Public-Key Technology

HELP.BCSECDISI

Release 4.6C

Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of
Microsoft Corporation.

IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.

ORACLE® is a registered trademark of ORACLE Corporation.

INFORMIX®-OnLine for SAP and Informix® Dynamic Server™ are registered trademarks of Informix Software Incorporated.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium,
Massachusetts Institute of Technology.

JAVA® is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
⚠	Caution
💬	Example
➡	Note
🧭	Recommendation
SYN	Syntax
💡	Tip

Contents

Public-Key Technology

This topic describes the basic principles behind the public-key technology that is used to produce digital signatures and digital envelopes in SAP Systems.

Public and Private Keys

Characteristics of Public and Private Keys

The secret behind public-key technology lies in the relationship between two keys, a public key and a private key. The person or component that wants to "sign" owns these two keys. These two keys have the following characteristics:

- The keys are pairs; they belong together.
- You cannot obtain the private key from the public key.
- As the name suggests, the public key is to be made public. The owner of the keys distributes the public key as necessary. A recipient of a signed document needs to have knowledge of this key in order to verify the digital signature. Also, to send an encrypted document (digital envelope), the sender needs to know the recipient's public-key.
- The private key is to be kept secret. The owner of the keys uses the private key to generate his or her digital signature and to decrypt messages protected with a digital envelope. Therefore, the owner of the keys needs to make sure that **no** unauthorized person has access to his or her private key.

In the rest of the documentation, we refer to the owner of the keys as the signer and the piece of information to sign as a document.

Generating and Assigning Keys

To be able to sign digitally, the signer needs a pair of keys. A central instance, called a Certification Authority (CA), generates these keys and assigns them to the owner. You can compare this to a central office that distributes identification cards. These keys then "belong" to the owner and can be used for identification purposes.

As an alternative method to receiving your keys, you can generate them yourself and then send your public key to the CA to be certified.

Using a Digital Signature

Signing a Document

Then, to sign a document, the signer uses his or her private key to create his or her digital signature. We describe this process in Digitally Signing a Digital Document [Page 8].

The document, along with the signature, is passed on to the recipient.

Public-Key Technology

Verifying a Digital Signature

The recipient of the document then uses the signer's public key to verify the signature and the integrity of the document (that it has not been changed since being signed). This is explained in Verifying a Digital Signature [Page 10].

Using a Digital Envelope

Creating a Digital Envelope

To create a digital envelope, you use a secret message key to "wrap" the document in a secure "envelope". The recipient of the message also needs knowledge of this key to be able to decrypt the message. Therefore, you encrypt this message key using the recipient's public key and send it along with the document. See Creating a Digital Envelope [Page 14].

"Opening" a Digital Envelope

The recipient of the document then uses his or her own private key to encrypt the secret key that was used to encrypt the document. He or she can then decrypt the document using this secret key. This is explained in "Opening" a Digital Envelope [Page 16].

The Public-Key Certificate

The questions still arise: "How do you know which public key belongs to whom?" and "How do you obtain the signer's public key?"

The answers lie in the public-key certificate.

We have mentioned that the signer needs to have a pair of keys. We also mentioned that a central instance, called a CA, assigns these keys to the owner. The CA assigns these keys by issuing a digital certificate. This digital certificate contains the information needed to ensure that the public key belongs to the person indicated. For a detailed description, see Public-Key Certificate [Page 19].

The signer distributes his or her public key by distributing his or her public-key certificate (for example, directly with an e-mail or by using X.500 Directory Services).

The recipient uses the information from the public-key certificate (namely the public key and which hash algorithm to use) to verify the signature of the signed document. The recipient also knows that this public key belongs to this person, because a CA has also signed the public-key certificate. (The recipient should also know of and trust this CA.) The recipient can also verify the validity of the CA's signature, because it's signature and it's public key are also included in the public-key certificate.

