encryption standard in the world. It is defined by the OpenPGP Working Group of the Internet Engineering Task Force (IETF) Proposed Standard RFC 2440. The OpenPGP standard was originally derived from PGP (Pretty Good Privacy), first created by Phil Zimmermann. For more on RFC 2440 please refer to the link: http://sunsite.icm.edu.pl/gnupg/rfc2440.html. The standard (partly) supported by Danske Bank is RFC4880 (www.ietf.org/rfc/rfc4880.txt)
Passphrase	OpenPGP private keys have two components: a file on customer's disk and a passphrase. The file on disk contains your private key (hidden to outsiders). A passphrase is much like a password, except that it is much longer and includes spaces. Whenever you work with your private key, the OpenPGP program will request your passphrase. Passphrase is private to you and it should not be shared. It is recommended to have a strong Passphrase.
PGP	Pretty Good Privacy (PGP) is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. PGP is often used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions and to increase the security of e-mail communications. The program was created by Phil Zimmermann in 1991.
PGP certificate	Certificates are in this context labelled as PGP Certificates (not x.509). In Danske Bank's interpretation of the OpenPGP standard a certificate should contain a user ID, an email address (optional), a public key for signing and a public key for encryption.
PIN	A one-time code used to implement OpenPGP in your system.
Pretty Good Privacy (PGP)	See 'PGP'
Private decryption key	A private decryption key is private to you. It should not be distributed to others. A private decryption key is used to decrypt cipher text.
Private sign key	A private sign key is private to you. It should not be distributed to others. A private sign key is used to sign an encrypted file, so that the recipient of the file can identify your signature.
Public encryption key	A public encryption key is shared with Danske Bank who you exchange OpenPGP secured messages with. It may be distributed to others as well. The public encryption key is used by the bank to encrypt plain text messages into cipher text sent to the customer.

Public sign key	A public sign key is shared with Danske Bank who you exchange OpenPGP secured messages with. It may be distributed to others as well. The public sign key is used by the bank to validate the signature on your OpenPGP secured messages that you send to the bank.
RSA – keys	RSA is an algorithm for public-key cryptography. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described it in 1977.
Symmetric cryptography	Symmetric-key algorithms are algorithms for cryptography that use the same cryptographic keys for both encryption of plain text and decryption of cipher text
Web of trust	In cryptography, a web of trust is a concept used in PGP, GnuPG, and other OpenPGP-compatible systems to establish the authenticity of the binding between a public key and its owner.

Appendix F: Code examples

In this section we show, through code examples, how to perform:
1) Onboarding,
2) Certificate renewal,
3) Signing and encrypting a file to the bank
4) Decrypting and verifying a file from the bank

The code examples below use Kleopatra/Gpg4Win (Version 3.1.7). Other tools have similar command syntax and it is not difficult to adapt the below code to them.



Kleopatra
Version 3.1.7-gpg4win-3.1.7

1) Onboarding (For Windows based systems)

Onboarding consists of four steps:
1) Ordering a PIN
2) Creating your key pair
3) Checking your certificate
4) Sending your public certificate to the bank.

Step 1) is of administrative character and no code is involved in performing it.

To make Step 2) work, create a batch file ***PGP gen-key.bat*** with the following commands:

```
gpg --batch --gen-key "%~f1"
pause
```

Then, create a text file ***TestCertificate.txt*** containing the following, replacing the example information with your real data. The information in the fields highlighted in **bold** must be changed:

```
Key-type: RSA
Key-length: 4096
Key-Usage: sign,auth
Name-Real: TEST Usr001 Sign and Encrypt cert 02
Name-comment: Second cert
Name-Email: TestTestersen@company.dk
Expire-Date: 2y
PassPhrase: login@1234
Subkey-Type: RSA
Subkey-Length: 4096
Subkey-Usage: encrypt
```

Drag and drop the text file on the above bat file, or run the below command in the command prompt, and the key pair will be created according to the data supplied in the text file.

```
"PGP gen-key.bat" TestCertificate.txt
```

The command above will output the KeyID of the new key, which is needed in the following steps, like in the following output example:

```
gpg: key 9979F7DB marked as ultimately trusted
```

To check your certificate, which constitutes Step 3), perform the following command:

```
gpg --batch --armor --output MyCert.txt --export 9979F7DB
```

In the line above, replace 9979F7DB with the actual last eight hexadecimal digits of the keyId of your new key.

