BankID Open B2B

BankID OpenB2B is a specific packaging of documentation, tools and example code on how to use BankID merchant certificates for B2B signing applications without the need to install or use any software from BankID

Note

BankID Open B2B is currently in **preview** status. Using BankID OpenB2B for production purposes will work, but documentation and example code may change depending on customer feedback. The service currently supports B2B siging. It may be extended to support B2B encryption in the future.

This page describes the BankID Open B2B packaging, ie. a set of documentation, tools and example code on how to use BankID merchant certificates in business-to-business scenarios. An Open B2B BankID consists of industry compliant X.509 certificates, and can be used without the need for installed BankID software. The target audience of this document is merchant project teams: Technical personnel designing and coding the use of BankID Open B2B. The rest of this document is organized as follows:

1 Concepts and abbreviations
2 BankID Open B2B scenarios
- 2.1 Scenario: Sender verifies BankID status
- 2.2 Scenario: Receiver verifies BankID status
- 2.3 Comparison of scenarios

3 Example code
4 BankID lifecycle overview
- 4.1 Create
  - 4.1.1 Order
    - 4.1.1.1 PREPROD
    - 4.1.1.2 PROD
  - 4.1.2 Activate
    - 4.1.2.1 PREPROD
    - 4.1.2.2 PROD
  - 4.1.3 Create keystore
- 4.2 Test
- 4.3 Use
- 4.4 Renew
  - 4.4.1 PREPROD
  - 4.4.2 PROD
  - 4.4.3 PREPROD
  - 4.4.4 PROD

5 Appendix A: Keystores and trust
- 5.1 Trust stores
  - 5.1.1 BankID CA trust
  - 5.1.2 BankID VA (OCSP responder) trust
  - 5.1.3 BankID VA SSL trust
- 5.2 Merchant Open B2B keystore

Concepts and abbreviations

Concept /abbreviation	
BankID CA	BankID Certification Authority: The originator's system for issuing BankIDs.
BankID RA	BankID Registration Authority: The originator's system for handling the lifecycle of BankIDs.
BankID VA	BankID Validation Authority: The originator's system for online checking the revocation status of a BankID.
CN	Common Name: A field in the X.509 certificates of an BankID identifying the merchant.
CSR	Certificate signing request: A request for a BankID CA to sign a certificate stating that the Common Name of an merchant is coupled to a BankID (i.e. coupled to a specific public/private key pair).
OCSP	Online Certificate Status Protocol.
Originator	Issuer of an BankID.
Sky-MAT	MAT = Merchant Activation Tool: Online service for activating an Open B2B BankID.

BankID Open B2B scenarios

An Open B2B BankID is a standard X.509 certificate and can be used in many ways. However, there are a couple of use cases that are especially relevant. Both use cases relates to signing data sent from merchant A to merchant B. Sending data signed with an Open B2B BankID ensures that the receiver can verify that:

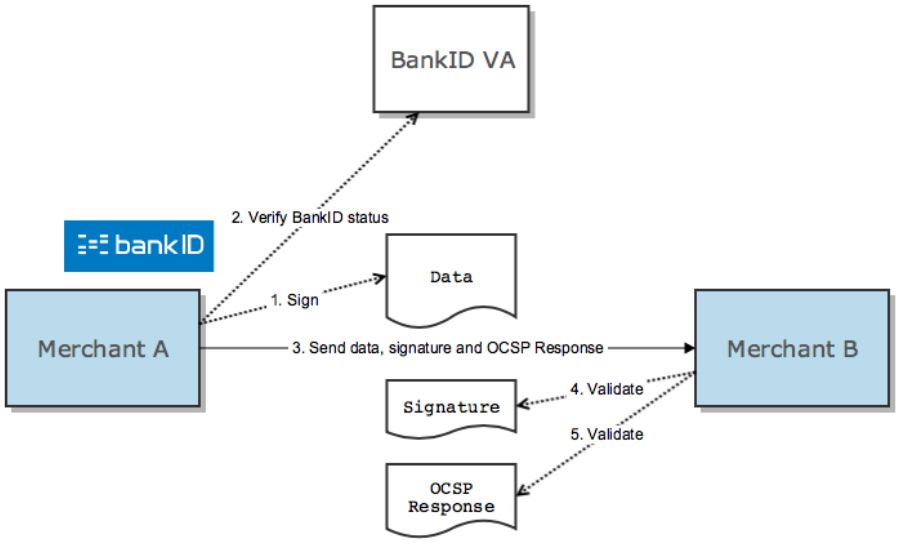
- The data are not changed under transport.
- The identity of the sender is genuine and non-reputable.

