TCVN TIÊU CHUẨN QUỐC GIA

TCVN 11167-15:2015
ISO/IEC 7816-15:2004 WITH AMENDMENT 1:2007 AND AMENMENT 2:2008

Xuất bản lần 1

THẺ DANH ĐỊNH - THẺ MẠCH TÍCH HỢP - PHẦN 15: ỨNG DỤNG THÔNG TIN MÃ HÓA

Identification cards - Integrated circuit cards - Part 15: Cryptographic information application

HÀ NỘI - 2015

Mục lục **Trang**

Lời nói đầu

TCVN 11167-15:2015 hoàn toàn tương đương với ISO/IEC 7816-15:2004, ISO/IEC 7816-15:2004/Amd.1:2007, ISO/IEC 7816-15:2004/Amd.2:2008, ISO/IEC 7816-15:2004/Cor.1:2004.

TCVN 11167-15:2015 do Tiểu Ban kỹ thuật tiêu chuẩn quốc gia TCVN/JTC 1/SC 17 *"Thẻ nhận dạng"* biên soạn, Tổng cục Tiêu chuẩn Đo lường Chất lượng đề nghị, Bộ Khoa học và Công nghệ công bố.

Bộ tiêu chuẩn TCVN 11167 (ISO/IEC 7816) *Thẻ định danh – Thẻ mạch tích hợp* gồm các tiêu chuẩn sau:

- Phần 1: Thẻ tiếp xúc - Đặc tính vật lý;
- Phần 2: Thẻ tiếp xúc - Kích thước và vị trí tiếp xúc;
- Phần 3: Thẻ tiếp xúc - Giao diện điện và giao thức truyền;
- Phần 4: Tổ chức, an ninh và lệnh trao đổi;
- Phần 5: Đăng ký của bên cung cấp ứng dụng;
- Phần 6: Phần tử dữ liệu liên ngành trong trao đổi;
- Phần 7: Lệnh liên ngành đối với ngôn ngữ truy vấn thẻ có cấu trúc;
- Phần 8: Lệnh đối với hoạt động an ninh;
- Phần 9: Lệnh đối với quản lý thẻ;
- Phần 10: Tín hiệu điện và trả lời để thiết lập lại cho thẻ đồng bộ;
- Phần 11: Xác minh cá nhân bằng phương pháp sinh trắc học;
- Phần 12: Thẻ tiếp xúc - Thủ tục vận hành và giao diện điện tử USB;
- Phần 13: Lệnh đối với quản lý ứng dụng trong môi trường đa ứng dụng;
- Phần 15: Ứng dụng thông tin mã hóa.

Thẻ định danh – Thẻ mạch tích hợp – Phần 15: Ứng dụng thông tin mã hóa

Identification cards – Integrated circuit cards with contacts – Part 15: Cryptographic information application

1 Phạm vi áp dụng

Tiêu chuẩn này quy định về ứng dụng trong một thẻ. Ứng dụng này có chứa thông tin về chức năng mã hóa. Tiêu chuẩn này định nghĩa cú pháp chung và định dạng cho thông tin mã hóa và cơ chế mã hóa để chia sẻ thông tin này khi thích hợp.

Phạm vi của tiêu chuẩn này nhằm đáp ứng:

- Dễ dàng liên tác giữa các thành phần chạy trên nhiều nền tảng (độc lập với nền tảng);
- Cho phép các ứng dụngngoài hệ thống tận dụng các sản phẩm và thành phần từ các bên sản xuất (độc lập với bên cung cấp);
- Cho phép tận dụng công nghệ mà không cần viết lại phần mềm mức ứng dụng (độc lập với ứng dụng);
- Duy trì tính kiên định với các tiêu chuẩn sẵn có, liên quan trong khi phát triển nhờ chúng chỉ khi cần thiết và trên thực tiễn.

Phạm vi của tiêu chuẩn này hỗ trợ các khả năng sau:

- Lưu trữ nhiều trường hợp thông tin mã hóa trên một thẻ;
- Sử dụng thông tin mã hóa;
- Thu nhận thông tin mã hóa, nhân tố chính cho việc này là khái niệm "tệp tin thư mục", nhằm tạo một lớp gián tiếp giữa đối tượng trên thẻ và định dạng thực tế của các đối tượng đó;
- Tham chiếu chéo của thông tin mã hóa với các DO được quy định trong các tiêu chuẩn tương ứng khác trong bộ TCVN 11167 (ISO/IEC 7816);
- Các cơ chế xác minh khác nhau;

- Thuật toán đa mã hóa (sự phù hợp của các thuật toán không được đề cập trong phạm vi của bộ tiêu chuẩn này).

Phạm vi của tiêu chuẩn này không đề cập về việc triển khai bên trong và/hoặc bên ngoài. Không bắt buộc phải triển khai sự phù hợp với tiêu chuẩn này khi hỗ trợ tất cả tùy chọn đã được mô tả.

Trong trường hợp có sự khác nhau giữa các quy định của ASN.1 trong nội dung văn bản và các mô đun trong Phụ lục A, Phụ lục A được ưu tiên.

2 Tài liệu viện dẫn

Các tài liệu viện dẫn sau rất cần thiết cho việc áp dụng tiêu chuẩn này. Đối với các tài liệu viện dẫn ghi năm công bố thì áp dụng phiên bản được nêu. Đối với các tài liệu viện dẫn không ghi năm công bố thì áp dụng phiên bản mới nhất, bao gồm cả các sửa đổi, bổ sung (nếu có).

TCVN 11167 (ISO/IEC 7816) Thẻ định danh - Thẻ mạch tích hợp tiếp xúc (tất cả các phần).

ISO/IEC 8824-1:1998, Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation

ISO/IEC 8824-2:1998, Information technology - Abstract Syntax Notation One (ASN.1): Information object specification

ISO/IEC 8824-3:1998, Information technology - Abstract Syntax Notation One (ASN.1): Constraint specification

ISO/IEC 8824-4:1998, Information technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications

ISO/IEC 8825-1:1998, Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

ISO 9564-1:2002, Banking - Personal IDentification Number (PIN) management and security - Part 1: Basic principles and requirements for online PIN handling in ATM and POS systems

ISO/IEC 9594-8:1998, Information technology - Open Systems Interconnection - The Directory: Authentication framework

ISO/IEC 10646-1:2000, Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane

ANSI X9.42-2001, Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography

ANSI X9.62-1998, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)

3 Thuật ngữ và định nghĩa

Trong phạm vi của tiêu chuẩn này, các thuật ngữ và định nghĩa sau được áp dụng.

3.1

Đường dẫn tuyệt đối (absolute path)

Đường dẫn bắt đầu với định danh tệp tin '3F00'.

3.2

Ứng dụng (application)

Cấu trúc dữ liệu, phần tử dữ liệu và mô-đun chương trình cần thiết để thực hiện một chức năng cụ thể.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.3

Định danh ứng dụng (application identifier)

Phần tử dữ liệu định danh một ứng dụng trong một thẻ.

CHÚ THÍCH Phù hợp với TCVN 11167-4 (ISO/IEC 7816-4).

3.4

Bên cung cấp ứng dụng (application provider)

Thực thể cung cấp các thành phần được yêu cầu nhằm thực hiện một ứng dụng trên thẻ.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.5

Đối tượng thông tin chứng thực (authentication information object)

Đối tượng thông tin mã hóa cung cấp thông tin về dữ liệu liên quan tới chứng thực, ví dụ: mật khẩu.

3.6

Tệp tin thư mục đối tượng chứng thực (authentication object directory file)

Tệp tin cơ bản bao gồm các đối tượng thông tin chứng thực.

3.7

Số thập phân được mã hóa nhị phân (binary coded decimal)

Biểu diễn số học trong đó một số được biểu diễn như một chuỗi các số thập phân và mỗi số thập phân này được mã hóa thành một số nhị phân 4 bit.

3.8

Chủ thẻ (cardholder)

Cá nhân sở hữu thẻ được phát hành.

3.9

Bên phát hành thẻ (card issuer)

Tổ chức hay thực thể phát hành thẻ.

3.10

tệp tin thư mục chứng chỉ (certificate directory file)

Tệp tin cơ bản chứa các đối tượng thông tin chứng chỉ.

3.11

Đối tượng thông tin chứng chỉ (certificate information object)

Đối tượng thông tin mã hóa cung cấp thông tin về một chứng chỉ.

3.12

Lệnh (command)

Thông điệp chỉ ra một hành động và kêu gọi một hồi đáp từ thẻ.

3.13

Ứng dụng thông tin mã hóa (cryptographic information application)

Ứng dụng trong một thẻ mà chứa thông tin về các đối tượng thông tin mã hóa, các phần tử dữ liệu an ninh khác và mục đích sử dụng của chúng.

3.14

Đối tượng thông tin mã hóa (cryptographic information object)

Thông tin có cấu trúc nằm trong một CIA, mô tả một phần tử dữ liệu mã hóa, ví dụ: khóa công khai hay chứng chỉ.

3.15

Đối tượng thông tin bộ chứa dữ liệu (data container information object)

Đối tượng thông tin mã hóa cung cấp thông tin về một bộ chứa dữ liệu, ví dụ: tệp tin.

3.16

Tệp tin thư mục đối tượng bộ chứa dữ liệu (data container object directory file)

Tệp tin cơ bản gồm các đối tượng thông tin bộ chứa dữ liệu.

3.17

Tệp tin chuyên dụng (dedicated file)

Cấu trúc bao gồm thông tin kiểm soát tệp tin và tính sẵn sàng định vị của bộ nhớ được tùy chọn.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.18

Tệp tin thư mục (DIR) (directory (DIR) file)

Tệp tin cơ bản tùy chọn bao gồm một danh sách các ứng dụng hỗ trợ bởi thẻ và các phần tử dữ liệu liên quan tùy chọn.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.19

Tệp tin cơ bản (elementary file)

Tập các đơn vị dữ liệu, bản ghi hay đối tượng dữ liệu chia sẻ cùng định danh tệp tin và có (các) thuộc tính giống nhau.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.20

Định danh tệp tin (file identifier)

Phần tử dữ liệu (2 byte) dùng để chỉ ra một tệp tin

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.21

Chức năng (function)

Quy trình xử lý được hoàn thiện bởi một hay nhiều lệnh và các hành động có kết quả.

3.22

Tệp tin chủ (master file)

MF

Tệp tin chuyên dụng đơn nhất thể hiện gốc trong một thẻ, sử dụng một phân nhánh của các tệp tin chuyên dụng.

[TCVN 11167-4 (ISO/IEC 7816-4)]

CHÚ THÍCH Tệp MF có định danh tệp tin là '3F00'.

3.23

Thông điệp (message)

Chuỗi các byte được truyền bởi thiết bị giao tiếp tới thẻ và ngược lại, không bao gồm các kí tự định hướng truyền.

3.24

Tệp tin thư mục đối tượng (object directory file)

Tệp tin cơ bản bắt buộc chứa các thông tin và các tệp tin thư mục CIA khác.

3.25

Mật khẩu (password)

Dữ liệu mà có thể được yêu cầu bởi ứng dụng được gửi tới thẻ bởi chính người dùng với mục đích chứng thực.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.26

Đường dẫn (path)

Tổ hợp các định danh tệp tin mà không cần phân định.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.27

Tệp tin thư mục khóa riêng (private key directory file)

Tệp tin cơ bản chứa các đối tượng thông tin khóa riêng.

3.28

Đối tượng thông tin khóa riêng (private key information object)

Đối tượng thông tin mã hóa, cung cấp thông tin về một khóa riêng.

3.29

Bên cung cấp (provider)

Người có thẩm quyền hay người có quyền tạo ra một tệp tin chuyên dụng trong thẻ.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.30

Tệp tin thư mục khóa công khai (public key directory file)

Tệp tin cơ bản bao gồm các đối tượng thông tin khóa công khai.

3.31

Đối tượng thông tin khóa công khai (public key information object)

Đối tượng thông tin mã hóa mà cung cấp thông tin về một khóa công khai.

3.32

Bản ghi (record)

Chuỗi các byte được tham chiếu và quản lý bởi thẻ trong một tệp tin cơ bản của cấu trúc bản ghi.

[TCVN 11167-4 (ISO/IEC 7816-4)]

3.33

Đường dẫn tương đối (relative path)

Đường dẫn bắt đầu với định danh tệp tin của DF hiện tại.

3.34

tệp tin thư mục khóa bí mật (secret key directory file)

Tệp tin cơ bản chứa các đối tượng thông tin khóa bí mật.

3.35

Đối tượng thông tin khóa bí mật (secret key information object)

Đối tượng thông tin mã hóa cung cấp thông tin về một khóa bí mật.

3.36

Khuôn mẫu (template)

Tập các đối tượng dữ liệu tạo thành trường giá trị của một đối tượng dữ liệu có cấu trúc.

CHÚ THÍCH Phù hợp với TCVN 11167-6 (ISO/IEC 7816-6).

4 Thuật ngữ viết tắt

4.1 Kí hiệu

DF.*x*	Tệp tin chuyên dụng x, trong đó x là từ viết tắt của tệp tin
EF.*x*	Tệp tin cơ bản x, trong đó x là từ viết tắt của tệp tin
'0' -'9' và 'A' - 'F'	Các số thuộc hệ cơ số 16

4.2 Thuật ngữ viết tắt

Thuật ngữ	Tiếng Anh	Tiếng Việt
AID	Application IDentifier	Mã định danh ứng dụng
AOD	Authentication object directory	Thư mục đối tượng chứng thực
BCD	Binary-coded decimal	Số thập phân mã hóa nhị phân
CD	Certificate directory	Thư mục chứng chỉ
CDE	Cryptographic data element	Phần tử dữ liệu mã hóa
CIA	Cryptographic information application	Ứng dụng thông tin mã hóa
CIO	Cryptographic information object	Đối tượng thông tin mã hóa
CV	Card-verifiable	Thẻ có thể xác minh
DCOD	Data container object directory	Thư mục đối tượng bộ chứa dữ liệu
DDO	Discretionary data object	Đối tượng dữ liệu tùy ý
DF	dedicated file	Tệp tin dành riêng
DH	Diffie-Hellman	Diffie-Hellman
DSA	Digital Signature Algorithm	Thuật toán chữ ký số
EC	Elliptic Curve	Cung e-líp
EF	Elementary file	Tệp tin cơ bản
IDO	Interindustry data object	Đối tượng dữ liệu liên ngành
IFD	Interface device	Thiết bị giao tiếp
KEA	Key Exchange Algorithm	Thuật toán trao đổi khóa
MF	Master file	Tệp tin chủ
OD	Object directory	Đường dẫn đối tượng
PKCS	Public-key cryptography standard	Tiêu chuẩn mã hóa khóa công khai
PrKD	Private key directory	Thư mục khóa riêng
PuKD	Public key directory	Thư mục khóa công khai

Thuật ngữ	Tiếng Anh	Tiếng Việt
RSA	Rivest-Shamir-Adleman	Rivest-Shamir-Adleman
SKD	Secret key directory	Thư mục khóa bí mật
SPKI	Simple Public Key Infrastructure	Hạ tầng khóa công khai đơn giản
UCS	Universal multiple-octet coded character set	Tập kí tự mã hóa đa octet phổ cập
URL	Uniform resource locator	Bộ định vị tài nguyên đồng nhất
UTC	Coordinated universal time	Giờ phối hợp quốc tế
UTF-8	UCS transformation format 8	Chuyển đổi UCS định dạng 8
WTLS	Wireless Application Protocol transport layer security	An ninh tầng giao vận giao thức ứng dụng không dây

5 Quy đổi

Tiêu chuẩn này trình bày kí hiệu ASN.1 dưới dạng **in đậm kiểu Helvetica**. Khi các loại và giá trị của ASN.1 được tham chiếu dưới dạng văn bản thông thường, tạo sự khác biệt giữa văn bản thông thường bằng cách thể hiện chúng theo dạng **in đậm kiểu Helvetica**. Tên của các lệnh, được tham chiếu chuẩn tắc khi quy định thông tin trao đổi giữa các thẻ và các IFD, tạo sự khác biệt với văn bản thông thường bằng cách hiển thị chúng dưới phông Courier.

Nếu các mục tin trong một danh sách được đánh số (khác với cách dùng "-" hoặc các chữ cái), thì các mục tin phải được xem xét như các bước trong một thủ tục.

6 Đối tượng thông tin mã hóa

6.1 Giới thiệu

Tiêu chuẩn này cung cấp:

- Mô tả các đối tượng mô tả thông tin mã hóa được chứa trong thẻ;
- Mô tả mục đích sử dụng của thông tin này;
- Cách thức lấy thông tin này (nếu phù hợp);
- Một cú pháp trừu tượng cho thông tin cung cấp cơ sở cho việc mã hóa;
- Một mô hình đối tượng cho thông tin.

Thông tin cũng bao gồm thông tin kiểm soát truy cập, được mô tả dưới dạng các CIO.

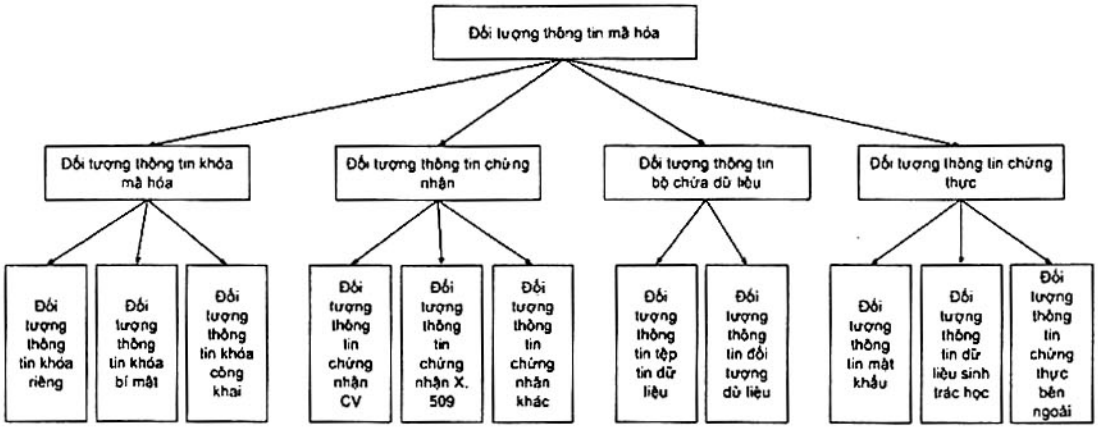
6.2 Lớp CIO

Tiêu chuẩn này định nghĩa 4 lớp của các CIO:

- Đối tượng thông tin khóa mã hóa;

- Đối tượng thông tin chứng chỉ;
- Đối tượng thông tin bộ chứa dữ liệu; và
- Đối tượng thông tin chứng thực.

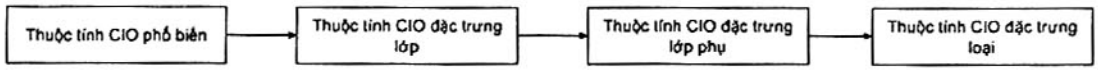
Cấu trúc logic của các CIO này được thể hiện trong Hình 1. Lớp đối tượng của các đối tượng thông tin khóa mã hóa có 3 lớp phụ: đối tượng thông tin của khóa riêng, khóa bí mật và khóa công khai. CIO kế thừa các thuộc tính từ các lớp mức cao và có thể được khởi tạo trên thẻ.



Hình 1 - Phân nhánh lớp CIO

6.3 Thuộc tính

Tất cả CIO có một số thuộc tính. Thuộc tính đặc trưng loại thường được xem xét. Thuộc tính đặc trưng nhóm và thuộc tính phổ thông với tất cả CIO có thể kế thừa được trình bày trong Hình 2. Các thuộc tính được quy định trong Điều 8.



Hình 2 - Khái niệm kế thừa thuộc tính

6.4 Hạn chế truy cập

Các CDE có thể được giữ bí mật, nghĩa là chúng được bảo vệ chống lại truy cập không chứng thực hay truy cập công khai. Quyền truy cập (đọc, viết, .v..v.) với các CDE cá nhân được mô tả bởi *Đối tượng thông tin chứng thực* (cũng bao gồm *Thủ tục chứng thực*). Truy cập có điều kiện (theo quan điểm của chủ thẻ) đạt được với thông tin người dùng dựa trên hiểu biết, thông tin người dùng sinh trắc học hay các cách thức mã hóa. Các CDE công khai không được bảo vệ khỏi quyền truy cập đọc.

7 Các điều kiện

7.1 Tổng quan

Một CIO chứa một tệp tin cơ bản và tham chiếu chung đến một CDE; một CIO có thể có một vài trường hợp chứa trực tiếp CDE. Một tệp tin chuyên dụng (DF.CIA) bao gồm các tệp tin cơ bản CIO. Các tệp tin CIO hiện tại có thể được xem xét dưới các tệp tin chuyên dụng khác trong trường hợp chúng được tham chiếu từ DF.CIA.

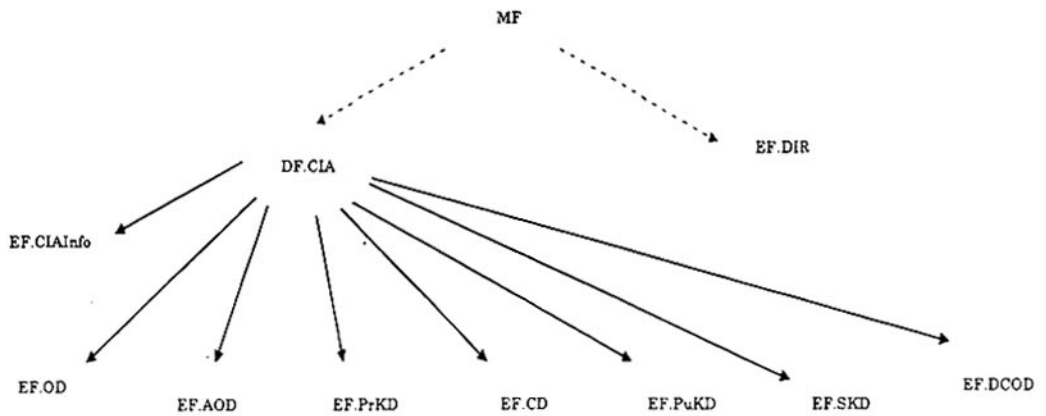
7.2 Yêu cầu thẻ IC

Các thẻ phải phù hợp với các phần của bộ tiêu chuẩn này, khi dùng:

- Các hệ thống tệp tin logic phân nhánh;
- Lựa chọn ứng dụng trực tiếp hay gián tiếp;
- Các cơ chế kiểm soát truy cập;
- Các thao tác đọc;
- Các thao tác mã hóa.

7.3 Cấu trúc tệp tin thẻ

Một thẻ điển hình hỗ trợ tiêu chuẩn này có bố cục như sau:



CHÚ THÍCH Trong phạm vi của tiêu chuẩn này, EF.DIR chỉ cần thiết trên các thẻ mà không hỗ trợ chọn lựa ứng dụng dùng AID như tên DF được quy định trong TCVN 11167-4 (ISO/IEC 7816-4) hoặc khi nhiều CIA nằm trên một thẻ riêng lẻ.

Hình 3 - Ví dụ về nội dung của DF.CIA

Các mô hình tô-pô khác được đề cập trong Phụ lục C. Nội dung và mục đích của mỗi tệp tin và đường dẫn được mô tả bên dưới.

7.4 EF.DIR

Tệp tin theo MF (định danh tệp tin: '2F00') phải gồm một hay một vài mẫu ứng dụng được quy định trong TCVN 11167-4 (ISO/IEC 7816-4). Mẫu ứng dụng (thẻ '61') với một CIA phải gồm ít nhất các IDO sau:

- Định danh ứng dụng (thẻ 'AF'), giá trị được quy định trong Điều 7.5.5.
- Đường dẫn (thẻ '51'), giá trị được cung cấp bởi bên cung cấp ứng dụng.

