Network Working Group
Request for Comments: 4683
Category: Standards Track

J. Park
J. Lee
H. Lee
KISA
S. Park
BCQRE
T. Polk
NIST

October 2006

Internet X.509 Public Key Infrastructure Subject Identification Method (SIM)

Status of This Memo

Copyright Notice

Abstract

This document defines the Subject Identification Method (SIM) for including a privacy-sensitive identifier in the subjectAltName extension of a certificate.

The SIM is an optional feature that may be used by relying parties to determine whether the subject of a particular certificate is also the person corresponding to a particular sensitive identifier.

Table of Contents

1. Introduction

A Certification Authority (CA) issues X.509 public key certificates to bind a public key to a subject. The subject is specified through one or more subject names in the "subject" or "subjectAltName" fields of a certificate. The "subject" field contains a hierarchically structured distinguished name. The "subjectAltName field" may contain an electronic mail address, IP address, or other name forms that correspond to the subject.

For each particular CA, a subject name corresponds to a unique person, device, group, or role. The CA will not knowingly issue certificates to multiple entities under the same subject name. That is, for a particular certificate issuer, all currently valid certificates asserting the same subject name(s) are bound to the same entity.

Where the subject is a person, the name that is specified in the subject field of the certificate may reflect the name of the individual and affiliated entities (e.g., their corporate affiliation). In reality, however, there are individuals or corporations that have the same or similar names. It may be difficult for a relying party (e.g., a person or application) to associate the certificate with a specific person or organization based solely on the subject name. This ambiguity presents a problem for many applications.

In some cases, applications or relying parties need to ensure that the subject of certificates issued by different CAs are in fact the same entity. This requirement may be met by including a "permanent identifier" in all certificates issued to the same subject, which is unique across multiple CAs. By comparing the "permanent identifier", the relying party may identify certificates from different CAs that are bound to the same subject. This solution is defined in [RFC 4043].

In many cases, a person's or corporation's identifier (e.g., a Social Security Number) is regarded as sensitive, private, or personal data. Such an identifier cannot simply be included as part of the subject field, since its disclosure may lead to misuse. Therefore, privacy-sensitive identifiers of this sort should not be included in certificates in plaintext form.

On the other hand, such an identifier is not actually a secret. People choose to disclose these identifiers for certain classes of transactions. For example, a person may disclose a Social Security Number to open a bank account or obtain a loan. This is typically corroborated by presenting physical credentials (e.g., a driver's license) that confirm the person's name or address.

To support such applications in an online environment, relying parties need to determine whether the subject of a particular certificate is also the person corresponding to a particular sensitive identifier. Ideally, applications would leverage the applicants' electronic credential (e.g., the X.509 public key certificate) to corroborate this identifier, but the subject field of a certificate often does not provide sufficient information.

To fulfill these demands, this specification defines the Subject Identification Method (SIM) and the Privacy-Enhanced Protected Subject Information (PEPSI) format for including a privacy sensitive identifier in a certificate. Although other solutions for binding privacy-sensitive identifiers to a certificate could be developed, the method specified in this document has especially attractive properties. This specification extends common PKI practices and

mechanisms to allow privacy-sensitive identifiers to be included in the certificate as well. The SIM mechanism also permits the subject to control exposure of the sensitive identifier; when the subject chooses to expose the sensitive identifier, relying parties can verify the binding. Specifically:

(1) A Public Key Infrastructure (PKI) depends upon a trusted third party -- the CA -- to bind one or more identities to a public key. Traditional PKI implementations bind X.501 distinguished names to the public key, but identity may also be specified in terms of RFC 822 addresses or DNS names. The SIM specification allows the same trusted third party -- the CA -- that binds a name to the public key to include a privacy-sensitive identifier in the certificate as well. Since the relying party (RP) already trusts the CA to issue certificates, it is a simple extension to cover verification and binding of a sensitive identifier as well. This binding could be established separately, by another trusted third party, but this would complicate the infrastructure.

(2) This specification leverages standard PKI extensions to achieve new functional goals with a minimum of new code. This specification encodes the sensitive identifier in the otherName field in the alternative subject name extension. Since otherName field is widely used, this solution leverages a certificate field that is often populated and processed. (For example, smart card logon implementations generally rely upon names encoded in this field.) Whereas implementations of this specification will require some SIM-specific code, an alternative format would increase cost without enhancing security. In addition, that has no impact on implementations that do not process sensitive identifiers.