There are two scenarios for verifying data integrity and sender identity. Both implies that the signature itself is validated, this is done offline by the receiver. Further, to be sure the signature is made with a valid BankID, the status of the signing BankID must be verified online with BankID VA.

For both scenarios it is the merchant verifying BankID status that are charged for the status check.

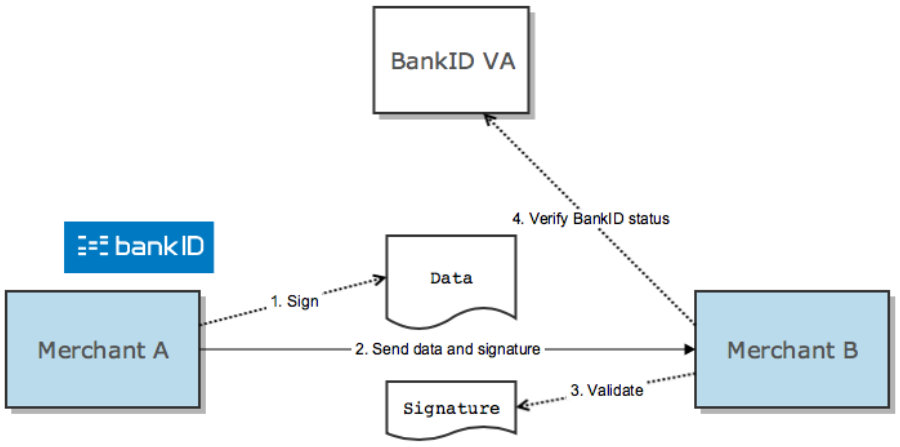
Scenario: Sender verifies BankID status

When merchant A checks its own BankID status with BankID VA (2), merchant A will get a signed OCSP response in return. The OCSP response is sent to merchant B together with the data and the signature itself. Merchant B must validate, offline, both the signature (4) and the OCSP response's signature (5).



Scenario: Receiver verifies BankID status

Merchant B checks merchant A's BankID status with BankID VA (4).



Comparison of scenarios

The two most important factors to be considered when choosing between these two scenarios are therefore:

- Which merchant should be charged for the status checks?
- Which merchant shall integrate with BankID VA?

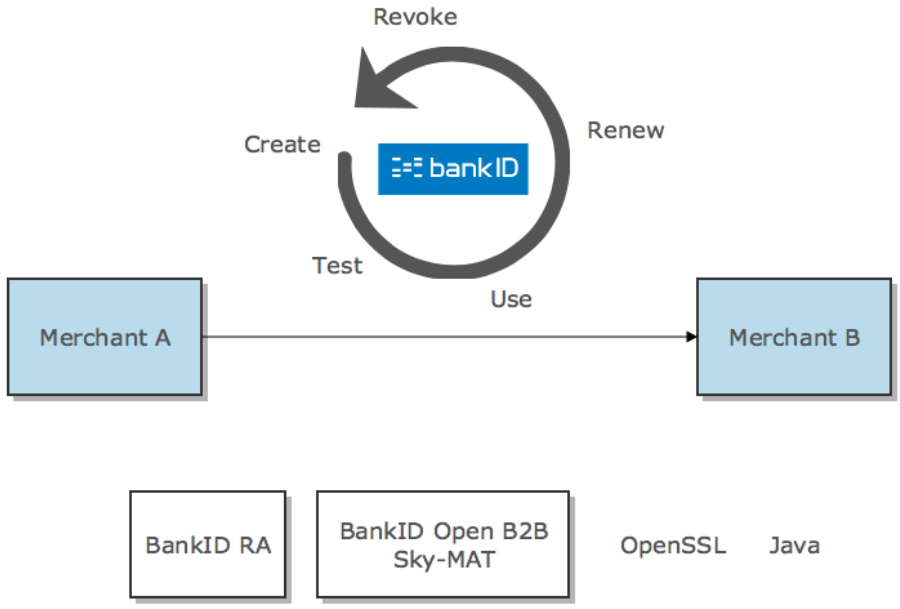
Another aspect to consider is the long term validity of the BankID status check, which is better with the "sender verifies" approach. This is because the status check is performed at the time of signing. If the signing BankID is revoked later, the OCSP response still shows the BankID was valid at time of signing. This difference is subtle, but may be a concern if the receiver, Merchant B, performs the status check long time (relatively) after Merchant A has sent the data. This can be relevant in batch oriented systems.

Example code

Example code supporting both the usage scenarios and the Open B2B BankID lifecycle are located in BankID Norge's GitHub repository, on the master branch. There are lifecycle examples both with Java's keytool and OpenSSL. The usage examples are in Java.