Các IDO khác của TCVN 11167-4 có thể có theo nhận thức của bên cung cấp ứng dụng. Thực tế, điều này được khuyến nghị cho bên cung cấp ứng dụng bao gồm vả đối tượng dữ liệu của "đối tượng dữ liệu tùy ý" (thẻ '73') và đối tượng dữ liệu của "nhãn ứng dụng" (thẻ '50'). Nhãn ứng dụng phải gồm một nhãn được mã hóa theo UTF-8 với các ứng dung, được chọn lựa bởi bên cung cấp ứng dụng. Đối tượng dữ liệu "đối tượng dữ liệu tùy ý" phải gồm một giá trị DER mã hóa (ISO/IEC 8825-1:1998) của ASN.1 loại **CIODDO**;

```
CIODDO ::= SEQUENCE {
    providerID          OBJECT IDENTIFIER OPTIONAL,
    odfPath         Path OPTIONAL,
    ciaInfoPath         [0] Path OPTIONAL,
    aid                 [APPLICATION 15] OCTET STRING (SIZE(1..16)),
                        (CONSTRAINED BY {-- Phải là một AID liên quan tới TCVN 11167-4
                        (ISO/IEC 7816-4)--})
                        OPTIONAL,
    ... -- Cho việc mở rộng sau này
    } -- Thẻ ngữ cảnh 1 theo lịch sử và không được dùng
```

CHÚ THÍCH 1 PKCS #15 dùng thẻ này.

CHÚ THÍCH 2 Theo TCVN 11167-4 (ISO/IEC 7816-4) và khi xem xét trong một mẫu ứng dụng, thẻ **[APPLICATION 19]**

('73') thay thế thẻ **CIODDO SEQUENCE** ('30') dựa vào việc đánh dấu hàm ẩn. Xem Điều D.8 với một ví dụ.

Thành phần **providerID** phải gồm một định danh đối tượng định danh đơn nhất bên cung cấp CIA. Các thành phần **odfPath** và **ciaInfoPath** phải gồm các đường dẫn tới các tệp tin cơ bản EF.OD và EF.CIAInfo tương ứng. Điều này đưa ra một cách thức với bên phát hành nhằm sử dụng các định danh tệp tin phi chuẩn với các tệp tin này mà không cần hi sinh khả năng liên vận. Nó cũng cung cấp bên phát hành thẻ với cơ hội chia sẻ các tệp tin CIAInfo giữa các CIA, khi một vài CIA nằm trên một thẻ. Thành phần **aID** phải chỉ ra ứng dụng cho CIA này áp dụng.

Việc dùng một tệp tin DIR sẽ chọn lựa ứng dụng đơn giản khi một vài CIA nằm trên một thẻ. Việc dùng các tệp tin DIR này được mô tả trong TCVN 11167-4 (ISO/IEC 7816-4).

7.5 Nội dung của DF.CIA

7.5.1 Tổng quan

Bảng 1 liệt kê các tệp tin cơ bản (bắt buộc và tùy chọn) trong DF.CIA cùng với các định danh tệp tin nghịch của chúng. Các loại tệp tin (bản ghi tuyến tính hoặc trong suốt) được chỉ ra trong cột cuối cùng

Bảng 1 - Tệp tin cơ bản trong DF.CIA

Tệp tin	Bắt buộc	Định danh tệp tin (mặc định)	Định danh EF ngắn	Loại tệp tin
CIAInfo	x	'5032'	'12'	Trong suốt
OD	x	'5031'	'11'	Bản ghi tuyến tính hoặc trong suốt
Các AOD				Bản ghi tuyến tính hoặc trong suốt
Các PrKD				Bản ghi tuyến tính hoặc trong suốt
Các PuKD				Bản ghi tuyến tính hoặc trong suốt
Các SKD				Bản ghi tuyến tính hoặc trong suốt
Các CD				Bản ghi tuyến tính hoặc trong suốt
Các DCOD				Bản ghi tuyến tính hoặc trong suốt
-		'5033'	-	Bản ghi tuyến tính hoặc trong suốt

7.5.2 EF CIAInfo

CIAInfo EF phải bao gồm thông tin về thẻ và các khả năng của nó, liên quan tới việc sử dụng các CIO. Thông tin sau phải luôn có:

- Số phiên bản; và
- Đặc tính thẻ.

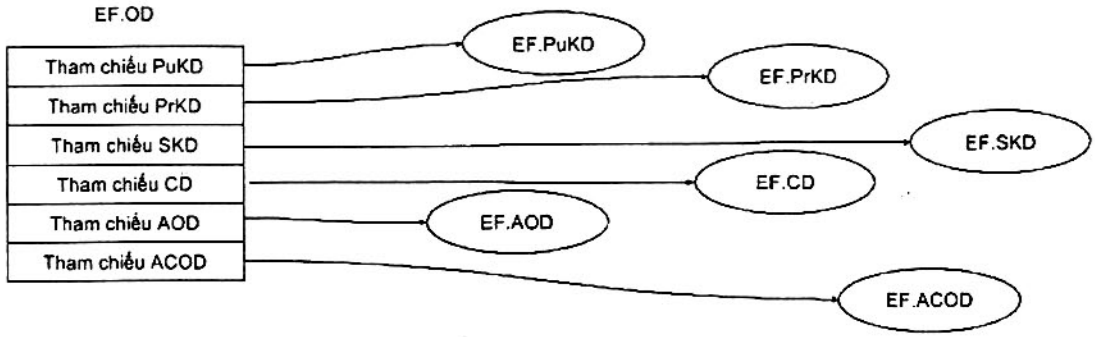
Thông tin sau có thể được tìm thấy:

- Số se-ri CIA;
- Định danh nhà sản xuất;
- Nhãn thẻ;
- Môi trường an ninh được định vị;
- Cấu trúc tệp tin;
- Thuật toán hỗ trợ;

- Định danh bên cung cấp;
- chứng thực người cấp; và
- Thời gian cập nhật cuối cùng.

7.5.3 EF.OD

tệp tin thư mục đối tượng (EF.DO) là một tệp tin cơ bản, có thể gồm các tham chiếu từ các CIO EF khác. Hình 4 thể hiện mối quan hệ giữa EF.DO và các CIO EF khác (vì nhiều lí do về tính đơn giản, chỉ một tệp tin tham chiếu của mỗi loại được thể hiện). Cú pháp ASN.1 với các nội dung của EF.OD được mô tả .trong 8.3



Hình 4 - Bù gián tiếp các CIO dùng EF.OD

7.5.4 Tệp tin thư mục CIO

Mỗi tệp tin thư mục CIO gồm các CIO của một loại nhất định:

- Tệp tin thư mục khóa riêng gồm các đối tượng thông tin khóa riêng;
- Tệp tin thư mục khóa công khai gồm các đối tượng thông tin khóa công khai;
- Tệp tin thư mục khóa bí mật gồm các đối tượng thông tin khóa bí mật;
- Tệp tin thư mục đối tượng bộ chứa dữ liệu gồm các đối tượng thông tin bộ chứa dữ liệu; và
- Tệp tin thư mục đối tượng chứng thực gồm các đối tượng thông tin chứng thực.

Nhiều tệp tin thư mục CIO của cùng loại có thẻ có trong một DF.CIA.

Tệp tin thư mục đối tượng EF.OD là đơn nhất và gồm các tham chiếu tới các tệp tin thư mục CIO.

CHÚ THÍCH 1 Nếu một tệp tin thư mục CIO của một loại nhất định tồn tại trong một DF.CIA, nó thường không được để trắng.

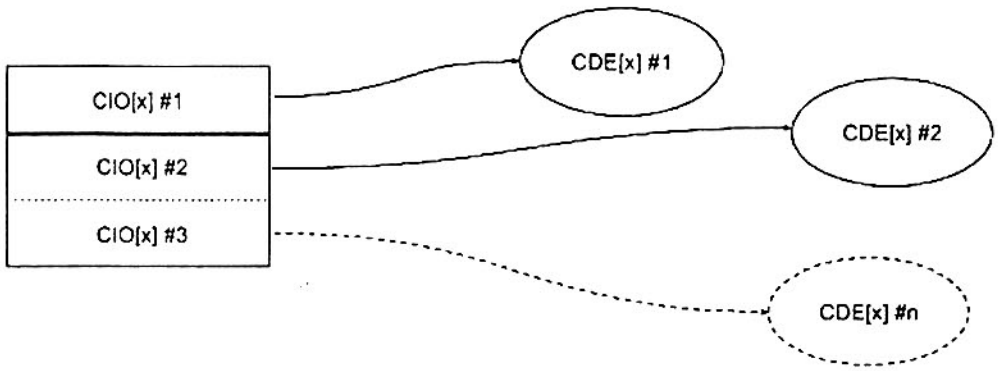
CHÚ THÍCH 2 Tệp tin thư mục CIO cũng được tìm thấy trong các DF khác trên thẻ.

CHÚ THÍCH 3 Các CIO có thể được lưu trữ trực tiếp trong một EF.OD (mà không có bất kì hành động gián tiếp nào) hoặc trong các tệp tin thư mục CIO. Các CDE có thể được lưa trữ trực tiếp trong các CIO hoặc được tham chiếu bởi chúng.

CHÚ THÍCH 4 Việc dùng hành động gián tiếp làm đơn giản sự cá nhân hóa, cho phép các quy tắc truy cập đáng tin cậy hơn và được khuyến nghị.

Khi các CIO tham chiếu các CDE bằng liên kết logic (ví dụ: một khóa riêng CIO và một khóa công khai CIO tương ứng) thì các CDE phải có cùng định danh CIO.

Hình 5 mô tả cấu trúc chung của các tệp tin này.



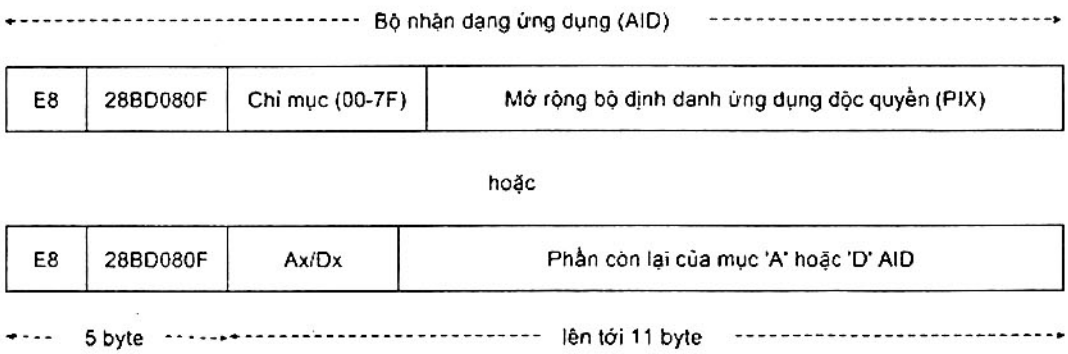
Hình 5 – Bù gián tiếp các CDE dùng CIO

7.5.5 Chọn lựa DF.CIA

AID của một DF.CIA gồm hai trường: định danh chuẩn E8 28 BD 08 0F (bắt buộc), được tùy chọn bởi cả:

- Một chỉ mục 1-byte trong dải '00' tới '7F' được tuân theo bởi một mở rộng định danh ứng dụng độc quyền (PIX); hay
- Một AID (có thể cắt ngắn) (ví dụ: của một ứng dụng dùng CIA này) dùng một định danh bên cung cấp ứng dụng đã đăng kí từ danh mục đăng kí từ 'A' tới 'Z' (xem TCVN 11167-4).

Độ dài của AID không được vượt quá 16 byte. Định danh của AID là:



Hình 6 - Định dạng AID

DF.CIA có thể được chọn lựa dùng AID của nó bởi các thẻ hỗ trợ việc chọn lựa ứng dụng trực tiếp.

CHÚ THÍCH Vì các lí do lịch sử, DF.CIA có thể được chọn dùng AID: A0 00 00 00 63 50 4B 43 53 2D 31 35.

Nếu việc chọn lựa ứng dụng trực tiếp là không khả thi, một tệp tin EF.DIR với nội dung được quy định trong Điều 7.4 phải được dùng.

Khi một vài DF.CIA nằm trên một thẻ, chung có thể phân biệt bởi thông tin trong mẫu thông tin trong EF.DIR. Điều này được khuyến nghị rằng nhãn ứng dụng (thẻ '50') cũng được xem xét nhằm đơn giản góa giao diện người-máy (ví dụ: tên bên cung cấp ở định dạng ngắn)

8 Cú pháp thông tin trong ASN.1

8.1 Hướng dẫn và chuyển đổi mã hóa

Tiêu chuẩn này dùng ASN.1 (ISO/IEC 8824:1998, tất cả các phần) nhằm mô tả các CIO. Khi được lưu trữ trên một thẻ, mã hóa DER của các giá trị CIO được giả định. Phụ lục A bao gồm một quy định kỹ thuật hoàn chỉnh trong ASN.1 của tất cả các CIO; văn bản của điều này chỉ giải thích.

Nội dung của tệp tin thư mục CIO là tổ hợp của 0, 1 và các giá trị DER mã hóa có cùng loại, xem ví dụ Phụ lục D.

8.2 Loại ASN.1 cơ bản được quy định

8.2.1 Định danh

IDentifier ::= OCTET STRING (SIZE (0..cia-ub-IDentifier))

Loại **IDentifier** được dùng như một định danh CIO. Đối với các mục đích tham chiếu ngang, hai nhay nhiều CIO có thể có cùng giá trị **IDentifier**. Một ví dụ của loại này là một khóa cá nhận và một hay. nhiều chứng chỉ tương ứng.

8.2.2 Tham chiếu

Reference ::= INTEGER (0..cia-ub-reference)

Loại này được dùng với các mục đích tham chiếu chung.

8.2.3 Nhãn

Label ::= UTF8String (SIZE(0..cia-ub-label))

Loại này được dùng với tất cả nhãn (ví dụ: người dùng được gán các tên đối tượng).

8.2.4 CredentialIDentifier

```
CredentialIDentifier {KEY-IDENTIFIER : IDentifierSet} ::= SEQUENCE {
        IDType KEY-IDENTIFIER.&ID ({IDentifierSet}),
        IDValue KEY-IDENTIFIER.&Value ({IDentifierSet}{@IDType})
        }
KeyIDentifiers KEY-IDENTIFIER ::= {
```

```
        issuerAndSerialNumber           |
        issuerAndSerialNumberHash       |
        subjectKeyID                    |
        subjectKeyHash                  |
        issuerKeyHash                   |
        issuerNameHash                  |
        subjectNameHash                 |
        pgp2KeyID                       |
        openPGPKeyID,
        certificateHolderReference,
        ...
        }
KEY-IDENTIFIER ::= CLASS {
        &ID INTEGER UNIQUE,
        &Value
        } WITH SYNTAX {
        SYNTAX &Value IDENTIFIED BY &ID
        }
```

Loại **CredentialIDentifier** được dùng nhằm định danh một khóa hay chứng chỉ cụ thể. Có 9 thành phần trong tập các định danh vơus các khóa cá nhân và chứng chỉ **KeyIDentifiers**:

- **issuerAndSerialNumber**: Giá trị của loại này phải là một **SEQUENCE** gồm tên phân biệt và số se-ri cả một chưgns nhận chứa khóa công khai tương ứng với khóa cá nhân của bên phát hành.
- **issuerAndSerialNumberHas**: Tương tự với **issuerAndSerialNumber**, nhưng giá trị là một **OCTET STRING** gồm một giá trị băm SHA-1 của thông tin này để bảo tòan không gian.
- **subjectKeyID**: Giá trị của loại này phải là một **OCTET STRING** với cùng giá trị như mở rộng chứng chỉ **subjectKeyIDentifier** trong một chứng chỉ của ISO/IEC 9594-8:1998, gồm khóa công khai liên quan tới khóa cá nhân. Định danh này có thể được dùng với các giao cắt chuỗi chứng chỉ.
- **subjectKeyHash**: Một **OCTET STRING** gồm băm SHA-1 của khóa công khai liên quan tới khóa cá nhân.
- **issuerKeyHash**: Một **OCTET STRING** gồm một băm SHA-1 của khóa công khai dùng để gán chứng chỉ được yêu cầu.
- **issuerNameHash**: Một **OCTET STRING** gồm một băm SHA-1 của tên bên phát hành khi nó xuất hiện trong chứng chỉ.

CHÚ THÍCH Định danh khóa có thể liên kết với định danh **subjectNameHash** cũng được dùng với xây dựng chuỗi chứng chỉ.

- **subjectNameHash**: Một **OCTET STRING** gồm một băm SHA-1 của tên nội dung khi nó xuất hiện trong chứng chỉ.
- **pgp2KeyID**: Một **OCTET STRING (SIZE(8))** gồm một định danh khóa PGP2.

CHÚ THÍCH Định danh khóa PGP2 được quy định trong IETF RFC 2440 (xem Thư mục tài liệu tham khảo).

- **openPGPKeyID**: Một **OCTET STRING (SIZE(8))** gồm một định danh khóa OpenPGP.

CHÚ THÍCH Định danh khóa PGP được quy định trong IETF RFC 2440 (xem Thư mục tài liệu tham khảo).

8.2.5 ReferencedValue và Path (đường dẫn)

```
ReferencedValue ::= CHOICE {
        path    Path,
        url     URL
        } -- Cú pháp của đối tượng được quy định theo ngữ cảnh
URL ::= CHOICE {
        url     CHOICE {printable PrintableString, ia5 IA5String},
        urlWithDigest [3] SEQUENCE {
                url     IA5String,
                digest  DigestInfoWithDefault
                }
        }
Path ::= SEQUENCE {
        efIDOrPath CHOICE {
                efIDOrPath OCTET STRING,
                tagRef [0] SEQUENCE {
                        tag OCTET STRING,
                        efIDOrPath OCTET STRING OPTIONAL
                        },
        appFileRef [1] SEQUENCE {
                aID [APPLICATION 15] OCTET STRING,
                efIDOrpath OCTET STRING
                },
        appTagRef [2] SEQUENCE {
                aID [APPLICATION 15] OCTET STRING,
                tag OCTET STRING,
                efIDOrPath OCTET STRING OPTIONAL
                }
        },
        index   INTEGER (0..cia-ub-index) OPTIONAL,
        length [0] INTEGER (0..cia-ub-index) OPTIONAL
        }( WITH COMPONENTS {..., index PRESENT, length PRESENT}|
        WITH COMPONENTS {..., index ABSENT, length ABSENT})
```

Một **ReferencedValue** là một tham chiếu với một giá trị CIO của một vài loại. Nó có thể là một vài tham chiếu ngoài (thu được bởi việc chọn lựa url) hay một tham chiếu tới một tệp tin trên thẻ (định danh **path**). Cú pháo của giá trị này được xác định theo ngữ cảnh.

Trong trường hợp **path**, định danh **index** và **length** có thể quy định một vị trí đặc trưng trong tệp tin. Nếu tệp tin là tệp tin bản ghi tuyến tính **index** phải quy định số bản ghi (trong định nghĩa của TCVN 11167-4) và **length** có thể được đặt từ **0** (nếu hệ điều hành thẻ cho phép một thông số L_e = '00' trong một lệnh 'READ RECORD') . Độ dài của các bản ghi cố định có thể được tìm thấy trong tệp tin **CIAInfo** (xem Điều 8.10). Nếu tệp tin là một tệp tin trong suốt **index** phải quy định một bộ đệm trong tệp tin và **length** - độ dài của phân vùng (**index** sẽ thành thông số P_1 và/hoặc P_2 và **length** - thông số L_e trong

một lệnh 'READ BINARY'). Bằng cách dùng **index** và **length**, vài đối tượng có thể được lưu trữ trong cùng tệp tin trong suốt.

CHÚ THÍCH Sau đó một **length** của 0 chỉ ra rằng tệp tin được trỏ tới bởi **efIDOrPath** là một tệp tin bản ghi tuyến tính.

Khi **efIDOrPath** là:

- Trống, không tệp tin nào được tham chiếu tới nó;
- Một byte dài, nó tham chiếu một định danh EF ngắn trong 5 bit có nghĩa nhất (các bit: b3, b2 và b1 phải bằng 0).
- Hai byte dài, nó tham chiếu một tệp tin bởi chính định danh tệp tin của nó;
- Dài hơn 2 byte và gồm một số byte chẵn, nó tham chiếu một tệp tin bởi một đường dẫn tương đối và tuyệt đối (ví dụ: tổ hợp của các định danh tệp tin);
- Dài hơn 2 byte và gồm một số byte lẻ, nó tham chiếu một đường dẫn định tính (xem TCVN 11167-4).

Trong trường hợp **url**, URL có thể là một URL đơn hay một URL kết hợp với một băm mã hóa của đối tượng ở vị trí sẵn có. Giả định rằng thẻ CIO được bảo vệ toàn vẹn, việc thống kế sẽ bảo vệ đối tượng được bảo vệ bên ngoài.

CHÚ THÍCH Cú pháp URL được quy định trong IETF RFC 2396 (xem Thư mục tài liệu tham khảo)

8.2.6 ObjectValue

```
ObjectValue { Type } ::= CHOICE {
        indirect ReferencedValue,
        direct  [0] Type,
        ... -- Cho việc mở rộng sau này
        }
```

Một giá trị đối tượng của loại **ObjectValue** phải được lưu trữ bởi tham chiếu gián tiếp, ngoại trừ các điều khác được đề cập tới (ví dụ: bằng cách trỏ tới vị trí khác mà giá trị thực tế bên trong)

8.2.7 PathOrObjects

```
PathOrObjects {ObjectType} ::= CHOICE {
        path    Path,
        objects [0] SEQUENCE OF ObjectType,
        ... -- Cho việc mở rộng sau này
        }
```

Loại **PathOrObjects** được dùng để tham chiếu các chuỗi đối tượng nằm trong OD hay trong tệp tin khác. Nếu **path** thay đổi được dùng tệp tin tham chiếu phải bao gồm tổ hợp của 0, 1 và các giá trị DER mã hóa của loại đã cho. Bất kỳ số nào của các octet 'FF' có thể xảy ra trước, giữa và sau các giá trị mà không có bất kỳ ý nghĩa nào (ví dụ: việc dồn không gian chưa dùng hoặc các giá trị bị xóa). **Path** thay đổi được khuyến nghị mạnh mẽ (xem CHÚ THÍCH 4 trong Điều 7.5.4).

8.2.8 CommonObjectAttributes

CHÚ THÍCH Loại này là một bộ chứa với các thuộc tính phổ biến với tất cả các CIO.

```
CommonObjectAttributes ::= SEQUENCE {
   label          Label OPTIONAL,
   flags          CommonObjectFlags OPTIONAL,
   authID                IDentifier OPTIONAL,
   userConsent INTEGER (1..cia-ub-userConsent) OPTIONAL,
   accessControlRules  SEQUENCE SIZE (1..MAX) OF AccessControlRule OPTIONAL,
      ...
      } (CONSTRAINED BY {-- authID nên được xem xét nếu flags.private được thiết lập.
      -- Nó phải bằng một authID theo một đối tượng chứng thực trong AOD -- })
CommonObjectFlags ::= BIT STRING {
      private          (0),
      modifiable       (1),
      internal(2)
      } -- Bit (2) được xem xét với các lí do lịch sử và không được dùng
AccessControlRule ::= SEQUENCE {
      accessMode             AccessMode,
      securityCondition      SecurityCondition,
      ... -- Cho việc mở rộng sau này
      }
AccessMode ::= BIT STRING {
      read           (0),
      update (1),
      execute        (2),
      delete  (3),
      attribute      (4),
      pso_cds        (5),
      pso_verif      (6),
      pso_dec        (7),
      pso_enc        (8),
      int_auth       (9),
      ext_auth       (10)
      }
SecurityCondition ::= CHOICE {
   always              NULL,
   authID              IDentifier,
   authReference       AuthReference,
   not            [0] SecurityCondition,
   and  [1] SEQUENCE SIZE (2..cia-ub-securityConditions) OF SecurityCondition,
```

```
    or          [2] SEQUENCE SIZE (2..cia-ub-securityConditions) OF SecurityCondition,
        ... -- Cho việc mở rộng sau này
        }
AuthReference ::= SEQUENCE {
        authMethod      AuthMethod,
        selDentifier    Reference OPTIONAL
        }
AuthMethod ::= BIT STRING {secureMessaging(0), extAuthentication(1), userAuthentication(2), always(3)}
```

Thành phần **label** hoàn toàn cho mục đích hiển thị (giao diện người-máy), ví dụ: khi một người sử dụng có một số giấy chứng chỉ cho một cặp khóa (ví dụ: "chứng chỉ ngân hàng", "chứng chỉ e-mail").