For more information, see:

- Digitally Signing a Digital Document [Page 8]
- Verifying a Digital Signature [Page 10]
- Creating a Digital Envelope [Page 14]
- "Opening" a Digital Envelope [Page 16]
- Public-Key Certificate [Page 19]

Digital Signature

Definition

The digital signature serves the same function for the processing of digital data as the handwritten signature does for paper documents.

Use

You use the digital signature to "sign" digital documents. Digital signatures specifically identify the "signer" of a digital document and also protect the integrity of the document. (Changes in signed documents are detected when verifying the digital signature.)

Integration

You can use digital signatures in SAP Systems either together with an external security product or without. By using an external security product with the SAP System, you can introduce features that are not directly available with the SAP System, such as digital envelopes [Page 13] or the authentication of individuals using smart cards.

However, for certain areas of application (for example, SAP ArchiveLink content server HTTP interface), the digital signature itself suffices without needing the extra features of an external product. For this reason, we deliver the SAP Security Library [Ext.] (SAPSECULIB) with the SAP System. For more information, see the area of application's documentation that uses the digital signature.

Example

To find examples of using digital signatures in SAP Systems, see any of the following:

- Digital Signature in QM [Ext.]
- PI Sheet Processing [Ext.] in the section Reporting of Actual Data [Ext.]
- SAP Content Server HTTP Interface 4.5 [Ext.] in the section secKey [Ext.]

Digitally Signing a Digital Document

Purpose

You digitally sign a digital document for the same reasons that you sign ordinary documents. In addition, because of the way you create a digital signature, you can also verify the integrity of the document. (If the document has been changed after being signed, then the process of verifying the signature [Page 10] will fail.)

The following are possible reasons for digitally signing documents:

- To state that you have read or approved the document (for example, approving requests).
- To obligate yourself to the terms of the document (for example, closing paperless contracts or purchasing products over an online catalog).
- To protect data integrity (for example, signing archives for auditing purposes).

Prerequisites

To create a digital signature, you must possess a pair of keys. One key is public; the other is private. How to obtain these keys depends on the public-key infrastructure of your organization.

You also need a digital document to sign.

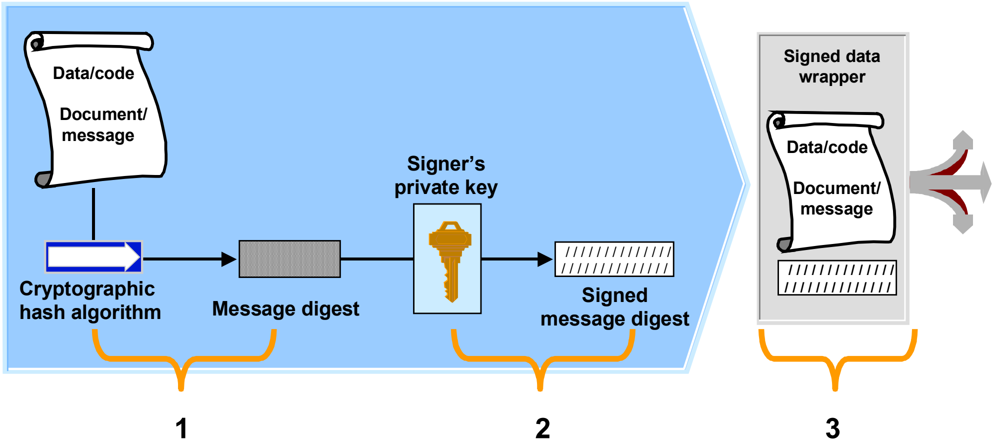
Process Flow

As an end user, you generally indicate that you want to "sign" a document and the system does the rest.

> ➡
>
> This step may also include a part of a business workflow where the system requests a digital signature before proceeding. You do need to give the system explicit access to your private key, for example, by providing a PIN or passphrase that allows the system to access the smart card or file where your secret key is stored.

The following graphic shows what happens when you digitally sign a document:

Digitally Signing a Digital Document



The following explains what happens at each step:

1. A hash algorithm is applied to the document or message to create a message digest for the document.

   The result is a message digest. This message digest represents a unique fingerprint for the document. If a cryptographic hash algorithm is used, then it should be impossible to compute another meaningful input message that will produce the same digest.

