Telecommunications and Information Exchange Between Systems

ISO/IEC JTC 1/SC 6

Document Number:	N13805
Date:	2008-11-18
Replaces:	
Document Type:	Other document (Defined)
Document Title:	The second working draft of the Amendment 2 on Edition 6 of Directory (ISO/IEC 9594 Amendment 2)
Document Source:	Project Editor
Project Number:	
Document Status:	For your information.
Action ID:	FYI
Due Date:	
No. of Pages:	42

ISO/IEC JTC1/SC6 Secretariat Ms. Jooran Lee, KSA (on behalf of KATS)

Korea Technology Center #701-7 Yeoksam-dong, Gangnam-gu, Seoul, 135-513, Republic of Korea ;

Telephone: +82 2 6009 4808 ; Facsimile: +82 2 6009 4819 ; Email : jooran@kisi.or.kr

INTERNATIONAL STANDARD
ITU-T RECOMMENDATION

Information technology – Open Systems Interconnection – The Directory

Amendment 2

Password policy

Summary

Password policy is a set of rules that controls how passwords are used and administered in the Directory. It improves the security of the Directory and makes it difficult for password cracking programs to break into the Directory. These rules ensure that users change their passwords periodically, that passwords meet quality requirements that re-use of old passwords is restricted, and that users are locked out after a certain number of failed attempts.

CONTENTS

ISO/IEC 9594-1: 2008, Information Technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services

1) Subclause A.3.8

Add at the end of the second paragraph:

A password policy can be used for administration of passwords to improve the security of the Directory and make it more difficult for password cracking programs to break into the Directory. A password policy defines the rules to ensure that users change their passwords periodically, that passwords meet quality requirements, that re-use of old passwords is restricted, and that users are locked out after a certain number of failed password comparison attempts. A password can only be changed by the owner of the entry or by an administrator of the Directory (for example when a user has lost his password).

ISO/IEC 9594-2: 2008, Information Technology – Open Systems Interconnection – The Directory: Models

No change

ISO/IEC 9594-3: 2008, Information Technology – Open Systems Interconnection – The Directory: Abstract service definition

1) Clause 8

Replace the title of clause 8 with:

8 Bind, Unbind and Change of password operations

2) Subclause 8.1.1 and Annex A

*Replace the ASN.1 definition of **DirectoryBindResult** with:*

```
DirectoryBindResult ::= SET {
        credentials [0] Credentials OPTIONAL,
        versions [1] Versions DEFAULT {v1},
        pwdResponseValue PwdResponseValue OPTIONAL }
```

*Insert after definition of **Versions**, the following definitions:*

```
PwdResponseValue ::= SEQUENCE {
        warning [0] CHOICE {
                timeLeft [0]            INTEGER (0..MAX),
                graceRemaining [1]      INTEGER (0..MAX)} OPTIONAL,
        error [1] ENUMERATED {
                passwordExpired (0),
                changeAfterReset(1) } OPTIONAL }
```

3) Subclause 8.1.3

Insert at end end the following text:

The following applies independently whether the DSA holds the responder's master entry or a replicated entry.

a) if the **warning.timeLeft** component is present and different from zero, the **error** component shall be absent;

b) if the **warning.graceRemaing** component is present, the **error.passwordExpired** may be set.

The following applies when the DSA holds the master entry for the requestor:

a) if **warning** is present with either the **timeLeft** set to zero or **gracesRemaining** set to zero and **error.passwordExpired** set, only a change-password operation is accepted;

b) if **error.changeAfterReset** is set, **warning** shall not be present. Only a change-password operation is accepted.

4) Subclause 8.1.4

*Replace the text after the sentence "A **securityError** or **serviceError** shall be supplied as follows:"*

– **securityError** **inappropriateAuthentication**
invalidCredentials
blockedCredentials
passwordPolicyRequired
passwordExpired

– **serviceError** **unavailable**
saslBindInProgress

5) New subclause 8.3

Add new subclause 8.3

8.3 Directory Change of password

8.3.1 Directory Change of password syntax

A Directory Change of password operation is used by a user to change a password to prevent password expiration or after password reset by an administrator. The password may be change at any time during an application-association. The user is allowed as many attempts as specified in the **pwdMaxCompareFailure** attribute. When this limit is reached, the DSA shall unbind the application-association and, if the **pwdCompareLockout** attribute is **TRUE**, lock the account for **pwdCompareLockoutDuration**.

```
changePassword OPERATION ::= {
        ARGUMENT ChangePasswordArgument
        RESULT ChangePasswordResult
        ERRORS { securityError | updateError }
        CODE id-opcode-changePassword }

ChangePasswordArgument ::= OPTIONALLY-PROTECTED-SEQ {
        SEQUENCE {
        oldPwd [0] SimpleCredentials,
        newPwd [1] SimpleCredentials } }

ChangePasswordResult ::= NULL
```

8.3.2 Directory Change of password arguments

The current password (**oldPwd** component) and the new password (**newPwd** component) have to be supplied in a Change of operation.

8.3.3 Directory Change of password results

If the password is changed successfully, the operation returns no information and normal communication may continue.

8.3.4 Directory Change of password errors

Should the request fail, a **securityError** or **updateError** shall be supplied as follows:

- **securityError** — **passwordModNotAllowed**
- **updateError** — **insufficientPasswordQuality**, **passwordInHistory**, **noPasswordSlot**

The circumstances under which other errors shall be reported are defined in clause 12.

6) Subclause 12.7 and Annex A

*Replace the definition of **SecurityProblem** with:*

```
SecurityProblem  ::=  INTEGER {
        inappropriateAuthentication   (1),
        invalidCredentials            (2),
        insufficientAccessRights      (3),
        invalidSignature              (4),
        protectionRequired            (5),
        noInformation                 (6),
        blockedCredentials            (7),
- -     invalidQOPMatch               (8), obsolete
        spkmError                     (9),
        passwordExpired               (10),
        passwordModNotAllowed         (11)}
```

Insert after item h) the new following items:

i) **passwordExpired** – The requestor cannot log onto the DSA because the password has expired. The password has to be reset by an administrator.

j) **passwordModNotAllowed** – The requestor has not the right to change his own password.

7) Subclause 12.9 and Annex A

*Replace the definition of **UpdateProblem** with:*

```
UpdateProblem ::= INTEGER {
    namingViolation                  (1),
    objectClassViolation             (2),
    notAllowedOnNonLeaf              (3),
    notAllowedOnRDN                  (4),
    entryAlreadyExists               (5),
    affectsMultipleDSAs              (6),
    objectClassModificationProhibited (7),
    noSuchSuperior                   (8),
    notAncestor                      (9),
    parentNotAncestor                (10),
    hierarchyRuleViolation           (11),
    familyRuleViolation              (12),
    insufficientPasswordQuality      (13),
    passwordInHistory                (14),
    noPasswordSlot                   (15) }
```

Insert after item l) the new following items:

m) **insufficientPasswordQuality** – The new password does not satisfy the quality rules (no trivial passwords, mixture of characters, too short, etc) imposed by the Directory.

n) **passwordInHistory** – The new password has been found in the history kept by the Directory.

o) **noPasswordSlot** – There are no free slots left in the password history.