Thành phần **flags** chỉ ra cho dù các đối tượng cụ thể là cá nhân hay không, và liệu nó là kiểu chỉ đọc hay không. Một đối tượng **private** chỉ có thể được truy cập sau chứng thực thông thường (ví dụ: xác minh mật khẩu). Nếu đối tượng được đánh dấu là **modifiable**, nó phải có khả năng cập nhật giá trị của các đối tượng. Nếu một đối tượng là cả **private** và **có thể sửa đổi**, tuy nhiên cập nhật chỉ được phép sau khi chứng thực thành công.

Thành phần **authID** tạo, trong trường hợp của một đối tượng riêng, một tham chiếu chéo về đối tượng chứng thực được sử dụng để bảo vệ đối tượng này (đối với một mô tả các đối tượng chứng thực, xem Điều 8.9).

Thành phần **userConsent** tạo, trong trường hợp của một đối tượng riêng (hoặc một đối tượng mà điều kiện truy cập đã được chỉ định), số lần một ứng dụng có thể truy cập các đối tượng mà không có sự cho phép của người dùng (ví dụ: một giá trị của **3** chỉ ra rằng một chứng thực mới sẽ được yêu cầu trước truy cập thứ 4, thứ 7). Thẻ có thể thực thi các giá trị này, ví dụ: thông qua việc sử dụng "đối tượng truy cập" (xem TCVN 11167-8 (ISO/IEC 7816-8). Một giá trị **1** có nghĩa là một chứng thực mới là cần thiết trước mỗi truy cập.

Thành phần **accessControlRules** đưa ra một cách thay thế, dễ hơn để thông báo một ứng dụng máy chủ về điều kiện an ninh cho các phương pháp khác nhau của việc truy cập các đối tượng theo câu hỏi. Bất kỳ biểu thức Boolean nào trong các phương pháp chứng thực có được phép. Nếu một chế độ truy cập nhất định là không được phép, thì sẽ không có quy tắc kiểm soát truy cập cho nó (tức là nó là tiềm ẩn). Nếu thành phần này không được hiển thị, quy tắc kiểm soát truy cập sẽ phải được suy luận bằng các cách tiện khác. Tùy chọn **authReference** cho phép cho một kết nối chặt chẽ hơn với các phần khác trong bộ tiêu chuẩn TCVN 11167 (ISO/IEC 7816), thông qua các tham chiếu đến môi trường an ninh và việc định danh các lớp học của phương pháp chứng thực (**authMethod**).

CHÚ THÍCH 1 Khi các thành phần: **accessControlRules** và **authID** đều được thể hiện, thông tin trong thành phần **accessControlRule** được ưu tiên. Điều này có thể xảy ra vì những lý do tương thích ngược.

CHÚ THÍCH 2 Khi các đặc tính liên quan đến kiểm soát truy cập có thể được suy luận, ví dụ: bằng cách nghiên cứu các EF FCI, thông tin này là tùy chọn và không cần thiết khi các trường hợp được áp dụng (xem TCVN 11167-4).

CHÚ THÍCH 3 Thông tin kiểm soát truy cập được trình bày trong cấu trúc này thể hiện quy tắc kiểm soát truy cập trong thẻ, nhưng không nhất thiết phải sử dụng như vậy bởi các thẻ.

8.2.9 CommonKeyAttributes

```
CommonKeyAttributes ::= SEQUENCE {
        ID              IDentifier,
        usage           KeyUsageFlags,
        native          BOOLEAN DEFAULT TRUE,
        accessFlags     KeyAccessFlags OPTIONAL,
        keyReference    KeyReference OPTIONAL,
        startDate               GeneralizedTime OPTIONAL,
        endDate                 [0] GeneralizedTime OPTIONAL,
        algReference    [1] SEQUENCE OF Reference OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
KeyUsageFlags ::= BIT STRING {
        encipher                (0),
        decipher                (1),
        sign                    (2),
        signRecover     (3),
        keyEncipher             (4),
        keyDecipher             (5),
        verify                  (6),
        verifyRecover   (7),
        derive          (8),
        nonRepudiation          (9)
        }
KeyAccessFlags ::= BIT STRING {
        sensitive               (0),
        extractable             (1),
        alwaysSensitive         (2),
        neverExtractable        (3),
        cardGenerated (4)
        }
KeyReference ::= INTEGER
```

Thành phần **ID** phải là duy nhất cho mỗi đối tượng thông tin khóa, trừ khi một đối tượng thông tin khóa công khai và đối tượng thông tin khóa riêng tương ứng của nó được lưu trữ trên cùng một thẻ. Trong trường hợp này, các đối tượng thông tin phải chia sẻ cùng một định danh (cũng có thể được chia sẻ với một hay một số đối tượng thông tin chứng chỉ, xem Điều 8.2.15).

Thành phần **usage (encipher, decipher, sign, signRecover, keyEncipher, keyDecipher, verify, verifyRecover, derive** và **nonRepudiation**) báo hiệu việc sử dụng có thể có của các khóa. Các thuật toán và phương pháp thực tế được sử dụng cho các hoạt động này được ngầm hiểu và không được quy định nghĩa trong tiêu chuẩn này. Để ánh xạ giữa ISO/IEC 9594-8, cờ **keyUsage** cho các khóa công khai, các cờ CIO cho khóa công khai và cờ CIO cho khóa riêng sử dụng bảng sau:

Bảng 2 - Ánh xạ giữa thẻ dùng khóa CIO và thẻ dùng khóa của ISO/IEC 9594-8

Thẻ dùng khóa công khai trong chứng chỉ khóa công khai của ISO/IEC 9594-8	Thẻ dùng khóa CIO tương ứng với khóa công khai	Thẻ dùng khóa CIO tương ứng với khóa cá nhân
DataEncipherment	Encipher	Decipher
DigitalSignature, keyCertSign, cRLSign (các thuật toán chữ ký không cần phục hồi thông điệp)	Verify	Sign
DigitalSignature, keyCertSign, cRLSign (các thuật toán chữ ký có phục hồi thông điệp)	VerifyRecover	SignRecover
KeyAgreement	Derive	Derive
KeyEncipherment	KeyEncipher	KeyDecipher
NonRepudiation	NonRepudiation	NonRepudiation

Thành phần **native** chỉ ra liệu các thuật toán mã hóa kết hợp với khóa được thực hiện trong phần cứng thẻ.

Việc biên dịch các bit **KeyAccessFlags** phải được quy định như sau:

- **sensitive** chỉ ra rằng tài liệu khóa không thể được tiết lộ theo văn bản thường ra ngoài thẻ;
- nếu **extractable** không được đặt, tài liệu khóa không thể được trích xuất từ thẻ, thậm chí dưới dạng mã hoá;
- **alwaysSensitive** chỉ ra rằng khóa luôn là **sensitive**;
- **neverExtractable** chỉ ra rằng khóa chưa bao giờ được **extractable**; và
- **cardGenerated** chỉ ra rằng khóa được tạo ra một cách ngẫu nhiên trên thẻ.

Thành phần **accessFlags** có thể vắng mặt trong trường hợp giá trị của nó có thể được suy luận bằng các cách thức khác.

Thành phần **keyReference** chỉ áp dụng cho thẻ với khả năng mã hóa. Nếu có, nó có chứa một tham chiếu đặc trưng thẻ làm khóa theo câu hỏi (để biết thêm thông tin xem TCVN 11167-4 và TCVN 11167-8).

Chú thích Giá trị của thành phần **keyReference** được dùng trong tham chiếu khóa của các DO (TCVN 11167-4), và bất kỳ giá trị, gồm cả giá trị âm là có thể hiểu được.

Thành phần **startDate** và **endDate**, nếu có, chỉ rõ khoảng thời gian mà khóa là hiệu lực sử dụng.

Thành phần **algReference** chỉ ra mà khóa được dùng bởi việc tham chiếu các giá trị **supportedAlgorithm** từ tệp tin EF.CIAInfo.

8.2.10 CommonPrivateKeyAttributes

```
CommonPrivateKeyAttributes ::= SEQUENCE {
   name                Name OPTIONAL,
   keyIDentifiers      [0] SEQUENCE OF CredentialIDentifier {{KeyIDentifiers}} OPTIONAL,
   generalName         [1] GeneralNames OPTIONAL,
   ... -- Cho việc mở rộng sau này
   }
```

Thành phần **name**, khi được trình bày, gọi tên chủ khóa, được quy định trong thành phần **subject** của chứng chỉ tương ứng.

Giá trị của thành phần **keyIDentifiers** có thể kết hợp với các định danh từ thông điệp bên ngoài hoặc các giao thức chọn khóa thích hợp cho một hoạt động nhất định. Các giá trị này cũng được truyền tới bên nhận để chỉ ra khóa nào được sử dụng. Một số cơ chế xác định một khóa được hỗ trợ (xem Điều 8.2.4).

Thành phần **generalName**, khi được trình bày, cung cấp nhiều cách khác để xác định chủ khoá.

8.2.11 CommonPublicKeyAttributes

```
CommonPublicKeyAttributes ::= SEQUENCE {
   name                Name OPTIONAL,
   trustedUsage        [0] Usage OPTIONAL,
   generalName         [1] GeneralNames OPTIONAL,
   keyIDentifiers      [2] SEQUENCE OF CredentialIDentifier {{KeyIDentifiers}} OPTIONAL,
      ... -- Cho việc mở rộng sau này
      }
```

Việc biên dịch **name**, **generalName** và các thành phần của **keyIDentifiers** của loại **CommonPublicKeyAttributes** phải giống như các thành phần tương ứng của **CommonPrivateKeyAttributes**.

Thành phần **trustedUsage** chỉ ra một hay nhiều mục đích với khóa công khai được tin dùng bởi người giữ thẻ (Xem Điều 8.2.15)

CHÚ THÍCH Các ngữ nghĩa chính xác của "sự tin tưởng" nằm ngoài pahmj vi của bộ tiêu chuẩn này.

8.2.12 CommonSecretKeyAttributes

```
CommonSecretKeyAttributes ::= SEQUENCE {
        keyLen INTEGER OPTIONAL, -- keylength (in bits)
        ... -- Cho việc mở rộng sau này
        }
```

Thành phần **keyLen** tùy chọn thông báo độ dài khóa được dùng, trong các trường hợp khi một thuật toán cụ thể có thể có một độ dài khóa thay đổi.

8.2.13 GenericKeyAttributes

```
GenericKeyAttributes ::= SEQUENCE {
        keyType         CIO-ALGORITHM.&objectIDentifier ({AllowedAlgorithms}),
        keyAttr         CIO-ALGORITHM.&Parameters ({AllowedAlgorithms}{@keyType})
        }
AllowedAlgorithms CIO-ALGORITHM ::= {...}
```

Loại này nhằm chứa thông tin đặc trưng với một loại cho sẵn. Định nghĩa tập đối tượng thông tin **AllowedAlgorithms** được trả sau, mặc dù các hồ sơ chuẩn hóa hay cho các tình huống phù hợp thiết lập giao thức. Tập này được yêu cầu nhằm quy định một bảng bắt buộc theo các thành phần của **GenericKeyAttributes**.

8.2.14 KeyInfo

```
KeyInfo {ParameterType, OperationsType} ::= CHOICE {
        paramsAndOps SEQUENCE {
                parameters      ParameterType,
                operations      OperationsType OPTIONAL
                },
        reference       Reference -- Theo lịch sử, không được dùng
        }
```

CHÚ THÍCH PKCS # 15 dùng tùy chọn **reference**.

Loại này là một phần tùy chọn của từng loại khóa riêng và khóa công khai, gồm cả những chi tiết đặc trưng thuật toán về các thông số của khóa và các hoạt động được hỗ trợ bởi thẻ hoặc một tham chiếu đến thông tin đó. Nếu được trình bày, các giá trị đặc trưng thuật toán ghi đè bất kỳ giá trị tham chiếu bởi thành phần **CommonKeyAttributes.algReference**.

8.2.15 CommonCertificateAttributes

```
CommonCertificateAttributes ::= SEQUENCE {
   ID                   IDentifier,
   authority    BOOLEAN DEFAULT FALSE,
```

```
    IDentifier    CredentialIDentifier {{KeyIDentifiers}} OPTIONAL,
    certHash              [0] CertHash OPTIONAL,
    trustedUsage          [1] Usage OPTIONAL,
    Identifiers   [2] SEQUENCE OF CredentialIDentifier {{KeyIDentifiers}} OPTIONAL,
    validity              [4] validity OPTIONAL,
    ...
        }
```

CHÚ THÍCH PKCS #15 dùng thẻ ngữ cảnh [3].

```
Usage ::= SEQUENCE {
        keyUsage KeyUsage OPTIONAL,
        extKeyUsage SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL,
        ...
        } (WITH COMPONENTS {..., keyUsage PRESENT} | WITH COMPONENTS {..., extKeyUsage
        PRESENT})
```

Khi một khóa công khai trong một chứng chỉ được tham chiếu bởi một đối tượng thông tin chứng chỉ tương ứng với một khóa riêng được tham chiếu bởi một đối tượng thông tin khóa riêng, thì các đối tượng thông tin phải chia sẻ cùng giá trị cho thành phần **ID**. Yêu cầu này sẽ đơn giản hóa các tìm kiếm một khóa riêng tương ứng với một chứng chỉ đặc biệt và ngược lại. Nhiều chứng chỉ cho các khóa tương tự phải chia sẻ cùng giá trị cho thành phần **ID**.

Thành phần **authority** chỉ ra dù chứng chỉ cho một cơ quan (ví dụ: cơ quan chứng chỉ) hay không. Thành phần **IDentifier** được trình bày chỉ vì lý do lịch sử và thành phần **IDentifiers** phải sử dụng thay thế.

Thành phần **certHash** hữu ích theo một góc độ an ninh khi một chứng chỉ được lưu trữ bên ngoài thẻ (việc chọn **url** của **ReferencedValue**), khi cho phép người dùng xác minh rằng không có ai bị giả mạo chứng chỉ.

Thành phần **trustedUsage** chỉ ra một hay nhiều mục đích mà khóa công khai được chứng chỉ là đáng tin cậy bởi chủ thẻ. Định danh đối tượng cho thành phần **extKeyUsage** có thể được xác định bởi bất kỳ tổ chức nào theo nhu cầu. Đối với sử dụng thực tế, các giao điểm của việc sử dụng chỉ định trong thành phần này, và việc mở rộng **keyUsage** (nếu có) trong chính chứng chỉ cần được thực hiện. Nếu thành phần **trustedUsage** thiếu, tất cả việc sử dụng là khả thi.

CHÚ THÍCH 1 Ngữ nghĩa chính xác của "niềm tin" nằm ngoài phạm vi của tiêu chuẩn này.

CHÚ THÍCH 2 Để tìm một chứng chỉ cho chủ thẻ cho một sử dụng cụ thể, dùng thành phần **commonKeyAttributes.usage**, theo tham chiếu chéo (**commonKeyAttributes.ID**) cho một chứng chỉ phù hợp.

Thành phần **identifiers** đơn giản hoá việc tìm kiếm một chứng chỉ đặc biệt, khi người yêu cầu biết (và chuyển tải) một số thông tin để phân biệt về chứng chỉ được yêu cầu. Điều này có thể được dùng, ví dụ: khi một chứng chỉ người dùng được chọn và gửi đến một máy chủ như là một phần của chứng thực người dùng, và máy chủ cung cấp cho khách hàng với việc phân biệt thông tin

cho một chứng chỉ đặc biệt. Sử dụng thay thế **subjectNameHash** và **issuerNameHash** cũng tạo thuận lợi cho xây dựng chuỗi nhanh chóng.

Thành phần **validity** cung cấp thông tin về thời hạn hiệu lực của chứng chỉ.

8.2.16 GenericCertificateAttributes

```
GenericCertificateAttributes ::= SEQUENCE {
        certType        CIO-OPAQUE.&ID ({AllowedCertificates}),
        certAttr        CIO-OPAQUE.&Type ({AllowedCertificates}{@certType})
        }
AllowedCertificates CIO-OPAQUE ::= {...}
```

Loại này được thiết kế nhằm chứa thông tin đặc trưng cho bất kỳ loại chứng chỉ nào. Việc xác định tập đối tượng thông tin của **AllowedCertificates** được trả sau, dù hồ sơ được chuẩn hóa hay cho các trường hợ phù hợp cho việc triển khai giao thức. Tập này cần thiết để xác định một bảng hạn chế theo các thành phần của **GenericCertificateAttributes**.

8.2.17 CommonDataContainerObjectAttributes

```
CommonDataContainerObjectAttributes ::= SEQUENCE {
        applicationName         Label OPTIONAL,
        applicationOID OBJECT IDENTIFIER OPTIONAL,
        ID                      IDentifier OPTIONAL,
        ... -- Cho việc mở rộng sau này
        } (WITH COMPONENTS {..., applicationName PRESENT}
        | WITH COMPONENTS {..., applicationOID PRESENT})
```

Thành phần **applicationName** dùng chứa tên hay định danh đối tượng đăng ký cho ứng dụng mà đối tượng chứa dữ liệu theo câu hỏi "phụ thuộc". Để tránh xung đột tên ứng dụng, ít nhất việc thay thế **applicationOID** được khuyến khích. Như đã nêu trong ASN.1, ít nhất một trong những thành phần đã được trình bày trong một giá trị kiểu **CommonDataContainerObjectAttributes**.

Thành phần **ID** có thể được dùng kết hợp một đối tượng chứa dữ liệu nhất định với một vài CIO khác, ví dụ: một đối tượng thông tin khóa riêng.

8.2.18 CommonAuthenticationObjectAttributes

```
CommonAuthenticationObjectAttributes ::= SEQUENCE {
        authID                  IDentifier OPTIONAL,
        authReference  Reference OPTIONAL,
        selDentifier            [0] Reference OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

authId phải là một định danh duy nhất. Nó được dùng cho mục đích tham chiếu chéo từ các CIO riêng.

Thành phần **authReference**, khi được trình bày, phải có một giá trị của một đối tượng "tham chiếu khóa" (xem TCVN 11167-4), đây là cách để tham chiếu các khóa này trong môi trường an ninh.

Thành phần **selIdentifier**, khi được trình bày, định danh môi trường an ninh cho đối tượng nó phụ thuộc.

8.2.19 Loại CIO

Loại này là một mẫu cho tất cả các loại CIO. Nó được tham chiếu với các thuộc tính lớp đối tượng, thuộc tính lớp phụ đối tượng và các thuộc tính kiểu đối tượng.

```
CIO {ClassAttributes, SubClassAttributes, TypeAttributes} ::= SEQUENCE {
        commonObjectAttributes      CommonObjectAttributes,
        classAttributes             ClassAttributes,
        subClassAttributes          [0] SubClassAttributes OPTIONAL,
        typeAttributes              [1] TypeAttributes
        }
```

8.3 Loại CIOChoice

```
CIOChoice ::= CHOICE {
        privateKeys                 [0] PrivateKeys,
        publicKeys                  [1] PublicKeys,
        trustedPublicKeys           [2] PublicKeys,
        secretKeys                  [3] SecretKeys,
        certificates                [4] Certificates,
        trustedCertificates         [5] Certificates,
        usefulCertificates          [6] Certificates,
        dataContainerObjects        [7] DataContainerObjects,
        authObjects                 [8] AuthObjects,
        ... -- Cho việc mở rộng sau này
        }
PrivateKeys ::=                     PathOrObjects {PrivateKeyChoice}
PublicKeys ::=                      PathOrObjects {PublicKeyChoice}
SecretKeys ::=                      PathOrObjects {SecretKeyChoice}
Certificates ::=                    PathOrObjects {CertificateChoice}
DataContainerObjects ::=            PathOrObjects {DataContainerObjectChoice}
AuthObjects ::=                     PathOrObjects {AuthenticationObjectChoice}
```

EF.OD phải gồm tổ hợp của 0, 1 hay nhiều giá trị CIOChoice được mã hóa kiểu DER. Bất kỳ các octet 'FF' nào có thể xuất hiện trước, trong và sau các giá trị mà không có bất kỳ ý nghĩa nào (v dụ: việc đệm cho không gian chưa sử dụng hoặc xóa các giá trị). Một lựa chọn cụ thể có thể xuấ hiện nhiều hơn một lần trong tệp tin (trong đó có thể được hoàn thành, ví dụ: áp dụng các quy tắ kiểm soát truy cập khác nhau để phân chia tập hợp các đối tượng cùng loại).

Mong đợi rằng một mục EF.OD thường tham chiếu một tệp tin riêng biệt (lựa chọn **path** của **PathOrObjects**) chứa các CIO của loại chỉ định. Tuy nhiên, một mục có thể giữ các CIO trực tiếp (lựa chọn **objects** của **PathOrObjects**), nếu các đối tượng và tệp tin EF.OD có yêu cầu kiểm soát truy cập tương tự.

Thành phần **trustedPublicKeys** tham chiếu các đối tượng thông tin khóa công khai mô tả các khóa công khai được tin cậy bởi chủ thẻ với một số mục đích, như là điểm tin (gốc) để xử lý đường dẫn chứng chỉ.

Lựa chọn **certificates** tham chiếu các đối tượng thông tin chứng chỉ mô tả chứng chỉ được phát hành cho thẻ hay chủ thẻ.

Thành phần **trustedCertificates** tham chiếu các đối tượng thông tin chứng chỉ mô tả các chứng chỉ đáng tin cậy bởi chủ thẻ vì các mục đích chỉ định của họ. Ví dụ: chứng chỉ CA được tham chiếu bởi thành phần này có thể được dùng như điểm tin (gốc) trong việc xử lý đường dẫn chứng chỉ.

CHÚ THÍCH Để duy trì sự tin tưởng mong muốn trong chứng chỉ và/hoặc khóa công khai cho trước, các CIO tương ứng của chúng trong các thành phần: **trustedCertificates** và/hoặc **trustedPublicKeys** cần được bảo vệ thích hợp chống lại việc chỉnh sửa (tức các kiểm soát truy cập thích hợp). Việc bảo vệ này phải áp dụng cho tệp tin EF.OD, bất kỳ tệp tin CIO được tham chiếu bởi thành phần: **trustedCertificates** hay **trustedPublicKeys** và bất kỳ khóa hay tệp tin chứng chỉ thực tế nào được tham chiếu từ các CIO riêng lẻ.

Thành phần **usefulCertificates** tham chiếu các đối tượng chứng nhận mô tả các chứng chỉ mà không thuộc về một trong hai thành phần: **trustedCertificates** hay **certificates**. Nó được sử dụng để lưu trữ hay là thực thể kết thúc hay là các chứng chỉ CA hữu dụng, ví dụ: một chứng chỉ cho khóa mã hóa của một đồng nghiệp hay các chứng chỉ CA trung gian nhằm đơn giản hóa việc xử lý đường dẫn chứng chỉ.

8.4 Đối tượng thông tin khóa cá nhân

8.4.1 PrivateKeyChoice

```
PrivateKeyChoice ::= CHOICE {
        privateRSAKey       PrivateKeyObject {PrivateRSAKeyAttributes},
        privateECKey        [0] PrivateKeyObject {PrivateECKeyAttributes},
        privateDHKey        [1] PrivateKeyObject {PrivateDHKeyAttributes},
        privateDSAKey       [2] PrivateKeyObject {PrivateDSAKeyAttributes},
        privateKEAKey               [3] PrivateKeyObject {PrivateKEAKeyAttributes},
        genericPrivateKey   [4] PrivateKeyObject {GenericKeyAttributes},
        ... -- Cho việc mở rộng sau này
        }
PrivateKeyObject {KeyAttributes} ::= CIO {
        CommonKeyAttributes, CommonPrivateKeyAttributes, KeyAttributes}
```

Loại này gồm thông tin liên quan đến một khóa riêng. Mỗi giá trị gồm các thuộc tính chung cho bất kỳ đối tượng nào, bất kỳ khóa nào, bất kỳ khóa riêng và các thuộc tính cụ thể cho khóa.

8.4.2 Thuộc tính khóa RSA cá nhân

```
PrivateRSAKeyAttributes ::= SEQUENCE {
    value                Path,
    modulusLength INTEGER, -- modulus length in bits, e.g. 1024
    keyInfo          KeyInfo {NULL, PublicKeyOperations} OPTIONAL,
    ... -- Cho việc mở rộng sau này
    }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PrivateRSAKeyAttributes.value**: Giá trị phải là một đường dẫn đến một tệp tin có chứa một khóa RSA riêng. Nếu không có nhu cầu xác định một đường dẫn đến một tệp tin, giá trị đường dẫn có thể được thiết lập là đường dẫn trống.
- **PrivateRSAKeyAttributes.modulusLength**: Trên nhiều thẻ, một thẻ cần phải định dạng dữ liệu được gán trước khi gửi dữ liệu cho thẻ. Để có thể định dạng dữ liệu một cách chính xác, độ dài của khóa phải được biết đến. Chiều dài phải được thể hiện bằng bit, ví dụ: 1024.
- **PrivateRSAKeyAttributes.keyInfo**: Thông tin về thông số áp dụng cho khóa này và các vận hành của thẻ có thể thực hiện với nó. Các giá trị ghi đè lên bất kỳ giá trị **CIAInfo.supportedAlgorithms** nào được tham chiếu bởi thành phần **CommonKeyAttributes.algReference**. Thành phần này không cần thiết nếu thông tin sẵn có thông qua các cách thức khác.

