PKI based Access Control with Attribute Certificates for Data held on Smartcards

Lutz Suhrbier, Thomas Hildmann
Technical University of Berlin
Research Center for Network and Multimedia Technology
PRZ / FSP-PV

29. May 2002

Abstract

Common smartcard systems are not capable of providing effective Data Access Control in distributed IT-infrastructures with high configuration dynamics. The crucial points of that approach are resource consumption and inflexibility. The storage capacity of actual smartcards is clearly insufficient to store large certificate databases required by distributed services and applications. The exchange of CA-certificates using a remote update mechanism on smartcards is unrealisable with conventional smartcard based public key infrastructures (PKI) because of security issues.

In one contrast, the use of attribute certificates minimizes the expenses of certificate storage and allows to implement PKI based access control mechanisms executed on the card. In other access control contexts attribute certificates were copiously discussed.

Thus the employment of smartcards in large public facilities becomes feasible even considering the constraints of Data Protection Acts. Attribute certificates allow flexible and configurable access on specific personal or other sensible data on smartcards by different service providers. A multiapplication smartcard platform, e.g. Java Card, permits actualisation of existent modules within a secure PKI environment. Furthermore dynamic enhancement of smartcard functionality will be possible using such mechanism.

This contribution describes a solution of the problem and reports the experiences concerning the realisation in the context of the project Campuskarte of the Technical University Berlin.

Contents

1 State of the art

Common smartcard systems are using smartcards to store information (e.g. ordinary phone or health insurance cards), as access control tokens, corporate identity card or to provide digital signature functionality. Due to the nature of data (internal corporate information) or paradigm shift over the years informations stored on smartcard are not touched by Data Protection Act. Today, health insurance cards are beneath contempt because they support no access control for the sensible personal information was implemented.

To provide access control or corporate identity functionality the card holders are identified using an identification number stored on the card or by the application of symmetric or asymmetric authentication mechanism.

For the support of digital signatures smartcards are storing between one and three asymmetric key pairs. The private keys do not leave the smartcard and are used for authentication, signature and encryption. Public Keys are stored as X.509 certificates and are used to establish secure communications e.g. for a secure email application.

1.1 Issues of existing solutions

Most Smartcards do offer additional functions like saving certificates of communication partners. However, this kind of application is limited by resource problems which are preventing the usage of these smartcard systems in distributed IT-infrastructures. Depending on the smartcard's memory capacity the range of storable certificates is between five or eventually two or three dozen. A look at the address book of any professional's email-client proves that this amount of storable certificates is by far not sufficient to establish a secure channel to all communication partners.

Using smartcards in distributed IT-infrastructures for secure communications in conjunction with mutual authentication mechanisms reveals the mentioned problem because of the high number of potential servers. One solution of this problem is to build up an infrastructure for directory services. This makes it possible to get the certificates of the communication partners using such directory services via dedicated software. In this case the smartcard only needs to know the certificate of the certification authority of the corresponding public key infrastructure (PKI). Thus verifying the integrity of the certificates distributed by the directory service becomes feasible. Obviously this approach minimises the resource consumption of storing certificates.

This solution would comply with the needs for distributed IT-infrastructures of single organisations. It reduces the address book problem to the number of certification authorities belonging to the PKIs of the communication partners. The increasing importance of those systems would probably conclue in a rampant amount of PKI-providers, posing the same problem.

In public area this problem becomes more important due to the requirements of the Data Protection Act. Applications like the health insurance card are extremely sensible. Multifunctional smartcards allow one to integrate several applications on a single card. Therefore it unauthorizes third parties to read stored data and make systems vulnerable.

1.2 Newer Approaches

How can personal information stored on smartcards be secured against third parties?

The project Campuskarte of the Technical University Berlin prepares and introduces a digital identitity card for all members (e.g. students, employees of the institution). The question above represents one of the problems to be solved during the system specification phase. A main application of the project, the re-registration of students before the beginning of each semester needs write access to modify the validity data element stored on the smartcard. How to guarantee that only the application “re-registration” is able to manipulate the corresponding data elements?