(3) By explicitly binding the public key to the identifier, this specification allows the relying party to confirm the claimant’s identifier and confirm that the claimant is the subject of that identifier. That is, proof of possession of the private key confirms that the claimant is the same person whose identity was confirmed by the PKI (CA or RA, depending upon the architecture).

To achieve the same goal in a separate message (e.g., a signed and encrypted Secure MIME (S/MIME) object), the message would need to be bound to the certificate or an identity in the certificate (e.g., the X.501 distinguished name). The former solution is problematic, since certificates expire. The latter solution may cause problems if names are ever reused in the infrastructure. An explicit binding in the certificate is a simpler solution, and more reliable.

(4) This specification allows the subject of the privacy-sensitive identifier to control the distribution and level of security applied to the identifier. The identifier is only disclosed when the subject chooses to disclose it, even if the certificate is posted in a public directory. By choosing a strong password, the subject can ensure that the identifier is protected against brute force attacks. This specification permits subjects to selectively disclose an identifier where they deem it appropriate, which is consistent with common use of such identifiers.

(5) Certificates that contain a sensitive identifier may still be used to support other applications. A party that obtains a certificate containing a sensitive identifier, but to whom the subject does not choose to disclose the identifier, must perform a brute force attack to obtain the identifier. By selecting a strong hash algorithm, this attack becomes computationally infeasible. Moreover, when certificates include privacy-sensitive identifiers as described in this specification, each certificate must be attacked separately. Finally, the subjects can use this mechanism to prove they possess a certificate containing a particular type of identifier without actually disclosing it to the relying party.

This feature MUST be used only in conjunction with protocols that make use of digital signatures generated using the subject's private key.

In addition, this document defines an Encrypted PEPSI (EPEPSI) so that sensitive identifier information can be exchanged during certificate issuance processes without disclosing the identifier to an eavesdropper.

This document is organized as follows:

- Section 3 establishes security and usability requirements;
- Section 4 provides an overview of the mechanism;
- Section 5 defines syntax and generation rules; and
- Section 6 provides example use cases.

1.1. Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Symbols

The following cryptography symbols are defined in this document.

H()	Cryptographically secure hash algorithm. SHA-1 [FIPS 180-1] or a more secure hash function is required.
SII	Sensitive Identification Information (e.g., Social Security Number).
SIItype	Object Identifier that identifies the type of SII.
P	A user-chosen password.
R	The random number value generated by a Registration Authority (RA).
PEPSI	Privacy-Enhanced Protected Subject Information. Calculated from the input value P, R, SIItype, SII using two iteration of H().
E()	The encryption algorithm to encrypt the PEPSI value.
EPEPSI	Encrypted PEPSI.
D()	The decryption algorithm to decrypt the EPEPSI.

3. Requirements

3.1. Security Requirements

We make the following assumptions about the context in which SIM and PEPSI are to be employed:

- Alice, a certificate holder, with a sensitive identifier SIIa (such as her Social Security Number)
- Bob, a relying party who will require knowledge of Alice’s SIIa
- Eve, an attacker who acquires Alice’s certificate
- An RA to whom Alice must divulge her SIIa
- A CA who will issue Alice’s certificate

We wish to design SIM and PEPSI, using a password that Alice chooses, that has the following properties:

- Alice can prove her SII, SIIa to Bob.

- Eve has a large work factor to determine Alice's SIIa from Alice's certificate, even if Alice chooses a weak password, and a very large work factor if Alice chooses a good password.
- Even if Eve can determine SIIa, she has an equally hard problem to find any other SII values from any other PEPSI; that is, there is nothing she can pre-compute that helps her attack PEPSIs in other certificates, and nothing she learns from a successful attack that helps in any other attack.
- The CA does not learn Alice's SIIa except in the case where the CA needs to validate the SII passed by the RA.
- The CA can treat the SIM as an additional name form in the "subjectAltName" extension with no special processing.
- Alice cannot find another SII (SIIx), and a password (P), that will allow her to use her certificate to assert a false SII.

3.2. Usability Requirements

In addition to the security properties stated above, we have the following usability requirements:

- When SIM and PEPSI are used, any custom processing occurs at the relying party. Alice can use commercial off-the-shelf software (e.g., a standard browser) without modification in conjunction with a certificate containing a SIM value.