8.4.3 Thuộc tính khóa Private Elliptic Curve

```
PrivateECKeyAttributes ::= SEQUENCE {
    value          Path,
    keyInfo        KeyInfo {Parameters, PublicKeyOperations} OPTIONAL,
    ... -- Cho việc mở rộng sau này
    }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PrivateECKeyAttributes.value**: Giá trị phải là một đường dẫn đến một tệp tin có chứa một khóa e-llip đường cong riêng. Nếu không có nhu cầu xác định một đường dẫn đến một tệp tin, giá trị đường dẫn có thể được thiết lập là đường dẫn trống.
- **PrivateECKeyAttributes.keyInfo**: Xem thành phần tương ứng trong Điều 8.4.2.

8.4.4 Thuộc tính khóa Private Diffie-Hellman

```
PrivateDHKeyAttributes ::= SEQUENCE {
    value          Path,
    keyInfo        KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
    ... -- Cho việc mở rộng sau này
```

}

Việc biên dịch các thành phần phải được quy định như sau:

- **PrivateDHKeyAttributes.value**: Giá trị phải là một đường dẫn đến một tệp tin một khóa Diffie-Hellman riêng. Nếu không có nhu cầu xác định một đường dẫn đến một tệp tin, giá trị đường dẫn có thể được thiết lập là đường dẫn trống.
- **PrivateDHKeyAttributes.keyInfo**: Xem thành phần tương ứng trong Điều 8.4.2.

8.4.5 Thuộc tính khóa Private DSA

```
PrivateDSAKeyAttributes ::= SEQUENCE {
        value           Path,
        keyInfo KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PrivateDSAKeyAttributes.value**: Giá trị phải là một đường dẫn đến một tệp tin một khóa DSA riêng. Nếu không có nhu cầu xác định một đường dẫn đến một tệp tin, giá trị đường dẫn có thể được thiết lập là đường dẫn trống.
- **PrivateDSAKeyAttributes.keyInfo:** Xem thành phần tương ứng trong Điều 8.4.2.

8.4.6 Thuộc tính khóa Private KEA

```
PrivateKEAKeyAttributes ::= SEQUENCE {
        value           Path,
        keyInfo         KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PrivateKEAKeyAttributes.value**: Giá trị phải là một đường dẫn đến một tệp tin một khóa KEA riêng. Nếu không có nhu cầu xác định một đường dẫn đến một tệp tin, giá trị đường dẫn có thể được thiết lập là đường dẫn trống.
- **PrivateKEAKeyAttributes.keyInfo**: Xem thành phần tương ứng trong Điều 8.4.2.

8.4.7 Đối tượng thông tin khóa Generic Private

Loại này nhằm chứa thông tin đặc trưng với một khóa cá nhân của bất kỳ loại nào. Xem Điều 8.2.13.

8.5 Đối tượng thông tin khóa công khai

8.5.1 PublicKeyChoice

```
PublicKeyChoice ::= CHOICE {
        publicRSAKey  PublicKeyObject {PublicRSAKeyAttributes},
        publicECKey   [0] PublicKeyObject {PublicECKeyAttributes},
        publicDHKey   [1] PublicKeyObject {PublicDHKeyAttributes},
        publicDSAKey  [2] PublicKeyObject {PublicDSAKeyAttributes},
        publicKEAKey [3] PublicKeyObject {PublicKEAKeyAttributes},
        genericPublicKey        [4] PublicKeyObject{GenericKeyAttributes},
        ... -- Cho việc mở rộng sau này
        }
PublicKeyObject {KeyAttributes} ::= CIO {
        CommonKeyAttributes, CommonPublicKeyAttributes, KeyAttributes}
```

Loại này chứa thông tin liên quan đến một khóa công khai. Mỗi giá trị gồm các thuộc tính chung cho bất kỳ đối tượng nào, bất kỳ khóa nào, bất kỳ khóa công khai nào và các thuộc tính cụ thể cho khóa.

8.5.2 Thuộc tính khóa Public RSA

```
PublicRSAKeyAttributes ::= SEQUENCE {
        value                   ObjectValue {RSAPublicKeyChoice},
        modulusLength INTEGER, -- modulus length in bits, e.g. 1024
        keyInfo         KeyInfo {NULL, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
RSAPublicKeyChoice ::= CHOICE {
        raw     RSAPublicKey,
        spki    [1] SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa RSA công khai.
        ...
        }
RSAPublicKey ::= SEQUENCE {
        modulus                 INTEGER,
        publicExponent INTEGER
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PublicRSAKeyAttributes.value:** Giá trị phải là một đường dẫn đến một tệp tin gồm cả giá trị **RSAPublicKeyChoice** hay (một vài thể hiện đặc trưng thẻ của) một khóa RSA công khai.
- **PublicRSAKeyAttributes.modulusLength:** Trên nhiều thẻ, một thẻ phải định dạng dữ liệu được mã hóa trước khi gửi dữ liệu vào thẻ. Để có thể định dạng các dữ liệu một cách chính xác, độ dài của khóa phải được biết đến. Chiều dài phải được thể hiện bằng bit, ví dụ: 1024.

- **PublicRSAKeyAttributes.keyInfo:** Xem thành phần tương ứng trong Điều 8.4.2.

8.5.3 Thuộc tính khóa Public Elliptic Curve

```
PublicECKeyAttributes ::= SEQUENCE {
        value           ObjectValue {ECPublicKeyChoice},
        keyInfo         KeyInfo {Parameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
ECPublicKeyChoice ::= CHOICE {
        raw     ECPoint, -- See ANSI X9.62,
        spki    SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa vòng e-líp công khai.
        ...
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PublicECKeyAttributes.value:** Giá trị phải là một đường dẫn đến một tệp tin gồm cả giá trị **ECPublicKeyChoice** hay (một vài thể hiện đặc trưng thẻ của) một khóa elip vòng cung công khai.
- **PublicECKeyAttributes.keyInfo:** Xem thành phần tương ứng trong Điều 8.4.2.

8.5.4 Thuộc tính khóa Public Diffie-Hellman

```
PublicDHKeyAttributes ::= SEQUENCE {
        value           ObjectValue {DHPublicKeyChoice},
        keyInfo         KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
DHPublicKeyChoice ::= CHOICE {
        raw     DHPublicNumber,
        spki    SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa D-H công khai.
        ...
        }
DHPublicNumber ::= INTEGER
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PublicDHKeyAttributes.value:** Giá trị phải là một đường dẫn đến một tệp tin gồm cả giá trị **DHPublicKeyChoice** hay (một vài thể hiện đặc trưng thẻ của) một khóa D-H công khai.
- **PublicDHKeyAttributes.keyInfo:** Xem thành phần tương ứng trong Điều 8.4.2.

8.5.5 Thuộc tính khóa Public DSA

```
PublicDSAKeyAttributes ::= SEQUENCE {
        value           ObjectValue {DSAPublicKeyChoice},
        keyInfo         KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
```

```
        ... -- Cho việc mở rộng sau này
        }
DSAPublicKeyChoice ::= CHOICE {
        raw     DSAPublicKey,
        spki    SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa DSA công khai.
        ...
        }
DSAPublicKey ::= INTEGER
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PublicDSAKeyAttributes.value:** Giá trị phải là một đường dẫn đến một tệp tin gồm cả giá trị **DSAPublicKeyChoice** hay (một vài thể hiện đặc trưng thẻ của) một khóa DSA công khai.
- **PublicDSAKeyAttributes.keyInfo**: Xem thành phần tương ứng trong Điều 8.4.2.

8.5.6 Thuộc tính khóa Public KEA

```
PublicKEAKeyAttributes ::= SEQUENCE {
        value           ObjectValue {KEAPublicKeyChoice},
        keyInfo         KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
KEAPublicKeyChoice ::= CHOICE {
        raw     KEAPublicKey,
        spki    SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa KEA công khai.
        ...
        }
KEAPublicKey ::= INTEGER
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PublicKEAKeyAttributes.value:** Giá trị phải là một đường dẫn đến một tệp tin gồm cả giá trị **KEAPublicKeyChoice** hay (một vài thể hiện đặc trưng thẻ của) một khóa KEA công khai.
- **PublicKEAKeyAttributes.keyInfo:** Xem thành phần tương ứng trong Điều 8.4.2.

8.5.7 Đối tượng thông tin khóa công khai chung

Loại này nhằm chứa thông tin đặc trưng về một khóa công khai của bất kỳ loại nào. Xem Điều 8.2.13.

8.6 Đối tượng thông tin khóa bí mật

8.6.1 SecretKeyChoice

```
SecretKeyChoice ::= CHOICE {
        algIndependentKey       SecretKeyObject {SecretKeyAttributes},
        genericSecretKey                [15] SecretKeyObject {GenericKeyAttributes},
```

```
    ... -- Cho việc mở rộng sau này
    } -- Chú thích: Các thẻ đánh dấu ngữ cảnh [0] - [14] theo lịch sử và không được dùng
```

CHÚ THÍCH PKCS #1 dùng các thẻ này.

```
SecretKeyObject {KeyAttributes} ::= CIO {
    CommonKeyAttributes, CommonSecretKeyAttributes, KeyAttributes}
```

Loại này chứa thông tin liên quan đến một khóa bí mật. Mỗi giá trị gồm các thuộc tính chung cho bất kỳ đối tượng nào, bất kỳ khóa nào, bất kỳ khóa bí mật và các thuộc tính cụ thể cho khóa.

8.6.2 Thuật toán khóa thuật toán độc lập

Các đối tượng thể hiện các khóa bí mật sẵn có cho việc dùng trong nhiều thuật toán khác nhau hay cho việc biến đổi của các khóa bí mật.

```
SecretKeyAttributes ::= SEQUENCE {
    value    ObjectValue { OCTET STRING },
    ... -- Cho việc mở rộng sau này
    }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **SecretKeyAttributes.value**: Giá trị phải là một đường dẫn đến một tệp tin gồm cả một **OCTET STRING** (trong trường hợp một thẻ có thể thực hiện các thao tác khóa bí mật) một vài thể hiện đặc trưng thẻ của khóa.

8.6.3 Loại GenericSecretKey

Loại này nhằm chứa thông tin đặc trưng về một khóa bí mật với bất kỳ loại nào. Xem Điều 8.2.13.

8.7 Đối tượng thông tin chứng chỉ

8.7.1 CertificateChoice

```
CertificateChoice ::= CHOICE {
    x509Certificate         CertificateObject {X509CertificateAttributes},
    x509AttributeCertificate    [0] CertificateObject {X509AttributeCertificateAttributes},
    spkiCertificate         [1] CertificateObject {SPKICertificateAttributes},
    pgpCertificate          [2] CertificateObject {PGPCertificateAttributes},
    wtlsCertificate         [3] CertificateObject {WTLSCertificateAttributes},
    x9-68Certificate            [4] CertificateObject {X9-68CertificateAttributes},
    cvCertificate           [5] CertificateObject {CVCertificateAttributes},
    genericCertificateObject    [6] CertificateObject {GenericCertificateAttributes},
    ... -- Cho việc mở rộng sau này
    }
CertificateObject {CertAttributes} ::= CIO {
    CommonCertificateAttributes, NULL, CertAttributes}
```

Loại này chứa thông tin liên quan đến một chứng chỉ. Mỗi giá trị gồm các thuộc tính chung cho bất kỳ đối tượng nào, bất kỳ chứng chỉ nào và các thuộc tính cụ thể cho chứng chỉ.

8.7.2 Thuộc tính chứng chỉ X.509

```
X509CertificateAttributes ::= SEQUENCE {
        value                   ObjectValue { Certificate },
        subject         Name OPTIONAL,
        issuer                  [0] Name OPTIONAL,
        serialNumber    CertificateSerialNumber OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **X509CertificateAttributes.value:** Giá trị phải là một **ReferencedValue** hoặc là việc xác định một tệp tin chứa một chứng chỉ DER mã hóa tại địa điểm nhất định, hoặc một URL trỏ đến một số địa điểm mà chứng chỉ có thể được tìm thấy.
- **X509CertificateAttributes.subject** **X509CertificateAttributes.issuer** và **X509CertificateAttributes.serialNumber:** Ngữ nghĩa của thành phần này cũng giống như với thành phần tương ứng trong ISO/IEC 9594-8:1998. Các giá trị của các thành phần này phải giống chính xác như đối với các thành phần tương ứng trong chính chứng chỉ. Lý do để làm cho chúng tùy chọn là cung cấp một vài hiệu quả không gian, vì chúng được trình bày trong chính chứng chỉ.

8.7.3 Thuộc tính chứng chỉ thuộc tính X.509

```
X509AttributeCertificateAttributes ::= SEQUENCE {
        value                   ObjectValue { AttributeCertificate },
        issuer                  GeneralNames OPTIONAL,
        serialNumber    CertificateSerialNumber OPTIONAL,
        attrTypes               [0] SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **X509AttributeCertificateAttributes.value:** Giá trị phải là một **ReferencedValue** hoặc là việc xác định một tệp tin chứa một chứng chỉ DER mã hóa tại địa điểm nhất định, hoặc một URL trỏ đến một số địa điểm mà chứng chỉ có thể được tìm thấy.
- **X509AttributeCertificateAttributes.issuer** và **X509AttributeCertificateAttributes.serialNumber:** Giá trị của các thành phần này phải giống chính xác như với các thành phần tương ứng trong chính chứng chỉ thuộc tính. Chúng có thể được lưu trữ tường minh để tra cứu dễ dàng hơn.
- **X509AttributeCertificateAttributes.attrTypes:** Thành phần tùy chọn này phải, khi được trình bày, gồm một danh sách định danh đối tượng cho các thuộc tính được trình bày trong chứng

chỉ thuộc tính này. Điều này tạo cơ hội cho các ứng dụng tìm kiếm một chứng chỉ thuộc tính cụ thể mà không cần đọc và phân tích chính chứng chỉ đó.

8.7.4 Thuộc tính chứng chỉ SPKI

CHÚ THÍCH Các chứng chỉ SPKI được quy định trong IETF RFC 269 (xem Danh mục thư mục tài liệu tham khảo)

```
SPKICertificateAttributes ::= SEQUENCE {
        value    ObjectValue { CIO-OPAQUE.&Type },
        ... -- Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **SPKICertificateAttributes.value:** Giá trị phải là một **ReferencedValue** hoặc là việc xác định một tệp tin chứa một chứng chỉ SPKI tại địa điểm nhất định, hoặc một URL trỏ đến một số địa điểm mà chứng chỉ có thể được tìm thấy.

8.7.5 Thuộc tính chứng chỉ PGP (Pretty Good Privacy)

CHÚ THÍCH Các chứng nhận PGP được quy định trong IETF RFC 2440 (xem Danh mục thư mục tài liệu tham khảo)

```
PGPCertificateAttributes ::= SEQUENCE {
        value    ObjectValue { CIO-OPAQUE.&Type },
        ... -- Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PGPCertificateAttributes.value:** Giá trị phải là một **ReferencedValue** hoặc là việc xác định một tệp tin chứa một chứng chỉ PGP tại địa điểm nhất định, hoặc một URL trỏ đến một số địa điểm mà chứng chỉ có thể được tìm thấy.

8.7.6 Thuộc tính chứng chỉ WTLS

CHÚ THÍCH Các chứng nhận WTLS được quy định trong quy định kỹ thuật "Wireless Transport Layer Security Protocol" (xem Danh mục thư mục tài liệu tham khảo)

```
WTLSCertificateAttributes ::= SEQUENCE {
        value    ObjectValue { CIO-OPAQUE.&Type },
        ... -- Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **WTLSCertificateAttributes.value:** Giá trị phải là một **ReferencedValue** định danh cả một tệp tin chứa một chứng chỉ WTLS được mã hóa tại địa điểm nhất định, hoặc một URL trỏ đến một số địa điểm mà chứng chỉ có thể được tìm thấy.

8.7.7 Thuộc tính chứng chỉ miền ANSI X9.68

CHÚ THÍCH Các chứng nhận tên miền ANSI X9.68 được quy định trong ANSI X9.68-2001 (xem Danh mục thư mục tài liệu tham khảo)

```
X9-68CertificateAttributes ::= SEQUENCE {
        value    ObjectValue { CIO-OPAQUE.&Type },
        ... -- Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **X9-68CertificateAttributes.value:** Giá trị phải là một **ReferencedValue** định danh cả một tệp tin chứa một chứng chỉ tên miền ANSI X9.68 được mã hóa kiểu DER hay PER(ISO/IEC 8825-2:1998) tại địa điểm nhất định, hoặc một URL trỏ đến một số địa điểm mà chứng chỉ có thể được tìm thấy.

8.7.8 Thuộc tính chứng chỉ có thể xác minh thẻ

CHÚ THÍCH Các chứng nhận Card Verifiable Certificate được quy định trong TCVN 11167-8. Mục đích sử dụng chính của chúng là trong các phương pháp chứng thực thẻ dựa trên khóa công khai.

```
CVCertificateAttributes ::= SEQUENCE{
        value    ObjectValue {CIO-OPAQUE.&Type},
        certificationAuthorityReference OCTET STRING OPTIONAL
        ... - Cho việc mở rộng sau này
        }
```

Việc biên dịch các thành phần phải được quy định như sau:

- **CVCertificateAttributes.value:** Giá trị phải là một **ReferencedValue** định danh cả một tệp tin chứa một chứng chỉ có thể xác minh thẻ theo TCVN 11167-8 tại địa điểm nhất định, hoặc một URL trỏ đến một số địa điểm mà chứng chỉ có thể được tìm thấy.

8.7.9 Thuộc tính chứng chỉ chung

Loại này nhằm chứa thông tin đặc trưng về một chứng chỉ của bất kỳ loại nào. Xem Điều 8.2.16.

8.8 Đối tượng thông tin bộ chứa dữ liệu

8.8.1 DataContainerObjectChoice

```
DataContainerObjectChoice ::= CHOICE {
        opaqueDO       DataContainerObject {OpaqueDOAttributes},
        iso7816DO      [0] DataContainerObject {ISO7816DOAttributes},
        oIDDO [1] DataContainerObject {OIDDOAttributes},
        ... -- Cho việc mở rộng sau này
        }
DataContainerObject {DataObjectAttributes} ::= CIO {
        CommonDataContainerObjectAttributes, NULL, DataObjectAttributes}
```

Loại này chứa thông tin liên quan đến một đối tượng chứa dữ liệu. Mỗi giá trị gồm các thuộc tính chung cho bất kỳ đối tượng nào, bất kỳ đối tượng chứa dữ liệu nào và thuộc tính cụ thể đến các đối tượng chứa dữ liệu.

8.8.2 Thuộc tính đối tượng bộ chứa dữ liệu

Việc biên dịch các đối tượng này nằm trong việc truy cập ứng dụng của chúng.

```
OpaqueDOAttributes ::= ObjectValue {CIO-OPAQUE.&Type}
```

8.8.3 Thuộc tính đối tượng dữ liệu của bộ TCVN 11167

EF.DCOD có thể chứa thông tin về một hay một số các IDO. Các đối tượng này phải được tuân theo một kịch bản phân bổ thẻ tương thích được quy định trong TCVN 11167-4 (ISO/IEC 7816-4).

```
ISO7816DOAttributes ::= ObjectValue {CIO-OPAQUE.&Type}
        (CONSTRAINED BY {-- Tất cả các đối tượng dữ liệu này phải được quy định phụ thuộc vào TCVN 11167-4
        (ISO/IEC 7816-4)-})
```

Mỗi mục **iso7816DO** trong một EF.DCOD sẽ tham chiếu một tệp tin, mà phải phù hợp với TCVN 11167-4 (ISO/IEC 7816-4). Bằng cách dùng các đối tượng chứa dữ liệu này, các ứng dụng tăng cường khả năng tương tác.

Khi CDE được tham chiếu là một đối tượng dữ liệu được lấy ra, ví dụ: trong một lệnh 'GET DATA', lựa chọn **direct** của **ObjectValue** phải được dùng và giá trị **CIO-OPAQUE.&Type** phải là thẻ của đối tượng dữ liệu.

8.8.4 Đối tượng thông tin bộ chứa dữ liệu được định danh bởi OBJECT IDENTIFIERS

Loại này cung cấp một cách thức lưu trữ, tìm kiếm và lấy các đối tượng chứa dữ liệu với định danh đối các định danh đối tượng được gán. Một ví dụ cho loại thông tin này là bất kỳ **ATTRIBUTE** của ASN.1 (xem ISO/IEC 9594-6:1998).