Principally, no restriction regarding the potential number of supported applications requesting access to data elements stored on the Smartcard should exist. Furthermore a solution not based on the storage of an apropriate number of certificates is required.

Referring to the practice gathered in the context of the project Campuskarte [CAMPUSKARTE], this contribution provides a new solution of these problems using attribute certificates.

2 Attribute certificates

Usually, public key certificates represent public keys signed by a certification authority. These certificates are related to a subject by using additional information. Public key certificates are issued for persons, services, servers or program code. The meaning of the certificate and its elements defining the additional information are specified by the security policy of the certification authority.

Attribute certificates are used to store information like group membership, roles or rights of access in certificates [ATTRCERT], [CERTAC].

Beside common information like subject-name (distinguished name of the certificate holder), issuer name (distinguished name of the certification authority), validity etc. the standard X.509v3 supports the addition of non standard extensions.

Unknown certificate attributes are ignored by most applications. The ASN.1 coding of certificates avoids conflicts concerning the specification of attribute types. For that, the Technical University Berlin owns the internationally registrered Object Identifier (OID) 1.3.6.1.4.1.10238. Starting from this OID it is possible to organise a hierarchical structure of subordinate identifiers. This approach allows the addition of all kind of information. The interpretation of these information is performed by the application.

The combination of public keys and attribute certificates opens up a way to encrypt data elements of the data model with the public key included in the certificate. This will be done after the verification of rights of access specified in the attributes of the particular certificate. Thus, only authorised instances have access to the corresponding data elements.

Several options of relation between attributes and public key certificates are discussed [BINDING]. We are using the concept of monolithic binding. Both public keys and attributes are signed together by a single certification authority.

3 Right of access to the smartcard

A data model specifies the data types and data stored on the smartcard. The access model defines the rights of access to specific data elements of this data model. Table 1 is showing the access model of the project Campuskarte [DATENMODELL].

Data element	Visible Methods	Card issuing			University internal users	Foreign users			Card holder
		Student's administration	Staff administration	Guest administration		University consolidation	University depending users	Others	
Application Profile (AP)	cwr	cwr	cwr	cwr	r	r	r	r	r
Card ID (CID)	cr	cr	cr	cr	r	r			r
Certificate card holder – encrypt (CHE)	cwrv	r	cwr	cwr					rv
Certificate card holder – sign (CHS)	cwrv	cwr	cwr	cwr					rv
Certificate card holder – auth (CHA)	cwrv	cwr	cwr	cwr					rv
Certificate Trust Center (CTC)	cwrv	cwr	cwr	cwr					rv
Date of Expire – Begin (DEB)	cwr	cwr	cwr	cwr	r	r	r	r	r
Date of Expire – End (DEE)	cwr	cwr	cwr	cwr	r	r	r	r	r
List of DES-Keys (DES)	cw	cw	cw	cw					
Card holder identification (OM)	cr	cr	cr	cr	r	r			r
Status-group (PS)	cwr	cwr	cwr	cwr	r	r			r
Personal Identification Number (PIN)	cw	c	c	c					w
PIN Failure counter (FVZ)	c	c	c	c					
Secret Key – encrypt (SKE)	cwd		cw	cw					d
Secret Key – sign (SKS)	cs	c	c	c					s
Secret Key – auth (SKA)	cwa	cw	cw	cw					a

Table 1 Data Model – Access Control

Rows are specifying the data elements. Columns are showing possible users. In the cells the rights of access of the user to the data element are defined. "c" is read as "create", "w" means "write", "r" is "read", "v" is "verify", "s" stands for "sign", "e" means "encrypt", "d" is "decrypt" and "a" is read as "authenticate".

For example the "student's administration" is allowed to create, write and read (cwr) the "application profile (AP)". On the other hand the "card owner" is able to decrypt (d) data using his "secret key (SKE)".

4 Security level

How are rights of access mapped to the different data elements? An important aspect regarding this question is revealed by the technology dependent lack of resources on smartcards.

