Secure Network Communications and Secure Store & Forward Mechanisms with the SAP R/3® System

SAP AG ● Neurottstraße 16 ● 69190 Walldorf ● Germany

Contents

Secure Network Communications and Secure Store & Forward Mechanisms with the SAP R/3 System

Abstract

Security in the sense of data protection is gaining more and more importance with SAP R/3 customers. SAP supports strong security mechanisms to protect the interests of the user, but has decided not to include cryptographic modules in its own software. Instead, external products can be integrated which have been developed by security professionals.

Both Secure Network Communications (SNC) and Secure Store and Forward Mechanisms (SSF) support state-of-the-art authentication, data integrity, and confidentiality services for the R/3 System by integrating an external security product using well-defined application programming interfaces. At the present time, SECUDE is the only product that is commercially available to SAP customers, but additional product certification will begin in early 1998.

Secure Network Communications (SNC)

Secure Network Communications (SNC) provides protection for the communication links between the distributed components of an R/3 System. With SNC, SAP R/3 can support products which adhere to the *GSS-API Version 2* Standard.

Features which are supported by SNC include:

- Application level, end-to-end security
- Smartcard authentication
- Single Sign-On

Cooperation projects have been productive at MIT since September 1996 and at Volkswagen AG since March 1997.

Secure Store & Forward Mechanisms (SSF)

Secure Store & Forward Mechanisms (SSF) provides the required support to protect R/3 data and documents as independent data units. This is done via:

- digital signatures and
- digital envelopes.

The standard used for these secure data formats is PKCS#7. When applying these secure formats, the authenticated data is placed in an envelope (security wrapper) before the data is stored or transmitted.

Both SNC and SSF offer increased security for the customer, while maintaining a high level of flexibility and expansion capabilities. By integrating external products, individual requirements can be met, and SAP can ensure that the customer receives the security protection which best fits his needs.

Introduction

With the increasing market penetration of business software, data security and information security are becoming of utmost importance to the customer. When a company uses SAP's R/3 System to model and execute its most important business processes, R/3 becomes a mission-critical application. To meet these demands, SAP provides a variety of security mechanisms directly in the R/3 System:

- authentication of all R/3 users by means of passwords,
- authorization concept, and
- activity logging.

With the increasing distribution of systems and the resulting use of computer networks and global integration of IT systems, new potential security trouble spots have emerged. As a response to the need for protecting distributed transactions and application components communicating over network connections, SAP provides additional security support in the R/3 System through:

- **Secure Network Communications (SNC) and**
- **Secure Store & Forward (SSF).**

With SNC, protection of the communication links between the distributed components of an R/3 System is achieved. In addition, SNC enables the use of cryptographic mechanisms and smartcards to securely authenticate users.

With SSF, digital signatures and encryption are provided to R/3 applications for protecting business data and documents as independent data units.

Where possible, SNC and SSF integrate third-party vendor security products via standard interfaces to achieve their goals. In this manner, we will provide our customers with the most technically advanced and scientifically sound security products that are available on the market. SAP wants to support strong cryptography, authentication, and firewall mechanisms to protect the legitimate security interests of customers, but has decided not to include cryptographic modules in its own software. Instead, products can be integrated which have been developed by security professionals. This strategy ensures that R/3 remains internationally importable and exportable by providing the necessary interfaces to such third-party security products.

To be able to support a variety of different security products, the use of standardized interfaces is necessary. The use of such interfaces enables the user to install a security product of his own choice, set up a security policy according to his requirements, and use cryptographic algorithms that he finds powerful enough to protect his data. The security products can be replaced at any time to accommodate changing requirements or modifications in security software.

This strategy has a number of advantages:

- Application level, end-to-end security is provided. This has certain benefits, such as transport independence or transparent firewall traversal.
- Single Sign-On can be implemented (see description on Page 6).
- The R/3 software remains competitive in the international market.
- Each customer can use his favorite security product with well analyzed protocols and algorithms.
- Security algorithms and protocols can be changed without affecting the business applications.