3.3. Solution

We define SIM as: R || PEPSI
where PEPSI = H(H(P || R || SIItype || SII))

The following steps describe construction and use of SIM:

1. Alice picks a password P, and gives P, SIItype, and SII to the RA (via a secure channel).
2. The RA validates SIItype and SII; i.e., it determines that the SII value is correctly associated with the subject and the SIItype is correct.
3. The RA generates a random value R.
4. The RA generates the SIM = (R || PEPSI) where PEPSI = H(H(P || R || SIItype || SII)).
5. The RA sends the SIM to Alice by some out-of-band means and also passes it to the CA.
6. Alice sends a certRequest to CA. The CA generates Alice's certificate including the SIM as a form of otherName from the GeneralName structure in the subjectAltName extension.
7. Alice sends Bob her Cert, as well as P, SIItype, and SII. The latter values must be communicated via a secure communication channel, to preserve their confidentiality.

8. Bob can compute PEPSI' = H(H(P || R || SIItype || SII)) and compare SIM' = R || PEPSI' to the SIM value in Alice's certificate, thereby verifying SII.

If Alice's SII value is not required by Bob (Bob already knows Alice's SII and does not require it), then steps 7 and 8 are as follows:

7. Alice sends Bob her Cert and P. P must be sent via a secure communication channel, to preserve its confidentiality.
8. Bob can compute PEPSI' = H(H(P || R || SIItype || SII)) and compare SIM' = R || PEPSI' to the value in the SIM, thereby verifying SII.

If Alice wishes to prove she is the subject of an RA-validated identifier, without disclosing her identifier to Bob, then steps 7 and 8 are as follows:

7. Alice sends the intermediate value H(P || R || SIItype || SII) and her certificate to Bob.
8. Bob can get R from the SIM in the certificate, then compute H (intermediate value) and compare it to the value in SIM, thereby verifying Alice's knowledge of P and SII.

Eve has to exhaustively search the H(P || R || SIItype || SII) space to find Alice's SII. This is a fairly hard problem even if Alice uses a poor password, because of the size of R (as specified later), and a really hard problem if Alice uses a fairly good password (see Section 8).

Even if Eve finds Alice's P and SII, or constructs a massive dictionary of P and SII values, it does not help find any other SII values, because a new R is used for each PEPSI and SIM.

4. Procedures

4.1. SII and SIItype

The user presents evidence that a particular SII has been assigned to him/her. The SIItype is an Object Identifier (OID) that defines the format and scope of the SII value. For example, in Korea, one SIItype is defined as follows:

```
-- KISA specific arc
id-KISA OBJECT IDENTIFIER ::=
  {iso(1) member-body(2) korea(410) kisa(200004)}
```

```
-- KISA specific OIDs
id-npki OBJECT IDENTIFIER ::= {id-KISA 10}
id-attribute OBJECT IDENTIFIER ::= {id-npki 1}
id-kisa-identifyData OBJECT IDENTIFIER ::= {id-attribute 1}
id-VID OBJECT IDENTIFIER ::= {id-kisa-identifyData 10}
id-SII OBJECT IDENTIFIER ::= {id-VID 1}
```

For closed communities, the SIItype value may be assigned by the CA itself, but it is still recommended that the OID be registered.

4.2. User Chosen Password

The user selects a password as one of the input values for computing the SIM. The strength of the password is critical to protection of the user's SII, in the following sense. If an attacker has a candidate SII value, and wants to determine whether the SIM value in a specific subject certificate, P is the only protection for the SIM. The user should be encouraged to select passwords that will be difficult to be guessed, and long enough to protect against brute force attacks.

Implementations of this specification MUST permit a user to select passwords of up to 28 characters. RAs SHOULD implement password filter rules to prevent user selection of trivial passwords. See [FIPS 112] and [FIPS 180-1] for security criteria for passwords and an automated password generator algorithm that randomly creates simple pronounceable syllables as passwords.

4.3. Random Number Generation

The RA generates a random number, R. A new R MUST be generated for each SIM. The length of R MUST be the same as the length of the output of the hash algorithm H. For example, if H is SHA-1, the random number MUST be 160 bits.

A Random Number Generator (RNG) that meets the requirements defined in [FIPS 140-2] and its use is strongly recommended.