8) Annex A

In the IMPORTS clause replace:

-- from ITU-T Rec. X.519 | ISO/IEC 9594-5

```
    Code, ERROR, id-errcode-abandoned, id-errcode-abandonFailed,  id-errcode-attributeError,
    id-errcode-nameError, id-errcode-referral, id-errcode-securityError, id-errcode-serviceError,
    id-errcode-updateError, id-opcode-abandon, id-opcode-addEntry, id-opcode-compare,
    id-opcode-list,  id-opcode-modifyDN, id-opcode-modifyEntry, id-opcode-read,
    id-opcode-removeEntry, id-opcode-search, InvokeId, OPERATION
        FROM CommonProtocolSpecification commonProtocolSpecification
```

with:

-- from ITU-T Rec. X.519 | ISO/IEC 9594-5

```
    Code, ERROR, id-errcode-abandoned, id-errcode-abandonFailed,  id-errcode-attributeError,
    id-errcode-nameError, id-errcode-referral, id-errcode-securityError, id-errcode-serviceError,
    id-errcode-updateError, id-opcode-abandon, id-opcode-addEntry, id-opcode-compare,
    id-opcode-list,  id-opcode-modifyDN, id-opcode-modifyEntry, id-opcode-read,
    id-opcode-removeEntry, id-opcode-search, id-opcode-changePassword, InvokeId, OPERATION
        FROM CommonProtocolSpecification commonProtocolSpecification
```

Insert after -- Operations, arguments, and results –

```
changePassword OPERATION ::= {
    ARGUMENT ChangePasswordArgument
    RESULT ChangePasswordResult
    ERRORS { securityError | updateError }
    CODE id-opcode-changePassword }

ChangePasswordArgument ::= OPTIONALLY-PROTECTED-SEQ {
    SEQUENCE {
        oldPwd [0] SimpleCredentials,
        newPwd [1] SimpleCredentials }}

ChangePasswordResult ::= NULL
```

ISO/IEC 9594-4: 2008, Information Technology – Open Systems Interconnection – The Directory: Procedures for distributed operation

1) Annex A

Replace the following text:

-- from ITU-T Rec. X.511 | ISO/IEC 9594-3
abandon, addEntry, CommonResults, compare, directoryBind, list,
modifyDN, modifyEntry, read, referral, removeEntry, search, SecurityParameters
FROM DirectoryAbstractService directoryAbstractService

with:

-- from ITU-T Rec. X.511 | ISO/IEC 9594-3
abandon, addEntry, changePassword, CommonResults, compare, directoryBind, list,
modifyDN, modifyEntry, read, referral, removeEntry, search, SecurityParameters
FROM DirectoryAbstractService directoryAbstractService

Add, after ***chainedModifyDN****, the following definition:*

chainedChangePassword OPERATION ::= chained { changePassword }

ISO/IEC 9594-5: 2008, Information Technology – Open Systems Interconnection – The Directory: Protocol specifications

1) Subclause 6.4.1 and Annex A

*Add after **id-opcode-modifyDN**, the following definition:*

```
id-opcode-changePassword   Code  ::= local:10
```

ISO/IEC 9594-6: 2008, Information Technology – Open Systems Interconnection – The Directory: Selected attribute types

1) Annex A

Add the following definitions after the line beginning with: "-- id-at-permission":

```
-- id-at-userPwd                OBJECT IDENTIFIER   ::=   {id-at 83}
-- id-at-encUserPwd             OBJECT IDENTIFIER   ::=   {id-at 84}
```

Add the following definitions after the line beginning with: "-- id-mr-dualStringMatch":

```
-- id-mr-pwdHistoryMatch        OBJECT IDENTIFIER   ::=   {id-mr 70}   X.509|Part8
-- id-mr-encUserPwdMatch        OBJECT IDENTIFIER   ::=   {id-mr 71}   X.509|Part8
```

ISO/IEC 9594-7: 2008, Information Technology – Open Systems Interconnection – The Directory: Selected object classes

1) Subclause 5.4 and Annex A

Replace the current definition of ***OrganizationalAttributeSet*** *with:*

```
OrganizationalAttributeSet  ATTRIBUTE  ::=  {
        description |
        LocaleAttributeSet |
        PostalAttributeSet |
        TelecommunicationAttributeSet |
        businessCategory |
        seeAlso |
        searchGuide |
        userPassword |
        userPwd |
        userEncPwd}
```

2) Subclause 6.5 and Annex A

Replace the current definition of ***person*** *with:*

```
person  OBJECT-CLASS  ::=  {
        SUBCLASS OF     { top }
        MUST CONTAIN    { commonName | surname }
        MAY CONTAIN     { description |
                          telephoneNumber |
                          userPassword |
                          userPwd |
                          userEncPwd |
                          seeAlso }
        ID              id-oc-person }
```

3) Annex A

Replace the third part of the ***IMPORTS*** *clause with:*

-- from ITU-T Rec. X.509 | ISO/IEC 9594-8

```
        authorityRevocationList, cACertificate, certificateRevocationList, crossCertificatePair,
        deltaRevocationList, supportedAlgorithms, userCertificate, userPassword, userPwd, userEncPwd
                FROM AuthenticationFramework authenticationFramework
```

ISO/IEC 9594-8: 2008, Information Technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks

1) Subclause 3.3

Add the following definitions after 3.3.37 and renumber the existing subclauses 3.3.38 to 3.3.60 as 3.3.41 to 3.3.63 :

3.3.38 Password expiration: the situation where a user password has reached the end of its validity period: the account is locked and the user has to change the password before doing any other directory operation.

3.3.39 Password quality rules: rules that specify how a password shall be constructed. Password quality rules include things like minimum length, mixture of characters (uppercase, lowercase, figures, punctuations, etc), and avoidance of trivial passwords.

3.3.40 Password history: list of old passwords and the times they were inserted in the history..

2) Subclause 18.1.3

Replace the existing subclause 18.1.3 with:

18.1.3 Password policy

Password policy is a set of rules that controls how passwords are used and administered in the Directory. It improves the security of the Directory and makes it difficult for password cracking programs to break into the Directory. These rules ensure that users change their passwords periodically, that passwords meet quality requirements, that re-use of old password is restricted, and that users are locked out after a certain number of failed attempts. This policy also forces the user to update its password after it has been set for the first time, or has been reset by a password administrator. However, in some cases, it is desirable to disallow users from adding and updating their own passwords.

A password is supposed not to be well known. If a password is frequently changed, the chance of misuse is minimized. Password policy administrators may deploy a password policy that causes passwords to expire after a given amount of time thus forcing users to change their passwords periodically. There must be a way to make users aware of the need to change their password before being locked out of their accounts. One or both of the following methods could be used:

- A warning may be returned to the user sometime before the password is due to expire. If the user ignores this warning before the expiration time, the account will be locked.
- The user may Bind to the directory a certain number of times after the password has expired. If the user fails to change the password following one of the 'grace' authentications, the account will be locked.

Password quality rules are rules for how a password shall be constructed. It is not the intention to provide specification for password qualities, as requirements on quality may change over time. Password quality includes things like:

- minimum length;
- mixture of characters (uppercase, lowercase, figures, punctuations, etc.); and
- avoidance of trivial passwords

A particular quality rule requires specialised code within the implementation. It may therefore be advantage to standardise password quality rules and assign object identifiers to such rules. An implementation may then claim support to one or more of such standardised quality rules.

An intruder may try to guess a password to get access to protected information. Currently, two different safeguards have been identified:

- Specification of the maximum number of failed attempts before a successful attempt within a given time span (which could be indefinitely). This approach allows for "denial of service attacks". One or more genuine users could have their access to directory barred by action of an attacker.
- The other mechanism is to insert a delay before returning information on authentication failure, and increasing this delay for repeated failed authentications on the same connection. This approach slows authentication, and makes brute force attacks impractical.

Password history is a mechanism to prevent password re-use. Previously used passwords should be stored to allow the Directory to ensure that a new password has not been previously used. Old passwords are stored for a time specified by the password policy, and after this time a password may be re-used. The history is maintained in a **pwdHistory** multi-

valued operational attribute. A value is purged after a specific time, and the purged password may in principle be reused. This time period a password is kept in the in **pwdHistory** attribute is specified in the **timeInHistory** operational attribute. The number of passwords stored is limited by the **pwdHistorySlots** operational attribute and the password cannot be changed if there is no free slot in the history, so a user cannot revert to a "preferred password" simply by making lots of password changes.

The password policy can be used with clear passwords (using the **userPwd** attribute) or with encrypted passwords (using the **encUserPwd** attribute). All entries in the same administrative domain shall use either **userPwd** or **encUserPwd** attributes.

The password policy uses specific operational attributes to register policy parameters, times and dates related to password management.

When a password value is first stored in the directory, in either the **userPwd** (figure 10) or **encUserPwd** (figure 11) attribute, the **pwdCreationTime** or **encPwdCreationTime** operational attribute respectively may be set. The **pwdExpiryDate** (or **encPwdExpiryDate** resp.) operational attribute may either be automatically computed from the **pwdExpiryAge** (**encPwdExpiryAge** resp.) operational attribute or set by explicit administrator action. The **pwdEndDate** (or **encPwdEndDate** resp.) operational attribute may either be automatically computed from the **pwdMaxAge** (or **encPwdMaxAge** resp.) operational attribute or by explicit administrator action. If the **pwdExpiryWarning** (or **encPwdExpiryWarning** resp.) operational attribute is set, then the user should be informed during this period that his or her password is about to expire.