```
OIDDOAttributes ::= SEQUENCE {
        ID      CIO-OPAQUE.&ID ({AllowedOIDDOs}),
        value   CIO-OPAQUE.&Type ({AllowedOIDDOs}{@ID})
        }
AllowedOIDDOs CIO-OPAQUE ::= {...}
```

8.9 Đối tượng thông tin chứng thực

8.9.1 AuthenticationObjectChoice

```
AuthenticationObjectChoice ::= CHOICE {
        pwd                 AuthenticationObject { PasswordAttributes },
        biometricTemplate   [0] AuthenticationObject{ BiometricAttributes},
        authKey             [1] AuthenticationObject {AuthKeyAttributes},
        external            [2] AuthenticationObject {ExternalAuthObjectAttributes},
        ... -- Cho việc mở rộng sau này
        }
AuthenticationObject {AuthObjectAttributes} ::= CIO {
```

CommonAuthenticationObjectAttributes, NULL, AuthObjectAttributes}

Loại này chứa thông tin về một phương pháp chứng thực cụ thể. Mỗi đối tượng chứng thực phải có một **CommonAuthenticationObjectAttributes.authID** khác biệt, cho phép tra cứu đối tượng chứng thực rõ ràng cho các đối tượng riêng.

8.9.2 Thuộc tính mật khẩu

```
PasswordAttributes ::= SEQUENCE {
  pwdFlags          PasswordFlags,
  pwdType           PasswordType,
  minLength         INTEGER (cia-lb-minPasswordLength..cia-ub-minPasswordLength),
  storedLength INTEGER (0..cia-ub-storedPasswordLength),
  maxLength   INTEGER OPTIONAL,
  pwdReference      [0] Reference DEFAULT 0,
  padChar           OCTET STRING (SIZE(1)) OPTIONAL,
  lastPasswordChange GeneralizedTime OPTIONAL,
  path          Path OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
PasswordFlags ::= BIT STRING {
      case-sensitive          (0),
      local                   (1),
      change-disabled         (2),
      unblock-disabled        (3),
      initialized             (4),
      needs-padding           (5),
      unblockingPassword      (6),
      soPassword              (7),
      disable-allowed         (8),
      integrity-protected     (9),
      conflDentiality-protected (10),
      exchangeRefData         (11),
      resetRetryCounter1      (12),
      resetRetryCounter2      (13)
      } (CONSTRAINED BY { -- 'unblockingPassword' và 'soPassword' không thể cùng thiết lập -- })
PasswordType ::= ENUMERATED {bcd, ascii-numeric, utf8, half-nibble-bcd, iso9564-1, ...}
```

Việc biên dịch các thành phần phải được quy định như sau:

- **PasswordAttributes.pwdFlags:** Thành phần này ra dấu nếu mật khẩu:
  - là **case-sensitive** nghĩa là một mật khẩu cho người dùng phải không được chuyển đổi thành viết hoa trước khi được trình bày trong thẻ (xem bên dưới)

- là **local** nghĩa là mật khẩu là cục bộ với ứng dụng mà nó thuộc về.

CHÚ THÍCH Một mật khẩu không phải "cục bộ", được coi là "toàn cầu". Một mật khẩu cục bộ chỉ được dùng để bảo vệ dữ liệu trong một ứng dụng nhất định. Đối với một mật khẩu cục bộ, vòng đời xác minh không được bảo đảm và nó có thể được tái xác minh theo mỗi lần sử dụng. Ngược lại với điều này, một xác minh thành công của mật khẩu toàn cầu có nghĩa là xác minh vẫn có hiệu lực cho đến khi thẻ được gỡ bỏ hoặc thiết lập lại, hoặc cho đến khi xác minh mới của cùng một mật khẩu không thành công. Một ứng dụng, trong đó xác nhận một mật khẩu toàn cầu, có thể giả định rằng mật khẩu còn hiệu lực, ngay cả khi các ứng dụng khác xác minh các mật khẩu cục bộ của chính nó, chọn các DF khác, v..v

- là **change-disabled** nghĩa là nó không thể thay đổi mật khẩu;
- là **unblock-disabled** nghĩa là nó không thể mở mật khẩu;
- là **initialized** nghĩa là mật khẩu đã được khởi tạo;
- **needs-padding** nghĩa là phụ thuộc độ dài của mật khẩu đã cho và độ dài lưu trữ, mật khẩu cần được chèn trước khi được trình bày trong thẻ;
- là một **unblockingPassword** (TCVN 11167-4 *đặt lại mã*) nghĩa là mật khẩu này có thể dùng cho mục đích mở như: đặt lại bộ đếm thử lại của đối tượng chứng thực liên quan cho giá trị ban đầu;
- là một **soPassword** nghĩa là mật khẩu là một mật khẩu (quản trị) của nhân viên an ninh;

CHÚ THÍCH Khi mật khẩu được mô tả bởi các CIO, các đối tượng chứng thực khác có thể bảo vệ chúng. Điều này đưa ra một cách để xác định mật khẩu có thể được dùng để mở khóa (tức là: thiết lập lại bộ đếm thử lại cho) mật khẩu khác - để **authID** của một đối tượng thông tin mật khẩu cho một đối tượng chứng thực mật khẩu mở khóa.

- là **disable-allowed** nghĩa là mật khẩu có thể bị tắt;
- phải được trình bày trong thẻ với thông điệp an ninh (**integrity-protected**);
- phải được trình bày trong thẻ với thẻ được mã hóa (**confidentiality-protected**);
- có thể thay đổi bởi bằng việc trình bày dữ liệu tham chiếu mới cho thẻ hay cả dữ liệu tham chiếu cũ và mới cần được trình bày. Nếu bit được thiết lập và dữ liệu tham chiếu cũ và mới phải được trình bày; ngoặc lại chỉ dữ liệu tham chiếu mới cần được trình bày (**exchangeRefData**).

– **PasswordAttributes.pwdType:** thành phần này xác định loại mật khẩu sau:

- **bcd** (thập phân mã hóa nhị phân, mỗi nhóm 4 bit của một byte phải gồm một số của mật khẩu);
- **ascii-numeric** (mỗi byte của mật khẩu gồm một ASCII (ANSI X3.4, xem danh mục thư viện tài liệu tham khảo) số mã hóa);
- **utf8** (mỗi kí tự được mã hóa phụ thuộc UTF-8);

- **half-nibble-bcd** (nhóm4 bit thấp nhất của một byte phải gồm một số của mật khẩu, nhóm 4 bit cao nhất phải gồm 'F')l hoặc
- **iso9564-1** (mã hóa phụ thuộc vào ISO 9564-1:1996).

- **PasswordAttributes.minLength:** Chiều dài tối thiểu (bằng ký tự) của mật khẩu mới (nếu được phép thay đổi).
- **PasswordAttributes.storedLength:** Chiều dài lưu trữ trên thẻ (theo byte). Được dùng để suy ra số ký tự chèn cần thiết. Giá trị có thể được đặt là 0 và bỏ qua nếu **pwdFlags** chỉ ra rằng việc chèn là không cần thiết (tức là không có ký tự chèn được gửi vào thẻ).
- **PasswordAttributes.maxLength:** Trên một số thẻ, mật khẩu là không được chèn và do đó có một nhu cầu để biết chiều dài mật khẩu tối đa (bằng ký tự) được phép.
- **PasswordAttributes.pwdReference**: Thành phần này là một tham chiếu đặc trưng thẻ cho mật khẩu. Dự đoán rằng nó có thể được dùng như một tham số 'P2' trong lệnh 'VERIFY' của TCVN 11167-4 (ISO/IEC 7816-4) khi áp dụng. Nếu không được thể hiện, nó có giá trị mặc định là **0**.
- **PasswordAttributes.padChar**: Kí tự chèn được dùng (thường là 'FF' hoặc '00'). Không cần thiết nếu **pwdFlags** chỉ ra rằng việc chèn là không cần thiết cho thẻ này. Nếu **asswordAttributes.pwdType** là loại **bcd**, thì **padChar** nên bao gồm hai nhóm 4 bit có cùng giá trị, bất kỳ nhóm 4 bit nào được sử dụng như là "chèn nhóm 4 bit". Ví dụ, '55' được phép, nghĩa là chèn với '0101_2', nhưng '34' là bất hợp pháp.
- **PasswordAttributes.lastPasswordChange:** Thành phần này được thiết kế để nhằm sử dụng trong các ứng dụng đòi hỏi kiến thức về thời gian mà mật khẩu cuối được thay đổi (ví dụ: để thực thi các chính sách gia hạn mật khẩu). Khi mật khẩu không được thiết lập (hoặc không bao giờ được thay đổi) giá trị phải là (dùng ký hiệu giá trị được quy định nghĩa trong ISO/IEC 8824-1:1998) **'000000000000Z'**. Một ví dụ khác: một mật khẩu thay đổi vào ngày 6 tháng 1 năm 1999 ở năm 1934 (lúc 7:34 PM) UTC sẽ có một giá trị **lastPasswordChange** của **'19990106193400Z'**.
- **PasswordAttributes.path**: Đường dẫn tới DF trong đó các mật khẩu nằm tại đó. Đường dẫn này phải được chọn bởi một ứng dụng má chủ trước khi thực hiện một thao tác mật khẩu, để cho phép một ngữ cảnh chứng thực thích hợp cho các thao tác mật khẩu. Nếu không được trình bày, việc xác minh chủ thẻ phải luôn có thể thực hiện mà không có một thao tác 'SELECT' trước đó.

8.9.2.1 Mã hóa một mật khẩu được cung cấp

Các bước thực hiện bởi một ứng dụng máy chủ để mã hóa mật khẩu được người dùng cung cấp về điều gì trình bày cho thẻ được quy định như sau:

a) Chuyển đổi mật khẩu theo loại mật khẩu sau:

1) Nếu mật khẩu là mật khẩu **utf8**, biến đổi nó thành UTF-8: *x = UTF8 (mật khẩu)*. Sau đó, nếu bit **case-sensitive** bị tắt, chuyển đổi x thành chữ hoa: *x = NLSUPPERCASE (x)(NLSUPPERCASE* = định vị chữ hoa liên quan)

2) Nếu mật khẩu là mật khẩu **bcd**, xác minh mỗi ký tự là một chữ số và mã hóa các ký tự thành các số BCD: *x = BCD (mật khẩu)*

3) Nếu mật khẩu là mật khẩu **ascii-numeric** hay **iso9564-1**, xác minh mỗi ký tự là một chữ số dưới dạng trang mã hiện tại và - nếu cần, mã hóa các ký tự thành chữ số ASCII: *x = ASCII (mật khẩu)*

4) Nếu mật khẩu là mật khẩu **half-nibble-bcd**, xác minh mỗi ký tự là một chữ số và mã hóa các ký tự như BCD trong nửa dưới của mỗi byte, thiết lập mỗi nhóm 4 bit thành 'F_{16}': *x = Half-BCD (mật khẩu)*

b) Nếu chỉ ra trong thành phần **pwdFlags**, chèn x bên phải với các ký tự chèn *padChar*, để lưu trữ dài *storedLength*: *x = PAD (x, padChar, storedLength)*.

c) Nếu các bit **pwdFlags.integrity-protected** hay **pwdFlags.confidentiality-protected** được thiết lập, áp dụng các thuật toán và các khóa tương ứng cho mật khẩu đã được chuyển đổi và định dạng.

d) Trình bày các mật khẩu cho thẻ.

Ví dụ (ascii-) Mạt khẩu số 1234, lưu trữ chiều dài 8 byte, và ký tự chèn 'FF' cho giá trị được trình bày trên là '31323334FFFFFFFF'

8.9.3 Thuộc tính dữ liệu tham chiếu sinh trắc học

Loại này chỉ liên quan đến thẻ có khả năng thực hiện các chứng thực bằng cách so sánh dữ liệu tham chiếu sinh trắc học được lưu trữ với dữ liệu xác minh sinh trắc học được trình bày, có chứa thông tin về dữ liệu tham chiếu sinh trắc học được lưu trữ ("mẫu").

```
BiometricAttributes ::= CHOICE {
  biometricTemplateAttributes BiometricTemplateAttributes,
  bit                         [APPLICATION 96] BiometricInformationTemplate,
  bitGroup                    [APPLICATION 97] BiometricInformationTemplateGroup
  }
BiometricInformationTemplate ::= OCTET STRING
-- Phải gồm một giá trị mẫu thông tin sinh trắc học của TCVN 11167-11
BiometricInformationTemplateGroup ::= OCTET STRING
-- Phải gồm một giá trị mẫu nhóm thông tin sinh trắc học của TCVN 11167-11
BiometricTemplateAttributes ::= SEQUENCE {
  bioFlags            BiometricFlags,
  templateID   BiometricTemplateIDentifier,
  bioType             BiometricType,
  bioReference        Reference DEFAULT 0,
```

```
    lastChange    GeneralizedTime OPTIONAL,
    path          Path OPTIONAL,
    ... -- Cho việc mở rộng sau này
    }
BiometricTemplateIDentifier ::= CHOICE {
    oID           OBJECT IDENTIFIER,
    issuerID      OCTET STRING,
    ... -- Cho việc mở rộng sau này
    }
BiometricFlags ::= BIT STRING {
        local                       (1),
        change-disabled             (2),
        unblock-disabled            (3),
        initialized                 (4),
        disable-allowed         (8),
        integrity-protected         (9),
        conflDentiality-protected   (10)
        }
BiometricType ::= CHOICE {
        fingerPrint     FingerPrintInformation,
        iris            [0] IrisInformation,
        chained [1] SEQUENCE SIZE (2..cia-ub-biometricTypes) OF BiometricType,
        ... -- Cho việc mở rộng sau này
        }
FingerPrintInformation ::= SEQUENCE {
        hand    ENUMERATED {left, right},
        finger  ENUMERATED {thumb, pointerFinger, mIDdleFinger, ringFinger, littleFinger},
        }
IrisInformation ::= SEQUENCE {
        eye     ENUMERATED {left, right},
        ... -- Cho việc mở rộng sau này
        }
```

Loại **BiometricAttributes** đưa ra cho hai cách khác nhau nhằm trình bày thông tin về dữ liệu tham chiếu sinh trắc học được lưu trữ:

- qua các thông tin cụ thể cho tiêu chuẩn này (thành phần **biometricTemplateAttributes**); hoặc
- qua thông tin được quy định trong TCVN 11167-11 (các thành phần **bit** và **bitGroup**).

Ngữ nghĩa của các thành phần của loại BiometricTemplateAttributes là như sau:

- **BiometricAttributes.bioFlags**: Tương tự với **PasswordAttributes.pwdFlags**, nhưng thay thế "mật khẩu" thành "dữ liệu tham chiếu sinh trắc học".
- **BiometricAttributes.templateId**: Thành phần này chỉ ra cấu trúc dữ liệu đã được gửi vào thẻ.
- **BiometricAttributes.bioType**: Thành phần này xác định loại thông tin sinh trắc học được lưu trữ trong thẻ, ví dụ: ngón tay trỏ phải. Thành phần **"chained"** có nghĩa là nhiều hơn một chức năng sinh trắc học đã được trình bày trong cùng quy trình xác minh, có thể bằng cách sử dụng lệnh xâu chuỗi, để chứng thực thành công.
- **BiometricAttributes.bioReference**, **BiometricAttributes.lastChange** và **BiometricAttributes.path**: Khi cho các thành phần tương ứng trong **PasswordAttributes** nhưng thay thế "mật khẩu" với "dữ liệu tham chiếu sinh trắc học".

8.9.4 Đối tượng chứng thực đối với chứng thực bên ngoài

CHÚ THÍCH Điều này chỉ mô tả các cách thức chứng thực thẻ chứng thực bên trong hay bên ngoài không được nói tới.

```
ExternalAuthObjectAttributes ::= CHOICE {
        authKeyAttributes               AuthKeyAttributes,
        certBasedAttributes     [0] CertBasedAuthenticationAttributes,
        ... -- Cho việc mở rộng sau này
        }
AuthKeyAttributes ::= SEQUENCE {
        derivedKey      BOOLEAN DEFAULT TRUE,
        authKeyID       IDentifier,
        ... -- Cho việc mở rộng sau này
        }
CertBasedAuthenticationAttributes ::= SEQUENCE {
        cha     OCTET STRING,
        ...
        }
```

Việc biên dịch các loại nà phải tuân theo:

- **AuthKeyAttributes.derivedKey:** Thành phần này quy định xem khóa chứng thực được lưu trữ trong thẻ là một khóa phân phối (là một khóa cá nhân), một khóa nhóm, hay một khóa cái sử dụng tạo các khóa cá nhân.
- **AuthKeyAttributes.authKeyId:** Thành phần này quy định các định danh (**CommonKeyAttribute.ID**) của khóa chứng thực như được mô tả trong một EF.SKD.
- **CertBasedAuthenticationAttributes.cha:** Thành phần này quy định việc uỷ quyền cấp chứng chỉ như được trình bày trong chứng chỉ thẻ có thể kiểm chứng (xem TCVN 11167-8). Nếu chứng chỉ thẻ có thể kiểm chứng có chứa giá trị này được xác minh và thủ tục chứng thực với

cặp khóa tương ứng được hoàn tất, thì **cha** được thiết lập là hợp lệ, và truy cập tới các đối tượng riêng được bảo vệ trong ủy quyền cấp chứng chỉ.

8.10 Tệp tin thông tin mã hóa: EF.CIAInfo

Loại này gồm thông tin chung về DF.CIA và thẻ.

```
CIAInfo ::= SEQUENCE {
   version                    INTEGER {v1(0),v2(1)} (v1|v2,...),
   serialNumber               OCTET STRING OPTIONAL,
   manufacturerID             Label OPTIONAL,
   label                [0] Label OPTIONAL,
   cardflags                  CardFlags,
   seInfo                     SEQUENCE OF SecurityEnvironmentInfo OPTIONAL,
   recordInfo                 [1] RecordInfo OPTIONAL,
   supportedAlgorithms [2] SEQUENCE OF AlgorithmInfo OPTIONAL,
   issuerID                   [3] Label OPTIONAL,
   holderID                   [4] Label OPTIONAL,
   lastUpdate                 [5] LastUpdate OPTIONAL,
   preferredLanguage   PrintableString OPTIONAL, -- In accordance with IETF RFC 1766
   profileIndication          [6] SEQUENCE OF ProfileIndication OPTIONAL,
   ...
   } (CONSTRAINED BY { -- Mỗi giá trị AlgorithmInfo.reference phải là đơn nhất -})
CardFlags ::= BIT STRING {
   readonly           (0),
   authRequired       (1),
   prnGeneration      (2)
   } -- Bit (3) được bảo toàn vì lí do lịch sử
SecurityEnvironmentInfo ::= SEQUENCE {
   se    INTEGER,
   owner         OBJECT IDENTIFIER OPTIONAL,
   aID   OCTET STRING
   (CONSTRAINED BY {-- Phải được mã hóa theo TCVN 11167-4 --}) OPTIONAL,
   ... -- Cho việc mở rộng sau này
   }
RecordInfo ::= SEQUENCE {
   oDRecordLength             [0] INTEGER (0..cia-ub-recordLength) OPTIONAL,
   prKDRecordLength  [1] INTEGER (0..cia-ub-recordLength) OPTIONAL,
   puKDRecordLength  [2] INTEGER (0..cia-ub-recordLength) OPTIONAL,
   sKDRecordLength   [3] INTEGER (0..cia-ub-recordLength) OPTIONAL,
   cDRecordLength             [4] INTEGER (0..cia-ub-recordLength) OPTIONAL,
   dCODRecordLength  [5] INTEGER (0..cia-ub-recordLength) OPTIONAL,
```

```
  aODRecordLength    [6] INTEGER (0..cia-ub-recordLength) OPTIONAL
  }
AlgorithmInfo ::= SEQUENCE {
  reference                  Reference,
  algorithm                  CIO-ALGORITHM.&ID({AlgorithmSet}),
  parameters          CIO-ALGORITHM.&Parameters({AlgorithmSet}{@algorithm}),
  supportedOperations CIO-ALGORITHM.&Operations({AlgorithmSet}{@algorithm}),
  objID                      CIO-ALGORITHM.&objectIDentifier ({AlgorithmSet}{@algorithm}),
  algRef                     Reference OPTIONAL
  }
LastUpdate ::= CHOICE {
  generalizedTime     GeneralizedTime,
  referencedTime      ReferencedValue,
  ... -- Cho việc mở rộng sau này
  }(CONSTRAINED BY {-- referencedValue phải là giá trị theo loại GeneralizedTime --})
ProfileIndication ::= CHOICE {
  profileOID          OBJECT IDENTIFIER,
  profileName UTF8String,
  ... -- Cho việc mở rộng sau này
  }
```

EF.CIAInfo phải có một giá trị DER mã hóa thuộc loại **CIAInfo**.

Việc biên dịch loại **CIAInfo** được quy định như sau:

- **CIAInfo.version**: Thành phần này phải được thiết lập để **v2** cho phiên bản này của tiêu chuẩn. Phiên bản dự kiến có thể sử dụng các giá trị khác. Một giá trị CIAInfo phải không bị từ chối chỉ vì nó có một số phiên bản không rõ.

CHÚ THÍCH Số phiên bản **v1** được sử dụng trong các cấu trúc tương đương trong PKCS # 15.

- **CIAInfo.serialNumber**: Thành phần này phải có số seri đơn nhất của CIA, được chọn bởi các bên cung cấp ứng dụng.
- **CIAInfo.manufacturerID**: Thành phần tùy chọn này phải chứa thông tin định danh về các nhà sản xuất thẻ, UTF-8 được mã hóa.
- **CIAInfo.label**: Thành phần tùy chọn này phải chứa thông tin định danh về các ứng dụng.
- **CIAInfo.cardflags**: Thành phần này chứa thông tin về thẻ *per se*. Các thẻ bao gồm: Nếu thẻ chỉ đọc, nếu có chức năng mã hóa yêu cầu người dùng được chứng thực, và nếu các thẻ hỗ trợ tạo số giả ngẫu nhiên.
- **CIAInfo.seInfo**: Thành phần tùy chọn này được thiết kế để truyền đạt thông tin về môi trường an ninh thiết lập sẵn trên thẻ, và chủ sở hữu của các môi trường này. Định nghĩa của các môi

trường này hiện nằm ngoài phạm vi của tiêu chuẩn, xem thêm TCVN 11167-4 (ISO/IEC 7816-4). Thành phần **aID** chỉ ra ứng dụng (thẻ) với môi trường an ninh được áp dụng.

- **CIAInfo.recordInfo**: Thành phần tùy chọn này có hai mục đích:
  - cho biết các tệp tin cơ bản: EF.OD, EF.PrKD, EF.PuKD, EF.SKD, EF.CD, EF.DCOD và EF.AOD là các tệp tin ghi tuyến tính hay các tệp tin minh bạch (nếu các thành phần có mặt, chúng phải là các tệp tin ghi tuyến tính, nếu không chúng phải là các tệp tin minh bạch); và
  - nếu chúng là các tệp tin ghi tuyến tính, cho dù chúng có chiều dài cố định hay không (nếu chúng có chiều dài cố định, giá trị tương ứng trong **RecordInfo** được xem xét và không bằng 0 và chỉ ra độ dài bản ghi. Nếu một số tệp tin là tập bản ghi tuyến tính nhưng không có độ dài cố định, sau đó các giá trị tương ứng trong **RecordInfo** phải được đặt là 0.
- **CIAInfo.supportedAlgorithms**: Mục đích của thành phần tùy chọn này là để chỉ ra các thuật toán mã hóa, thông số, hoạt động liên quan và các định dạng đầu vào thuật toán được hỗ trợ bởi thẻ. Thành phần **reference** của **AlgorithmInfo** là một tham chiếu đơn nhất được sử dụng cho mục đích tham chiếu chéo từ các PrKD và PuKD. Giá trị cho các thành phần **algorithm** được sử dụng cá nhân. Giá trị của thành phần **supportedOperations** (**compute-checksum**, compute-signature, **verify-checksum, verify-signature, encipher, decipher, hash** và **derive-key**) xác định các hoạt động của *thẻ* có thể thực hiện với một thuật toán cụ thể. Thành phần **objID** chỉ ra định danh đối tượng với các thuật toán. Thành phần **algRef** chỉ ra định danh được sử dụng bởi các thẻ cho việc biểu thị thuật toán này (và xảy ra tại giao diện thẻ như một tham số, ví dụ: một lệnh "EXTERNAL AUTHENTICATE").

CHÚ THÍCH Giá trị đối với thành phần **algorithm** có thể được lựa chọn từ, và được biện dịch như số lượng cơ chế trong PKCS # 11 (xem Danh mục tài liệu tham khảo).

- **CIAInfo.issuerID**: Thành phần tùy chọn này chứa thông tin định danh về tổ chức phát hành thẻ (ví dụ: các công ty phát hành thẻ).
- **CIAInfo.holderID**: Thành phần tùy chọn này chứa thông tin định danh về chủ thẻ (ví dụ: chủ thẻ).
- **CIAInfo.lastUpdate**: Thành phần tùy chọn này chứa (hoặc đề cập tới) ngày cập nhật cuối cùng của các tệp tin trong CIA. Sự hiện diện của thành phần nà, cùng với thành phần **CIAInfo.serialNumber** cho phép các ứng dụng phía chủ nhanh chóng tìm hiểu xem chúng phải đọc EF.OD, EF.CD, .v..v. hoặc nếu chúng có thể sử dụng bản sao cache (nếu có). Các **referencedTime** thay thế các loại **LastUpdate** dành cho những trường hợp khi **EF.CIAInfo** cần ghi-bảo vệ.
- **CIAInfo.preferredLanguage**: Các ngôn ngữ ưa thích của chủ thẻ, được mã hóa phù hợp với IETF RFC 1766.
- **CIAInfo.profileIndication**: Thành phần tùy chọn này chỉ ra các hồ sơ của tiêu chuẩn này, mà thẻ đã được phát hành theo sự phù hợp với chuẩn.

CHÚ THÍCH Hãy để cho các thông số kỹ thuật khác nhằm xác định các hồ sơ chuẩn hóa của tiêu chuẩn này.

Phụ lục A

(tham khảo)

Mô đun ASN.1

Phụ lục này bao gồm tất cả các loại, giá trị và định nghĩa lớp đối tượng thông tin của ASN.1 được đề cập trong tiêu chuẩn này, dưới dạng mô-đun **CryptographicInformationFramework**.