Look at the file. It should start with the text “-----BEGIN PGP PUBLIC KEY BLOCK-----". Now type:

```
gpg --list-packet MyCert.txt
```

As a result you should see something like this:

```
:public key packet:
        version 4, algo 1, created 1688392409, expires 0
        pkey[0]: [4096 bits]
        pkey[1]: [17 bits]
        keyid: E4DE06A2219DEC3C
# off=528 ctb=b4 tag=13 hlen=2 plen=77
:user ID packet: "TEST Usr001 Sign and Encrypt cert 02 (Second cert)
<TestTestersen@company.dk>"
# off=607 ctb=89 tag=2 hlen=3 plen=596
:signature packet: algo 1, keyid E4DE06A2219DEC3C
        version 4, created 1688392409, md5len 0, sigclass 0x13
        digest algo 8, begin of digest d7 f7
        hashed subpkt 33 len 21 (issuer fpr v4
5C1DFC4CF67069E46A2CEE71E4DE06A2219DEC3C)
        hashed subpkt 2 len 4 (sig created 2023-07-03)
        hashed subpkt 27 len 1 (key flags: 23)
        hashed subpkt 9 len 4 (key expires after 2y0d0h0m)
        hashed subpkt 11 len 4 (pref-sym-algos: 9 8 7 2)
        hashed subpkt 21 len 5 (pref-hash-algos: 10 9 8 11 2)
        hashed subpkt 22 len 3 (pref-zip-algos: 2 3 1)
        hashed subpkt 30 len 1 (features: 01)
        hashed subpkt 23 len 1 (keyserver preferences: 80)
        subpkt 16 len 8 (issuer key ID E4DE06A2219DEC3C)
        data: [4096 bits]
# off=1206 ctb=b9 tag=14 hlen=3 plen=525
:public sub key packet:
        version 4, algo 1, created 1688392409, expires 0
        pkey[0]: [4096 bits]
        pkey[1]: [17 bits]
        keyid: EA8FA6AFF9B703ED
# off=1734 ctb=89 tag=2 hlen=3 plen=572
:signature packet: algo 1, keyid E4DE06A2219DEC3C
        version 4, created 1688392409, md5len 0, sigclass 0x18
        digest algo 8, begin of digest f5 1e
        hashed subpkt 33 len 21 (issuer fpr v4
5C1DFC4CF67069E46A2CEE71E4DE06A2219DEC3C)
        hashed subpkt 2 len 4 (sig created 2023-07-03)
        hashed subpkt 27 len 1 (key flags: 0C)
        hashed subpkt 9 len 4 (key expires after 2y0d0h0m)
        subpkt 16 len 8 (issuer key ID E4DE06A2219DEC3C)
        data: [4095 bits]
```

There should be a value “key expires after …” (see above). The keys should be of appropriate length (4096).

To complete Step 4), import the bank encryption key into Kleopatra/Gpg4Win using the following line:

```
gpg --import DanskeBankOpenPGPpublickeys.asc
```

Then, create a batch file named ***Export Cert with PIN.bat*** with the following content, and run it:

```
ECHO OFF
CLS
REM
REM Export certificate
REM Prepend pincode to certificate
REM Encrypt certificate without signature
REM
IF EXIST tmp1.txt DEL tmp1.txt
IF EXIST tmp2.txt DEL tmp2.txt
REM
REM Get userid
set /P inpuser=[Enter District UserId : ]
REM Get cert keyid
set /P inpcert=[Enter certificate's keyid : ]
REM Get pin-code
set /P inppin=[Enter pin-code : ]
REM
REM Delete Output file
REM
IF EXIST "cert-%inpcert%-user-%inpuser%.asc" DEL "cert-%inpcert%-user-
%inpuser%.asc"
REM
IF [%inpuser%]==[] (
SET commentuser=
) ELSE (
SET commentuser=--comment "<UserId>%inpuser%</UserId>"
)
REM
gpg --batch ^
--armor ^
--output tmp1.txt ^
--export %inpcert%
SET returncode=%ERRORLEVEL%
IF %returncode%==0 (
ECHO.
ECHO Successfull export!
ECHO.
) ELSE (
ECHO.
ECHO Unsuccessfull export - errorlevel %returncode% !
ECHO See messages above...
ECHO.
GOTO Byebye
)
ECHO Encrypt :)
ECHO %inppin%; >> tmp2.txt
ECHO %inpuser%; >> tmp2.txt
TYPE tmp1.txt >> tmp2.txt
gpg --batch ^
-r 6F557906 ^
--trust-model always ^
--armor ^
--cipher-algo AES256 ^
--digest-algo SHA256 ^
%commentuser% --comment "<Action>OnboardKeys</Action>" ^
--output "cert-%inpcert%-user-%inpuser%.asc" ^
-e "tmp2.txt"
SET returncode=%ERRORLEVEL%
IF %returncode%==0 (
ECHO.
ECHO Successfull encrypt!
ECHO.
IF EXIST tmp1.txt DEL tmp1.txt
IF EXIST tmp2.txt DEL tmp2.txt
) ELSE (
ECHO.
ECHO Unsuccessfull encrypt - errorlevel %returncode% !
ECHO See messages above...
ECHO.
GOTO Byebye
)
:ByeBye
ECHO.
Pause
```