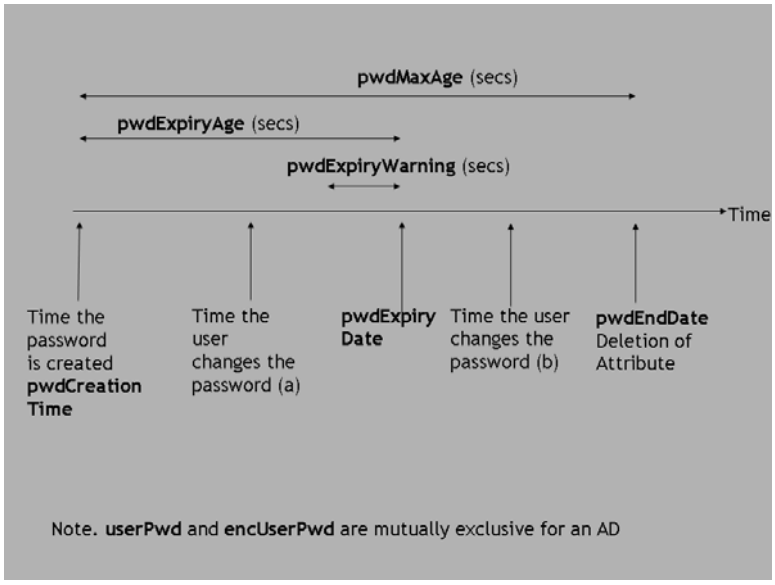

Figure 10 – UserPwd attribute

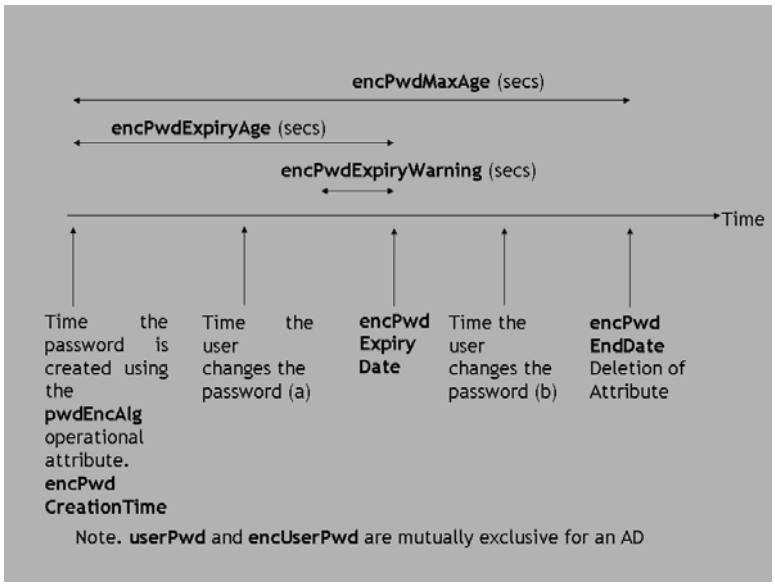

Figure 11 – encUserPwd attribute

When the user (or an administrator acting on behalf of the user) changes the value of their password, the old value should be copied into the recently expired password attribute. (The **userPwd** attribute is copied into the **recentlyExpiredPwd** as shown in figure 12 attribute and the **userEncPwd** is copied into the **recentlyExpiredEncPwd**. as shown in figure 13) The recently expired password creation time (**expPwdCreationTime** or **expEncPwdCreationTime** resp.) should be set. When the recently expired password duration time has finished, the recently expired password attribute (**recentlyExpiredPwd** or **recentlyExpiredEncPwd** resp.) should be deleted. If the user (or an administrator acting on behalf of the user) changes their password again during the recently expired password duration time, then their recently expired password should be overwritten and the duration should be set to start again. Thus a recently expired password will only be kept in the recently expired password attribute for the shorter of the recently expired password duration time or until the user changes their password again. However, it will be kept in the password history table.

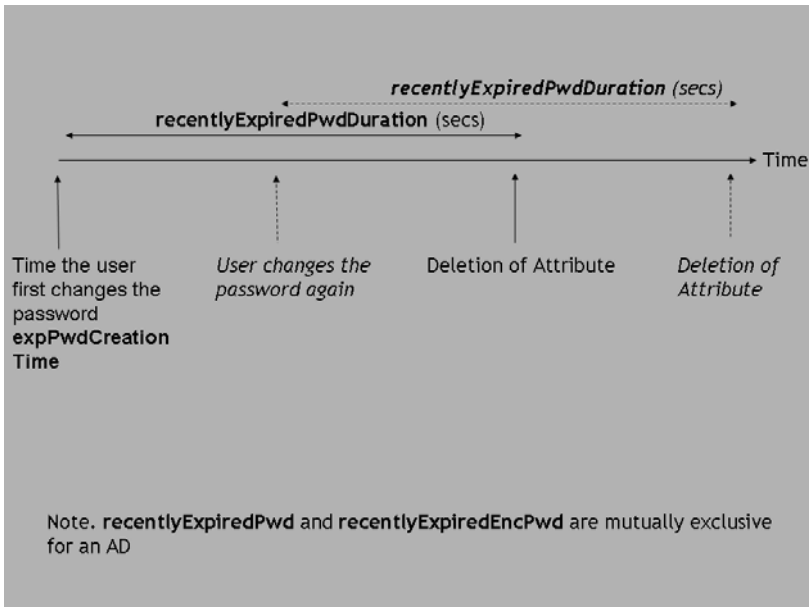

Figure 12 – recentlyExpiredPwd attribute

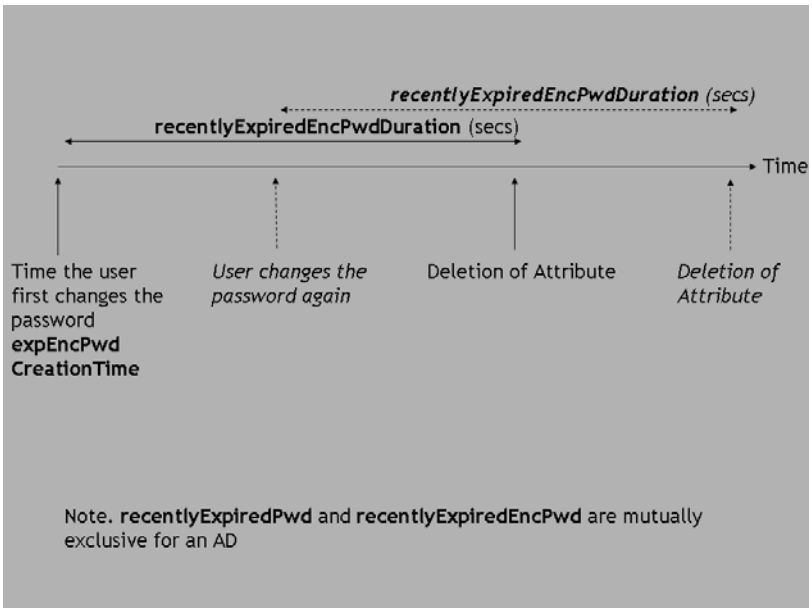

Figure 13 – recentlyExpiredEncPwd attribute

The password history attribute is used to prevent password re-use, by storing old values of the user's password so that the user cannot re-use the same password again whilst it is stored in the password history (see figure 14). When the user (or an administrator acting on behalf of the user) changes their password, it may be copied into the password history (**pwdHistory**) operational attribute along with the time that the password was changed. The password time in history attribute specifies the duration (in seconds) that a password should remain in the password history. Once this time has expired for a particular password, then it is removed from the password history, and the user may use this password again.

The number of slots in the password history table (or password history attribute values) is defined in the **pwdHistorySlots** operational attribute. When all the slots are filled, the user is not allowed to change his password again until one of the old passwords has expired. If the user forgets his password when all the history slots are full, then the administrator must free two slots in the history table (i.e. delete two attribute values), reset the user's password to a temporary value (which is copied into the history), leaving one spare slot for the user to choose their own new password.

