SPECIFICATION SDN.702
Revision 1.4     1989-08-01

# SDNS

## Secure Data Network System

## SDNS Directory Specifications for Utilization with the SDNS Message Security Protocol

Source: SDNS Protocol and Signaling Working Group

Introductory note :

This document provides the framework for the SDNS Directory Server Specification. This document is being circulated for comment and approval. It is subject to change during the development phase of SDNS.

## 0. Introduction

This document specifies additions to the Directory System described in the 1988 X.500 series of CCITT Recommendations to support some key management functions, both for general use by SDNS components and, in particular, for use by X.400 messages protected by SDNS Message Security Protocol (MSP). This document describes the new attribute types and object classes for inclusion in the Directory Information Base (DIB) in support of these functions. These new attributes will be manipulated using the Directory Access Protocol (DAP), but no new operations are required to manipulate these attributes. None of these attributes are interpreted by the Directory, and none are used for naming.

## 1. Scope and Field of Application

In order to support key distribution for X.400 messages protected by MSP, the DIB must store some attributes, which are not currently provided as Directory attributes, in Directory entries associated with mail system users .

It is anticipated that, in normal operation, a Directory User Agent (DUA) would query a Directory System Agent (DSA), using the DAP to retrieve attributes associated with one or more entries, based on asserted attribute values that are sufficient to identify the intended recipient(s).

It is assumed that both private and administrative Directories may be employed in support of SDNS electronic messages and that both Type I and Type II users will be supported.

## 2. References

IS 7498/1 Information Processing Systems - Open Systems Interconnection - Basic Reference Model.

IS 7498/2 Information Processing Systems - Open Systems Interconnection - Security Architecture.

CCITT X.500 The Directory - Overview of Concepts, Models, and Services.

CCITT X.501 The Directory - Models.

CCITT X.509 The Directory - Authentication Framework.

CCITT X.511 The Directory - Abstract Service Definition.

CCITT X.518 The Directory - Procedures for Distributed Operation.

CCITT X.519 The Directory - Protocol Specifications.

CCITT X.520 The Directory - Selected Attribute Types.

CCITT X.521 The Directory - Selected Object Classes.

SDN.801 SDNS Access Control Concept Document.

SDN.802 SDNS Access Control Specification.

## 3. Definitions and Abbreviations

This document contains terms and abbreviations defined in CCITT X.500 and ISO DIS 9594 1-8. In addition, this document contains the following:

KMID - key material identifier.

SDNS - Secure Data Network System.

Ukm - User's individual keying material.

## 4. Specification of Attributes

### 4.1 Certificate

Each of these attributes consists of a certificate (Type I or II) associated with a message system user.

```
type1Certificate            ATTRIBUTE
                                WITH ATTRIBUTE-SYNTAX
                                        OCTETSTRING
                                MATCHES FOR EQUALITY
                                :: = {sdnsAttributeType 1}

type2Certificate            ATTRIBUTE
                                WITH ATTRIBUTE-SYNTAX
                                        OCTET STRING
                                MATCHES FOR EQUALITY
                                :: = {sdnsAttributeType 2}
```

### 4.2 Signature Certificate

These attributes consist of a certificate which can be used to validate digital signatures formulated by a user employing a signature key (Type I or II).

```
signature1Certificate           ATTRIBUTE
                                WITH ATTRIBUTE-SYNTAX
                                        OCTET STRING
                                MATCHES FOR EQUALITY
                                :: = {sdnsAttributeType 3}
```

```
signature2Certificate          ATTRIBUTE
                               WITH ATTRIBUTE-SYNTAX
                                    OCTET STRING
                               MATCHES FOR EQUALITY
                               :: = {sdnsAttributeType 4}
```

### 4.3 Auxiliary Vector

This attribute contains information used to support the access control and authentication information contained in the Certificate.

```
auxVector                      ATTRIBUTE
                               WITH ATTRIBUTE-SYNTAX
                                    OCTET STRING
                               MATCHES FOR EQUALITY
                               :: = {sdnsAttributeType 5}
```

### 4.4 Ukm

These attributes consist of values, each of which is a SignedUkm. The signature can be validated using the signature certificate stored with the Directory entry. The intent is that each Ukm is valid for a period of time and may be changed at the owner's discretion by modifying the appropriate attribute value. At any time, the user may delete the old attribute value; however, he should maintain the old Ukm locally for use in deciphering messages which were encrypted using the old Ukm. Users desiring to transmit secure mail should query the Directory and retrieve the appropriate SignedUkm.

```
signedUkm1             ATTRIBUTE
                           WITH ATTRIBUTE-SYNTAX
                                   SEQUENCE OF SignedUkm
                           MATCHES FOR EQUALITY
                           :: = {sdnsAttributeType 6}


signedUkm2             ATTRIBUTE
                           WITH ATTRIBUTE-SYNTAX
                                   SEQUENCE OF SignedUkm
                           MATCHES FOR EQUALITY
                           :: = {sdnsAttributeType 7}
```