In the following, we will explain in detail how both SNC and SSF work, the functions they provide, and the security gains achieved.

Secure Network Communications (SNC)

SNC Mechanisms and Functionality

Secure Network Communications (SNC) provides protection for the communication links between the distributed components of an R/3 System. In addition, SNC enables the use of cryptographic mechanisms and smartcards to securely authenticate users. This is accomplished by integrating an external security product with the R/3 System. The external product can provide state-of-the-art authentication, data integrity, and data confidentiality services for the R/3 System.

When we refer to SNC, we are referring to the interface between R/3 and products which adhere to the standardized *Generic Security Services Application Programming Interface (GSS-API) Version 2* [1;2]. This standard has been developed by the "Common Authentication Technologies" (CAT) working group of the Internet Engineering Task Force (IETF)[1].

The frontend SAP Graphical User Interface (SAPgui) is traditionally used to access the R/3 System from or across "open" TCP/IP networks where physical security of the data transmitted over the wire is impossible to maintain. Using the SNC interface, the user can integrate the features of an external security tool with his R/3 System (for example, **Single Sign-On** as shown in Fig. 1).

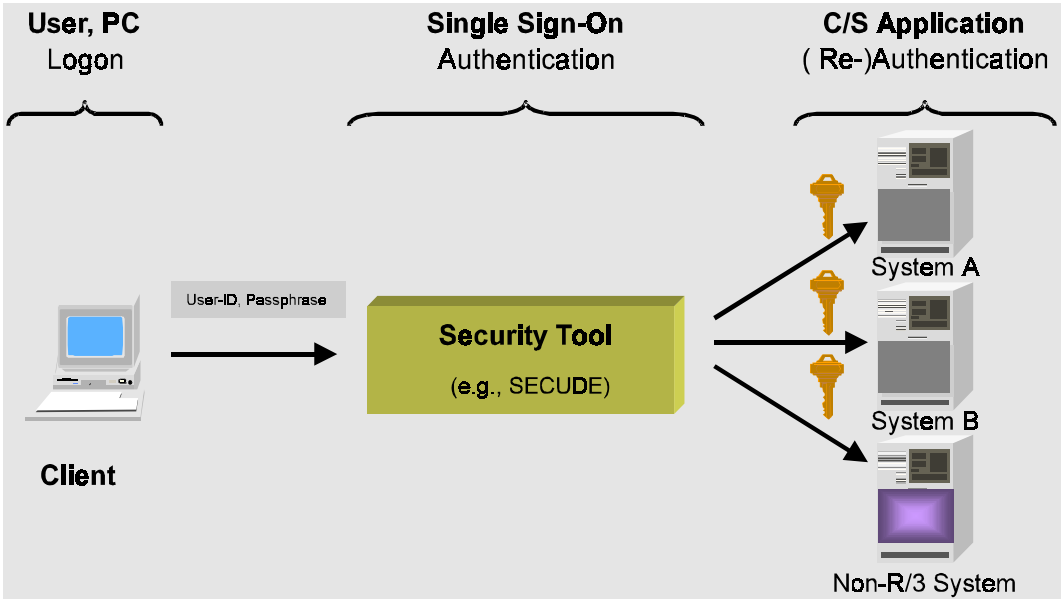

Fig. 1: Single Sign-On

[1] IETF is the standards body and open forum that defines and improves the protocols for the global Internet.

Several advantages for the customer offered by SNC are:

- Application level, end-to-end security is provided. This has certain benefits such as transport independence or transparent firewall traversal.[2]
- The use of **smartcards**[3] for authentication is supported by certain products.
- Many network security systems implement **Single Sign-On,** so that a user's initial authentication permits further automatic (re-)authentications of the user to distributed services. The ability to automatically (re-)authenticate is usually limited either to a period of time or by the presence of the smartcard in the reader.
- Each customer can use his favorite security product, and this can be replaced at any time without affecting the application.
- The transmission of passwords or passphrases[4] over untrusted networks has been eliminated.