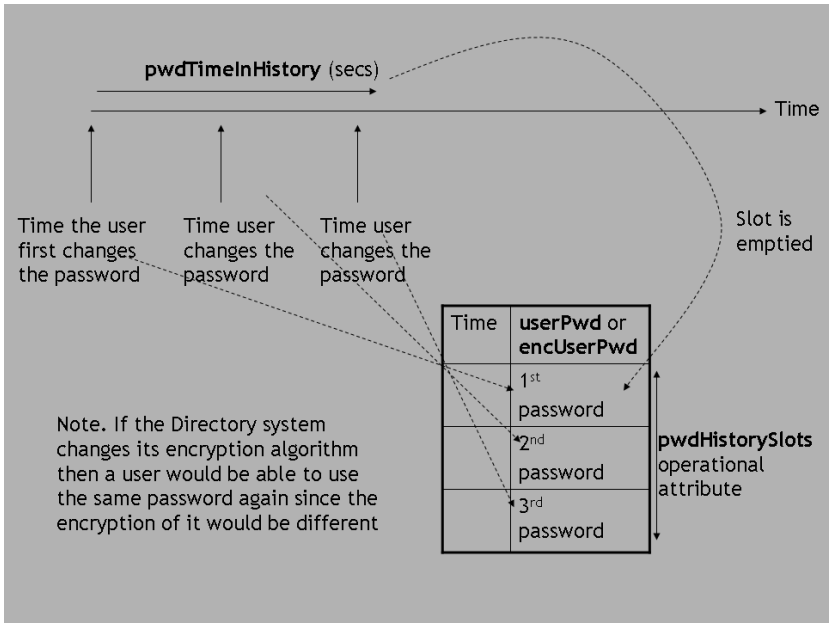

Figure 14 – pwdHistory attribute

3) Subclause 18.1.3

*Replace the current text preceding the **userPassword** definition with:*

18.1.3 User Passwords attribute type

The multi-valued User Passwords attribute type contains the current and possibly previous passwords of an object. An attribute value for a user password is a string specified by the object.

4) Subclause 18.1.4

Add a new subclause 18.1.4

18.1.4 Simple Authentication attributes held by object entries

18.1.4.1 User Password attribute

The **userPwd** attribute type contains the current password of an object. The attribute value for this single-valued user password is an octet string. During password rollover, the old password value may be copied into the **recentlyExpiredPwd** attribute value.

```
userPwd    ATTRIBUTE::=    {
      WITH SYNTAX                    OCTET STRING
      EQUALITY MATCHING RULE         octetStringMatch
      SINGLE VALUE                   TRUE
      ID                             id-at-userPwd}
```

18.1.4.2 Encrypted User Password attribute

The **EncUserPwd** attribute type contains the encrypted password of an object. The attribute value for this single-valued attribute is an octet string containing the encrypted value, with the encryption algorithm identifier. During password rollover, the old encrypted password value may be copied into the **recentlyExpiredEncPwd** attribute value.

```
encUserPwd      ATTRIBUTE::=    {
      WITH SYNTAX                    EncUserPwd
      EQUALITY MATCHING RULE         encUserPwdMatch
```

```
        SINGLE VALUE                    TRUE
        ID                              id-at-encUserPassword }

EncUserPwd ::= SEQUENCE {
        encryptedString   OCTET STRING,
        algorithmIdentifier AlgorithmIdentifier{{SupportedAlgorithms}}}
```

Annex L contains examples of two encryption methods.

18.1.4.2 Password Creation Time

The **pwdCreationTime** operational attribute indicates when the password has been created for the object represented by the entry in which the attribute is present.

```
pwdCreationTime ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
        ORDERING MATCHING RULE          generalizedTimeOrderingMatch
        SINGLE VALUE                    TRUE
        USAGE                           directoryOperation
        ID                              id-oa-pwdCreationTime }
```

18.1.4.3 Encrypted Password Creation Time

The **encPwdCreationTime** operational attribute indicates when the encrypted password has been created for the object represented by the entry in which the attribute is present.

```
encPwdCreationTime ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
        ORDERING MATCHING RULE          generalizedTimeOrderingMatch
        SINGLE VALUE                    TRUE
        USAGE                           directoryOperation
        ID                              id-oa-encPwdCreationTime }
```

18.1.4.4 Expiry Password Creation Time

The **expPwdCreationTime** operational attribute indicates when the password was latest changed for the object represented by the entry in which the attribute is present.

```
expPwdCreationTime ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
        ORDERING MATCHING RULE          generalizedTimeOrderingMatch
        SINGLE VALUE                    TRUE
        USAGE                           directoryOperation
        ID                              id-oa-expPwdCreationTime }
```

18.1.4.5 Expiry Encrypted Password Creation Time

The **expEncPwdCreationTime** operational attribute indicates when the encrypted password was latest changed for the object represented by the entry in which the attribute is present.

```
expEncPwdCreationTime ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
        ORDERING MATCHING RULE          generalizedTimeOrderingMatch
        SINGLE VALUE                    TRUE
        USAGE                           directoryOperation
        ID                              id-oa-expEncPwdCreationTime }
```

18.1.4.6 Password Expiry Date

The **pwdExpiryDate** operational attribute indicates when the password will expires for the object represented by the entry in which the attribute is present. Its value may be obtained by addition of the **pwdExpiryAge** to the **pwdCreationTime** of the entry or set by an administrator.

```
pwdExpiryDate ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
```

```
    ORDERING MATCHING RULE      generalizedTimeOrderingMatch
    SINGLE VALUE                TRUE
    USAGE                       directoryOperation
    ID                          id-oa-pwdExpiryDate }
```

18.1.4.7 Encrypted Password Expiry Date

The **encPwdExpiryDate** operational attribute indicates when the encrypted password will expires for the object represented by the entry in which the attribute is present. Its value may be obtained by addition of the **encPwdExpiryAge** to the **expPwdCreationTime** of the entry or set by an administrator.

```
encPwdExpiryDate ATTRIBUTE ::= {
    WITH SYNTAX                 GeneralizedTime
    EQUALITY MATCHING RULE      generalizedTimeMatch
    ORDERING MATCHING RULE      generalizedTimeOrderingMatch
    SINGLE VALUE                TRUE
    USAGE                       directoryOperation
    ID                          id-oa-encPwdExpiryDate }
```

18.1.4.8 Password End Date

The **pwdEndDate** operational attribute indicates when the password will be no longer valid for the object represented by the entry in which the attribute is present. Its value may be obtained by addition of the **pwdMaxAge** to the **pwdCreationTime** of the entry or set by an administrator.

```
pwdEndDate ATTRIBUTE ::= {
    WITH SYNTAX GeneralizedTime
    EQUALITY MATCHING RULE generalizedTimeMatch
    ORDERING MATCHING RULE generalizedTimeOrderingMatch
    SINGLE VALUE TRUE
    USAGE directoryOperation
    ID id-oa-pwdEndDate }
```

18.1.4.9 Encrypted Password End Date

The **encPwdEndDate** operational attribute indicates when the encrypted password will be no longer valid for the object represented by the entry in which the attribute is present. Its value may be obtained by addition of the **encPwdMaxAge** to the **expPwdCreationTime** of the entry or set by an administrator.

```
encPwdEndDate ATTRIBUTE ::= {
    WITH SYNTAX                 GeneralizedTime
    EQUALITY MATCHING RULE      generalizedTimeMatch
    ORDERING MATCHING RULE      generalizedTimeOrderingMatch
    SINGLE VALUE                TRUE
    USAGE                       directoryOperation
    ID                          id-oa-encPwdEndDate }
```

18.1.4.10 Password Bind Fails attribute

The **pwdBindFails** operational attribute specifies the current number of consecutive failed bind attempts. The value of this attribute is incremented by one after a failed bind attempt and is reset to zero after a successful authentication.

```
pwdBindFails    ATTRIBUTE ::= {
    WITH SYNTAX                 INTEGER (0..MAX)
    EQUALITY MATCHING RULE      integerMatch
    SINGLE VALUE                TRUE
    USAGE                       directoryOperation
    ID                          id-oa-pwdBindFails }
```