The representation of the rights of access (following called security level) should be compact and must be usable without increasing resource consumption due to an extensive need of program libraries on the card.

A compact representation uses a bit-mask containing one bit for each method for each data element (see also Figure 2). The verification of a bit-mask is realisable with only low effort concerning the program code to load on the smartcard.

We decided to use the certificate's subject-name to hold the security level. X.509v3 certificates are making it necessary to represent the bitmap in a distinguished name which only contains readable ASCII-characters. At the same time the transformation must be platform independent. Base64-encoding [BASE64] is often used in this context. Unfortunately this encoding can not be used because of the format of the distinguished name, since some characters appearing in base64 are used as special characters in the distinguished name encoding. Additionally the code of a base64 library functions would be too extensive to fulfil the requirements for an application on smartcards.

4.1 Security level representation

Thus, we designed our own 6-bit-coding of the security level [SECLEVEL] held in the subject-name of X.509v3 certificates which fits our needs.

The following lines are showing the coding of the security level in Extended Backus Naur form:

<SL-CODE> ::= [<PRINTABLE>]16
<PRINTABLE> ::= [0-9] | [A-Z] | [a-z] | '@' | '?'

Table 2 describes the complete code-set of the security level representation:

ASCII-char	6-Bit decimal	6-Bit binary	ASCII-decimal	ASCII-hex	ASCII-char	6-Bit decimal	6-Bit binary	ASCII-decimal	ASCII-hex
0	0	000000	48	30	U	32	100000	85	55
1	1	000001	49	31	V	33	100001	86	56
2	2	000010	50	32	W	34	100010	87	57
3	3	000011	51	33	X	35	100011	88	58
4	4	000100	52	34	Y	36	100100	89	59
5	5	000101	53	35	Z	37	100101	90	5A
6	6	000110	54	36	a	38	100110	97	61
7	7	000111	55	37	b	39	100111	98	62
8	8	001000	56	38	c	40	101000	99	63
9	9	001001	57	39	d	41	101001	100	64
?	10	001010	63	3F	e	42	101010	101	65
@	11	001011	64	40	f	43	101011	102	66
A	12	001100	65	41	g	44	101100	103	67
B	13	001101	66	42	h	45	101101	104	68
C	14	001110	67	43	i	46	101110	105	69
D	15	001111	68	44	j	47	101111	106	6A
E	16	010000	69	45	k	48	110000	107	6B
F	17	010001	70	46	l	49	110001	108	6C
G	18	010010	71	47	m	50	110010	109	6D
H	19	010011	72	48	n	51	110011	110	6E

I	20	010100	73	49	o	52	110100	111	6F
J	21	010101	74	4A	p	53	110101	112	70
K	22	010110	75	4B	q	54	110110	113	71
L	23	010111	76	4C	r	55	110111	114	72
M	24	011000	77	4D	s	56	111000	115	73
N	25	011001	78	4E	t	57	111001	116	74
O	26	011010	79	4F	u	58	111010	117	75
P	27	011011	80	50	v	59	111011	118	76
Q	28	011100	81	51	w	60	111100	119	77
R	29	011101	82	52	x	61	111101	120	78
S	30	011110	83	53	y	62	111110	121	79
T	31	011111	84	54	z	63	111111	122	7A

Table 2 Code-Table for coding security levels

The 6-bit representation of the certificates is transformed to a 8-bit representation directly on the smartcard: i.e. the 6-bit characters are concatenated and newly interpreted as 8-bit characters (see Figure 1), four 6-bit characters are transformed to three 8-bit characters using a simple concatenation.