The frontend SAP Graphical User Interface (SAPgui) and the SAP Line Printer Daemon (SAPlpd) were the first components in Release 3.1 with the new security option to use other vendors' network security products. The provisions needed to link distant R/3 Systems securely will be available in Release 4.0.

An interface to link and synchronize the user database of a network security system with the master user records within R/3 is under development. One part of the link will contain product specific code; however, due to the lack of a common standard, it may not be immediately available when a security product passes the certification procedure (See SNC Interface Certification on Page 9).

---

[2] The benefits obtained are dependent on the capabilities of the security product implemented.

[3] A smartcard looks like a credit card and contains a small cryptographic microprocessor and a small amount of memory. A part of that memory is "Write-Only" and is used to hold and safely guard the secret key of the card owner. Only the microprocessor on board can access the secret key and create mathematical proofs for the knowledge of this key. The smartcard uses the self-created mathematical proofs to confirm the authentication, without revealing the key. In this way, the smartcard is non-replicable. A special hardware device called a "smartcard reader" is needed to link the smartcard with the computer. Most smartcards perform an additional authentication check by requiring the user to supply a Personal Identification Number (PIN) each time they are inserted into a smartcard reader.

[4] Passphrases are becoming more and more popular. They are similar to passwords; however, they consist of complete phrases, allowing for more possible combinations.

SNC Secured Communications and Supported Platforms

SNC Secured Networks

SNC provides network communication security between the R/3 clients and application servers. The SNC-protected clients and application servers can exist in either a LAN or WAN, but to provide a higher level of security, the application servers and database servers should exist in a single secured LAN. This configuration is shown in Fig. 2.

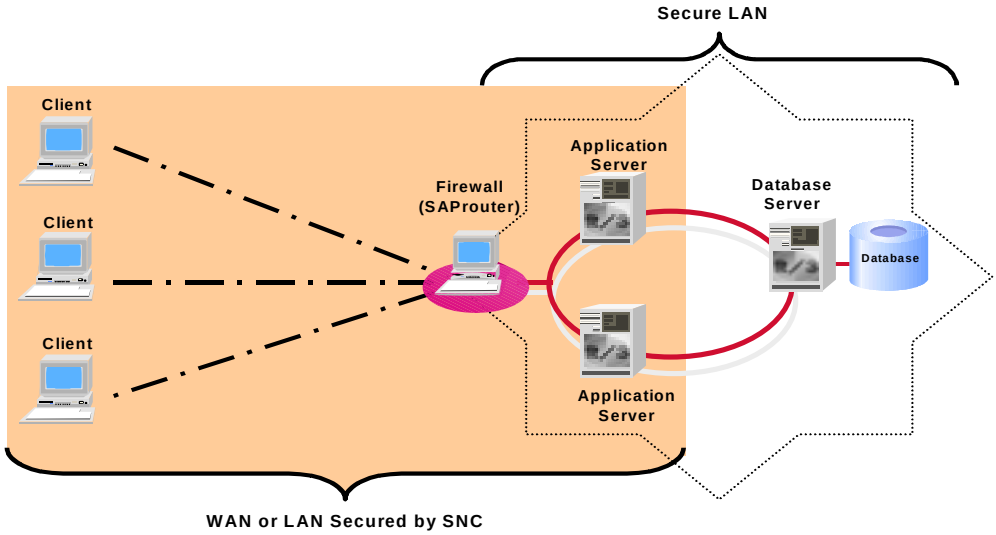

Fig. 2: SNC Secured Communications

Supported Platforms

Secure Network Communication is available on the following R/3 Release 3.1 platforms:

