On the development of a security toolkit for open networks
- New security features in SECUDE

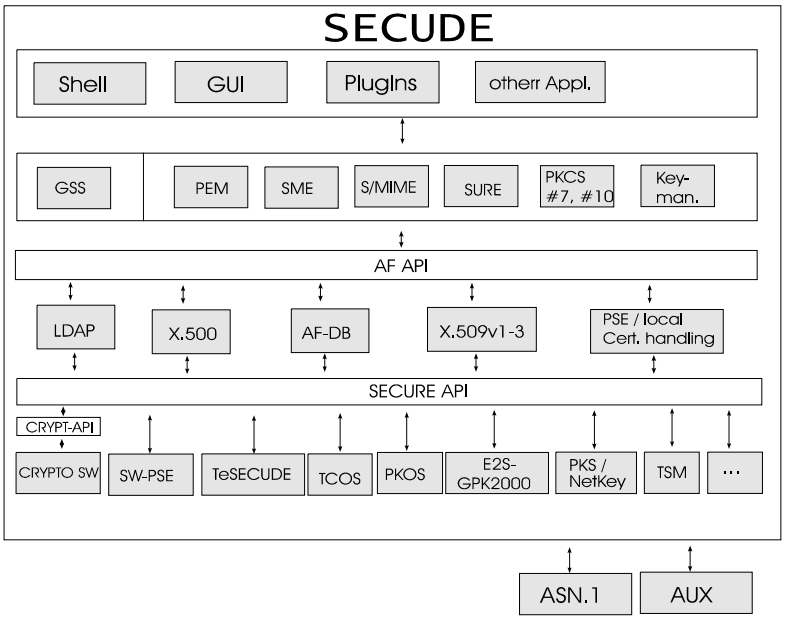
D. Hühnlein
secunet Security Networks GmbH
Schlosserstrasse 23, 60322 Frankfurt am Main
U. Faltin, P. Glöckner, U. Viebeg, A. Berger, Giehl, S. Kolletzki, T. Surkau
GMD - TKT.SIT Security Technology
Dolivostr. 15, 64293 Darmstadt

ABSTRACT: In this article we will discuss the requirements of security toolkits for open networks, explain some important technical details and give a perspective on modern security technology. To illustrate these issues we will focus on the current and future development of SECUDE. We will give a brief overview of the SECUDE **[16]** structure, emphasize the latest developments and new security APIs, such as improvements in the CRYPT-API, the integration of new smartcards, the Directory access via **LDAP**, the support of **X.509v3** certificates and new security features like **GSSv2**, **PKCS#7,10**, **S/MIME, BAKO** and **SURE.**

1 Introduction

Since global and local network-services are increasingly being used by the general public there is a strong demand for authentic, confidential and non-repudiable communication. These key issues of open telecooperation can be achieved through cryptographic primitives, like *encryption* and *digital signatures*. The trustworthiness of these mechanisms rest in the public control of the algorithms, rather than in hiding the design principles. Therefore SECUDE provides a variety of well known and intensively studied crypto-algorithms as security basis for higher level applications. Furthermore there is a need for the secure storage of private keys. SECUDE provides two possible solutions of a **P**ersonal **S**ecurity **E**nvironment (PSE) for this purpose; an encrypted directory (SW-PSE) and interfaces to a variety of smart-cards (SC-PSE). It is preferable to have a technology-independent interface used by higher level applications, because the security of the underlying algorithms and the chipcard-technology is subject to change due to further research. SECUDE contains the SECURE API, which provides access to the secure processing and secure storage module. Another important API is needed to achieve authentic communication. The functions for the management of public-keys are accessible through the **A**uthentification **F**ramework API. On top of these basic modules there are higher level APIs such as e.g. PEM, PKCS, GSS and S/MIME. Finally these functions are utilized in security-plugins for existing software products, like SAP/R3 or MS-Exchange, just to name the prominent ones.

These aforementioned issues lead canonically to the structure of SECUDE as shown in the following figure.



2 New Security-Features in SECUDE

The functionality of these APIs is addressed in the following, where we focus on the latest developments, new features and give perspectives for further research.

2.1 CRYPT API

This API provides multi-precision integer arithmetic, modulo arithmetic, random number generation and implementations of symmetric and asymmetric cryptoalgorithms.