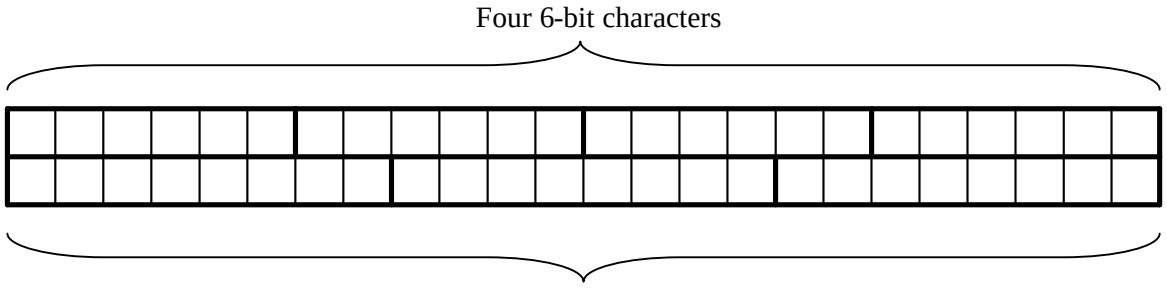

Figure 1 Coding of the security level

This 8-bit characters interpreted as a byte-array represent the security level. For our example re-registration (see Figure 2) it corresponds to the following string:

„00000001fAngkH08“

To fulfill further requirements we extended the real length of the bitmap from 50 to 96 bit by adding 46 null-bits at the beginning of the 50 defined data-bits.

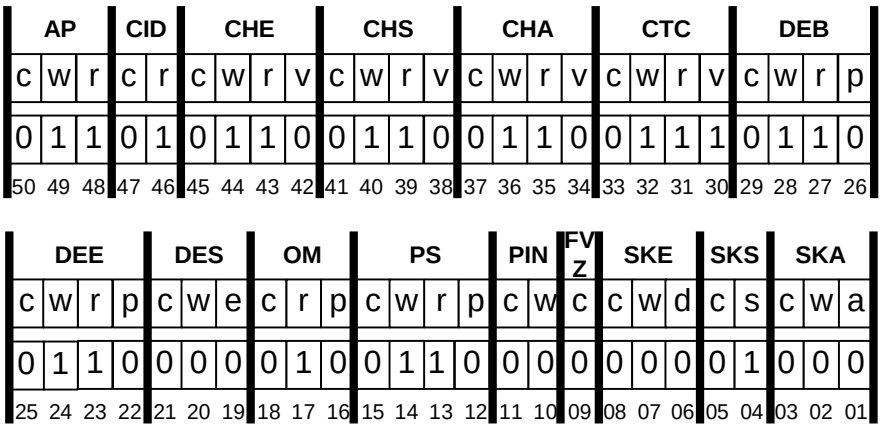

Figure 2 Security level re-registration

4.2 Integration of the security level into certificates

The current implementation extends the distinguished name (DN) of the subject of the X.509v3-certificate with an addditional element (SL: security level). This is a no standard token and will be interpreted as a comment. To be precise we are talking about attribute certificates but we do not add any attributes to the certificate; we just extended the existing attribute subject-name.

The advantage of this kind of coding is the reusability of the method of evaluation of the subject-name to also parse the security level. Another advantage (different from ASN.1 coded binary in the certificate) is that the security level is "human readable".

The subject-name is used as a key field in a directory server. If the content of the directory is queried the only field shown to the user is the subject-name. The separated attribute field can only be read by downloading the certificate from the directory service and analyse it with a special viewer.

Using our solution it is not necessary to define new attributes and therefore to adapt the policy of the certification authority.

On the other hand our representation uses 25% more storage than a binary ASN.1 coding of the certificate. The current representation is not directly human readable (e.g. *„00000001fAngkH08"*) and can only be decoded using a string parsing algorithm. Using a respective ASN.1-parser the attribute-variant would be by all means faster. However it would be more difficult to generate standard compatible certificates but it is possible for example with OpenSSL.

An attribute extension would solve the problem. In our case it would save four bytes per certificate. For this reason we decided to use the ASCII-format in the existing subject field.

5 Operating sequence : Data access

Access to the data elements of the data model stored on smartcard regarding the security level was granted following these steps:

1. Principially all requests to the smartcard are encrypted (*cryptreg*) with a session key (*skey*). This session key is the result of a successful mutual authentication between the smartcard an the backend system (the authentication mechanism is discussed in [VIS2001]). A successful authentication is required for every access to data elements of the data model on smartcard. The request (*req*) and the certificate of the accessor (*rcert*) are transmitted simultaneously and decrypted using the session key (*skey*).