The script exports your public key encrypted with the Danske Bank public encryption key, and supplies the appropriate <comment> fields as described earlier in this document. The script requires the following input (Case Sensitive – use capital letters):
1) Your UserID
2) The Key ID of the key you created in step 2 above
3) The PIN-code

You should now be ready to send your public certificate to the bank.

2) Certificate renewal (For Windows based systems)

Renewal of a certificate consists of the following steps:
1) Generate private and public keys for signing and encryption.
2) Check your certificate
3) Export a PGP certificate containing the certificate with the public keys from step 1)
4) Sign it with your old signing key and encrypt it with the Danske Bank's public encryption key. Note that this has to be done in **a single step.**

Steps 1) and 2) require you to create a new key pair and check it. These steps are the same as Steps 2) and 3) in the Onboarding process above.

The remaining steps, 3) and 4), of the above procedure are completed in one go by creating a batch file **ExportCertWithSign.bat** containing the commands cited below. The script does three things:

1) Asks for various input information (more on that below)
2) Exports the newly created certificate into a file
3) Signs and encrypts it with corresponding keys in a single step.

In regard to the first, the script asks for the following information:
a) User ID (the user ID that you got when you enrolled with your first certificate)
b) Key ID of the new key, created in Step 1) above
c) Key ID of the key with which we want to sign the new certificate (related to the previous, valid certificate).
Second, the certificate created in Step 1) above is exported to a temporary file.
Third, the certificate stored in the temporary file is signed with the given certificate and encrypted with the key indicated with the last eight alphanumeric characters of its fingerprint (see 6F557906 below). This string must be replaced by last eight digits of the Encryption Subkey fingerprint of our production key for the resulting certificate to be accepted. Also edit the password in the script to match the password of the signing key.
Please note that signing and encryption are performed in a **single operation** (gpg -se), and not in separate steps.
After these edits, run the script and supply the various information that the script is asking for, as stated above. The result will be the new certificate signed with the old one and encrypted with Danske Bank public encryption key, as required.

```
ECHO OFF
CLS
REM
REM Export certificate
REM Encrypt and sign certificate
REM
IF EXIST tmp1.txt DEL tmp1.txt
IF EXIST tmp2.txt DEL tmp2.txt
REM
REM Get userid
set /P inpuser=[Enter UserId : ]
REM Get cert keyid
set /P inpcert=[Enter new certificate's keyid : ]
REM Get signing certificate
set /P inpsign=[Enter signing certificate's keyid : ]
REM
ECHO Export:
gpg --batch ^
--armor ^
--output tmp1.txt ^
--export %inpcert%
SET returncode=%ERRORLEVEL%
IF %returncode%==0 (
ECHO.
ECHO Successfull export!
ECHO.
) ELSE (
ECHO.
ECHO Unsuccessfull export - errorlevel %returncode% !
ECHO See messages above...
ECHO.
GOTO Byebye
)
ECHO Encrypt :
TYPE tmp1.txt >> tmp2.txt
gpg --batch ^
-r 6F557906 ^
--local-user %inpsign% ^
--passphrase login@1234 ^
--trust-model always ^
--armor ^
--cipher-algo AES256 ^
--digest-algo SHA256 ^
%commentuser% --comment "<Action>RenewKeys</Action>" ^
--output "cert-%inpcert%-user-%inpuser%.asc" ^
-se "tmp2.txt"
SET returncode=%ERRORLEVEL%
IF %returncode%==0 (
ECHO.
ECHO Successfull encrypt!
ECHO.
IF EXIST tmp1.txt DEL tmp1.txt
IF EXIST tmp2.txt DEL tmp2.txt
) ELSE (
ECHO.
ECHO Unsuccessfull encrypt - errorlevel %returncode% !
ECHO See messages above...
ECHO.
GOTO Byebye
)
:ByeBye
ECHO.
Pause
```