4.4. Generation of SIM

The SIM in the subjectAltName extension within a certificate identifies an entity, even if multiple subjectAltNames appear in a certificate. RAs MUST calculate the SIM value with the designated inputs according to the following algorithm:

```
SIM = R || PEPSI
   where PEPSI = H(H(P || R || SIItype || SII))
```

The SII is made known to an RA at user enrollment. Both SHA-1 and SHA-256 MUST be supported for generation and verification of PEPSI values. This specification does not preclude use of other one-way hash functions, but SHA-1 or SHA-256 SHOULD be used wherever interoperability is a concern.

Note that a secure communication channel MUST be used to pass P and SII passing from the end entity to the RA, to protect them from disclosure or modification.

The syntax and the associated OID for SIM are also provided in the ASN.1 modules in Section 5.1. Also, Section 5.2 describes the syntax for PEPSI in the ASN.1 modules.

4.5. Encryption of PEPSI

It may be required that the CA (not just the RA) verifies SII before issuing a certificate. To meet this requirement, RA SHOULD encrypt the SIItype, SII, and SIM and send the result to the CA by a secure channel. The user SHOULD also encrypt the same values and send the result to the CA in his or her certificate request message. Then the CA compares these two results for verifying the user's SII.

Where the results from RA and the user are the EPEPSI.

```
EPEPSI = E(SIItype || SII || SIM)
```

When the EPEPSI is used in a user certificate request, it is in regInfo of [RFC4211] and [RFC2986].

Note: Specific encryption/decryption methods are not defined in this document. For transmission of the PEPSI value from a user to a CA, the certificate request protocol employed defines how encryption is performed. For transmission of this data between an RA and a CA, the details of how encryption is performed is a local matter.

The syntax and the associated OID for EPEPSI is provided in the ASN.1 modules in Section 5.3.

4.6. Certification Request

As described above, a certificate request message MAY contain the SIM. [RFC2986] and [RFC4211] are widely used message syntaxes for certificate requests.

Basically, a PKCS#10 message consists of a distinguished name, a public key, and an optional set of attributes, collectively signed by

the end entity. The SIM alternative name MUST be placed in the subjectAltName extension if this certificate request format is used. If a CA verifies SII before issuing the certificate, the value of SIM in the certification request MUST be conveyed in the EPEPSI form and provided by the subject.

4.7. Certification

A CA that issues certificates containing the SIM includes the SIM as a form of otherName from the GeneralName structure in the "subjectAltName" extension.

In an environment where a CA verifies SII before issuing the certificate, a CA decrypts the EPEPSI values it receives from both the user and the RA, and compares them. It then validates that the SII value is correctly bound to the subject.

```
SIItype, SII, SIM = D(EPEPSI)
```

5. Definition

5.1. SIM Syntax

This section specifies the syntax for the SIM name form included in the subjectAltName extension. The SIM is composed of the three fields: the hash algorithm identifier, the authority-chosen random value, and the value of the PEPSI itself.

```
id-pkix     OBJECT IDENTIFIER  ::=
      { iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) }

id-on       OBJECT IDENTIFIER ::= { id-pkix 8 }
id-on-SIM   OBJECT IDENTIFIER ::= { id-on 6 }

  SIM ::= SEQUENCE {
      hashAlg          AlgorithmIdentifier,
      authorityRandom  OCTET STRING,   -- RA-chosen random number
                                       -- used in computation of
                                       -- pEPSI
      pEPSI            OCTET STRING    -- hash of HashContent
                                       -- with algorithm hashAlg
  }
```

5.2. PEPSI

This section specifies the syntax for the PEPSI. The PEPSI is generated by performing the same hash function twice. The PEPSI is generated over the ASN.1 structure HashContent. HashContent has four values: the user-selected password, the authority-chosen random number, the identifier type, and the identifier itself.

```
HashContent ::= SEQUENCE {
   userPassword     UTF8String,
                    -- user-supplied password
   authorityRandom  OCTET STRING,
                    -- RA-chosen random number
   identifierType   OBJECT IDENTIFIER,  -- SIItype
   identifier       UTF8String          -- SII
}
```

Before calculating a PEPSI, conforming implementations MUST process the userPassword with the six-step [LDAPBIS STRPREP] string preparation algorithm, with the following changes:

* In step 2, Map, the mapping shall include processing of characters commonly mapped to nothing, as specified in Appendix B.1 of [RFC3454].
* Omit step 6, Insignificant Character Removal.