18.1.4.11 Password Compare Fails attribute

The **pwdCompareFails** operational attribute specifies the current number of consecutive compare fails with the userPwd or **encUserPwd** attribute of the object represented by the entry in which the attribute is present.

```
pwdCompareFails     ATTRIBUTE ::= {
    WITH SYNTAX                 INTEGER (0..MAX)
    EQUALITY MATCHING RULE      integerMatch
    SINGLE VALUE                TRUE
    USAGE                       directoryOperation
    ID                          id-oa-pwdCompareFails }
```

18.1.4.12 Remaining Authentication Attempts attribute

The **remAuthAttempts** operational attribute specifies the number of remaining authentication attempts with an expired password before this password will be unusable. The value of this attribute is set to the value of **pwdGraces** attribute when the password is changed and decremented by one after successful authentication using an expired password. When the value reaches 0, the password is unusable.

```
remAuthAttempts ATTRIBUTE ::= {
    WITH SYNTAX                 INTEGER (0..MAX)
    EQUALITY MATCHING RULE      integerMatch
    SINGLE VALUE                TRUE
    USAGE                       directoryOperation
    ID                          id-oa-remAuthAttempts }
```

18.1.4.13 Password History

The **pwdHistory** operational attribute is used to hold previous passwords for the user represented by the entry in which the attribute is present.

```
pwdHistory ATTRIBUTE ::= {
    WITH SYNTAX                 PwdHistory
    EQUALITY MATCHING RULE      pwdHistoryMatch
    USAGE                       directoryOperation
    ID                          id-oa-pwdHistory }

PwdHistory ::= SEQUENCE {
    time        GeneralizedTime,
    password    OCTET STRING }
```

This attribute is multi-valued. Each value consists of a sequence of the time the password was put in the history and the password.

18.1.4.14 Recently Expired Password

The **recentlyExpiredPwd** attribute type contains the old user password after it has been replaced during the **recentlyExpiredPwdDuration** period. During this period, this password and the **UserPwd** attribute are both considered to be valid. This attribute is removed when the **recentlyExpiredPwdDuration** period expires.

```
recentlyExpiredPwd ATTRIBUTE ::= {
    WITH SYNTAX                 OCTET STRING
    EQUALITY MATCHING RULE      octetStringMatch
    SINGLE VALUE                TRUE
    USAGE                       directoryOperation
    ID                          id-oa-recentlyExpiredPwd }
```

18.1.4.15 Recently Expired Encrypted Password

The **recentlyExpiredEncPwd** attribute type contains the old encrypted user password after it has been replaced during the **recentlyExpiredEncPwdDuration**. During this period, this encrypted password and the **encUserPwd** attribute are both considered to be valid. This attribute is removed when the **recentlyExpiredEncPwdDuration** expires.

```
recentlyExpiredEncPwd ATTRIBUTE ::= {
    WITH SYNTAX                 EncUserPwd
    EQUALITY MATCHING RULE      encUserPwdMatch
    USAGE                       directoryOperation
    ID                          id-oa-recentlyExpiredEncPwd }
```

18.1.4.16 Password Policy Subentry

The **pwdSubentry** operational attribute identifies the password policy subentry that governs the entry. It shall be available in every entry within the scope of a subentry.

```
pwdSubentry ATTRIBUTE ::= {
    WITH SYNTAX                 DistinguishedName
    EQUALITY MATCHING RULE      distinguishedNameMatch
    SINGLE VALUE                TRUE
    NO USER MODIFICATION        TRUE
    USAGE                       directoryOperation
    ID                          id-oa-pwdSubentry }
```

5) Subclause 18.1.5

Add a new subclause 18.1.5

18.1.5 Simple Authentication attributes held by object entries or subentries

Attributes of this type may be placed in an object entry and/or in a subentry. If an object entry holds such an attribute and is also within the scope of a password policy subentry, the value of the attribute in the object entry itself takes precedence.

18.1.5.1 ModifyEntry Password Allowed attribute

The **modifyEnryPwdAllowed** operational attribute specifies if the password or the encrypted password of an entry can be modified by an Administrattor with a Modify Entry operation. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, then the password or the encrypted password cannot be modified with a Modify Entry operation.

```
modifyEntryPwdAllowed ATTRIBUTE ::= {
        WITH SYNTAX                BOOLEAN
        EQUALITY MATCHING RULE     booleanMatch
        SINGLE VALUE               TRUE
        USAGE                      directoryOperation
        ID                         id-oa-modifyEntryPwdAllowed }
```

18.1.5.2 Change Password Allowed attribute

The **changePwdAllowed** operational attribute specifies if the password or the encrypted password of an entry can be modified by the owner of that entry with a Change Password operation. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, then the password or the encrypted password cannot be modified with a Change Password operation.

```
changePwdAllowed ATTRIBUTE ::= {
        WITH SYNTAX                BOOLEAN
        EQUALITY MATCHING RULE     booleanMatch
        SINGLE VALUE               TRUE
        USAGE                      directoryOperation
        ID                         id-oa-changePwdAllowed }
```

18.1.5.3 Password Maximum Age attribute

The **pwdMaxAge** operational attribute holds the number of seconds after which a modified password will be no longer available. It shall have a value greater than zero.

If an entry does not hold an attribute of this type and is not within the scope of a password policy subentry holding such an attribute, then a possible password is not under the rules of password policy.

```
pwdMaxAge ATTRIBUTE ::= {
        WITH SYNTAX                INTEGER (1 .. MAX)
        EQUALITY MATCHING RULE     integerMatch
        SINGLE VALUE               TRUE
        USAGE                      directoryOperation
        ID                         id-oa-pwdMaxAge }
```

18.1.5.4 Encrypted Password Maximum Age attribute

The **encPwdMaxAge** operational attribute holds the number of seconds after which a modified encrypted password will be no longer available. It shall have a value greater than zero.

If an entry does not hold an attribute of this type and is not within the scope of a password policy subentry holding such an attribute, then a possible encrypted password is not under the rules of password policy.

```
encPwdMaxAge ATTRIBUTE ::= {
        WITH SYNTAX                INTEGER (1 .. MAX)
        EQUALITY MATCHING RULE     integerMatch
        SINGLE VALUE               TRUE
        USAGE                      directoryOperation
        ID                         id-oa-encPwdMaxAge }
```

18.1.5.5 Password Expiry Age attribute

The **pwdExpiryAge** operational attribute holds the number of seconds after which a modified password will expire. It shall have a value greater than zero.

If an entry does not hold an attribute of this type and is not within the scope of a password policy subentry holding such an attribute, then a possible password is not under the rules of password policy.

```
pwdExpiryAge ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1 .. MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdExpiryAge }
```

18.1.5.6 Encrypted Password Expiry Age attribute

The **encPwdExpiryAge** operational attribute holds the number of seconds after which a modified encrypted password will expire. It shall have a value greater than zero.

If an entry does not hold an attribute of this type and is not within the scope of a password policy subentry holding such an attribute, then a possible encrypted password is not under the rules of password policy.

```
encPwdExpiryAge ATTRIBUTE ::= {
        WITH SYNTAX                  INTEGER (1 .. MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-encPwdExpiryAge }
```

18.1.5.7 Passwords Quality Rule attribute

The **pwdQualityRule** operational attribute holds an identification of a password quality rule.

```
pwdQualityRule ATTRIBUTE ::= {
        WITH SYNTAX                 OBJECT IDENTIFIER
        EQUALITY MATCHING RULE      objectIdentifierMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdQualityRule }
```

18.1.5.8 Password Expiry Warning attribute

The **pwdExpiryWarning** operational attribute specifies a period in seconds before password expiration. During this period a warning indication shall be returned whenever an authenticating requestor binds. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, a warning indication shall not be returned.

```
pwdExpiryWarning ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdExpiryWarning }
```

If the user does not attempt to bind during period, the account should be locked, but the user should have a chance to change the password.

18.1.5.9 Encrypted Password Expiry Warning attribute

The **encPwdExpiryWarning** operational attribute specifies a period in seconds before password expiration. During this period a warning indication shall be returned whenever an authenticating requestor binds. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, a warning indication shall not be returned.

```
encPwdExpiryWarning ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
```

```
        EQUALITY MATCHING RULE  integerMatch
        SINGLE VALUE            TRUE
        USAGE                   directoryOperation
        ID                      id-oa-pwdExpiryWarning }
```

If the user does not attempt to bind during period, the account should be locked, but the user should have a chance to change the password.