3) Signing and encrypting a file to the bank (For Windows based systems)

To sign your message with your private key and encrypt it with the Danske Bank public encryption key, create a batch file named ***SignAndEncrypt.bat***, with the content listed below. Make the following changes in it:
1) Replace the value of localuser 1234ABCD with the last eight alphanumeric characters of the KeyID of your private key for signing.
2) Replace the example value of password 12345678 with the password associated with your private key.

```
REM Encryption and signing of files using GnuPG
REM = = = = = = = = = = = = = = = = = = = = = = =
REM Prepend user in comment field
REM
ECHO OFF
CLS
REM
REM Setup variables that apply to all commands :
REM
SET extension=%~x1
IF %extension% == .pgp (
GOTO AlreadyEncrypted
)
IF %extension% == .PGP (
GOTO AlreadyEncrypted
)
IF %extension% == .asc (
GOTO AlreadyEncrypted
)
IF %extension% == .ASC (
GOTO AlreadyEncrypted
)
IF %extension% == .txt (
SET i01=%~f1
SET o01=%~dpn1.asc
GOTO Encrypt
)
IF %extension% == .TXT (
SET i01=%~f1
SET o01=%~dpn1.ASC
GOTO Encrypt
) ELSE (
SET i01=%~f1
SET o01=%~f1.ASC
GOTO Encrypt
)
:Encrypt
SET localuser=--local-user 1234ABCD
SET passphrase=--passphrase 12345678
SET recipient=-r 6F557906
set trustModel=--trust-model always
set armor=--armor
set AES256=--cipher-algo AES256
set SHA256=--digest-algo SHA256
set SHA512=--digest-algo SHA512
set NOZIP=--compress-algo Uncompressed
set ZIP=--compress-algo ZIP
set defaultPreferences=%armor% %AES256% %SHA256% %ZIP%
REM
REM DELETE THE OUTPUT-FILE
REM
IF EXIST "%o01%" (
DEL "%o01%"
)
REM
REM NOW DO THE TRICK!!!
REM
gpg -se --batch %localuser% %passphrase% %recipient% %trustModel%
%defaultpreferences% --output "%o01%" "%i01%"
SET returncode=%ERRORLEVEL%
IF %returncode%==0 (
ECHO.
ECHO.
ECHO Successfull encryption!
) ELSE (
ECHO.
ECHO.
ECHO Unsuccessfull encryption - errorlevel %returncode% !
ECHO See messages above...
)
GOTO ByeBye
:AlreadyEncrypted
ECHO Input files extension indicates that it is already encrypted.
:ByeBye
ECHO.
ECHO.
pause
```

Place your message in a text file, and run the above script, stating this file as the only command line parameter. The result will be a new file, signed and encrypted as required.

4) Decrypting and verifying a file from the bank (For Windows based systems)

(Note that it is optional to receive OpenPGP signed/encrypted files from the bank).

To decrypt and verify a message from the bank, using the bank's public verification key to verify and your private key to decrypt, create a batch file named ***DecryptAndVerify.bat***, with the content listed below. As in the previous example, make the following changes in it:
1) Replace `1929FCC6` with the last eight alphanumeric characters of the Key ID of the current Danske Bank public key for verification of signatures.