The available symmetric algorithms[1] are DES, Triple DES and IDEA. The public key algorithms contained in the CRYPT API are the Diffie Hellman Key Agreement, the NIST DSS and the RSA algorithm. Furthermore there is a need for cryptographically strong hash functions for signature generation. SECUDE provides the hash-functions MD2, MD4, MD5, SHA-0, SHA-1 and RIPE-MD160 **[14]**. MD2 and MD4 are totally unsuitable for signature generation, MD5 is suspected to be broken soon (c.f. **[2]**, **[3]**) and SHA-0 bears some weaknesses in the expand-function. Therefore we recommend SHA-1 and RIPE-MD160 as the most secure hash functions for signature generation. The other hash-functions are left in SECUDE, to keep compatibility with standards and yesterdays signatures.

[1] A description and further references for all crypto-algorithms (except RIPE MD160) may be found in **[15].**

2.2 SECURE - API

Like mentioned in the introduction, SECUDE provides a technology independent SECURE - API, which connects the CRYPT-API and the PSE-handling to the higher-level API's. SECUDE supports two PSE-types. That is an encrypted directory (SW-PSE) and interfaces to different smartcards (SC-PSE's). Both PSE-types are PIN-protected. The used PSE- and smartcard-type(s) may be selected during the configuration process. Only the desired SC-interface is linked dynamically.

At the time of writing SECUDE supports the STARCOS and the TCOS smartcard systems. Interfaces to the GEMPLUS smartcard GPK2000 and the G&D PKOS smartcard are in the development phase.

2.3 Authentication Framework API

The AF module adds X.509 certification functionality to SECUDE. Former SECUDE versions supported the X.509 version 1 certificates, while the next SECUDE release will contain the X.509 version 3 certificates.

Both local (i.e. PSE-located) certificates and Directory-located certificates can be addressed. Obtaining public security information, like public keys, certificates, crosscertificates and certificate revocation lists used to be done using the X.500 Directory Access Protocol (DAP). SECUDE 5 now uses the Lightweight Directory Access Protocol (LDAP) instead of DAP to retrieve security information from Directory servers.

2.3.1 X.509 version 3 certificates

The experience gained in attempts to deploy X.509 v1 certificates made it clear that the v1 and v2[2] certificate formats are deficient and too restrictive. With X.509 v3, which is standardized in **[20]**, most of the requirements of RFC 1422 **[10]** can be addressed using certificate extensions, without a need to restrict the CA structures used. In particular, the certificate extensions relating to certificate policies obviate the need for **P**olicy **C**ertification **A**uthorities and the constraint extensions obviate the need for the name subordination rule, because the certificate contains information (***basic constraints*** - field) to distinguish between user- and CA-certificates. The certificate may contain ***alternative names***, like mail-adresses or URLs for the issuer (CA) and the subject (user). The distribution and retrieval of **C**ertificate **R**evocation **L**ists is made easier by storing ***CRL distribution points*** in the certificate and the ***Key usage*** may be restricted. Besides this standard-extension, which are discussed in **[20]** more detailed, it is possible to use ***private extensions*** for application specific needs.

2.3.2 The Lightweight Directory Access Protocol

The Lightweight Directory Access Protocol (LDAP **[12]**,**[13]**) was designed to overcome the problems resulting from the requirements of the X.500 DAP. While queries and answers are still encoded using ASN.1, LDAP makes restrictions on the

[2] The difference between v1 and v2 - certificates is just the presence of two more fields for Directory access control.

type and format. Another simplification is the use of string notation in most of the attributes. On the transport side TCP connections are usually used to communicate with an LDAP server, eliminating the need for an OSI protocol stack. This all leads to smaller code and more acceptance on the implementors' side.

In SECUDE there are two possibilities to access X.500 Directories. Either using the X.500 ISODE ICR2.1 library or the access via LDAP. The latter is possible, if the server uses an LDAP-to-X.500 adapter. This allowes the client software to remain "light", moving the overhead upstream to the server. LDAPv2 is supported in SECUDE's various Unix ports using the LDAP reference implementation library of the University of Michigan. In the SECUDE for Windows NT/95 version the Dynamic Link Library (DLL) of the same package is used **[9]**.

2.4 Generic Security Services API

The Network Working Group of the **I**nternet **E**ngineering **T**ask **F**orce defined in 1993 a general interface to security systems. The *Generic Security Services* (GSS) API is a set of functions and data structures to incorporate security into a program independent of the underlying security and communication protocols **[4]**, **[5]**.

There are (to our knowledge) currently three underlying security mechanisms available, which support the GSS-API. First the well known ***Kerberos V5*** using a trusted authentication-server and DES-encryption, ***S**imple **P**ublic **K**ey **M**echanism* **[6]** using X.509 certificates and a variety of algorithms for authentication and confidentiality. Finally there is the SECUDE-mechanism, which also uses X.509 certificates, but is restricted to RSA, DES and IDEA. The a priori restriction to certain algorithms removes the inherent negotiation-overhead in SPKM. The GSS-API and this three underlying mechanisms are discussed in **[8]** more detailed. Currently the SECUDE-mechanism is available[3] and SPKM is under development. Kerberos won't be supported, because public-key-mechanisms in this context are generally preferable.