18.1.5.10 Password Grace Limit attribute

The **pwdGraces** operational attribute specifies the number of times an expired password can be used to authenticate. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, authentication shall fail.

```
pwdGraces ATTRIBUTE ::= {
        WITH SYNTAX             INTEGER (0..MAX)
        EQUALITY MATCHING RULE  integerMatch
        SINGLE VALUE            TRUE
        USAGE                   directoryOperation
        ID                      id-oa-pwdGraces }
```

18.1.5.11 Password Bind Lockout Duration attribute

The **pwdBindLockoutDuration** operational attribute holds the number of seconds that the password cannot be used to authenticate due to too many successive failed bind attempts (more than the limit specified by **pwdMaxBindFailures** operational attribute). If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, the password cannot be used to authenticate until reset by a password administrator.

```
pwdBindLockoutDuration ATTRIBUTE ::= {
        WITH SYNTAX             INTEGER (0..MAX)
        EQUALITY MATCHING RULE  integerMatch
        SINGLE VALUE            TRUE
        USAGE                   directoryOperation
        ID                      id-oa-pwdBindLockoutDuration }
```

18.1.5.12 Password Compare Lockout Duration attribute

The **pwdCompareLockoutDuration** operational attribute holds the number of seconds that the password cannot be used in a compare operation due to too many successive failed compare attempts (more than the limit specified by **pwdMaxCompareFailures** operational attribute). If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, there is no restriction on the number of failed compare operations.

```
pwdCompareLockoutDuration ATTRIBUTE ::= {
        WITH SYNTAX             INTEGER (0..MAX)
        EQUALITY MATCHING RULE  integerMatch
        SINGLE VALUE            TRUE
        USAGE                   directoryOperation
        ID                      id-oa-pwdCompareLockoutDuration }
```

18.1.5.13 Password Maximum Bind Failures attribute

The **pwdMaxBindFailures** operational attribute specifies the number of consecutive failed bind attempts after which the password may not be used to authenticate. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, there is no limit on failed attempts.

```
pwdMaxBindFailures ATTRIBUTE ::= {
        WITH SYNTAX             INTEGER (1..MAX)
        EQUALITY MATCHING RULE  integerMatch
        SINGLE VALUE            TRUE
        USAGE                   directoryOperation
        ID                      id-oa-pwdMaxBindFailures }
```

18.1.5.14 Password Maximum Compare Failures attribute

The **pwdMaxCompareFailures** operational attribute specifies the number of consecutive failed compare attempts after which the password may not be used in compare operations. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, there is no limit on failed compare operations.

```
pwdMaxCompareFailures ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdMaxCompareFailures }
```

18.1.5.15 PasswordTime in History attribute

The **pwdTimeInHistory** operational attribute specifies the delay, in number of seconds, during which a replaced password is kept within the **pwdHistory** operational attribute. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, then an implementation defined default should be used.

```
pwdTimeInHistory ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdHimeInHistory }
```

18.1.5.16 Password History slots attribute

The **pwdHistorySlots** operational attribute specifies the number of slots in the history which can be used to store replace passwords. The minimum number of slots is 2 because two slots are needed when an administrator has to reset a password.

```
pwdHistorySlots ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (2..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdHistorySlots }
```

18.1.5.17 Recently Expired Password Duration

The **recentlyExpiredPwdDuration** attribute type defines the period during which an expired password is kept in the **recentlyExpiredPwd** attribute.

```
recentlyExpiredPwdDuration ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-recentlyExpiredPwdDuration }
```

18.1.5.18 Recently Expired Encrypted Password Duration

The **recentlyExpiredEncPwdDuration** attribute type defines the period during which an expired encrypted password is kept in the **recentlyExpiredEncPwd** attribute.

```
recentlyExpiredEncPwdDuration ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-recentlyExpiredEncPwdDuration }
```

18.15.19 Password Bind Lockout attribute

The **pwdBindLockout** operational attribute indicates, when its value is **TRUE**, that the account shall be locked when a number consecutive failing bind attempts as specified in **pwdMaxBindFailures**. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, then the account shall not be locked.

```
pwdBindLockout ATTRIBUTE ::= {
        WITH SYNTAX                 BOOLEAN
```

```
        EQUALITY MATCHING RULE  booleanMatch
        SINGLE VALUE            TRUE
        USAGE                   directoryOperation
        ID                      id-oa-pwdBindLockout }
```

18.15.20 Password Compare Lockout attribute

The **pwdCompareLockout** operational attribute indicates, when its value is **TRUE**, that the account shall be locked when a number consecutive failing compare attempts as specified in **pwdMaxCompareFailures**. If an object entry does not hold such an attribute and it is not within the scope of a password policy subentry holding such an attribute, then the account shall not be locked.

```
pwdCompareLockout ATTRIBUTE ::= {
        WITH SYNTAX             BOOLEAN
        EQUALITY MATCHING RULE  booleanMatch
        SINGLE VALUE            TRUE
        USAGE                   directoryOperation
        ID                      id-oa-pwdCompareLockout }
```

18.15.21 Password Encryption Algorithm attribute

The **pwdEncryptionAlg** operational attribute indicates the algorithm to be used during the creation of an encrypted password.

```
pwdEncAlg ATTRIBUTE ::= {
        WITH SYNTAX             PwdEncAlg
        EQUALITY MATCHING RULE  pwdEncAlgMatch
        SINGLE VALUE            TRUE
        USAGE                   directoryOperation
        ID                      id-oa-pwdEncAlg }

PwdEncAlg ::= AlgorithmIdentifier{{SupportedAlgorithms}}
```

6) Subclause 18.1.6

Add a new subclause 18.1.6

18.1.6 Password History matching rule

The **pwdHistoryMatch** rule determines whether a proposed password is within the **pwdHistory** attribute by comparing the proposed password with the **password** component of each value of the **pwdHistory** attribute.

```
pwdHistoryMatch MATCHING-RULE ::= {
        SYNTAX    OCTET STRING
        ID        id-mr-pwdHistoryMatch }
```

7) Subclause 18.1.7

Add a new subclause 18.1.7

18.1.7 Encrypted User Password matching rule

The **encUserPwdMatch** rule determines whether a clear text password matches an encrypted password stored in the Directory.

```
encUserPwdMatch MATCHING-RULE ::= {
        SYNTAX    OCTET STRING
        ID        id-mr-encUserPwdMatch }
```

The presented octet string is encrypted using the encryption algorithm stored in the **encUserPwd** attribute and then compared with the octet string stored in the **encUserPwd** attribute.

8) Subclause 18.1.8

Add a new subclause 18.1.8

18.1.8 Password Encryption Algorithm matching rule

The **pwdEncAlgMatch** rule compares for equality a presented value with an attribute value of type **pwdEncAlg**. This rule returns TRUE if the algorithms, defined as object identifier values, are equal as if the parameters, defined as bit string values, are also equal.

```
pwdEncAlgMatch MATCHING-RULE ::= {
        SYNTAX      PwdEncAlg
        ID          id-mr-pwdEncAlgMatch }
```

9) Subclause 18.1.9

Add a new subclause 18.1.9

18.1.9 Password Policy Subentry object class

If a subentry contains password policy information, then its **objectClass** attribute shall contain the value **id-sc-pwdPolicySubentry .**

```
pwdPolicySubentry OBJECT-CLASS ::= {
        KIND  auxiliary
        ID    id-sc-pwdPolicySubentry }
```

A subentry of object class **pwdPolicySubentry** may contain password policy attributes relevant for the scope of the subentry. If, in case of subtree overlapping, several subentries are applicable to the same entry, the password policy of this entry is defined by the most restrictive conditions:

- **modifyEntryPwdAllowed**: **TRUE** if all subentries specify **TRUE, FALSE otherwise.**
- **changePwdAllowed**: **TRUE** if all subentries specify **TRUE, FALSE otherwise.**
- **pwdMaxAge**: minimum of the values.
- **encPwdMaxAge**: minimum of the values.
- **pwdExpiryAge**: minimum of the values.
- **encPwdExpiryAge**: minimum of the values.
- **pwdQualityRule**: the more restrictive rules (maximum of lengths, intersection of character sets, etc).
- **pwdExpiryWarning**: maximum of the values.
- **encPwdExpiryWarning**: maximum of the values.
- **pwdGraces**: minimum of the values.
- **pwdBindLockoutDuration**: maximum of the values.
- **pwdCompareLockoutDuration**: maximum of the values.
- **pwdMaxBindFailures**: minimum of the values.
- **pwdMaxCompareFailures**: minimum of the values.
- **pwdTimeInHistory**: maximum of the values.
- **pwdHistorySlots**: maximum of the values.
- **recentlyExpiredPwdDuration**: minimum of the values.
- **recentlyExpiredEncPwdDuration**: minimum of the values.
- **pwdBindLockout**: **FALSE** if all subentries specify **FALSE, TRUE otherwise.**
- **pwdCompareLockout**: **FALSE** if all subentries specify **FALSE, TRUE otherwise.**
- **pwdEncAlg**: the more secure algorithm.