```
CryptographicInformationFramework {iso(1) standard(0) 7816 15 1}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
IMPORTS
informationFramework, authenticationFramework, certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t(2) ds(5) module(1) usefulDefinitions(0) 3}
Name
        FROM InformationFramework informationFramework
Certificate, AttributeCertificate, CertificateSerialNumber, SubjectPublicKeyInfo, AlgorithmIDentifier, validity
        FROM AuthenticationFramework authenticationFramework
GeneralName, GeneralNames, KeyUsage
        FROM CertificateExtensions certificateExtensions
ECPoint, Parameters
        FROM ANSI-X9-62 {iso(1) member-body(2) us(840) ansi-x962(10045) module(4) 1}
DomainParameters
        FROM ANSI-X9-42 {iso(1) member-body(2) us(840) ansi-x942(10046) module(5) 1};
```

A.1 Giới hạn chữ viết hoa và chữ viết thường

```
cia-ub-IDentifier               INTEGER ::= 255
cia-ub-reference                INTEGER ::= 255
cia-ub-index                            INTEGER ::= 65535
cia-ub-label                            INTEGER ::= cia-ub-IDentifier
cia-lb-minPasswordLength                INTEGER ::= 4
cia-ub-minPasswordLength                INTEGER ::= 8
cia-ub-storedPasswordLength     INTEGER ::= 64
cia-ub-recordLength                     INTEGER ::= 16383
cia-ub-userConsent                      INTEGER ::= 15
cia-ub-securityConditions       INTEGER ::= 255
cia-ub-biometricTypes           INTEGER ::= 127
```

A.2 Loại cơ bản

A.2.1

```
IDentifier ::= OCTET STRING (SIZE (0..cia-ub-IDentifier))
```

A.2.2

```
Reference ::= INTEGER (0..cia-ub-reference)
```

A.2.3

```
Label ::= UTF8String (SIZE(0..cia-ub-label))
```

A.2.4

```
CredentialIDentifier {KEY-IDENTIFIER : IDentifierSet} ::= SEQUENCE {
        IDType KEY-IDENTIFIER.&ID ({IDentifierSet}),
        IDValue KEY-IDENTIFIER.&Value ({IDentifierSet}{@IDType})
        }
KeyIdentifiers KEY-IDENTIFIER ::= {
        issuerAndSerialNumber           |
        issuerAndSerialNumberHash       |
        subjectKeyid                            |
        subjectKeyHash                  |
        issuerKeyHash                   |
        issuerNameHash                  |
        subjectNameHash                         |
        pgp2KeyID                               |
        openPGPKeyID                    |
        certificateHolderReference,
        ...
        }
KEY-IDENTIFIER ::= CLASS {
        &ID INTEGER UNIQUE,
        &Value
        } WITH SYNTAX {
        SYNTAX &Value IDENTIFIED BY &ID
        }
IssuerAndSerialNumber ::= SEQUENCE {
        issuer Name,
        serialNumber CertificateSerialNumber
        }
issuerAndSerialNumber KEY-IDENTIFIER::=
        {SYNTAX IssuerAndSerialNumber IDENTIFIED BY 1}
issuerAndSerialNumberHash KEY-IDENTIFIER ::=
        {SYNTAX OCTET STRING IDENTIFIED BY 3}
        -- Mã băm SHA-1 của mã hóa DER với IssuerAndSerialNumber
subjectKeyID KEY-IDENTIFIER ::=
        {SYNTAX OCTET STRING IDENTIFIED BY 2}
        -- Từ mở rộng chứng chỉ của ISO/IEC 9594-8:1998
subjectKeyHash KEY-IDENTIFIER ::=
        {SYNTAX OCTET STRING IDENTIFIED BY 4}
issuerKeyHash KEY-IDENTIFIER ::=
        {SYNTAX OCTET STRING IDENTIFIED BY 5}
issuerNameHash KEY-IDENTIFIER ::=
        {SYNTAX OCTET STRING IDENTIFIED BY 6}
        -- Hàm băm SHA-1 của tên bên phát hành DER mã hóa
subjectNameHash KEY-IDENTIFIER ::=
        {SYNTAX OCTET STRING IDENTIFIED BY 7}
        -- Hàm băm SHA-1 của tên nội dung DER mã hóa
pgp2KeyID KEY-IDENTIFIER ::=
        {SYNTAX OCTET STRING (SIZE(8)) IDENTIFIED BY 8}
openPGPKeyID KEY-IDENTIFIER ::=
        {SYNTAX OCTET STRING (SIZE(8)) IDENTIFIED BY 9}
```

A.2.5

```
ReferencedValue ::= CHOICE {
        path    Path,
        url     URL
        } -- Cú pháp của đối tượng được xác định theo ngữ cảnh
URL ::= CHOICE {
        url             CHOICE {printable PrintableString, ia5 IA5String},
        urlWithDigest   [3] SEQUENCE {
                url     IA5String,
                digest  DigestInfoWithDefault
                }
        }
alg-ID-sha1 AlgorithmIDentifier ::= {
        algorithm       ID-sha1,
        parameters      SHA1Parameters : NULL
        }
ID-sha1 OBJECT IDENTIFIER ::= {iso(1) IDentified-organization(3) oiw(14) secsig(3) algorithms(2) 26 }
SHA1Parameters ::= NULL
DigestInfoWithDefault ::= SEQUENCE {
        digestAlg       AlgorithmIDentifier DEFAULT alg-ID-sha1,
        digest          OCTET STRING (SIZE(8..128))
        }


Path ::= SEQUENCE { efidOrTagChoice CHOICE
         { efidOrPath OCTET STRING, tagRef [0]
         SEQUENCE { tag OCTET STRING,
                        efidOrPath OCTET STRING OPTIONAL
             },
              appFileRef [1]     SEQUENCE {
                        aid [APPLICATION 15] OCTET STRING,
                efidOrpath OCTET STRING
             },
             appTagRef [2] SEQUENCE {
                        aid [APPLICATION 15] OCTET STRING,
                        tag OCTET STRING,
                        efidOrPath OCTET STRING OPTIONAL
             }
         },
         index INTEGER (0 .. cia-ub-index) OPTIONAL, length [0]
         INTEGER (0 .. cia-ub-index) OPTIONAL
} ( WITH COMPONENTS {..., index PRESENT, length PRESENT}| WITH COMPONENTS {..., index ABSENT, length
ABSENT} )
```

A.2.6

```
ObjectValue { Type } ::= CHOICE {
        indirect ReferencedValue,
        direct          [0] Type
        }
```

A.2.7

```
PathOrObjects {ObjectType} ::= CHOICE {
        path            Path,
```

```
        objects  [0] SEQUENCE OF ObjectType,
        ... -- Cho việc mở rộng sau này
        }
```

A.2.8

```
CommonObjectAttributes ::= SEQUENCE {
        label     Label OPTIONAL,
        flags     CommonObjectFlags OPTIONAL,
        authID  IDentifier OPTIONAL,
        userConsent       INTEGER (1..cia-ub-userConsent) OPTIONAL,
        accessControlRules SEQUENCE SIZE (1..MAX) OF AccessControlRule OPTIONAL,
        ... -- Cho việc mở rộng sau này
        } (CONSTRAINED BY {-- authID should be present if flags.private is set.
        -- Nó phải bằng một authID trong một đối tượng chứng thực trong AOD -- })

CommonObjectFlags ::= BIT STRING {
        private           (0),
        modifiable        (1),
        internal (2)
        } -- Bit (2) được trình bày theo lí do lịch sử và không được dùng

AccessControlRule ::= SEQUENCE {
        accessMode                AccessMode,
        securityCondition         SecurityCondition,
        ... -- Cho việc mở rộng sau này
        }

AccessMode ::= BIT STRING { read
                  (0),
        update (1),
        execute (2),
        delete    (3),
        attribute (4),
        pso_cds (5),
        pso_verif (6),
        pso_dec  (7),
        pso_enc  (8),
         int_auth (9),
         ext_auth (10)
}

SecurityCondition ::= CHOICE {
  always                  NULL,
  authID                  IDentifier,
  authReference AuthReference,
  not             [0] SecurityCondition,
  and             [1] SEQUENCE SIZE (2..cia-ub-securityConditions) OF SecurityCondition,
  or              [2] SEQUENCE SIZE (2..cia-ub-securityConditions) OF SecurityCondition,
  ... -- Cho việc mở rộng sau này
  }

AuthReference ::= SEQUENCE {
  authMethod    AuthMethod,
  seIDentifier    INTEGER OPTIONAL
  }

AuthMethod ::= BIT STRING {secureMessaging(0), extAuthentication(1), userAuthentication(2), always(3)}
```

A.2.9

```
CommonKeyAttributes ::= SEQUENCE {
  ID                    IDentifier,
  usage          KeyUsageFlags,
  native         BOOLEAN DEFAULT TRUE,
  accessFlags    KeyAccessFlags OPTIONAL,
  KeyReference  OPTIONAL,
  startDate             GeneralizedTime OPTIONAL,
  endDate               [0] GeneralizedTime OPTIONAL,
  algReference   [1] SEQUENCE OF Reference OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
KeyUsageFlags ::= BIT STRING {
        encipher                (0),
        decipher                (1),
        sign                    (2),
        signRecover             (3),
        keyEncipher             (4),
        keyDecipher             (5),
        verify                  (6),
        verifyRecover   (7),
        derive                  (8),
        nonRepudiation  (9)
        }
KeyAccessFlags ::= BIT STRING {
        sensitive               (0),
        extractable             (1),
        alwaysSensitive  (2),
        neverExtractable        (3),
        cardGenerated   (4)
        }
KeyReference ::= INTEGER
```

A.2.10

```
CommonPrivateKeyAttributes ::= SEQUENCE {
  name          Name OPTIONAL,
  keyIDentifiers [0] SEQUENCE OF CredentialIDentifier {{KeyIDentifiers}} OPTIONAL,
  generalName   [1] GeneralNames OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
```

A.2.11

```
CommonPublicKeyAttributes ::= SEQUENCE {
  name          Name OPTIONAL,
  trustedUsage   [0] Usage OPTIONAL,
  generalName   [1] GeneralNames OPTIONAL,
  keyIDentifiers [2] SEQUENCE OF CredentialIDentifier {{KeyIDentifiers}} OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
```

A.2.12

```
CommonSecretKeyAttributes ::= SEQUENCE {
  keyLen        INTEGER OPTIONAL, -- keylength (in bits)
  ... -- Cho việc mở rộng sau này
```

```
  }
```

A.2.13

```
GenericKeyAttributes ::= SEQUENCE {
  keyType      CIO-ALGORITHM.&objectIDentifier ({AllowedAlgorithms}),
  keyAttr      CIO-ALGORITHM.&Parameters ({AllowedAlgorithms}{@keyType})
  }
AllowedAlgorithms CIO-ALGORITHM ::= {...}
```

A.2.14

```
KeyInfo {ParameterType, OperationsType} ::= CHOICE {
  paramsAndOps        SEQUENCE {
      parameters      ParameterType,
      operations      OperationsType OPTIONAL
  },
  reference       Reference -- Theo lịch sử, không được dùng
  }
```

A.2.15

```
CommonCertificateAttributes ::= SEQUENCE {
  ID                  IDentifier,
  authority           BOOLEAN DEFAULT FALSE,
  IDentifier          CredentialIDentifier {{KeyIDentifiers}} OPTIONAL,
  certHash            [0] CertHash OPTIONAL,
  trustedUsage    [1] Usage OPTIONAL,
  IDentifiers         [2] SEQUENCE OF CredentialIDentifier {{KeyIDentifiers}} OPTIONAL,
  validity            [4] validity OPTIONAL,
  ...
  } -- Thẻ ngữ cảnh [3] được bảo toàn vì lí do lịch sử
Usage ::= SEQUENCE {
  keyUsage            KeyUsage OPTIONAL,
  extKeyUsage   SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL,
  ...
      } (WITH COMPONENTS {..., keyUsage PRESENT} | WITH COMPONENTS {..., extKeyUsage PRESENT})
CertHash ::= SEQUENCE {
  hashAlg       [0] EXPLICIT AlgorithmIDentifier OPTIONAL,
  certID        [1] EXPLICIT CertID OPTIONAL,
  hashVal BIT STRING
      }(CONSTRAINED BY {-- hashVal được tính toán với toàn bộ chứng chỉ DER mã hóa --})
CertID ::= SEQUENCE {
  issuer        GeneralName,
  serialNumber  CertificateSerialNumber
  }
```

A.2.16

```
GenericCertificateAttributes ::= SEQUENCE {
  certType CIO-OPAQUE.&ID ({AllowedCertificates}),
  certAttr CIO-OPAQUE.&Type ({AllowedCertificates}{@certType})
  }
AllowedCertificates CIO-OPAQUE ::= {...}
```

A.2.17

```
CommonDataContainerObjectAttributes ::= SEQUENCE {
  applicationName     Label OPTIONAL,
  applicationOID      OBJECT IDENTIFIER OPTIONAL,
```

```
  ID                    IDentifier OPTIONAL,
  ... -- Cho việc mở rộng sau này
      } (WITH COMPONENTS {..., applicationName PRESENT}| WITH COMPONENTS {..., applicationOID
      PRESENT})
```

A.2.18

```
CommonAuthenticationObjectAttributes ::= SEQUENCE {
  authID                IDentifier OPTIONAL,
  authReference Reference OPTIONAL,
  seIDentifier          [0] Reference OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
```

A.2.19

```
CIO {ClassAttributes, SubClassAttributes, TypeAttributes} ::= SEQUENCE {
  commonObjectAttributes      CommonObjectAttributes,
  classAttributes             ClassAttributes,
  subClassAttributes          [0] SubClassAttributes OPTIONAL,

  typeAttributes              [1] TypeAttributes
  }
```

A.3 Các CIO

```
CIOChoice ::= CHOICE {
  privateKeys                 [0] PrivateKeys,
  publicKeys                  [1] PublicKeys,
  trustedPublicKeys           [2] PublicKeys,
  secretKeys                  [3] SecretKeys,
  certificates                [4] Certificates,
  trustedCertificates         [5] Certificates,
  usefulCertificates          [6] Certificates,
  dataContainerObjects  [7] DataContainerObjects,
  authObjects           [8] AuthObjects,
  ... -- Cho việc mở rộng sau này
  }
PrivateKeys ::= PathOrObjects {PrivateKeyChoice}
PublicKeys ::= PathOrObjects {PublicKeyChoice}
SecretKeys ::= PathOrObjects {SecretKeyChoice}
Certificates ::= PathOrObjects {CertificateChoice}
DataContainerObjects ::= PathOrObjects {DataContainerObjectChoice}
AuthObjects ::= PathOrObjects {AuthenticationObjectChoice}
```

A.4 Các đối tượng thông tin khóa riêng

A.4.1

```
PrivateKeyChoice ::= CHOICE {
      privateRSAKey   PrivateKeyObject {PrivateRSAKeyAttributes},
      privateECKey          [0] PrivateKeyObject {PrivateECKeyAttributes},
      privateDHKey    [1] PrivateKeyObject {PrivateDHKeyAttributes},
      privateDSAKey   [2] PrivateKeyObject {PrivateDSAKeyAttributes},
      privateKEAKey   [3] PrivateKeyObject {PrivateKEAKeyAttributes},
      genericPrivateKey     [4] PrivateKeyObject {GenericKeyAttributes},
      ... -- Cho việc mở rộng sau này
      }
```

```
PrivateKeyObject {KeyAttributes} ::= CIO {
        CommonKeyAttributes, CommonPrivateKeyAttributes, KeyAttributes}
```

A.4.2

```
PrivateRSAKeyAttributes ::= SEQUENCE {
        value                   Path,
        modulusLength   INTEGER, -- modulus length in bits, e.g. 1024
        keyInfo                 KeyInfo {NULL, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

A.4.3

```
PrivateECKeyAttributes ::= SEQUENCE {
        value                   Path,
        keyInfo  KeyInfo {Parameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

A.4.4

```
PrivateDHKeyAttributes ::= SEQUENCE {
        value                   Path,
        keyInfo  KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

A.4.5

```
PrivateDSAKeyAttributes ::= SEQUENCE {
        value                   Path,
        keyInfo  KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

A.4.6

```
PrivateKEAKeyAttributes ::= SEQUENCE {
        value                   Path,
        keyInfo  KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

A.5 Đối tượng thông tin khóa công khai

A.5.1

```
PublicKeyChoice ::= CHOICE {
        publicRSAKey            PublicKeyObject {PublicRSAKeyAttributes},
        publicECKey             [0] PublicKeyObject {PublicECKeyAttributes},
        publicDHKey             [1] PublicKeyObject {PublicDHKeyAttributes},
        publicDSAKey            [2] PublicKeyObject {PublicDSAKeyAttributes},
        publicKEAKey            [3] PublicKeyObject {PublicKEAKeyAttributes},
        genericPublicKey[4] PublicKeyObject{GenericKeyAttributes},
        ... -- Cho việc mở rộng sau này
        }
PublicKeyObject {KeyAttributes} ::= CIO {
        CommonKeyAttributes, CommonPublicKeyAttributes, KeyAttributes}
```

A.5.2

```
PublicRSAKeyAttributes ::= SEQUENCE {
  value          ObjectValue {RSAPublicKeyChoice},
  modulusLength  INTEGER, -- modulus length in bits, e.g. 1024
  keyInfo        KeyInfo {NULL, PublicKeyOperations} OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
RSAPublicKeyChoice ::= CHOICE {
  raw   RSAPublicKey,
  spki  [1] SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa RSA công khai.
  ...
  }
RSAPublicKey ::= SEQUENCE {
  modulus        INTEGER,
  publicExponent INTEGER
  }
```

A.5.3

```
PublicECKeyAttributes ::= SEQUENCE {
  value ObjectValue {ECPublicKeyChoice},
  keyInfo        KeyInfo {Parameters, PublicKeyOperations} OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
ECPublicKeyChoice ::= CHOICE {
  raw   ECPoint, -- See ANSI X9.62,
  spki  SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa elliptic curve công khai.
  ...
  }
```

A.5.4

```
PublicDHKeyAttributes ::= SEQUENCE {
  value ObjectValue {DHPublicKeyChoice},
  keyInfo        KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
DHPublicKeyChoice ::= CHOICE {
  raw   DHPublicNumber,
  spki  SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa D-H công khai.
  ...
  }
DHPublicNumber ::= INTEGER
```

A.5.5

```
PublicDSAKeyAttributes ::= SEQUENCE {
  value ObjectValue {DSAPublicKeyChoice},
  keyInfo        KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
DSAPublicKeyChoice ::= CHOICE {
  raw   DSAPublicKey,
  spki  SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa DSA công khai.
  ...
  }
DSAPublicKey ::= INTEGER
```

A.5.6

```
PublicKEAKeyAttributes ::= SEQUENCE {
  value ObjectValue {KEAPublicKeyChoice},
  keyInfo        KeyInfo {DomainParameters, PublicKeyOperations} OPTIONAL,
  ... -- Cho việc mở rộng sau này
  }
KEAPublicKeyChoice ::= CHOICE {
  raw   KEAPublicKey,
  spki  SubjectPublicKeyInfo, -- Xem ISO/IEC 9594-8:1998. Phải gồm một khóa KEA công khai.
  ...
  }
KEAPublicKey ::= INTEGER
```

A.6 Đối tượng thông tin khóa bí mật

A.6.1

```
SecretKeyChoice ::= CHOICE {
  algIndependentKey     SecretKeyObject {SecretKeyAttributes},
  genericSecretKey              [15] SecretKeyObject {GenericKeyAttributes},
  ... -- Cho việc mở rộng sau này
  } -- Chú thích: Thẻ ngữ cảnh [0] - [14] theo lịch sử và không được dùng
SecretKeyObject {KeyAttributes} ::= CIO {
  CommonKeyAttributes, CommonSecretKeyAttributes, KeyAttributes}
```

A.6.2

```
SecretKeyAttributes ::= SEQUENCE {
  value ObjectValue { OCTET STRING },
  ... -- Cho việc mở rộng sau này
  }
```

A.7 Đối tượng thông tin chứng chỉ

A.7.1

```
CertificateChoice ::= CHOICE {
        x509Certificate          CertificateObject {X509CertificateAttributes},
        x509AttributeCertificate [0] CertificateObject {X509AttributeCertificateAttributes},
        spkiCertificate          [1] CertificateObject {SPKICertificateAttributes},
        pgpCertificate                   [2] CertificateObject {PGPCertificateAttributes},
        wtlsCertificate          [3] CertificateObject {WTLSCertificateAttributes},
        x9-68Certificate         [4] CertificateObject {X9-68CertificateAttributes},
        cvCertificate                    [5] CertificateObject {CVCertificateAttributes},
        genericCertificateObject [6] CertificateObject {GenericCertificateAttributes},
        ... -- Cho việc mở rộng sau này
        }
CertificateObject {CertAttributes} ::= CIO {
        CommonCertificateAttributes, NULL, CertAttributes}
```

A.7.2

```
X509CertificateAttributes ::= SEQUENCE {
        value          ObjectValue { Certificate },
        subject  Name OPTIONAL,
        issuer         [0] Name OPTIONAL,
        serialNumber   CertificateSerialNumber OPTIONAL,
        ... -- Cho việc mở rộng sau này
```

```
}
```

A.7.3

```
X509AttributeCertificateAttributes ::= SEQUENCE {
        value           ObjectValue { AttributeCertificate },
        issuer          GeneralNames OPTIONAL,
        serialNumber    CertificateSerialNumber OPTIONAL,
        attrTypes       [0] SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
```

A.7.4

```
SPKICertificateAttributes ::= SEQUENCE {
        value    ObjectValue { CIO-OPAQUE.&Type },
        ... -- Cho việc mở rộng sau này
        }
```

A.7.5

```
PGPCertificateAttributes ::= SEQUENCE {
        value    ObjectValue { CIO-OPAQUE.&Type },
        ... -- Cho việc mở rộng sau này
        }
```

A.7.6

```
WTLSCertificateAttributes ::= SEQUENCE {
        value    ObjectValue { CIO-OPAQUE.&Type }
        ... -- Cho việc mở rộng sau này
        }
```

A.7.7

```
X9-68CertificateAttributes ::= SEQUENCE {
        value    ObjectValue { CIO-OPAQUE.&Type },
        ... -- Cho việc mở rộng sau này
        }
```