2.5 PKCS API

While the family of PKCS-standards **[11]** comprises standards for RSA encryption, DH Key Agreement and other cryptographic primitives that are already available in earlier versions of SECUDE, we will focus on PKCS#7 and PKCS#10.

2.5.1 Signing and encrypting data with PKCS #7

The PKCS #7 standard describes a general syntax for data that may have cryptography applied to it, such as digital signatures and digital envelopes. The syntax admits recursion, so that, for example, one envelope can be nested inside another, or one party can sign some previously enveloped digital data. It also allows arbitrary attributes, such as signing time, to be authenticated along with the content of a message, and provides for other attributes such as countersignatures to be associated with a signature. A degenerate case of the syntax provides means for disseminating certificates and certificate-revocation lists. The PKCS #7 API of SECUDE consists of

[3] E.g. it is used to secure the SAP / R3 application program.

functions for creating, developing, and verifying such enveloped data. The types *Signed* and *Signed-and-enveloped* are interoperable with Privacy Enhanced Mail **[10]**. SECUDE includes conversion functions for these types.

2.5.2 Certification Requesting with PKCS #10

The PKCS #10 standard describes a syntax for certification requests. A certification request consists of a distinguished name, a public key, and optionally a set of attributes, collectively signed by the entity requesting certification. Certification requests are sent to a certification authority, who transforms the request to an X.509 public-key certificate. SECUDE includes several programs and functions dealing with PKCS #7 ContentInfo and PKCS #10 CertificationRequests format.

2.5.3 Signing and encrypting multi media messages with S/MIME

The S/MIME Message Specification **[17]** combines the security enhancement of PEM and the multi-purpose content-types of MIME. It defines the MIME content types application/x-pkcs7-mime and application/x-pkcs7-signature for cryptographically enhanced MIME bodies according to PKCS #7 and application/x-pkcs10 for submitting a certification request. SECUDE provides API functions for producing and parsing such S/MIME messages. The main advantage compared to PGP is the scaleable certification infrastructure. I.e. it is possible to use either a hierarchical certification infrastructure combined with cross-certificates or a network of trust like in PGP.

3 Applications

3.1 The European ICE-TEL Project - TrustFactory Digital ID Center

Within the European research project '**I**nterworking Public Key **C**ertification Infrastructure for **E**urope' (ICE-TEL) more than 17 countries are working together to establish a common certification infrastructure not only for the R&D community, but also for interested partners in the governments, administrations and industry.

In this context GMD is running a **C**ertification **A**uthority; the 'TrustFactory Digital ID Center' as a European counterpart to the US american 'VeriSign Digital ID Center'.

3.1.1 Privacy Enhanced Mail Plugin for Microsoft Exchange

Another add-on application is the PEM plugin for MS Exchange where outgoing e-mail may be signed and encrypted and incoming messages may be decrypted and validated. The plugin provides full attachment support and the german MailTrusT PEM Specification **[19]**. This includes correct processing of raw binary data.

3.1.2 Privacy Enhanced Mail Shell Extension for Microsoft Windows Explorer

The same PEM functionality as described in (3.1.2) is avaiable as a shell extension for the new file manager of MS Windows, the Windows Explorer. Local files may be signed and encrypted or decrypted and validated.

3.2 Security Add-on for SAP R/3

In cooperation with SAP an interface for the R/3 Client/Server system was developed, which makes the use of security technology via GSS-API possible.

3.3 BAKO with SURE extension

3.3.1 Basic cooperation protocol

Another application of the SECUDE toolkit is BAKO **[1]** - a basic cooperation protocol for secure business transactions over open networks. In contrast to host-to-host session- or packet-based security mechanisms, BAKO can be applied where complete transactions need to be authentic and non-repudiable, and where ***documents*** need to be produced that are integer and confidential, like bank-orders for example. There will be BAKO - plugin for the WWW available to achieve secure web-transactions.

3.3.2 Signed Unique References - a BAKO extension

The original BAKO has two unsolved issues: minimizing the danger of replay attacks, and minimizing the network load. Even if the information is security-enhanced, a single or multiple resending of protocol units may cause trouble on both sides if the session becomes insecure. The second issue is caused by BAKO's nesting of complete transaction steps. If large objects, e.g. images, are to be transported, a high information overhead occurs: objects are sent twice or even three times. A proposed BAKO extension specifies the replacement of objects with signed and unique references ('SURE') **[18]** to the object as soon it has been transmitted or received. The reference additionally contains a timestamp consisting of a negotiated time base and the protocol data unit's time-to-live.