2) Replace the value of localuser with the last eight alphanumeric characters of the KeyID of your private key for decryption.
3) Replace the example value of password 12345678 with the password associated with your private key cited in point 2) above.

```
REM Decryption and authentication of files using GnuPG
REM = = = = = = = = = = = = = = = = = = = = = = = = = =
REM
REM
ECHO OFF
CLS
REM
REM Setup variables that apply to all commands :
REM
SET extension=%~x1
SET i01=%~f1
SET o01=%~dpn1.decrypt.txt
SET localuser=--local-user 1234ABCD
SET passphrase=--passphrase 12345678
SET recipient=-r 1929FCC6
REM
REM DELETE THE OUTPUT-FILE
REM
IF EXIST "%o01%" (
DEL "%o01%"
)
REM
REM NOW DO THE TRICK!!!
REM
gpg -d --batch %localuser% %passphrase% --output "%o01%" "%i01%"
SET returncode=%ERRORLEVEL%
IF %returncode%==0 (
ECHO.
ECHO.
ECHO Successfull decryption!
) ELSE (
ECHO.
ECHO.
ECHO Unsuccessfull decryption - errorlevel %returncode% !
ECHO See messages above...
)
```

```
ECHO.
ECHO.
:ByeBye
PAUSE
```

Place the message you received in a text file, and run the above script, stating this file as the only command line parameter. The result will be a new file, decrypted and verified as required.

5) Example of encrypting the initial certificate (For non-Windows based systems)

UserId is the technical user you been provided by Danske Bank.
MyCert is your unencrypted public certificate with the pin code and user
Encrypted-Cert is the output, which is your encrypted public certificate, that must be sent to the bank.

```
gpg -r 6F557906 --trust-model always --armor --cipher-algo AES256 --digest-
algo SHA256 --comment "<UserId>123456</UserId>" --comment
"<Action>OnboardKeys</Action>" --output "Encrypted-Cert.asc" -e
"MyCert.asc"
```

6) Example of signing and encrypting a certificate for renewal (For non-Windows based systems)

Local-user is the key id of your current certificate.
Passphrase is the password associated with your certificate.
MyCertNew.asc is your new public certificate.
RenewalCert is your new public certificate, which has been signed and encrypted and must be sent to the bank.

```
gpg -r 6F557906 --trust-model always --local-user 12345678 --passphrase
1234 --armor --cipher-algo AES256 --digest-algo SHA256 --comment
"<Action>RenewKeys</Action>" --output "RenewalCert.asc" -se "MyCertNew.asc"
```

7) Example of signing and encrypting a payment file (For non-Windows based systems)

Local-user is the key id of your current certificate.
Passphrase is the password associated with your certificate.

PaymentFile is the input file you want to sign and encrypt.
Encrypted_PaymentFile is the output, which is a signed and encrypted payment file.

```
gpg -r 6F557906 --trust-model always --local-user 12345678 --passphrase
1234 --armor --cipher-algo AES256 --digest-algo SHA256 --output
"Encrypted_PaymentFile.asc" -se "PaymentFile.xml"
```

Change log

Version number	Date	Change
1.0	16.05.2023	Document created
1.1	24.07.2023	Added appendix F Code examples
1.2	30.08.2023	Minor changes
1.3	27.09.2023	Added the chapters Test scenarios and Contact information + minor changes
1.4	16.11.2023	Added further code examples and additional test scenario + minor changes
1.5	07.02.2024	Minor changes
1.6	18.06.2024	Minor changes
1.7	20.01.2025	Updated sub-key to new Danske Bank public key
1.8	21.02.2025	Removed requirement for adding S; and H; when onboarding and renewal of certificates. Removed no support for Service bureau set-up
1.9	31.03.2025	Inserted new text on page 5. Customer can acquire the bank's public PGP keys from DanskeBank.com/INTS
1.10	20.02.2026	Updated version where guide is done more clear

Disclaimer

This guide has been prepared by Danske Bank A/S ("Danske Bank") for information purposes only. Whilst reasonable care has been taken to ensure that the contents of the guide are comprehensive, accurate and not misleading, Danske Bank accepts no responsibility for its completeness and compatibility with your software, hardware, IT- and operational security solutions. Danske Bank accepts no liability for any loss arising from reliance on this guide.

The guide should not be relied upon as investment, legal, tax, or financial advice under no circumstances — please consult with your professional advisors as to the legal, tax, financial or other relevant matters. The guide is not an offer or solicitation of any offer to purchase or sell any product, service or financial instrument.

The guide has been prepared for Danske Bank customers using OpenPGP in all countries where OpenPGP is available and should be read together with other applicable documentation.

The guide may not be further distributed or shared with any third party without Danske Bank A/S' prior written consent.

Any information contained herein is not intended for distribution to or use by any person in any jurisdiction or country where such distribution or use would be unlawful.