2. The signer's private key is applied to the message digest to create a signed message digest.
3. The document (in plain text), is packed together with the signed message digest to create a digitally signed document.

Result

The result is a digitally signed document that you can process in the same way as any other document. (For example, you can send it, save it, or archive it.) By verifying the digital signature [Page 10], you can then prove who the signer of the document was, as well as the integrity of the document.

Verifying a Digital Signature

Purpose

There are several reasons that you may want to verify a digital signature. For example:

- You have received a digitally signed document and you want to verify the identity of the sender.
- You want to verify the integrity of a signed document (for example, when auditing archives).

Prerequisites

Before you can verify a digital signature, you need the following:

- You need to have a signed document that you want to verify.
- You must also know the hash algorithm that the signer used for his or her signature.
- You need to have access to the signer's public key.

Process Flow

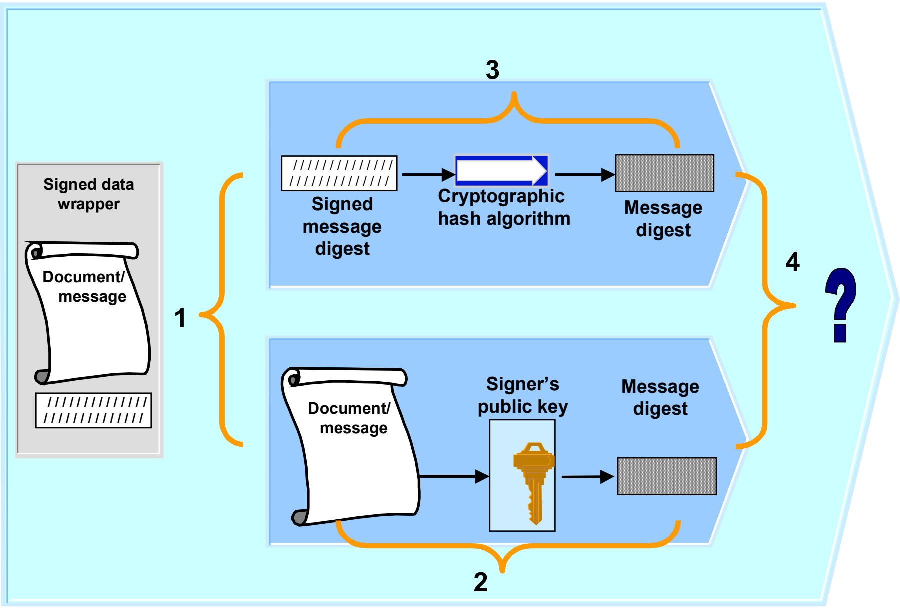
Generally, you indicate that you want to "verify" a digital signature, and the system does the rest.

⇨

This may also include a part of a business workflow where the system requests the verification of a digital signature before proceeding.

The following diagram shows what happens when you verify a digital signature.

Verifying a Digital Signature



The following explains what happens at each step:

1. The digitally signed document is divided into its components: the signed message digest and the document itself.
2. The public key is applied to the signed message digest.

   The result is the message digest from the original document.
3. The same hash algorithm that was used in the signing process is then applied to the document to be verified.

   The result is the message digest for the signed document.
4. The two message digests are compared.

Result

The result is either the acceptance or denial of the digital signature, based on the following conclusions:

- If the message digests are identical, then:
  - The signer is who you think it is (that is, the signer is the owner of the private key that corresponds to the public key that you used to verify the signature).
  - The document has not been altered after being signed.
- If the two message digests are not identical, then:
  - Either the document has been altered, or

- The signer is not who you think it is (that is, the message was signed with a key other than the private key that corresponds to the public key that you used in the verification).

Digital Envelope

Definition

A digital envelope is a secure "data wrapper" that provides privacy protection for a data packet. It protects the contents of the data packet from being viewed by anyone other than the intended recipient.