Application Server	Client
❑ UNIX	❑ Unix/Motif
❍ DEC	❑ OS/2 Presentation Manager
❍ HP	❑ Microsoft Windows 95
❍ IBM	❑ Microsoft Windows NT
❍ SNI	❑ Apple Macintosh
❍ Sun	
❑ IBM AS/400	
❑ Microsoft Windows NT	

SNC Interface Certification

SAP is currently working on validation procedures to certify interoperability and support of third-party network security products with R/3. A part of the validation procedure, which is based on the functionality of *GSS-API Version 2*, will include platform-specific interface details for binary compatibility at the shared library level. The technical validation of the third-party product in reference to the GSS-API functionality needed by R/3 will be accomplished with a testframe that verifies correctness, robustness, and inter-platform compatibility. The testframe will be available at the end of the first quarter in 1998 (as part of the certification process).

SNC Standards – GSS-API Version 2

Products for establishing company-wide network security have been available for quite some time. However, their installation *alone* does not affect the (in)security of traditional applications. The "native" APIs of existing network security products are different in several respects. Some APIs impose their own communication protocols, inflict architectural characteristics on the application, or require the application to deal with characteristics of the underlying cryptographic algorithms. Some APIs require an irreversible marriage of code and tools between the application and the network security product.

The leading standardization today is the *Generic Security Service Application Program Interface (GSS-API) Version 2*. It was developed by the CAT working group from the IETF to minimize the changes to and impact on existing applications when trying to secure its network communication. Because GSS-API provides abstract services, independent of the underlying technology such as cryptographic algorithms, it can be supported by a wide range of products. Additionally, GSS-API **does not** impose a transport protocol.[5] It only needs a few additional message exchanges over an arbitrary communication protocol of the application's choice to perform the authentication and enable the message protection services. Consequently, backwards compatibility and smooth migration is an implementation decision of the application writer and a security policy decision of the consumer.

In addition to the authentication and message protection services of base GSS-API, certain products offer **extensions** for delegation of credentials and central management of authorization/privilege information. A common standard for these extensions with significant acceptance is not yet apparent, but there is considerable interest in developing such a standard.

---

[5] Although the GSS-API specification is completely transport independent, several products do exhibit restrictions. All Kerberos-derived network security systems perform under-cover communication to the Key Distribution Center (KDC), which currently requires direct IP connectivity — no matter what transport protocol the application uses. Not all network security products with GSS-API will work across (multiple) firewalls or non-TCP/IP communication channels, even when the application does support them.

However, because currently available implementations of GSS-API extensions can neither handle the possible size nor the level of detail of the existing R/3 authorization model, R/3 does not support any GSS-API extensions at this time. When a standard emerges in this area, a more detailed analysis will determine which level of integration might eventually be accomplished in future releases of R/3.

Other than the base GSS-API functionality, these extensions do not affect the (network) communication; instead, they largely affect how the application logic works, that is, how services are provided, how pieces of information are accessed, and how they are processed. If existing distributed applications use a different authorization model and delegation-like functionality, the necessary changes to adhere to any such GSS-API extensions may amount to a complete rewrite of the application from scratch.

Availability of Security Products

The following list shows the products available on the market that support *GSS-API Version 2* and that are supported by R/3:

Product	Vendor / Developer
SECUDE 5.0	GMD[6] GmbH, SECUDE GmbH
Kerberos 5	MIT[7], OpenVision (Veritas), Cygnus

The implementation in R/3 Release 3.1 only interoperates with MIT's Kerberos 5 [3] and GMD's SECUDE [4]. Cooperation projects with Kerberos and SECUDE went live at:

- MIT in September 1996 (with Kerberos 5), and
- Volkswagen AG in March 1997 (with SECUDE 5.0).