10) SubClause 18.2.1

Rename the figures 10 and 11 to 15 and 16

11) SubClause 18.2.2.1

Rename the figure 12, 13 and 14 to 17, 18 and 19

12) Annex A.1

Replace the first three parts of IMPORTS clause with:

```
        id-at, id-nf, id-oa, id-mr, id-oc, id-sc, informationFramework, selectedAttributeTypes,
        basicAccessControl,certificateExtensions
                FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 6}

        Name, ATTRIBUTE, OBJECT-CLASS, NAME-FORM, top, MATCHING-RULE, DistinguishedName
                FROM InformationFramework informationFramework

        UniqueIdentifier, octetStringMatch, generalizedTimeMatch,
        generalizedTimeOrderingMatch,integerMatch, distinguishedNameMatch, booleanMatch,
        objectIdentifierMatch, commonName, UnboundedDirectoryString
                FROM SelectedAttributeTypes selectedAttributeTypes
```

Add the following definitions after ***userPassword*** *definition:*

```
userPwd    ATTRIBUTE::=    {
        WITH SYNTAX                     OCTET STRING
        EQUALITY MATCHING RULE          octetStringMatch
        SINGLE VALUE                    TRUE
        ID                              id-at-userPwd }

encUserPwd      ATTRIBUTE::=    {
        WITH SYNTAX                     EncUserPwd
        EQUALITY MATCHING RULE          encUserPwdMatch
        SINGLE VALUE                    TRUE
        ID                              id-at-encUserPwd }

EncUserPwd ::= SEQUENCE {
        encryptedString         OCTET STRING,
        algorithmIdentifier     AlgorithmIdentifier{{SupportedAlgorithms}}}

-- Operational attributes --

pwdCreationTime ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
        ORDERING MATCHING RULE          generalizedTimeOrderingMatch
        SINGLE VALUE                    TRUE
        USAGE                           directoryOperation
        ID                              id-oa-pwdCreationTime }

encPwdCreationTime ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
        ORDERING MATCHING RULE          generalizedTimeOrderingMatch
        SINGLE VALUE                    TRUE
        USAGE                           directoryOperation
        ID                              id-oa-encPwdCreationTime }

expPwdCreationTime ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
        ORDERING MATCHING RULE          generalizedTimeOrderingMatch
        SINGLE VALUE                    TRUE
        USAGE                           directoryOperation
        ID                              id-oa-expPwdCreationTime }

expEncPwdCreationTime ATTRIBUTE ::= {
        WITH SYNTAX                     GeneralizedTime
        EQUALITY MATCHING RULE          generalizedTimeMatch
```

```
        ORDERING MATCHING RULE      generalizedTimeOrderingMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-expEncPwdCreationTime }

pwdExpiryDate ATTRIBUTE ::= {
        WITH SYNTAX                 GeneralizedTime
        EQUALITY MATCHING RULE      generalizedTimeMatch
        ORDERING MATCHING RULE      generalizedTimeOrderingMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdExpiryDate }

encPwdExpiryDate ATTRIBUTE ::= {
        WITH SYNTAX                 GeneralizedTime
        EQUALITY MATCHING RULE      generalizedTimeMatch
        ORDERING MATCHING RULE      generalizedTimeOrderingMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-encPwdExpiryDate }

pwdEndDate ATTRIBUTE ::= {
        WITH SYNTAX                 GeneralizedTime
        EQUALITY MATCHING RULE      generalizedTimeMatch
        ORDERING MATCHING RULE      generalizedTimeOrderingMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdEndDate }

encPwdEndDate ATTRIBUTE ::= {
        WITH SYNTAX                 GeneralizedTime
        EQUALITY MATCHING RULE      generalizedTimeMatch
        ORDERING MATCHING RULE      generalizedTimeOrderingMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-encPwdEndDate }

pwdBindFails      ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdBindFails }

pwdCompareFails       ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdCompareFails }

remAuthAttempts ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-remAuthAttempts }

pwdHistory ATTRIBUTE ::= {
        WITH SYNTAX                 PwdHistory
        EQUALITY MATCHING RULE      pwdHistoryMatch
        USAGE                       directoryOperation
        ID                          id-oa-pwdHistory }

PwdHistory ::= SEQUENCE {
        time                        GeneralizedTime,
        password                    OCTET STRING }

recentlyExpiredPwd ATTRIBUTE ::= {
        WITH SYNTAX                 OCTET STRING
```

```
        EQUALITY MATCHING RULE      octetStringMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-recentlyExpiredPwd }

recentlyExpiredEncPwd ATTRIBUTE ::= {
        WITH SYNTAX                 EncUserPwd
        EQUALITY MATCHING RULE      encUserPwdMatch
        USAGE                       directoryOperation
        ID                          id-oa-recentlyExpiredEncPwd }

pwdSubentry ATTRIBUTE ::= {
        WITH SYNTAX                 DistinguishedName
        EQUALITY MATCHING RULE      distinguishedNameMatch
        SINGLE VALUE                TRUE
        NO USER MODIFICATION        TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdSubentry }

modifyEntryPwdAllowed ATTRIBUTE ::= {
        WITH SYNTAX                 BOOLEAN
        EQUALITY MATCHING RULE      booleanMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-modifyEntryPwdAllowed }

changePwdAllowed ATTRIBUTE ::= {
        WITH SYNTAX                 BOOLEAN
        EQUALITY MATCHING RULE      booleanMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-changePwdAllowed }

pwdMaxAge ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1 .. MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdMaxAge }

encPwdMaxAge ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1 .. MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-encPwdMaxAge }

pwdExpiryAge ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1 .. MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdExpiryAge }

encPwdExpiryAge ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1 .. MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-encPwdExpiryAge }

pwdQualityRule ATTRIBUTE ::= {
        WITH SYNTAX                 OBJECT IDENTIFIER
        EQUALITY MATCHING RULE      objectIdentifierMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdQualityRule }

pwdExpiryWarning ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
```

```
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdExpiryWarning }

encPwdExpiryWarning ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdExpiryWarning }

pwdGraces ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdGraces }

pwdBindLockoutDuration ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdBindLockoutDuration }

pwdCompareLockoutDuration ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdCompareLockoutDuration }

pwdMaxBindFailures ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdMaxBindFailures }

pwdMaxCompareFailures ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdMaxCompareFailures }

pwdTimeInHistory ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (1..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdTimeInHistory }

pwdHistorySlots ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (2..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdHistorySlots }

recentlyExpiredPwdDuration ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-recentlyExpiredPwdDuration }

recentlyExpiredEncPwdDuration ATTRIBUTE ::= {
        WITH SYNTAX                 INTEGER (0..MAX)
```

```
        EQUALITY MATCHING RULE      integerMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-recentlyExpiredEncPwdDuration }

pwdBindLockout ATTRIBUTE ::= {
        WITH SYNTAX                 BOOLEAN
        EQUALITY MATCHING RULE      booleanMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdBindLockout }

pwdCompareLockout ATTRIBUTE ::= {
        WITH SYNTAX                 BOOLEAN
        EQUALITY MATCHING RULE      booleanMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdCompareLockout }

pwdEncAlg ATTRIBUTE ::= {
        WITH SYNTAX                 PwdEncAlg
        EQUALITY MATCHING RULE      pwdEncAlgMatch
        SINGLE VALUE                TRUE
        USAGE                       directoryOperation
        ID                          id-oa-pwdEncAlg }

PwdEncAlg ::= AlgorithmIdentifier{{SupportedAlgorithms}}

-- Password History matching Rule --

pwdHistoryMatch MATCHING-RULE ::= {
        SYNTAX    OCTET STRING,
        ID        id-mr-pwdHistoryMatch }

-- Encrypted User Password matching Rule --

encUserPwdMatch MATCHING-RULE ::= {
        SYNTAX    OCTET STRING,
        ID        id-mr-encUserPwdMatch }

-- Password Encryption Algorithm matching Rule --

pwdEncAlgMatch MATCHING-RULE ::= {
        SYNTAX    pwdEncAlg,
        ID        id-mr-pwdEncAlgMatch }

-- Password Subentry Object class --

pwdPolicySubentry OBJECT-CLASS ::= {
        KIND      auxiliary
        ID        id-sc-pwdPolicySubentry }
```