Use

You can use digital envelopes when storing or sending confidential data.

Integration

You need to use an external security product with the SAP System to be able to use digital envelopes. The security product must be certified by SAP and it must support the PKCS#7 standard data format and X.509 public-key certificates.

Creating a Digital Envelope

Purpose

You use a digital envelope to protect a digital document from being visible to anyone other than the intended recipient.

The following are possible reasons for using digital envelopes:

- Sending confidential data or documents across (possibly) insecure communication lines
- Storing confidential data or documents (for example, company-internal reports)

Prerequisites

To create a digital envelope, you need access to the intended recipient's public key. How to obtain access to the public key depends on the public-key infrastructure of your organization.

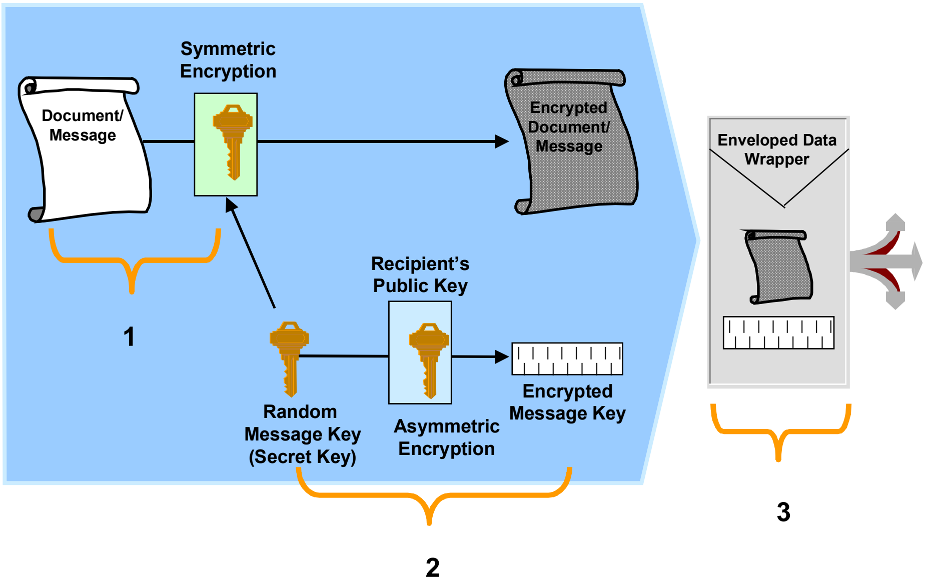
You also need the digital document that you want to protect.

Process Flow

As an end user, you generally indicate that you want to "create an envelope" for a document and the system does the rest.

The following graphic shows what happens when you create a digital envelope:

Creating a Digital Envelope



The following explains what happens at each step:

1. The message is encrypted using symmetric encryption. Typically, a newly generated random message key (secret key) is used for the encryption.

   Symmetric encryption means that the same key is used for both encryption and decryption (a secret key). Anyone wanting to decrypt the message needs access to this key.

2. To transfer the secret key between the parties, the secret key is encrypted using the recipient's public key.

3. The encrypted document and the encrypted message key are packed together in a single data packet to save or send to the intended recipient.

Result

The result is a secured digital document that only the owner of the corresponding private key can view (see "Opening" a Digital Envelope" [Page 16]).

'Opening' a Digital Envelope (Develope)

Purpose

You "open" a digital envelope if you are the intended recipient or viewer of a digital document that has been secured by a digital envelope.