Pending certification, **SECUDE** is the only supported commercial product *currently* available to SAP customers (MIT's Kerberos is not a commercial product). SECUDE is from a German company and therefore can provide strong cryptography both within and outside of the United States and Canada.[8]

Additional products provide an interface based on *GSS-API Version 1* with the indication to implement *GSS-API Version 2* as soon as the specification enters the standards track.[9]

---

[6] German National Research Center for Information Technology
[7] Massachusetts Institute of Technology
[8] Non-American products are **not** subject to strict American crypto export regulations. (More information concerning American crypto regulations can be found at the following WEB site: http://www.eff.org/pub/Privacy/ITAR_export.)
[9] For example: OSF DCE based products, SESAME 4 based products, Entrust.

Features of the above mentioned products:

- Secure authentication of users / programs / resources
- Smartcard-based authentication as supported by certain products (e.g., SECUDE)
- Single Sign-On for distributed applications and varying levels of integration with the logon procedure of the host operating system
- Cryptographic integrity protection of data
- Moderate to strong cryptographic confidentiality protection of data, depending on the crypto export and usage regulations of the vendors' or customers' home countries
  (Exception: International versions of OSF-DCE 1.1 as shipped by several vendors do not provide confidentiality services).

Secure Communication between the Customer and SAP

With the implementation of the SAProuter with a Firewall, SAP has improved the security of the network communication between the customer and SAP.[10] Communication data between customers and SAP is directed over the SAProuter, enabling controlled access to the LAN behind the Firewall.

With the official IP-Addresses from the SAProuters, the customer can use "open" addresses that are available from the network for his own purposes.

The *access controls* are maintained in *access control lists.* Here, the customer can maintain his own authority tables and distribute passwords. SAP cannot access these tables without permission from the customer. Normally, sessions are only established from the customer, providing for a "one-way street." The customer needs to grant permission only for departing connections and can deny all incoming requests.

Additionally, the SAProuter protects frontends that are connected to application servers on WANs. Here, the frontends communicate via their local SAProuter with the SAProuter that is connected to the network of the application server. The messages are then sent to the application server. The same process, in the reverse order, applies to messages sent from the application server to the frontends.

The most recent SAProuter can use the features of SNC to establish secure communication tunnels across untrusted networks. It is being used by a few customers, and will be generally available with Release 4.0. The new SAProuter is backwards compatible to earlier releases of R/3.

---

[10] A SAProuter is an application-level process that forwards and controls the data stream (SAPGUI, Remote Function Call, ...) transferred between IP sub-networks or within a single application-level sub-network. The SAProuter is located "above" the TCP and is included with every R/3 System since R/3 Release 2.2. Processes such as the SAProuter are also referred to as **Application Level Gateways**.

In this way, SAP can ensure a secure communication channel for remote support between SAP and its customers.

Secure Store & Forward Mechanisms (SSF)

SSF Mechanisms and Functionality

For today's business application software, it is increasingly important to support electronic authentication mechanisms and secure electronic transactions over public data communication networks. In the course of such electronic transactions, business data, such as electronic payments, or order and account information, leaves the secured realm of an R/3 System to be transmitted over insecure networks or to be stored on data carriers or archives.

The security requirements in the context of electronic business transactions are inherently different from the requirements for securing online communication between system components. The transmission of business data is done by computer systems and software processes, but the actual business transactions are carried out between persons or other subjects with a legal meaning. Therefore, additional security mechanisms at **application level** are required.

The support provided by "Secure Store & Forward (SSF)" enables the protection of R/3 data and documents when saved on external data carriers and when transmitted over possibly insecure communication paths, such as the Internet.[11] To facilitate this, digital signatures and encryption are utilized at the application level. In the process, the data is secured as independent data units, regardless of the type of its contents, and regardless of the selected transport procedure. The protection mechanisms provided with SSF are applied outside the context of an existing online connection.

With the SSF functions, R/3 data and documents are "wrapped" in secure formats - the so-called "security wrapper" - before they are saved on data carriers or transmitted via (insecure) communication links. A **digital signature** ensures that the data is not falsified, that the sender (signatory) can be clearly determined, and that proof of award of contract exists. The subsequently assigned **digital envelope** ensures that the contents of the data are only visible to the intended recipient(s). As a result, no security gaps arise, even if the data is temporarily stored during transport or at its destination.