   `(req, rcert) = sym_decrypt(cryptreq, skey)`

2. The certificate of the accessor (*rcert*) is verified using the certificate of the certification authority (CTC) stored on the smartcard.

   `verify_cert(rcert, CTC)`

3. The public key (*rpubkey*) from the certificate of the accessor (*rcert*) is temporarily stored for further disposition.

```
rpubkey = extract_public_key(rcert)
```

4. The signature of the request (*req*) is now verified applying the temporarily stored public key (*rpubkey*) of the accessor.

```
verify_data(req, rpubkey)
```

5. The security level (*seclevel*) is extracted out of the certificate of the accessor (*rcert*) and temporarily stored.

```
seclevel = extract_seclevel(rcert)
```

6. The request (*req*) is processed step-by-step, where every access to a data element of the data model is verified using the temporarily saved security level (*seclevel*).

```
return = execute_req(req, seclevel)
```

7. The return values of the request (*return*) are signed by the smartcard using the private authentication key (*SKA*).

```
signedreturn = sign(return, SKA)
```

8. The signed return values (*signedreturn*) are encrypted using the temporarily saved public key of the accessor (*rpubkey*).

```
cryptreturn = asym_encrypt(signedreturn, rpubkey)
```

6 Experiences and outlook

In the context of the project Campuskarte we made extensive positive experiences with the employment of attribute certificates. Within the scope of the access-issue of data elements stored on smartcards, the described solution is implemented and is working routinely without any problems within the general conditions of the different applications.

However some limitations restricting the universality of this solution showed up. These restrictions are subject of the next paragraphs.

6.1 Extension with self-contained units (universities)

The concept of the security level approach described above relies upon the assumption that all server entities with authorised access to data elements of the data model belong to a single certification authority. In the scope of Berlin universities each university can be viewed as a self-contained unit, each introducing its own organisational structures. So, e.g. the issuance of identity cards for students or employees is not subject to a central administration. This procedure was controlled by each university itself.

The result is a problem regarding e.g. the modification of the validity during re-registration concerning the control of rights of access by the smartcard. In the context of an university consolidation the access model described in chapter 3 could not identify the universitary membership of the re-registration server which desires modification of the validity data element.

In this case, all re-registration servers would have write access to the data elements DEB and DEE of the data model (see chapter 3) and all certificates would be signed by the unique university independent certification authority.

The approach to introduce one certificate authority for each university (one CTC data element each) leads to the resource consumption problem described at the top of this contribution. Another approach would be the introduction of certificate chains.

6.1.1 Approach: Certificate chain

For example, each university would build its own certification authority (CA) for smartcard access. To make the PKI comprehensive the certification authority of each university must be certified by the university-comprehensive CA. This means that the authenticy of the certificate of the CA of each university is guaranteed by the comprehensive CA.

The attribute certificates for accessing data elements on the smartcards would be signed by the CA of the corresponding university. The signature contains a copy of the certificate of its issuer.

Only the comprehensive CA certificate is stored on the smartcard. The first step is to prove the signature against the transmitted copy of the certificate of the university's CA. If this is correct the integrity of the certificate of the university's CA is verified using the certificate of the comprehensive CA stored on the smartcard.

This approach solves the given problem but implies the necessity of two certificate validations instead of only one. With nowadays smartcards this takes about 3-4 seconds. If the chain is longer the data-transfer also takes longer. However, higher transfer rates are just a question of technological progress.

6.1.2 Approach: Delegation of access rights and credentials

To decrease the cryptographically effort on the smartcard and to increase the flexibility of the access-control the following mechanism for delegation of rights is proposed.

Regarding a smartcard of any home-university, and a foreign-university of the university consolidation demanding access to data elements on the smartcard, the protocol could be as follows:

1. The foreign-university requires access to data elements.
2. The smartcard is generating a random number which should be returned signed by the home-university as a credential. This random number is sent to the foreign-university.
3. The foreign-university is sending the random number and its own smartcard request to the home-university requesting a credential.