A.7.8

```
CVCertificateAttributes ::= SEQUENCE{
        value                                   ObjectValue {CIO-OPAQUE.&Type},
        certificationAuthorityReference     OCTET STRING OPTIONAL
        ... - Cho việc mở rộng sau này,
        }
```

A.8 Đối tượng thông tin bộ chứa dữ liệu

A.8.1

```
DataContainerObjectChoice ::= CHOICE {
        opaqueDO        DataContainerObject {OpaqueDOAttributes},
        iso7816DO       [0] DataContainerObject {ISO7816DOAttributes},
        oIDDO           [1] DataContainerObject {OIDDOAttributes},
        ... -- Cho việc mở rộng sau này
        }
DataContainerObject {DataObjectAttributes} ::= CIO {
        CommonDataContainerObjectAttributes, NULL, DataObjectAttributes}
```

A.8.2

```
OpaqueDOAttributes ::= ObjectValue {CIO-OPAQUE.&Type}
```

A.8.3

```
ISO7816DOAttributes ::= ObjectValue {CIO-OPAQUE.&Type}
(CONSTRAINED BY {-- Tất cả đối tượng bộ chứa dữ liệu phải được quy định theo TCVN 11167-4 --})
```

A.8.4

```
OIDDOAttributes ::= SEQUENCE {
        ID      CIO-OPAQUE.&ID ({AllowedOIDDOs}),
        value   CIO-OPAQUE.&Type ({AllowedOIDDOs}{@ID})
        }
AllowedOIDDOs CIO-OPAQUE ::= {...}
```

A.9 Đối tượng thông tin chứng thực

A.9.1

```
AuthenticationObjectChoice ::= CHOICE {
        pwd                     AuthenticationObject { PasswordAttributes },
        biometricTemplate [0] AuthenticationObject{ BiometricAttributes},
        authKey                 [1] AuthenticationObject {AuthKeyAttributes},
        external                [2] AuthenticationObject {ExternalAuthObjectAttributes},
        ... -- Cho việc mở rộng sau này
        }
AuthenticationObject {AuthObjectAttributes} ::= CIO {
        CommonAuthenticationObjectAttributes, NULL, AuthObjectAttributes}
```

A.9.2

```
PasswordAttributes ::= SEQUENCE {
        pwdFlags                PasswordFlags,
        pwdType                 PasswordType,
        minLength               INTEGER (cia-lb-minPasswordLength..cia-ub-minPasswordLength),
        storedLength            INTEGER (0..cia-ub-storedPasswordLength),
        maxLength               INTEGER OPTIONAL,
        pwdReference    [0] Reference DEFAULT 0,
        padChar                 OCTET STRING (SIZE(1)) OPTIONAL,
        lastPasswordChange GeneralizedTime OPTIONAL,
        path                    Path OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }

PasswordFlags ::= BIT STRING { case-
      sensitive (0), local (1), change-
      disabled (2),
      unblock-disabled (3), initialized (4),
      needs-padding (5), unblockingPassword
      (6), soPassword (7), disable-allowed (8),
      integrity-protected (9), confidentiality-
      protected (10), exchangeRefData (11),
      resetRetryCounter1 (12),
      resetRetryCounter2 (13)
} (CONSTRAINED BY {-- 'unblockingPassword' và 'soPassword' không thể cùng thiết lập -- })

PasswordType ::= ENUMERATED {bcd, ascii-numeric, utf8, half-nibble-bcd, iso9564-1, ...}
```

A.9.3

```
BiometricAttributes ::= CHOICE {
        biometricTemplateAttributes     BiometricTemplateAttributes,
        bit                             [APPLICATION 96] BiometricInformationTemplate,
        bitGroup                        [APPLICATION 97] BiometricInformationTemplateGroup
        }
BiometricInformationTemplate ::= OCTET STRING
        -- Phải chứa một giá trị khuôn mẫu thông tin sinh trắc học của TCVN 11167-11 (ISO/IEC 7816-11)
BiometricInformationTemplateGroup ::= OCTET STRING
        -- Phải chứa một giá trị khuôn mẫu nhóm thông tin sinh trắc học của TCVN 11167-11 (ISO/IEC 7816-11)
BiometricTemplateAttributes ::= SEQUENCE {
        bioFlags BiometricFlags,
        templateID      BiometricTemplateIDentifier,
        bioType BiometricType,
        bioReference    Reference DEFAULT 0,
        lastChange      GeneralizedTime OPTIONAL,
        path            Path OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
BiometricTemplateIDentifier ::= CHOICE {
        oID             OBJECT IDENTIFIER,
        issuerID        OCTET STRING,
        ... -- Cho việc mở rộng sau này
        }
BiometricFlags ::= BIT STRING {
        local                           (1),
        change-disabled         (2),
        unblock-disabled        (3),
        initialized                     (4),
        disable-allowed         (8),
        integrity-protected             (9),
        conflDentiality-protected (10)
        }
BiometricType ::= CHOICE {
        fingerPrint     FingerPrintInformation,
        iris            [0] IrisInformation,
        chained [1] SEQUENCE SIZE (2..cia-ub-biometricTypes) OF BiometricType,
        ... -- Cho việc mở rộng sau này
        }
FingerPrintInformation ::= SEQUENCE {
        hand    ENUMERATED {left, right},
        finger  ENUMERATED {thumb, pointerFinger, mIDdleFinger, ringFinger, littleFinger}
}
IrisInformation ::= SEQUENCE {
        eye     ENUMERATED {left, right},
        ... -- Cho việc mở rộng sau này
        }
```

A.9.4

```
ExternalAuthObjectAttributes ::= CHOICE {
        authKeyAttributes       AuthKeyAttributes,
        certBasedAttributes     [0] CertBasedAuthenticationAttributes,
        ... -- Cho việc mở rộng sau này
        }
```

```
AuthKeyAttributes ::= SEQUENCE {
        derivedKey      BOOLEAN DEFAULT TRUE,
        authKeyID       IDentifier,
        ... -- Cho việc mở rộng sau này
        }
CertBasedAuthenticationAttributes ::= SEQUENCE {
        cha     OCTET STRING,
        ... -- Cho việc mở rộng sau này
        }
```

A.10 Thông tin thẻ và mã hóa

```
CIAInfo ::= SEQUENCE {
        version             INTEGER {v1(0),v2(1)} (v1|v2,...),
        serialNumber                OCTET STRING OPTIONAL,
        manufacturerID  Label OPTIONAL,
        label                       [0] Label OPTIONAL,
        cardflags                   CardFlags,
        seInfo                      SEQUENCE OF SecurityEnvironmentInfo OPTIONAL,
        recordInfo                  [1] RecordInfo OPTIONAL,
        supportedAlgorithms         [2] SEQUENCE OF AlgorithmInfo OPTIONAL,
        issuerID            [3] Label OPTIONAL,
        holderID                    [4] Label OPTIONAL,
        lastUpdate                  [5] LastUpdate OPTIONAL,
        preferredLanguage           PrintableString OPTIONAL, -- In accordance with IETF RFC 1766
        profileIndication [6] SEQUENCE OF ProfileIndication OPTIONAL,
        ...
        } (CONSTRAINED BY { -- Mỗi giá trị AlgorithmInfo.reference phải là đơn nhất --})
CardFlags ::= BIT STRING {
        readonly                    (0),
        authRequired                (1),
        prnGeneration   (2)
        } -- Bit (3) được bảo toàn vì lí do lịch sử
SecurityEnvironmentInfo ::= SEQUENCE {
        se      INTEGER,
        owner   OBJECT IDENTIFIER OPTIONAL,
        aID     OCTET STRING
        (CONSTRAINED BY {-- Phải được mã hóa theo TCVN 11167-4 --}) OPTIONAL,
        ... -- Cho việc mở rộng sau này
        }
RecordInfo ::= SEQUENCE {
        oDRecordLength [0] INTEGER (0..cia-ub-recordLength) OPTIONAL,
        prKDRecordLength        [1] INTEGER (0..cia-ub-recordLength) OPTIONAL,
        puKDRecordLength        [2] INTEGER (0..cia-ub-recordLength) OPTIONAL,
        sKDRecordLength         [3] INTEGER (0..cia-ub-recordLength) OPTIONAL,
        cDRecordLength [4] INTEGER (0..cia-ub-recordLength) OPTIONAL,
        dCODRecordLength        [5] INTEGER (0..cia-ub-recordLength) OPTIONAL,
        aODRecordLength         [6] INTEGER (0..cia-ub-recordLength) OPTIONAL
        }
AlgorithmInfo ::= SEQUENCE {
        reference                   Reference,
        algorithm                   CIO-ALGORITHM.&ID({AlgorithmSet}),
        parameters                  CIO-ALGORITHM.&Parameters({AlgorithmSet}{@algorithm}),
        supportedOperations         CIO-ALGORITHM.&Operations({AlgorithmSet}{@algorithm}),
        objID                       CIO-ALGORITHM.&objectIDentifier ({AlgorithmSet}{@algorithm}),
        algRef                      Reference OPTIONAL
```

```
        }
CIO-ALGORITHM ::= CLASS {
        &ID INTEGER UNIQUE,
        &Parameters,
        &Operations Operations,
        &objectIDentifier OBJECT IDENTIFIER OPTIONAL
        } WITH SYNTAX {
        PARAMETERS &Parameters OPERATIONS &Operations ID &ID [OID &objectIDentifier]
        }
CIO-OPAQUE ::= TYPE-IDENTIFIER
PublicKeyOperations ::= Operations
Operations ::= BIT STRING {
        compute-checksum        (0), -- Tính toán bộ kiểm tra phần cứng
        compute-signature       (1), -- Tính toán chữ ký phần cứng
        verify-checksum  (2), -- chứng thực bộ kiểm tra phần cứng
        verify-signature  (3), -- chứng thực chữ ký phần cứng
        encipher        (4), -- Mã hóa dữ liệu phần cứng
        decipher        (5), -- Giải mã dữ liệu phần cứng
        hash                (6), -- Băm phần cứng
        generate-key        (7) -- Tạo khóa phần cứng
        }
cia-alg-null CIO-ALGORITHM ::= {
        PARAMETERS NULL OPERATIONS {{generate-key}} ID -1}
AlgorithmSet CIO-ALGORITHM ::= {
        cia-alg-null,
        ... -- Xem PKCS #11 với các giá trị có thể đối với thành phần (và các thông số) &ID
        }
LastUpdate ::= CHOICE {
        generalizedTime  GeneralizedTime,
        referencedTime   ReferencedValue,
        ... -- Cho việc mở rộng sau này
        }(CONSTRAINED BY {-- Giá trị của referencedTime phải thuộc loại GeneralizedTime --})
ProfileIndication ::= CHOICE {
        profileOID      OBJECT IDENTIFIER,
        profileName     UTF8String,
        ... -- Cho việc mở rộng sau này
        }
```

A.11 CIO DDO

```
CIODDO ::= SEQUENCE {
        provIDerID      OBJECT IDENTIFIER OPTIONAL,
        odfPath Path OPTIONAL,
        ciaInfoPath     [0] Path OPTIONAL,
        aID             [APPLICATION 15] OCTET STRING
        (CONSTRAINED BY {-- Phải là một AID theo TCVN 11167-4--}) OPTIONAL,
        ... -- Cho việc mở rộng sau này
        } – Thẻ ngữ cảnh 1 là lịch sử và không được dùng
END
```

Phụ lục B

(tham khảo)

Ví dụ về CIA đối với thẻ có chữ ký số và chức năng chứng thực

B.1 Giới thiệu

Phụ lục này mô tả một ví dụ của CIA phù hợp với các mục đích và yêu cầu định danh điện tử đối với nó. Ví dụ này bao gồm các yêu cầu với cả các thẻ và ứng dụng bên host dùng thẻ.

B.2 Các CIO

- Khóa riêng: Một thẻ CIO nên chứa ít nhất hai phím cá nhân, trong đó một khóa nên chỉ được dùng cho các mục đích chữ ký số (cờ sử dụng khóa: bất kỳ sự kết hợp nào của **sign**, **signRecover** và **nonRepudiation**). Ít nhất một trong các khóa khác nên có thể sử dụng cho chứng thực khách hàng/máy chủ và có giá trị **sign** và/hoặc **decipher** đặt trong cờ sử dụng khóa của nó. chứng thực hoặc Mã hóa các CDE phải vệ tất cả các khóa riêng. Cách sử dụng của khóa chí có chữ ký cần yêu cầu xác minh chủ thẻ với một chứng chỉ CDE chỉ sử dụng cho khóa này. Chiều dài khóa phải phù hợp với mục đích sau này.

  Loại khóa riêng cho ví dụ này là: Khóa RSA, khóa Elliptic Curve (ví dụ này không đặt ra hạn chế theo các thông số miền khác với các khóa được đề cập ở trên); và các khóa DSA.

- Khóa bí mật: Các CDE của loại này có thể hoặc không có thể có mặt trên thẻ, tùy thuộc vào quyết định của bên cung cấp ứng dụng. Không có yêu cầu cho các ứng dụng phía chủ nhằm xử lý các khóa này.

- Khóa công khai: Các CDE của loại này có thể hoặc không có thể có mặt trên thẻ, tùy thuộc vào quyết định của bên cung cấp ứng dụng. Không có yêu cầu cho các ứng dụng phía chủ nhằm xử lý các khóa này.

- Chứng chỉ: Đối với mỗi khóa riêng ít nhất một giấy chứng chỉ tương ứng phải được lưu trữ trên thẻ. Chứng chỉ là loại **X509Certificate**. Nếu một bên cung cấp ứng dụng lưu trữ các chứng chỉ CA trên thẻ mà hỗ trợ tổ chức tệp tin logic của TCVN 11167-4 (ISO/IEC 7816-4) trong đó có các cơ chế truy cập tệp tin phù hợp, sau đó nó được khuyến nghị rằng chúng được lưu trữ trong một tệp tin được bảo vệ. Tệp tin này phải được trỏ đến bởi một tệp tin CD chỉ được thay đổi bởi các tổ chức phát hành thẻ (hoặc không được thay đổi gì cả). Điều này ngụ ý việc sử dụng lựa chọn **trustedCertificates** theo loại **CIOChoice**.

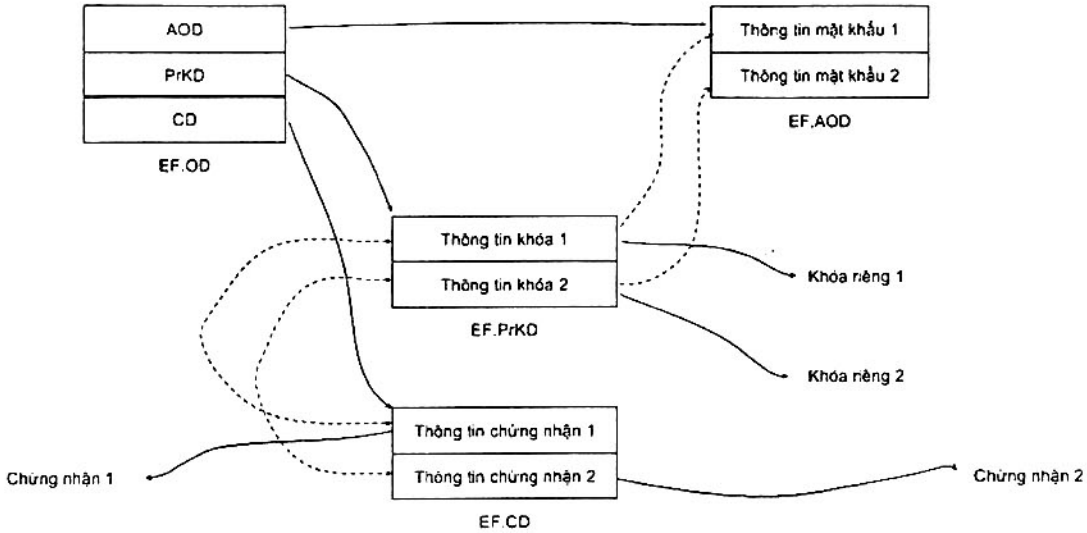
- Đối tượng bộ chứa dữ liệu: Các CDE của loại này có thể hoặc không có thể có mặt trên thẻ, tùy thuộc vào quyết định của bên cung cấp ứng dụng. Không có yêu cầu cho các ứng dụng phía chủ nhằm xử lý các đối tượng này.

- Đối tượng chứng thực: Ít nhất một chứng thực CDE phải có mặt trên thẻ, việc kiểm soát truy cập vào các CDE được bảo vệ. Một chứng thực CDE riêng biệt nên được sử dụng với các khóa chí có chữ ký, nếu một khóa như vậy tồn tại. Bất kỳ việc sử dụng khóa riêng chỉ có chữ

ký nên yêu cầu chứng thực một người dùng mới. Trong trường hợp của các mật khẩu, bất cứ xác minh tích cực của một mật khẩu phải không cho phép việc sử dụng các dịch vụ bảo mật liên quan đến mật khẩu khác.

Mật khẩu phải có ít nhất 4 ký tự dài (BCD, UTF-8 hoặc ASCII).

Khi một mật khẩu sẽ bị khóa sau các xác minh sai mật khẩu liên tiếp, mật khẩu chỉ có thể mở khóa thông qua một mã đặt lại hoặc một thủ tục mở khóa, được xác định bởi các tổ chức phát hành thẻ.



Hình B.1 - Mối quan hệ tệp tin trong DF.CIA. Các mũi tên chấm đứt chỉ ra các tham chiếu chéo.

B.3 Kiểm soát truy cập

Các khóa riêng phải là các đối tượng cá nhân và cần được đánh dấu là **nhạy cảm**. Các tệp tin chứa các khóa riêng cần được bảo vệ chống lại việc gỡ bỏ và/hoặc ghi đè. Các điều kiện truy cập sau phải được thiết lập với DF.CIA và các tệp tin cơ bản trong nó.

Bảng B.1 - Các điều kiện truy cập tệp tin được khuyến nghị

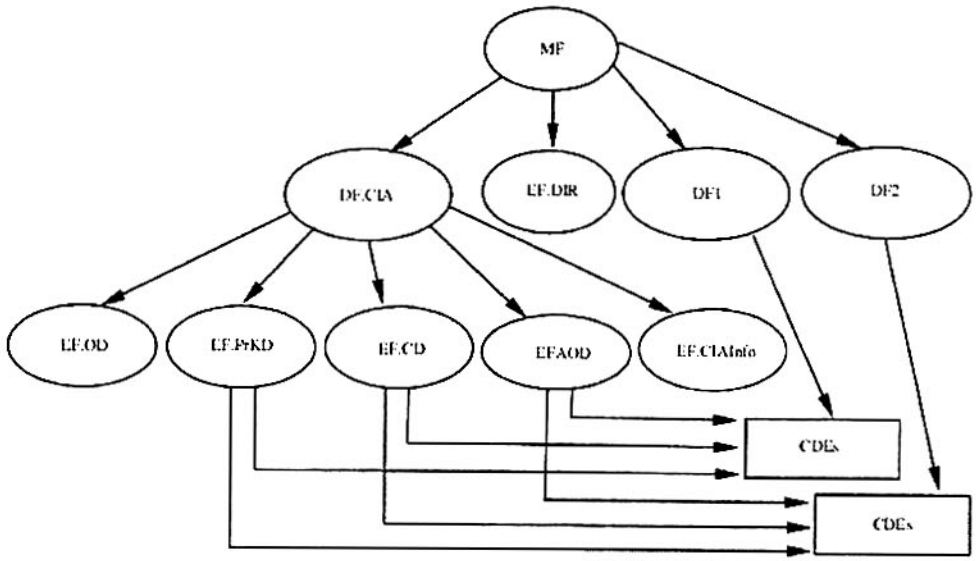
Tệp tin	Các điều kiện truy cập
DF.CIA	Tạo: chứng thực người dùng hoặc chứng thực bên ngoài Xóa: chứng thực bên ngoài
EF.CIAInfo	Đọc: Thường xuyên Cập nhật: chứng thực người dùng hoặc chứng thực bên ngoài hay Không cập nhật Ghi đè: Không được phép
EF.OD	Đọc: Thường xuyên Cập nhật: chứng thực bên ngoài Ghi đè: chứng thực bên ngoài
EF.AOD	Đọc: Thường xuyên Cập nhật: Không được phép Ghi đè: chứng thực người dùng hoặc chứng thực bên ngoài
EF.PrKD, EF.PuKD, EF.SKD, EF.CD và EF.DCOD	Đọc: Thường xuyên hoặc chứng thực người dùng Cập nhật: chứng thực bên ngoài hoặc Không được phép Ghi đè: Chứng chỉ ngoại bộ hoặc Không được phép
EF.CD bao gồm các tham chiếu với các chứng chỉ đáng tin cậy	Đọc: Thường xuyên Cập nhật: Chứng chỉ ngoại bộ hoặc Không được phép Ghi đè: Chứng chỉ ngoại bộ hoặc Không được phép
Các EF khác trong DF.CIA	Đọc: Thường xuyên hoặc chứng thực người dùng Cập nhật: chứng thực người dùng hoặc chứng thực bên ngoài hoặc Không được phép Ghi đè: chứng thực người dùng hoặc chứng thực bên ngoài hoặc Không được phép
CHÚ THÍCH 1 Chứng thực bên ngoài được mô tả trong TCVN 11167-4 (ISO/IEC 7816-4). CHÚ THÍCH 2 Chứng thực bên ngoài cần bao gồm thông điệp an ninh được mô tả trong TCVN 11167-4 (ISO/IEC 7816-4).	

CHÚ THÍCH Nếu một bên cung cấp ứng dụng muốn bảo vệ một tệp tin thư mục CIO với một đối tượng chứng thực thì đối tượng chứng thực đầu tiên trong EF.AOD phải được dùng mặc định. EF.OD và EF.AOD chắc chắn không được bảo vệ theo cách thức này.