To construct a digital signature for a given document or message (see Fig. 3), a hash function is applied, which delivers a so-called "message digest". The "message digest" represents an unambiguous fingerprint for the message, but is usually much shorter. If a cryptographic hash function is used, it should be impossible to compute another meaningful input message which will produce the same digest. The message digest is then transformed into a

[11] SSF provides security for the data contents. This is independent of any security mechanisms which may be applied to the transport medium.

signed message digest using the signer's private key. Anybody who has access to the corresponding public key of the signer can reverse the transformation and retrieve the message digest from the signed message digest. To verify the authenticity of the signature, and the integrity of the data, the same hash function is applied to the data and the result is compared with the message digest.

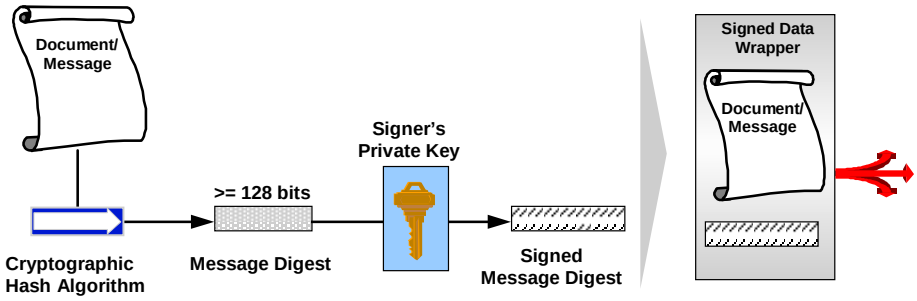

Fig. 3: Digital Signature

To wrap a document or message to be protected in a digital envelope (see Fig. 4), the message is first encrypted, so that only the intended recipient(s) are able to decrypt the message. Typically, the data is DES[12] encrypted using a newly generated DES key (message key). Then, the message key is encrypted using the recipient's public key. Only the owner of the corresponding private key, i.e., the intended recipient, is able to decrypt the message key and then to decrypt the data contained in the digital envelope.

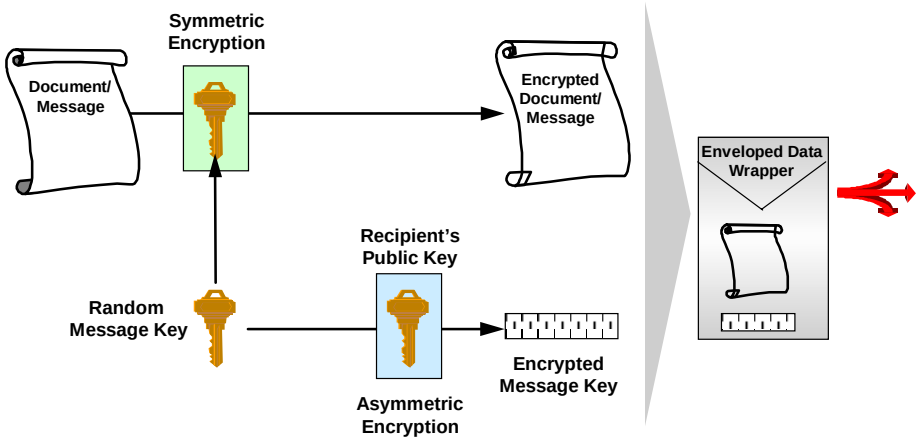

Fig. 4: Digital Envelope

[12] Data Encryption Standard - symmetric encryption algorithm

Protecting R/3 data and documents with SSF fulfills the following basic security requirements:

- **Integrity** of data (protection from falsification)
- **Confidentiality** of data (protection from unauthorized viewing)
- **Authenticity** of the sender (protection from counterfeiters)
- **Non-repudiation** (proof of award of contract)

In addition, the following SSF properties are also extremely relevant for electronic transactions:

- SSF is asynchronous - i.e., the creation, transmission, receipt, processing, and confirmation of business transactions are separate steps that can happen at different points in time without blocking the processing applications.
- Independence from the transport method - various transport media and procedures are possible (such as a public network, online service, Internet, tapes, disks), as well as various communication services and protocols (such as HTTP, FTP, e-mail, EDI).