Add the following definition before the line "--name forms --":

```
-- subentry classes --

id-sc-pwdPolicySubentry                 OBJECT IDENTIFIER   ::=   {id-sc 5}
```

Add the following definitions after id-at-pkiPath:

```
id-at-userPwd                           OBJECT IDENTIFIER   ::=   {id-at 83}
id-at-encUserPwd                        OBJECT IDENTIFIER   ::=   {id-at 84}
```

Add the following definitions before the line "END":

```
-- operational attributes --

id-oa-pwdCreationTime                   OBJECT IDENTIFIER   ::=   {id-oa 21}
id-oa-encPwdCreationTime                OBJECT IDENTIFIER   ::=   {id-oa 22}
id-oa-expPwdCreationTime                OBJECT IDENTIFIER   ::=   {id-oa 23}
```

```
id-oa-expEncPwdCreationTime          OBJECT IDENTIFIER   ::=   {id-oa 24}
id-oa-pwdExpiryDate                  OBJECT IDENTIFIER   ::=   {id-oa 25}
id-oa-encPwdExpiryDate               OBJECT IDENTIFIER   ::=   {id-oa 26}
id-oa-pwdEndDate                     OBJECT IDENTIFIER   ::=   {id-oa 27}
id-oa-encPwdEndDate                  OBJECT IDENTIFIER   ::=   {id-oa 28}
id-oa-pwdBindFails                   OBJECT IDENTIFIER   ::=   {id-oa 29}
id-oa-pwdCompareFails                OBJECT IDENTIFIER   ::=   {id-oa 30}
id-oa-remAuthAttempts                OBJECT IDENTIFIER   ::=   {id-oa 31}
id-oa-pwdHistory                     OBJECT IDENTIFIER   ::=   {id-oa 32}
id-oa-recentlyExpiredPwd             OBJECT IDENTIFIER   ::=   {id-oa 33}
id-oa-recentlyExpiredEncPwd          OBJECT IDENTIFIER   ::=   {id-oa 34}
id-oa-recentlyExpiredPwdDuration     OBJECT IDENTIFIER   ::=   {id-oa 35}
id-oa-recentlyExpiredEncPwdDuration  OBJECT IDENTIFIER   ::=   {id-oa 36}
id-oa-pwdSubentry                    OBJECT IDENTIFIER   ::=   {id-oa 37}
id-oa-modifyEntryPwdAllowed          OBJECT IDENTIFIER   ::=   {id-oa 38}
id-oa-changePwdAllowed               OBJECT IDENTIFIER   ::=   {id-oa 39}
id-oa-pwdMaxAge                      OBJECT IDENTIFIER   ::=   {id-oa 40}
id-oa-encPwdMaxAge                   OBJECT IDENTIFIER   ::=   {id-oa 41}
id-oa-pwdExpiryAge                   OBJECT IDENTIFIER   ::=   {id-oa 42}
id-oa-encPwdExpiryAge                OBJECT IDENTIFIER   ::=   {id-oa 43}
id-oa-pwdQualityRule                 OBJECT IDENTIFIER   ::=   {id-oa 44}
id-oa-pwdExpiryWarning               OBJECT IDENTIFIER   ::=   {id-oa 45}
id-oa-encPwdExpiryWarning            OBJECT IDENTIFIER   ::=   {id-oa 46}
id-oa-pwdGraces                      OBJECT IDENTIFIER   ::=   {id-oa 47}
id-oa-pwdBindLockoutDuration         OBJECT IDENTIFIER   ::=   {id-oa 48}
id-at-pwdCompareLockoutDuration      OBJECT IDENTIFIER   ::=   {id-oa 49}
id-oa-pwdMaxBindFailures             OBJECT IDENTIFIER   ::=   {id-oa 50}
id-oa-pwdMaxCompareFailures          OBJECT IDENTIFIER   ::=   {id-oa 51}
id-oa-pwdTimeInHistory               OBJECT IDENTIFIER   ::=   {id-oa 52}
id-oa-pwdHistorySlots                OBJECT IDENTIFIER   ::=   {id-oa 53}
id-oa-pwdBindLockout                 OBJECT IDENTIFIER   ::=   {id-oa 54}
id-oa-pwdCompareLockout              OBJECT IDENTIFIER   ::=   {id-oa 55}
-- matching rules
id-mr-pwdHistoryMatch                OBJECT IDENTIFIER   ::=   {id-mr 70}
id-mr-encUserPwdMatch                OBJECT IDENTIFIER   ::=   {id-mr 71}
id-mr-pwdEncAlgMatch                 OBJECT IDENTIFIER   ::=   {id-mr 72}
```

13) Annex L

Add a new Annex L containing the following text and rename current annexes L and M to M and N

Annex L

Examples of password encryption algorithms

L.1 SHA-1 method

The encrypted password is an octet string of 16 octets which is the MD5 digest of the concatenation of the clear password and the salt which is a bit string, parameter of the algorithm. This encryption method is defined by the following object:

```
mD5Algorithm ALGORITHM ::= { BIT STRING IDENTIFIED BY
                {iso(1) member-body(2) us(840) rsadsi(113549) digestAlgorithm(2) md5(5)}}
```

L.2 SHA-1 method

The encrypted password is an octet string of 20 octets which is the SHA-1 digest of the concatenation of the clear password and the salt which is a bit string, parameter of the algorithù. This encryption method is defined by the following object:

```
sha1Algorithm ALGORITHM ::= { BIT STRING IDENTIFIED BY
{iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 26}}
```

ISO/IEC 9594-9: 2008, Information Technology – Open Systems Interconnection – The Directory: Replication

1) Subclause 9.2.2

Add after the third paragraph the following text:

The following attributes shall be provided by the shadow supplier in the shadowed information (entries and subentries):

- modifyEntryPwdAllowed
- changePwdAllowed
- pwdMaxAge
- encPwdMaxAge
- pwdExpiryAge
- encPwdExpiryAge
- pwdQualityRule
- pwdExpiryWarning
- encPwdExpiryWarning
- pwdGraces
- pwdBindLockoutDuration
- pwdCompareLockoutDuration
- pwdMaxBindFailures
- pwdMaxCompareFailures
- pwdTimeInHistory
- pwdHistorySlots
- recentlyExpiredPwdDuration
- recentlyExpiredEncPwdDuration
- pwdBindLockout
- pwdCompareLockout
- pwdEncAlg

The following attributes shall be provided by the shadow supplier in the shadowed information (entries):

- pwdCreationTime
- encPwdCreationTime
- expPwdCreationTime
- expEncPwdCreationTime
- pwdExpiryDate
- encPwdExpiryDate
- pwdEndDate
- encPwdEndDate
- pwdBindFails
- pwdCompareFails
- remAuthAttempts
- pwdHistory
- recentlyExpiredPwd
- recentlyExpiredEncPwd
- pwdSubentry

2) Subclause 9.2.4

Replace the text of the existing subclause 9.2.4 with:

9.2.4 Subentries

Subentries are included in the unit of replication for access control, schema, collective attributes, contexts defaults, search-rules and password policy as described below.

3) Subclause 9.2.4.5

Add the new subclause 9.2.4.5

9.2.4.5 Password policy information

To have the password policy enforced by the shadow consumer, the **pwdPolicySubentry** subentries shall be included in the unit of replication.

ISO/IEC 9594-10: 2008, Information Technology – Open Systems Interconnection – The Directory: Use of systems management for administration of the Directory

No change