5.3. Encrypted PEPSI

This section describes the syntax for the Encrypted PEPSI. The Encrypted PEPSI has three fields: identifierType, identifier, and SIM.

```
EncryptedPEPSI ::= SEQUENCE {
   identifierType  OBJECT IDENTIFIER, -- SIItype
   identifier      UTF8String,        -- SII
   sIM             SIM                -- Value of the SIM
}
```

When it is used in a certificate request, the OID in 'regInfo' of [RFC4211] and [RFC2986] is as follows:

```
id-regEPEPSI OBJECT IDENTIFIER ::= { id-pkip 3 }
```

6. Example Usage of SIM

Depending on different security environments, there are three possible use cases with SIM.

1. When a relying party does not have any information about the certificate user.
2. When a relying party already knows the SII of the certificate user.
3. When the certificate user does not want to disclose his SII.

For the use case 1, the SII and a user-chosen password P (which only the user knows) must be sent to a relying party via a secure communication channel; the certificate including the SIM also must be transmitted. The relying party acquires R from the certificate. The relying party can verify that the SII was validated by the CA (or RA) and is associated with the entity that presented the password and certificate. In this case, the RP learns which SII is bound to the subject as a result of the procedure.

In case 2, a certificate user transmits only the password, P, and the certificate. The rest of the detailed procedure is the same as case 1, but here the relying party supplies the SII value, based on its external knowledge of that value. The purpose in this case is to enable the RP to verify that the subject is bound to the SII, presumably because the RP identifies the subject based on this SII.

In the last case, the certificate user does not want to disclose his or her SII because of privacy concerns. Here the only information sent by a certificate subject is the intermediate value of the PEPSI, H(R || P || SIItype || SII). This value MUST be transmitted via a secure channel, to preserve its confidentiality. Upon receiving this value, the relying party applies the hash function to the intermediate PEPSI value sent by the user, and matches it against the SIM value in the user’s certificate. The relying party does not learn the user’s SII value as a result of this processing, but the relying party can verify the fact that the user knows the right SII and password. This gives the relying party more confidence that the user is the certificate subject. Note that this form of user identity verification is NOT to be used in lieu of standard certificate validation procedures, but rather in addition to such procedures.

7. Name Constraints

The SIM value is stored as an otherName of a subject alternative name; however, there are no constraints that can be placed on this form of the name.

8. Security Considerations

Confidentiality for a SIM value is created by the iterated hashing of the R, P, and SII values. A SIM value depends on two properties of a hash function: the fact that it cannot be inverted and the fact that collisions (especially with formatted data) are rare. The current attacks by [WANG] are not applicable to SIM values since the end entity supplying the SII and SIItype values does not supply all of the data being hashed; i.e., the RA provides the R value.

In addition, a fairly good password is needed to protect against guessing attacks on SIMs. Due to the short length of many SIIs, it is possible that an attacker may be able to guess it with partial information about gender, age, and date of birth. SIItype values are very limited. Therefore, it is important for users to select a fairly good password to prevent an attacker from determining whether a guessed SII is accurate.

This protocol assumes that Bob is a trustworthy relying party who will not reuse the Alice’s information. Otherwise, Bob could "impersonate" Alice if only knowledge of P and SII were used to verify a subject’s claimed identity. Thus, this protocol MUST be used only with the protocols that make use of digital signatures generated using the subject’s private key.

Digital signatures are used by a message sender to demonstrate knowledge of the private key corresponding to the public key in a certificate, and thus to authenticate and bind his or her identity to a signed message. However, managing a private key is vulnerable under certain circumstances. It is not fully guaranteed that the claimed private key is bound to the subject of a certificate. So, the SIM can enhance verification of user identity.

Whenever a certificate needs to be updated, a new R SHOULD be generated and the SIM SHOULD be recomputed. Repeating the value of the SIM from a previous certificate permits an attacker to identify certificates associated with the same individual, which may be undesirable for personal privacy purposes.

9. Acknowledgements

Jim Schaad (Soaring Hawk Consulting), Seungjoo Kim, Jaeho Yoon, Baehyo Park (KISA), Bill Burr, Morrie Dworkin (NIST), and the Internet Security Technology Forum (ISTF) have significantly contributed to work on the SIM and PEPSI concept and identified a potential security attack. Also their comments on the set of desirable properties for the PEPSI and enhancements to the PEPSI were most illumination. Also, thanks to Russell Housley, Stephen Kent, and Denis Pinkas for their contributions to this document.