These properties are retained even after the data transmission is complete, as long as the data is saved in the secure format.

SSF Architecture and Supported Platforms

The software architecture and call sequence during actual use of the SSF functions are illustrated in Fig. 5. The R/3 applications (FI, SD, etc.)[13] access the SSF functions using various ABAP[14] function modules provided for the "SSF Call Interface" by the Basis software.

The security aspects (such as digital signature, encryption) of the data are passed on to the corresponding SSF ABAP function module, which in turn calls the appropriate "C" functions via an Application Programming Interface defined by SAP (SSF API). To enable use of the SSF API functions, a function library that provides the "C" functions integrated through SSF API is dynamically loaded at runtime.

The implementation through the "C" functions specified by the SSF API establish the connection to the security product ("security toolkit"). In the process, the specific API functions of the respective security product are called. After the return from the RFC, the secure data is passed from the SSF ABAP function module back to the application.

---

[13] Financials, Sales & Distribution, ... (business application modules of the R/3 System)

[14] Advanced Business Application Programming (the programming language for R/3 applications)

This entire process is indicated with ”Sign/Envelope” in Fig. 5. To verify secure data and make it readable again, the reverse process is applied (”Develope/Verify” in Fig. 5).

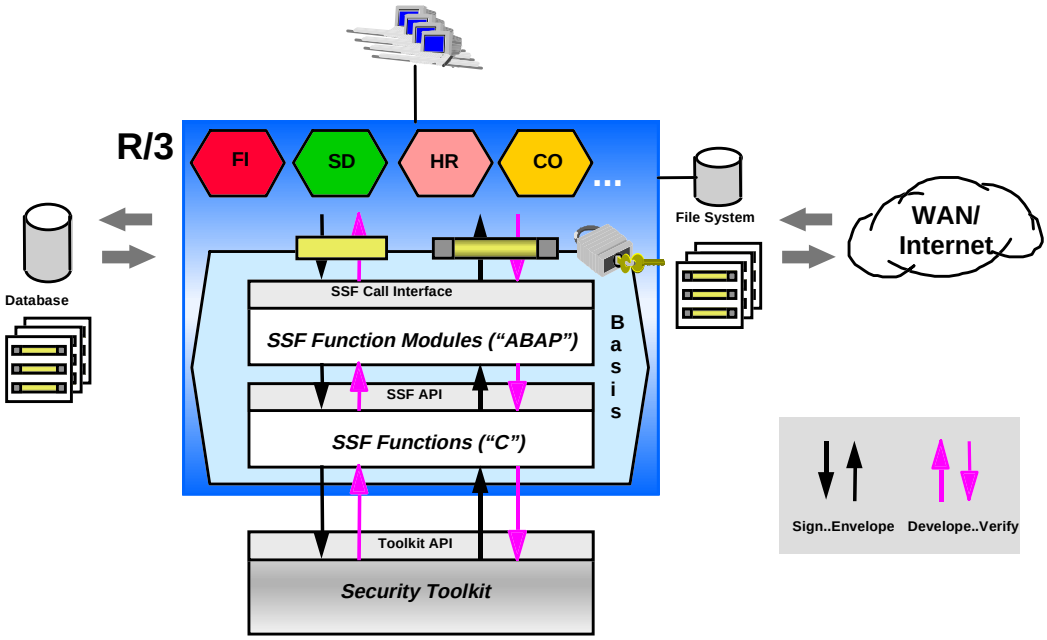

Fig. 5: SSF Architecture

Initially, SSF support for R/3 is provided for all common Unix operating system platforms, Windows NT servers, as well as for Windows NT and Windows 95 frontends. Additional platforms will follow as needed.