Prerequisites

The prerequisites are:

- The digital envelope to open
- Your private key

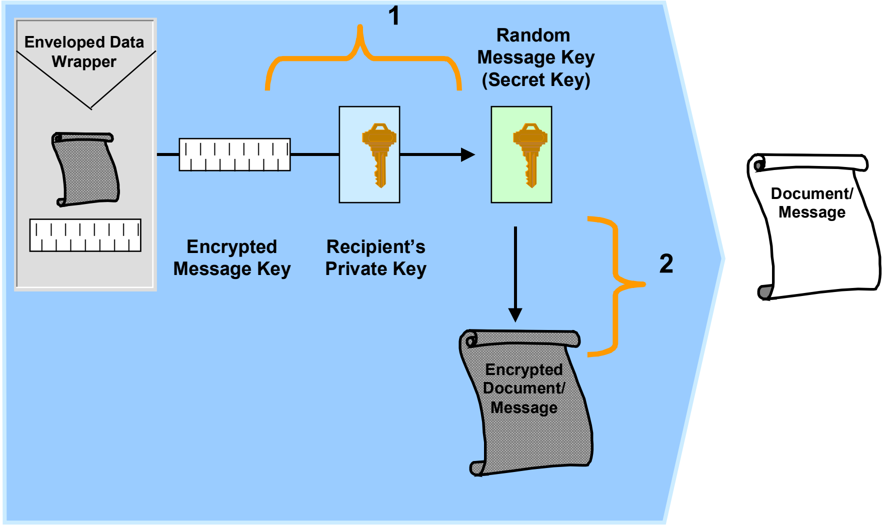
Process Flow

As an end user, you generally indicate that you want to "open" a digital envelope and the system does the rest.

> ➡
>
> This step may also include a part of a business workflow where the system requests the opening of a digital envelope before proceeding. You do need to give the system explicit access to your private key, for example, by providing a PIN or passphrase that allows the system to access the smart card or file where your secret key is stored.

The following graphic shows what happens when you "open" a digital envelope:

'Opening' a Digital Envelope



The following explains what happens at each step:

1. The recipient applies his or her private key to the encrypted message key.

   The result is the secret key that was originally used to encrypt the digital document.

2. The secret key that was revealed in the previous step is used to decrypt the digital document.

Result

The intended recipient (the owner of the corresponding private key) can view the contents of the digital document.

Personal Security Environment (PSE)

Definition

Secure location where a user or component's public-key information is stored. The PSE for a user or component is typically located in a protected directory in the file system or on a smart card. It contains both the public information (public-key certificate and private address book) as well as the private information (private key) for its owner. Therefore, only the owner of the information should be able to access his or her PSE.

For example, the SAP Security Library (SAPSECULIB) stores the application server's information in a PSE. In this case, the PSE contains both the private address book [Ext.] for the SAP System as well as the SSF profile [Ext.].

Use

The user or component's PSE contains the information needed to create and verify digital signatures and to create or "open" digital envelopes. As part of a system workflow, when the system creates a digital signature for a user, the user usually has to give the system explicit permission to access the information in his or her PSE. For example, he or she must enter the PIN (Personal Identification Number) or passphrase that protects the PSE.

Structure

The exact structure and contents of the PSE are determined by the product that you use. Typical contents of a PSE include the user's public-key certificate [Page 19], private address book, and private key.

Public-Key Certificate

Definition

The public-key certificate acts as a digital identification card that identifies a person or component.

Use

You use your own public-key certificate to identify yourself to others.

You can use someone else's public-key certificate to verify their digital signature.

Structure

A signer's public-key certificate contains the information you need to verify his or her digital signature, namely the public key and which algorithm was used. Additional information is also included so that you know that this public key does actually belong to the person or component.

There are various formats for storing this information; one standard that is commonly used is the X.509 certificate, which contains the following information:

- **General Information**
  - Version
  - Serial number
  - Validity period
- **Certificate Issuer's Information**
  - CA's Distinguished Name
- **Certificate Owner's Information**
  - Owner's Distinguished Name
  - Owner's public key
  - Asymmetric, cryptographic algorithm used
- **CA's Digital Signature**
  - Asymmetric, cryptographic algorithm used
  - CA's digital signature

    ➡

    Note that the CA's signature is also included in the public-key certificate as an additional (and necessary) measure to prove the authenticity of the certificate, the public key, and therefore, the digital signature.