BankID lifecycle overview

This chapter describes the lifecycle of an Open B2B BankID. Online tools and/or example code required in each phase of the lifecycle are also referenced here.



Create

Order

The first step to create a fully functional (active) BankID is to submit an order to the originator's BankID RA. Inputs to the order are the merchants organization name and Common Name for the BankID. The order results in an activation URL and a shared secret, both to be used for activation of the BankID, see next chapter.

PREPROD

A self-service BankID RA in preprod is available to issue BankID merchant certificates for testing. Note that this tool optionally support immediate activation of the certificate. Activation may also happen as a separate step via the Sky-MAT tool.

PROD

Production BankIDs must be ordered from a reseller or from BankID Norge.

Activate

Before the BankID can be used it must be activated. This means acquiring a X.509 certificate with the merchant's CommonName coupled to the merchant's public key, where the corresponding private key is known to the merchant only. This is done by:

1. Create a private/public key pair. See example code in BankID Norge's GitHub repository.
2. Use the private/public key pair to create a CSR for each of the two certificates a BankID consists of. Se example code in BankID Norge's GitHub repository.
3. Submit the CSRs to Sky-MAT, together with the activation URL and shared secret from the BankID order.

There are some requirements to the CSRs. If the CSRs do not conform to these, the activation will be rejected by Sky-MAT. The requirements are:

- Use two different private/public key pairs for the two CSRs (authentication/signing).
- Never use the same set of private/public key pairs for CSRs for two different BankIDs.
- The Common Name in the CSRs MUST be the same as in the BankID order.

The result of a successful activation is a standard X.509 certificate chain, one for authentication and one for signing purposes. The format of the chain is PKCS#7. The merchant's certificate is the end entity of this chain. The merchant's certificates has key usage:

- Authentication: *Digital Signature, Key Agreement*
- Signing: *Non Repudiation*

PREPROD

A web-based activation tool Sky-MAT for pre-preprod certificates is available at BankID Open B2B Sky-MAT PREPROD.

PROD

A web-based activation tool Sky-MAT for production certificates is available at BankID Open B2B Sky-MAT PROD.

Create keystore

The last step before the BankID can be used for authentication/signing purposes is coupling the X.509 certificate with the merchant's private key and save it in a keystore. See example code in BankID Norge's GitHub repository how to do this. Also see appendix A for details about keystore usage.

Test

The Open B2B BankID can be tested with the integration tests found in BankID Norge's GitHub repository.

Use

Typical tasks in the usage scenarios are:

- Sign data to be sent to another merchant
- Get signed status check of BankID
- Create detached signature: Package merchant signature and status check into a Cryptographic Message Syntax (PKCS#7) format
- Send data and detached signature to another merchant
- Check validity of senders BankID

Example code for all these tasks are found in BankID Norge's GitHub repository.

Renew

The BankID expires after 4 years. Before this happens the BankID must be renewed.

PREPROD

Renewal of preprod certificates is currently not supported via the Sky-MAT tool. Please use the legacy Java-based HAT tool for this.

PROD

Renewal of production certificates is currently not supported via the Sky-MAT tool. Please use the legacy Java-based HAT tool for this.

Revoke

If a BankID is not to be trusted anymore because it's integrity is broken, it must be revoked.

PREPROD

Please use the self-service BankID RA in preprod to revoke BankID merchant certificates for testing.

PROD

Please contact the your reseller or BankID Norge to revoke a BankID merchant certificate in production.

Appendix A: Keystores and trust

For general guidance how to use Java trust stores, see the X509TrustManager chapter in Java Secure Socket Extension (JSSE) Reference Guide.

Trust stores

BankID CA trust

Data sent from one merchant to another is signed with a merchant BankID. Verifying such signatures requires that the BankID root Certificate Authority (CA) is registered as a "trust anchor", see TrustAnchor chapter in Java PKI Programmer's Guide.

The example code loads the trust anchor from a hardcoded BankID root CA certificate.

BankID VA (OCSP responder) trust

BankID status online check is done with an OCSP request to the VA. Verifying the OCSP response requires the VA's certificate to be registered as "OCSP responder certificate".

The example code loads the OCSP responder from a hardcoded VA certificate. Denne mangler for PROD i eksempelkoden.

BankID VA SSL trust

BankID status online check is done with an OCSP request to the VA over Secure Socket Layer (SSL). The SSL-certificate for VA is issued by the BankID root Certificate Authority (CA).

The example code loads the BankID root CA certificate into the JSSE-stack from a Java keystore.

Merchant Open B2B keystore

When signing data (which can be arbitrary data for sending to another merchant or a VA OCSP request) a merchant must use it's private key.

The example code loads the merchant certificate and private key from a Java keystore.