SSF API Certification

The function library included in the implementation of the SSF API can come from various security product vendors, which means that many different security products can be used. To facilitate the implementation of the SSF API by security product vendors and integration with R/3, SAP plans to publish these SSF specifications as soon as they are available.

However, to guarantee proper interaction with the R/3 System, the security product must be certified by SAP for use with SSF. Currently, the only security product that can be used with SSF in R/3 Release 4.0A is SECUDE [4].

SSF Standards - PKCS#7

The format used for signed and/or encrypted data with SSF is PKCS#7 [5]. The use of SSF functions applies X.509 [6] as the standard for ”Public Key” certificates. These standards form a foundation that is currently the most widespread worldwide, while still maintaining maximum flexibility for future enhancements.

Application Scenarios

The SSF functions can be applied in various scenarios for protecting data and documents:

1. Clear-text data entered in the SAPgui is immediately transposed to the secure format by the application, and then saved in the R/3 database in that secure format. When the data is required again, it is read from the R/3 database and then verified and/or decrypted by the SSF functions before the actual use.

2. If necessary, the application can also save the data entered in the SAPgui directly in the file system in the secure format. When the data is required again, it is read from the file system and then verified and/or decrypted by the SSF functions before the actual use.

3. The data from an R/3 transaction is initially stored in the various tables of the R/3 System. In further transactions in the SAP processing, the data is then read from the R/3 database, a digital signature and/or encryption is applied, and the data is saved again in the R/3 database in the secure format.

4. Data is read from the R/3 database and prepared for external storage and/or transport/transmission. To do this, the data is initially transformed into the required external format and then secured with the SSF functions. Once the data is available in the secure format, it can be safely saved to disk or transmitted through insecure communication links, such as the Internet. The intended recipient can be another R/3 System, or a different system that supports the secured format used.

5. Data is received in secure format from the Internet (or through another method) and imported into the R/3 System. Note that the secure data does not necessarily need to have been generated with R/3 - it only needs to be available in the secure format used (PKCS#7). After decryption with the SSF functions, the data is available in clear text. In addition, the digital signature is verified as necessary.

As of Release 4.0, the SSF functions, following one or several of the above scenarios, are available for use by the R/3 applications. The application of SSF may also include the usage of chip card readers and smartcards as supported by the certified security products.

References

Generic Security Service API (GSS-API) Version 2	
[1] GSS-API Technical Information RFC 2078	ftp://ds1.internic.net/rfc/rfc2078.txt *
[2] C-Bindings for GSS-API v2 (working document)	ftp://ds1.internic.net/internet-drafts/draft-ietf-cat-gssv2-cbind-XX.txt **
Kerberos 5	
[3]	http://web.mit.edu/ http://web.mit.edu/kerberos/www/krb5-1.0/announce.html http://web.mit.edu/tytso/www/resume.html http://web.mit.edu/aellwood/www/thesis/areaexam.html
SECUDE 5.0	
[4]	http://www.secude.com http://www.darmstadt.gmd.de/secude/ http://www.darmstadt.gmd.de/TKT/security/commercial/ http://www.darmstadt.gmd.de/secude/Doc
SSF Standards	
[5] PKCS #7	RSA Laboratories, ”PKCS #7: Cryptographic Message Syntax Standard”, November 1993
[6] X.509 (”Public Key” Cerificates)	ITU Recommendation X.509, ”The Directory – Authentication Framework”, 1993
SAP R/3 Notes	
[7] OSS Note 66687	”Network security products Secude and Kerberos”

*An update with several changes is pending for RFC 2078, based on the discussion concerning C-Bindings. It contains clarifications and removes certain calls due to potential problems for layered multimechanisms and binary compatibility.

**XX=04 as of 14-Jul-97