4. The home-university is proving the permission for the access using its own security policy and agreements with the foreign-university. In case of a legitimated access the random number is signed by the home-university. The home-university's certificate with the needed security level is added and returned to the foreign-university as a credential.
5. The foreign-university is sending the credential to the smartcard.
6. The smartcard validates the random number and checks the signature made by the home-university. In this case it sends the requested data encrypted for the home-university to the foreign-university. This step is needed because the smartcard does not own the public key of the foreign-university. On the other hand it is also a correct behaviour to encrypt the result of a query for the requestor (home-university) and not a third party (foreign-university).
7. The foreign-university is sending the encrypted values to the home-university.
8. The home university decrypts the values, re-encrypts it with the public key of the foreign-university and sends it back.

This process act as deterrent at the first view. In reality this procedure is more flexible and faster then the one described in section 6.1.1 because of the outsourcing of computing-intensive processes to servers, which additionally will be able to handle dynamic events like date, time or external database-content etc. In principle the leak of date and time on the smartcard is a real problem. It makes it impossible to check the expire-date of certificates or use any kind of certificate-revocation-lists (CRL) on the smartcard itself.

6.2 Standardisation of smartcard specific attributes

Another problem is caused by the usage of different types of smartcards or of smartcards from different manufacturers. Even if only Java Card compatible smartcards are considered[JCVM211], [JCRE211], [JCAPI211], each manufacturer implements its own interpretation of standards which results in small but remarkable differences.

An approach would be the specification of smartcard specific ASN.1 object identifiers (OID) used for the security levels in certificates. Each smartcard would be initialised with its own OID. Thus, each smartcard would be able to extract and apply smartcard specific features from the security level by means of its OID.

This purpose requires standardisation of adequate OIDs.

7 Conclusions

Using combined attribute-/public key-certificates enables a flexible and effective protection of data elements on smartcards. Such an infrastructure is successfully implemented and approved at the TU-Berlin. For environments constituted by several organisations with different rights of access to the data elements two different solutions were outlined: certificate chains and delegation of access rights using credentials. For an extensive acceptance of smartcards, in particular multifunctional smartcards with different application providers mainly standards are missing. A real-time clock would make it possible to use expiration information of certificates and to implement certificate revocation lists on smartcards.

8 References

[CAMPUSKARTE]	Homepage des Projekts Campuskarte an der TU Berlin http://campuskarte.tu-berlin.de.
[DATENMODELL]	Suhrbier, Lutz: Smartcard-Datenmodell – Anforderungen, PRZ/TU Berlin, Berlin, September 2000.
[SECLEVEL]	Suhrbier, Lutz und Hamann, Klaus: Dokumentation Security Level, PRZ/TU Berlin, Berlin, Februar 2002.
[VIS2001]	Hildmann, Thomas: Vermeidung von Datenspuren bei smartcardbasierten Authentisierungssystemen, VIS 2001, Kiel, September 2001.
[BINDING]	Park, Joon S. and Sandhu, Ravi: Binding Identities and Attributes Using Digitally Sogned Certificates
[ATTRCERT]	Nykänen, Toni: Attribute Certificates in X.509
[CERTAC]	Thompson et al.: Certificate-Based Access Control for widely distributed Resources, Usenix, Washington DC, USA 1999
[BASE64]	RFC 2045: Multipurpose Internet Mail Extensions (MIME), Part One: Format of Internet Message Bodies
[JCVM211]	Sun Microsystems, Inc.: Java Card™ 2.1.1 Virtual Machine (JCVM) Specification, Revision 1.0 May 18, 2000
[JCAPI211]	Sun Microsystems, Inc.: Java Card™ 2.1.1 Application Programming Interface, Revision 1.0, May 18, 2000
[JCRE211]	Sun Microsystems, Inc.: Java Card™ 2.1.1 Runtime Environment (JCRE) Specification, Revision 1.0 May 18, 2000