4 Conclusion

In this paper we briefly discussed the requirements of a security toolkit for open networks, gave an overview of the security features available in SECUDE, illustrated some current and future developments and finally discussed a few secure end user-applications. SECUDE is designed to be a portable **SECU**rity **D**evelopment **E**nvironment and therefore well suited for security-plugins to any kind of applications. We will continue to upgrade SECUDE with upcoming new standards, crypto-algorithms, smartcard systems, and application requirements. An open research area, for instance, is the design of Public Key Infrastructures (PKIs) for various purposes, including cross-certification between different PKIs, and the interface between end user applications and the PKI. The SECUDE development will reflect the current standardization process in the IETF.

References:

[1] S. Kolletzki: „Secure Internet Banking with Privacy Enhanced Mail“, Computer Networks and ISDN Systems 28 (1996) 1891-1899

[2] H. Dobbertin: „Welche Hash-Funktionen sind für digitale Signaturen geeignet?“, Tagungsband „Digitale Signaturen“, Vieweg-Verlag, 1996, ISBN 3-528-05548-0, pp. 81-92

[3] H. Dobbertin: „Digitale Fingerabdrücke - Sichere Hashfunktionen für digitale Signaturen“, DuD 2/97, Vieweg, pp. 82-87, 1997

[4] J. Linn: „GSS API“ RFC's 1508 and 1509 (C-bindings), Sep. 93

[5] J. Linn: „The GSS API Version 2“ RFC 2078, Jan 97

[6] C. Adams: „The Simple Public-Key GSS-API Mechanism (SPKM)“, RFC 2025, Jan 96

[7] J. Kohl, C. Neumann: "The Kerberos Network Authentication Service (V5)", RFC 1510, Sep. 1993
J. Linn: " The Kerberos Version 5 GSS-API Mechanism ", RFC 1964, Juni 1996

[8] D. Hühnlein: "Generische Sicherheit - Die GSS-API und drei ihrer Mechanismen", to appear in FIFF-communication, 3/97

[9] University of Michigan Information Technology Division: „LDAP servers, client library and sample text based UNIX clients“
ftp://terminator.rs.itd.umich.edu/x500/ldap/ldap-3.3.tar.Z
„Windows Binary Distribution (contains LDAP32.DLL, LIB and header files)“
ftp://terminator.rs.itd.umich.edu/x500/ldap/windows

[10] J. Linn: „Message Encryption and Authent. Procedures“ RFC 1421, Feb 93
S. Kent: „Certificate Based Key Management“ RFC 1422, Feb 93
D. Balenson: „Algorithms modes and identifiers“ RFC 1423, Feb 93
B. Kaliski: „Key Certification and related Services“ RFC 1424, Feb 93

[11] RSA: „PKCS#1-#11: Public Key Cryptography Standards“, http://www.rsa.com, revised Nov. 1993

[12] M. Wahl, T. Howes, S.Killie: „Lightweight Directory Access Protocol (v3)“, 10/1996
ftp://ds.internic.net/internet-drafts/draft-ietf-asid-ldapv3-protocol-03.txt

[13] W. Yeong, T. Howes, S. Killie: „CURRENT LDAP Version2“, March1995
ftp://ds.internic.net/rfc/rfc1777.txt

[14] H. Dobbertin, A. Bosselaers, B. Preneel: „RIPEMD-160: A strengthened version of RIPEMD“, Fast Software Encryption, Cambridge Workshop, LNCS 1039, Springer, 1996, pp. 53-69 , corrected version via
ftp://esat.kuleuven.ac.be/pub/COSIC/bosselae/ripemd/

[15] B. Schneier: "Applied Cryptography - Protocols, Algorithms and Source Code in C", John Wiley & Sons, New York, 1994, ISBN 0-471-59756-2

[16] GMD: „SECUDE 5.0 - Hyperlink Documentation“, 1996,
http:www.darmstadt.gmd.de/secude/doc/index.htm

[17] RSA: „S/MIME Message Specification“, Feb 96, *smime-editor@rsa.com*

[18] P. Glöckner, S. Kolletzki, M. Wichert: „Signed Unique References“, to appear in the proceedings of JENC8

[19] F. Bauspieß (ed.): „MailTrusT Spezifikation, Version 1.1“, 12/96

[20] ISO/IEC JTC 1/SC 21/WG 4 and ITU-T Q15/7: „Final Text of Draft Amendment 1 to ISO/IEC 9594-8 on Certificate Extensions“, December 1996