## 5. Service Definitions

These additions to the DIB do not require any new services, but rather, specify new attributes which may be parameters to existing services as defined in X.511. Using these services, a DUA can retrieve the values of these attributes, modify the attribute values, and set the entry access control (A.C.) parameters associated with these attributes, subject to the A.C. restrictions. It is suggested that the default A.C. parameters of each of these new attributes be set to permit readItem access for all accessors, and modifyItem access for the owner(s) of each entry. Authorization to modify A.C. for these attributes should be granted to the Directory entry owner and/or system administrator.

Note: Access Control mechanisms for the Directory, as yet undefined by the International Standards, are a subject for further study.

## 6. Protocol Specification

These additions to the DIB require no new protocols. Rather, as in the service definitions above, the DAP is expected to operate on the newly defined attributes just as it would any other attribute. Since all of these attributes are represented as OCTET STRINGs in the Directory, no semantic processing is appropriate. Using DAP, a DUA can retrieve the values of these attributes, modify the attribute values, and set the entry A.C. parameters associated with these attributes, subject to the A.C. restrictions. It is suggested that the default A.C. parameters of each of these new attributes be set to permit readItem access for all accessors, and modifyItem access for the owner(s) of each entry. Authorization to modify A.C. for these attributes should be granted to the Directory entry owner and/or system administrator.

## 7. ASN.1 Notation

```
 SdnsAdditionsToDIB {48 3 9999}
 DEFINITIONS        ::=
 BEGIN

 IMPORTS

            ATTRIBUTE, ATTRIBUTE-SYNTAX, OBJECT-CLASS
                FROM InformationFramework {joint-ISO-CCITTds(5)modules(1)
                                                    informationFramework(1)},


--SDNS ObjectIDs

   sdnsAttributeType  OBJECT IDENTIFIER ::= {49}

-- SDNS Attributes

type1Certificate            ATTRIBUTE
                                WITH ATTRIBUTE-SYNTAX
                                        OCTETSTRING
                                MATCHES FOR EQUALITY
                                ::= {sdnsAttributeType 1}

type2Certificate            ATTRIBUTE
                                WITH ATTRIBUTE-SYNTAX
                                        OCTET STRING
                                MATCHES FOR EQUALITY
                                ::= {sdnsAttributeType 2}

signature1Certificate           ATTRIBUTE
                                WITH ATTRIBUTE-SYNTAX
                                        OCTET STRING
                                MATCHES FOR EQUALITY
                                ::= {sdnsAttributeType 3}

signature2Certificate           ATTRIBUTE
                                WITH ATTRIBUTE-SYNTAX
                                        OCTET STRING
                                MATCHES FOR EQUALITY
                                ::= {sdnsAttributeType 4}

auxVector                       ATTRIBUTE
                                WITH ATTRIBUTE-SYNTAX
                                        OCTET STRING
                                MATCHES FOR EQUALITY
                                ::= {sdnsAttributeType 5}
```

```
signedUkm1              ATTRIBUTE
                          WITH ATTRIBUTE-SYNTAX
                                SEQUENCE OF SignedUkm
                          MATCHES FOR EQUALITY
                          :: = {sdnsAttributeType 6}

signedUkm2              ATTRIBUTE
                          WITH ATTRIBUTE-SYNTAX
                                SEQUENCE OF SignedUkm
                          MATCHES FOR EQUALITY
                          :: = {sdnsAttributeType 7}

SignedUkm               :: =  SEQUENCE {
                          day                   INTEGER,
                          month                 INTEGER,
                          year                  INTEGER,
                          ukm                   OCTET STRING,
                          signatureValue        OCTET STRING}

type1Sdns                 ATTRIBUTE
                          WITH ATTRIBUTE-SYNTAX
                                SEQUENCE OF Type1Sdns
                          MATCHES FOR EQUALITY
                          :: = {sdnsAttributeType 8}

type2Sdns                 ATTRIBUTE
                          WITH ATTRIBUTE-SYNTAX
                                SEQUENCE OF Type2Sdns
                          MATCHES FOR EQUALITY
                          :: = {sdnsAttributeType 9}

Type1Sdns                 :: =  SEQUENCE{
                          type1Certificate         OCTET STRING,
                          signature1Certificate    OCTET STRING,
                          signedUkm1            }

Type2Sdns                 :: =  SEQUENCE{
                          type2Certificate         OCTET STRING,
                          signature2Certificate    OCTET STRING,
                          signedUkm2            }


END
```