10. IANA Considerations

In the future, IANA may be asked to establish a registry of object identifiers to promote interoperability in the specification of SII types.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, November 2000.

[RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, December 2002.

[RFC4043] Pinkas, D. and T. Gindin, "Internet X.509 Public Key Infrastructure Permanent Identifier", RFC 4043, May 2005.

[RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, September 2005.

11.2. Informative References

[LDAPBIS STRPREP] Zeilenga, K., "LDAP: Internationalized String Preparation", Work in Progress.

[FIPS 112] Fedreal Information Processing Standards Publication (FIPS PUB) 112, "Password Usage", 30 May 1985.

[FIPS 180-1] Federal Information Processing Standards Publication (FIPS PUB) 180-1, "Secure Hash Standard", 17 April 1995.

[FIPS 140-2] Federal Information Processing Standards Publication (FIPS PUB) 140-2, "Security Requirements for Cryptographic Modules", 25 May 2001.

[WANG] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu, "Finding Collisions in the Full SHA-1", Crypto'05. <http://www.infosec.sdu.edu.cn/paper/sha1-crypto-auth-new-2-yao.pdf>

Authors' Addresses

Jongwook Park
Korea Information Security Agency
78, Garak-Dong, Songpa-Gu, Seoul, 138-803
REPUBLIC OF KOREA

Phone: 2-405-5432
EMail: khopri@kisa.or.kr

Jaeil Lee
78, Garak-Dong, Songpa-Gu, Seoul, 138-803
REPUBLIC OF KOREA
Korea Information Security Agency

Phone: 2-405-5300
EMail: jilee@kisa.or.kr

Hongsub Lee
Korea Information Security Agency
78, Garak-Dong, Songpa-Gu, Seoul, 138-803
REPUBLIC OF KOREA

Phone: 2-405-5100
EMail: hslee@kisa.or.kr

Sangjoon Park
BCQRE Co.,Ltd
Yuil Bldg. Dogok-dong 411-14, Kangnam-ku, Seoul, 135-270
REPUBLIC OF KOREA

EMail: sjpark@bcqre.com

Tim Polk
National Institute of Standards and Technology
100 Bureau Drive, MS 8930
Gaithersburg, MD 20899

EMail: tim.polk@nist.gov

Appendix A. "Compilable" ASN.1 Module, 1988 Syntax

```
   PKIXSIM {iso(1) identified-organization(3) dod(6) internet(1)
      security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-sim2005(38) }

   DEFINITIONS EXPLICIT TAGS ::=

   BEGIN

   -- EXPORTS ALL

    IMPORTS

    AlgorithmIdentifier, AttributeTypeAndValue FROM PKIX1Explicit88
      {iso(1) identified-organization(3) dod(6) internet(1) security(5)
       mechanisms(5) pkix(7) id-mod(0) id-pkix1-explicit(18)}

   -- SIM

   -- SIM certificate OID

       id-pkix    OBJECT IDENTIFIER  ::=
            { iso(1) identified-organization(3) dod(6) internet(1)
              security(5) mechanisms(5) pkix(7) }

      id-on       OBJECT IDENTIFIER ::= { id-pkix 8 }
       id-on-SIM  OBJECT IDENTIFIER ::= { id-on 6 }

   -- Certificate Syntax

       SIM ::= SEQUENCE {
             hashAlg          AlgorithmIdentifier,
             authorityRandom  OCTET STRING,    -- RA-chosen random number
                                               -- used in computation of
                                               -- pEPSI
             pEPSI            OCTET STRING     -- hash of HashContent
                                               -- with algorithm hashAlg
         }

   -- PEPSI

       UTF8String ::= [UNIVERSAL 12] IMPLICIT OCTET STRING
       -- The content of this type conforms to RFC 2279

       HashContent ::= SEQUENCE {
            userPassword     UTF8String,
                             -- user-supplied password
            authorityRandom  OCTET STRING,
```

```
                            -- RA-chosen random number
            identifierType   OBJECT IDENTIFIER,  -- SIItype
            identifier       UTF8String          -- SII
         }

   -- Encrypted PEPSI

   -- OID for encapsulated content type

       id-regEPEPSI OBJECT IDENTIFIER ::= { id-pkip 3 }

         EncryptedPEPSI ::= SEQUENCE {
            identifierType  OBJECT IDENTIFIER, -- SIItype
            identifier      UTF8String,        -- SII
            sIM             SIM                -- Value of the SIM
         }

   END
```

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).