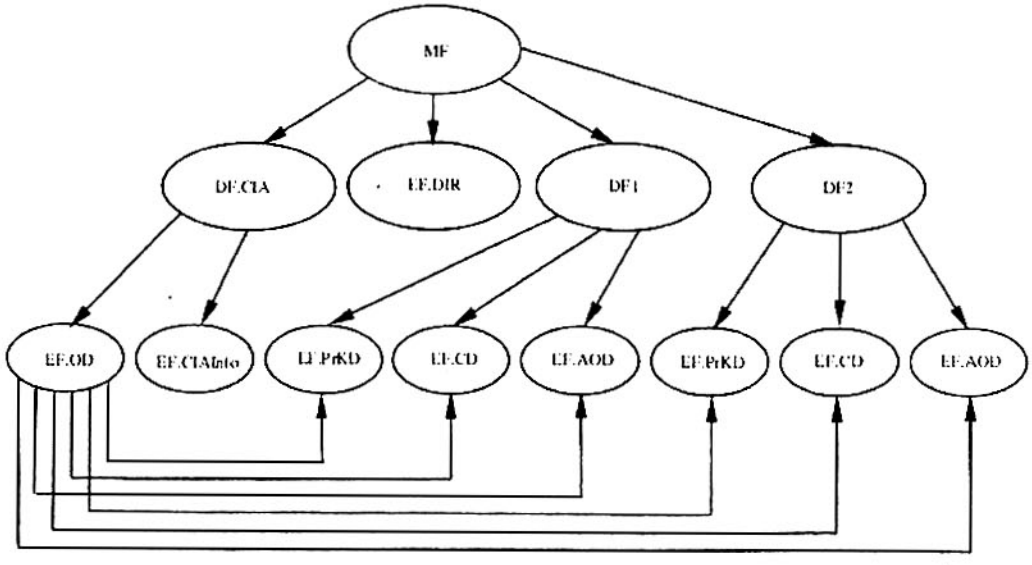
Phụ lục C

(tham khảo)

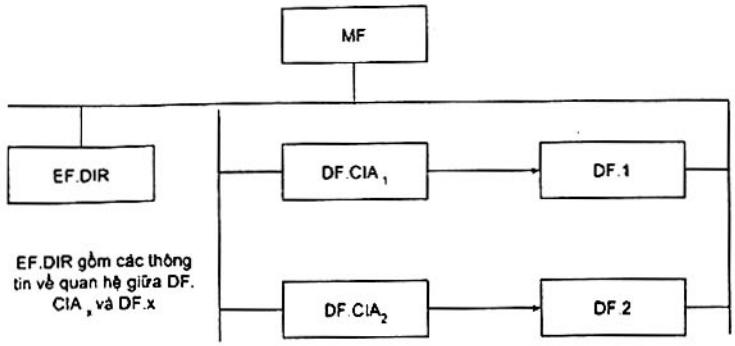
Ví dụ về mô hình tô pô



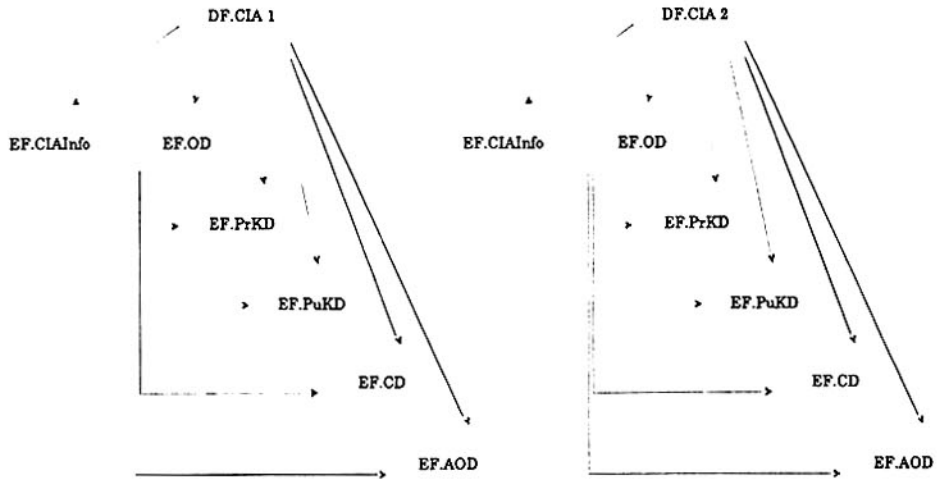
Hình C.1 - Ví dụ với ba ứng dụng. Các phần tử dữ liệu mã hóa được lưu trữ bên ngoài CIA



Hình C.2 - Ví dụ với ba ứng dụng. Chỉ có EF.OD và EF.CIAInfo trong DF.CIA



Hình C.3 - Ví dụ sử dụng EF.DIR



Hình C.4 – Ví dụ về hai ứng dụng trên thẻ không thấy EF.DIR trên giao diện.

Phụ lục D

(tham khảo)

Ví dụ về các giá trị và việc mã hóa CIO

D.1 Giới thiệu

Mỗi điều con của Phụ lục này, ví dụ D.x đề cập đến một trong các EFS được xác định trong cấu trúc tệp tin của thẻ. Mỗi điều phụ lần lượt bao gồm ba phần:

- Phần đầu tiên, ví dụ D.x.1 đưa ra các tên và giá trị mẫu của các DE liên quan tới việc xây dựng một DER theo ký hiệu giá trị ASN.1 được quy định trong ISO/IEC 8824-1: 1998.
- Phần thứ hai, ví dụ D.x.2 sát nhập theo cú pháp ASN.1 với các việc chỉ định có cấu trúc/nguyên thủy, các giá trị mẫu và chiều dài của nó.
- Phần thứ ba, ví dụ D.x3 đưa ra mã hóa DER hệ cơ số 16 thực tế DER như đọc từ tệp tin.

Ba phần hiển thị các giá trị theo các định dạng khác nhau, nhằm cho thấy những cách khác nhau để chúng có thể xuất hiện trong các thông số kỹ thuật dựa vào tiêu chuẩn này.

Trong phần đầu tiên, các dấu ngoặc kép đơn biểu thị một chuỗi thập lục phân, kí hiệu 7816 thông thường. Chúng được theo sau bởi 'H', mà còn là một ký hiệu thông thường, ví dụ: '0202'H. Dấu ngoặc kép biểu thị chuỗi UTF-8 (có thể in ra) ví dụ: "Ứng dụng CIA". Các dấu móc biểu thị một định danh đối tượng, ví dụ: {1 2 840 113549 1 15 4 1}, xem ví dụ của TCVN 11167-4 (ISO/IEC 7816-4) Phụ lục B để chuyển mã thành một chuỗi thập lục phân.

Trong phần thứ hai, tiền tố 0x biểu thị một chuỗi thập lục phân, một ký hiệu lập trình thông thường, ví dụ: 0x3f005015 tương đương với '3F005015' trong kí hiệu của TCVN 11167 thông thường. Một số mà không có tiền tố này có nghĩa là nó là một chữ số thập phân, ví dụ: một giá trị 12 tương đương với mã thập lục phân: 0x0c hoặc '0C'. Số thẻ được chỉ định giữa dấu ngoặc []. Số lượng thẻ được đưa ra trong hệ thập phân. Lớp thẻ được chỉ định trong dấu móc dấu ngoặc vuông, ngoại trừ với các lớp thẻ tag thuộcngữ cảnh cụ thể là được mặc định. Các thông tin nguyên thủy/có cấu trúc đưa ra giá trị của b6 trong thẻ thực tế được dùng trong phần thứ ba. Mỗi mức định danh chỉ ra một mức đóng gói.

Trong phần thứ ba, các byte thập lục được phân cách bằng dấu cách và không có dấu ngoặc kép. Mỗi mức định danh bắt đầu với một tiêu đề DO (độ dài thẻ), ngoại trừ mức cuối cùng, đó là giá trị của một DO nguyên thủy: các quy tắc thụt đầu dòng là tương tự như trong phần thứ hai. Như vậy, tất cả tiêu đề của các DO thuộc cùng một mẫu xuất hiện ở cùng một mức thụt đầu dòng.

Nhằm nâng cao kiến thức, đánh số dòng được thêm vào điều D.2.

D.2 EF.DO

D.2.1 Kí hiệu giá trị ASN.1

```
privateKeys :
        path : {
                efIDOrPath '4401'H
                },
certificates :
        path : {
                efIDOrPath '4402'H
                },
dataContainerObjects :
        path : {
                efIDOrPath '4403'H
                },
authObjects :
        path : {
                efIDOrPath '4404'H
                }
```

D.2.2 Mô tả, thẻ, độ dài và giá trị của ASN.1

```
CIOChoice CHOICE
 privateKeys : tag = [0] constructed; length = 6
  PrivateKeys CHOICE
        path Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 4
         efIDOrPath OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 2
          0x4401
CIOChoice CHOICE
 certificates : tag = [4] constructed; length = 6
  Certificates CHOICE
   path Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 4
         efIDOrPath OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 2
          0x4402
CIOChoice CHOICE
 dataContainerObjects : tag = [7] constructed; length = 6
  DataContainerObjects CHOICE
   path Path SEQUENCE:tag = [UNIVERSAL 16] constructed; length = 4
         efIDOrPath OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 2
          0x4403
CIOChoice CHOICE
 authObjects : tag = [8] constructed; length = 6
  AuthObjects CHOICE
   path Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 4
         efIDOrPat OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 2
          0x4404
```

D.2.3 Mã hóa DER hệ cơ số 16

```
A0 06
        30 04
                04 02
                        44 01
A4 06
        30 04
                04 02
```

```
                                    44 02
7       A7 06
8               30 04
9                       04 02
                                    44 03
10      A8 06
11              30 04
12                      04 02
                                    44 04
```

D.3 EF.CIAInfo

D.3.1 Kí hiệu giá trị ASN.1

```
ciaInfoExample CIAInfo ::= {
        version                 v2,
        serialNumber            '159752222515401240'H,
        manufacturerID          "Acme, Inc.",
        cardflags {
                prnGeneration
                }
        }
```

D.3.2 Mô tả, thẻ, độ dài và giá trị của ASN.1

```
CIAInfo SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 30
 version INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
  1
 serialNumber OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 9
  0x159752222515401240
 manufacturerID Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 10
  0x41636d652c20496e632e
 cardflags  CardFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length = 2
  0x0520
```

D.3.3 Mã hóa DER hệ cơ số 16

```
30 1E
        02 01
                01
        04 09
                15 97 52 22 25 15 40 12 40
        0C 0A
                41 63 6D 65 2C 20 49 6E 63 2E
        03 02
                05 20
```

D.4 EF.PrKD

Trong ví dụ này, hai khóa riêng được đề cập tới. Các tệp tin liên quan tới khác khác, ví dụ: EF.PuKD và EF.SKD, có cấu trúc giống nhau. Các khóa công khai liên quan cũng được tham chiếu trong EF.PuKD với cùng nhãn.

D.4.1 Kí hiệu giá trị ASN.1

```
privateRSAKey : {
        commonObjectAttributes {
                label "KEY1",
                flags { private },
                authID '01'H
                        },
        classAttributes {
                ID '45'H,
                usage { decipher, sign, keyDecipher }
                        },
        subClassAttributes {
                keyIDentifiers {
                        {
                                IDType 4,
                                IDValue ParameterString : '4321567890ABCDEF'H
                                }
                        }
                },
        typeAttributes {
                value {
                        efIDOrPath '4B01'H
                        },
                modulusLength 1024
                }
        },
privateRSAKey : {
        commonObjectAttributes {
                label "KEY2",
                flags { private },
                authID '02'H
                },
        classAttributes {
                ID '46'H,
                usage { sign, nonRepudiation }
                },
        subClassAttributes {
                keyIDentifiers {
                        {
                                IDType 4,
                                IDValue ParameterString : '1234567890ABCDEF'H
                                }
                        }
                },
```

```
typeAttributes {
        value {
                efIDOrPath '4B02'H
                },
        modulusLength 1024
        }
}
```

D.4.2 Mô tả, thẻ, độ dài và giá trị của ASN.1

```
PrivateKeyChoice CHOICE
 privateRSAKey SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 59
  commonObjectAttributes CommonObjectAttributes SEQUENCE:
   tag = [UNIVERSAL 16] constructed; length = 13
    label Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 4
          0x4b455931
    flags CommonObjectFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length = 2
      0x0780
     authID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
      0x01
        classAttributes CommonKeyAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length= 7
     ID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
      0x45
     usage KeyUsageFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length = 2
      0x0264
    subClassAttributes : tag = [0] constructed; length = 19
     CommonPrivateKeyAttributes SEQUENCE: tag=[UNIVERSAL 16] constructed; length=17
      keyIDentifiers SEQUENCE OF: tag = [0] constructed; length = 15
       SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 13
        IDType INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
         4
        IDValue OpenType
                  0x4321567890abcdef
      typeAttributes : tag = [1] constructed; length = 12
            PrivateRSAKeyAttributes SEQUENCE: tag=[UNIVERSAL 16] constructed; length=10
              value Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 4
           efIDOrPath OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 2
                      0x4b01
           modulusLength INTEGER: tag = [UNIVERSAL 2] primitive; length = 2
                      1024
       PrivateKeyChoice CHOICE
        privateRSAKey SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 60
         commonObjectAttributes CommonObjectAttributes SEQUENCE:
    tag = [UNIVERSAL 16] constructed; length = 13
              label Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 4
               0x4b455932
              flags CommonObjectFlags BIT STRING: tag = [UNIVERSAL 3] primitive;
               length=2
                0x0780
               authID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
            0x02
              classAttributes CommonKeyAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 8
               ID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
                0x46
               usage KeyUsageFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length = 3
```

```
                0x062040
        subClassAttributes : tag = [0] constructed; length = 19
            CommonPrivateKeyAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length= 17
              keyIDentifiers SEQUENCE OF: tag = [0] constructed; length = 15
               SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 13
                    IDType INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
                     4
                    IDValue OpenType
                     0x1234567890abcdef
              typeAttributes : tag = [1] constructed; length = 12
                    PrivateRSAKeyAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 10
                     value Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 4
                      efIDOrPath OCTET STRING: tag=[UNIVERSAL 4] primitive; length = 2
                       0x4b02
                     modulusLength INTEGER: tag = [UNIVERSAL 2] primitive; length = 2
                      1024
```

D.4.3 Mã hóa DER hệ cơ số 16

```
30 3B
        30 0D
                0C 04
                        4B 45 59 31
                03 02
                        07 80
                04 01
                        01
        30 07
                04 01
                        45
                03 02
                        02 64
A0 13
        30 11
                A0 0F
                        30 0D
                                02 01
                                        04
                                04 08
                                        43 21 56 78 90 AB CD EF
A1 0C
        30 0A
                30 04
                        04 02
                                4B 01
                        02 02
                                04 00
30 3C
        30 0D
                0C 04
                        4B 45 59 32
                03 02
                        07 80
                04 01
                        02
        30 08
```

```
                04 01
                        46
                03 03
                        06 20 40
        A0 13
                30 11
                        A0 0F
                                30 0D
                                        02 01
                                                04
                                        04 08
                                                12 34 56 78 90 AB CD EF
A1 0C
        30 0A
                30 04
                        04 02
                                4B 02
                                02 02
                                        04 00
```

D.5 EF.CD

D.5.1 Kí hiệu giá trị ASN.1

```
x509Certificate : {
        commonObjectAttributes {
                label "CERT1",
                flags { }
                },
        classAttributes {
                ID '45'H
                },
        typeAttributes {
                value indirect :
                        path : {
                                efIDOrPath '4331'H
                                }
                        }
                },
        x509Certificate : {
                commonObjectAttributes {
                label "CERT2",
                flags { }
                },
        classAttributes {
                        ID '46'H
                },
        typeAttributes {
                value indirect :
                        path : {
                                efIDOrPath '4332'H
                                }
                        }
```

}

D.5.2 Mô tả, thẻ, độ dài và giá trị ASN.1

```
CertificateChoice CHOICE
 x509Certificate SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 27
  commonObjectAttributes CommonObjectAttributes SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 10
        label Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 5
         0x4345525431
        flags CommonObjectFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length = 1
         0x00
        classAttributes CommonCertificateAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 3
   ID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
    0x45
   typeAttributes : tag = [1] constructed; length = 8
        X509CertificateAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 6
         value CHOICE
          indirect ReferencedValue CHOICE
               path Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 4
                efIDOrPath OCTET STRING: tag = [UNIVERSAL 4] primitive;
                 length = 2
                  0x4331
CertificateChoice CHOICE
 x509Certificate SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 27
  commonObjectAttributes CommonObjectAttributes SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 10
   label Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 5
    0x4345525432
       flags CommonObjectFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length=1
        0x00
  classAttributes CommonCertificateAttributes SEQUENCE: tag = [UNIVERSAL 16],                constructed; length = 3
   ID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
    0x46
  typeAttributes : tag = [1] constructed; length = 8
   X509CertificateAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 6
              value CHOICE
                   indirect ReferencedValue CHOICE
                        path Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length= 4
                             efIDOrPath OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 2
```

D.5.3 Mã hóa DER hệ cơ số 16

```
30 1B
      30 0A
            0C 05
                  43 45 52 54 31
            03 01
                  00
      30 03
            04 01
                  45
      A1 08
            30 06
                  30 04
                        04 02
                              43 31
```

```
30 1B
        30 0A
                0C 05
                        43 45 52 54 32
                03 01
                        00
        30 03
                04 01
                        46
        A1 08
                30 06
                        30 04
                                04 02 43 32
```

D.6 EF.AOD

D.6.1 Kí hiệu giá trị ASN.1

```
pwd : {
        commonObjectAttributes {
                label "PIN1",
                flags { private }
                },
        classAttributes {
                authID '01'H
                },
        typeAttributes {
                pwdFlags { change-disabled, initialized, needs-padding },
                pwdType bcd,
                        minLength 4,
                        storedLength 8,
                        padChar 'FF'H
                        }
                },
pwd : {
        commonObjectAttributes {
                label "PIN2",
                flags { private }
                },
        classAttributes {
                authID '02'H
                },
        typeAttributes {
                pwdFlags { change-disabled, initialized, needs-padding },
                pwdType bcd,
                minLength 4,
                storedLength 8,
                padChar 'FF'H,
                path {
                        efIDOrPath '3F0050150100'H
                        }
                }
        }
```

D.6.2 Mô tả, thẻ, độ dài và giá trị ASN.1

```
AuthenticationObjectChoice CHOICE
 pwd SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 37
  commonObjectAttributes CommonObjectAttributes SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 10
   label Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 4
    0x50494e31
   flags CommonObjectFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length=2
    0x0780
  classAttributes CommonAuthenticationObjectAttributes SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 3
   authID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
    0x01
  typeAttributes : tag = [1] constructed; length = 18
   PasswordAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 16
    pwdFlags PasswordFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length = 2
     0x022c
    pwdType PasswordType ENUMERATED: tag = [UNIVERSAL 10] primitive; length = 1
     0
    minLength INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
     4
    storedLength INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
     8
    padChar OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
     0xff
AuthenticationObjectChoice CHOICE
 pwd SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 47
  commonObjectAttributes CommonObjectAttributes SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 10
   label Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 4
    0x50494e32
   flags CommonObjectFlags BIT STRING: tag = [UNIVERSAL 3] primitive;      length = 2
    0x0780
  classAttributes CommonAuthenticationObjectAttributes SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 3
   authID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
    0x02
  typeAttributes : tag = [1] constructed; length = 28
   PasswordAttributes SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 26
    pwdFlags PasswordFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length = 2
     0x022c
    pwdType PasswordType ENUMERATED: tag = [UNIVERSAL 10] primitive; length = 1
     0
    minLength INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
     4
    storedLength INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
     8
    padChar OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
     0xff
    path Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 8
     efIDOrPath OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 6
      0x3f0050150100
```

D.6.3 Mã hóa DER hệ cơ số 16

```
30 25
        30 0A
                0C 04
                        50 49 4E 31
                03 02
                        07 80
        30 03
                04 01
                        01
        A1 12
                30 10
                        03 02
                                02 2C
                        0A 01
                                00
                        02 01
                                04
                        02 01
                                08
                        04 01
                                FF
30 2F
        30 0A
                0C 04
                        50 49 4E 32
                03 02
                        07 80
        30 03
                04 01
                        02
        A1 1C
                30 1A
                        03 02
                                02 2C
                        0A 01
                                00
                        02 01
                                04
                        02 01
                                08
                        04 01
                                FF
                                        30 08
                                                04 06
                                                        3F 00 50 15 01 00
```

D.7 EF.DCOD

D.7.1 Kí hiệu giá trị ASN.1

```
opaqueDO : {
        commonObjectAttributes {
                label "OBJECT1",
                flags { private, modifiable },
                authID '02'H
                },
        classAttributes {
                applicationName "APP"
                },
        typeAttributes indirect :
                path : {
                        efIDOrPath '4431'H,
                        index 64,
                        length 48
                        }
                }
```

D.7.2 Mô tả, thẻ, độ dài và giá trị ASN.1

```
DataContainerObjectChoice CHOICE
 opaqueDO SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 39
  commonObjectAttributes CommonObjectAttributes SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 16
   label Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 7
    0x4f424a45435431
   flags CommonObjectFlags BIT STRING: tag = [UNIVERSAL 3] primitive; length = 2
    0x06c0
   authID IDentifier OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 1
    0x02
  classAttributes CommonDataContainerObjectAttributes SEQUENCE:
  tag = [UNIVERSAL 16] constructed; length = 5
   applicationName Label UTF8String: tag = [UNIVERSAL 12] primitive; length = 3
    0x415050
  typeAttributes : tag = [1] constructed; length = 12
   OpaqueDOAttributes CHOICE
    indirect ReferencedValue CHOICE
     path Path SEQUENCE: tag = [UNIVERSAL 16] constructed; length = 10
      efIDOrPath OCTET STRING: tag = [UNIVERSAL 4] primitive; length = 2
       0x4431
      index INTEGER: tag = [UNIVERSAL 2] primitive; length = 1
       64
      length INTEGER: tag = [0] primitive; length = 1
       48
```

D.7.3 Mã hóa DER hệ cơ số 16 của DCOD

```
30 27
        30 10
        0C 07
                4F 42 4A 45 43 54 31
        03 02
                06 C0
```

```
        04 01
                02
30 05
        0C 03
                41 50 50
A1 0C
        30 0A
                04 02
                        44 31
                02 01
                        40
                80 01
                        30
```

D.8 Mẫu ứng dụng (trong EF.DIR)

Trong ví dụ này, chỉ một mẫu ứng dụng IDO (ví dụ: một **ApplicationTemplate**) được trình bày.

D.8.1 Kí hiệu giá trị ASN.1

```
applicationTemplateExample ApplicationTemplate ::= {
        aID     'A000000063504B43532D3135'H,
        label   "RSA DSI",
        path    '3F005015'H,
        ddo {
                provIDerID { 1 2 840 113549 1 15 4 1 },
                aID 'FAB123456789'H
                }
        }
```

D.8.2 Mô tả, thẻ, độ dài và giá trị ASN.1 trong ApplicationTemplate

```
ApplicationTemplate SET: tag = [APPLICATION 1] constructed; length = 53
 aID OCTET STRING: tag = [APPLICATION 15] primitive; length = 12
  0xa000000063504b43532d3135
 label UTF8String: tag = [APPLICATION 16] primitive; length = 7
  0x52534120445349
 path OCTET STRING: tag = [APPLICATION 17] primitive; length = 4
  0x3f005015
 ddo : tag = [APPLICATION 19] constructed; length = 12
  DDOTemplate OpenType
   provIDerID OBJECT IDENTIFIER: tag = [UNIVERSAL 6] primitive; length = 10
   { 1 2 840 113549 1 15 4 1 }
   aID OCTET STRING: tag = [APPLICATION 15] primitive; length = 6
```

D.8.3 Mã hóa DER hệ cơ số 16 của ApplicationTemplate

```
61 33
        4F 0C
                A0 00 00 00 63 50 4B 43 53 2D 31 35
        50 07
                52 53 41 20 44 53 49
```

```
51 04
        3F 00 50 15
73 14
        06 0A
                2A 86 48 86 F7 0D 01 0F 04 01
        4F 06
                FA B1 23 45 67 89
```

Phụ lục E

(tham khảo)

Ví dụ về việc sử dụng ứng dụng thông tin mã hóa

E.1 Giới thiệu

Mục đích của phụ lục tham khảo này nhằm cung cấp các ví dụ thực hành của việc sử dụng ứng dụng thông tin mã hóa. Bằng cách cung cấp mã chương trình mẫu với mỗi ví dụ, các lập trình viên có thể thấy được kết nối được lập trình giữa các biểu diễn ASN.1 mức cao và BER mức thấp, do đó tạo ta phần mềm tốt và hiệu quả hơn, sử dụng ứng dụng thông tin mã hóa.

Mỗi điều trong phụ lục này là một ví dụ dạng đứng tự do và bao gồm bốn lược đồ:

a) Mô tả ví dụ.

b) Một quy định kĩ thuật của ví dụ được mô tả dưới dạng lược đồ (1) theo cấu trúc ASN.1 của tiêu chuẩn này được bình luận, sử dụng kí hiệu giá trị hình thức được quy định trong ISO/IEC 8824-1.

c) Mã hóa trong ISO/IEC 9899 TC2 ngôn ngữ lập trình C đối với mã hóa và giải mã BER phụ thuộc vào quy định kĩ thuật ASN.1 của đoạn (2).

d) Mã hóa BER của ví dụ được cung cấp bởi bộ mã hóa đoạn (3). Hai ví dụ cũng bao gồm các hiển thị đồ họa của BER ở phần cuối của Phụ lục.

e) Mã nguồn được cung cấp trong đoạn (3) và được biện dịch và chạy nhằm tạo ra đầu ra được hiển thị trong đoạn (4).

Một mã của mã hóa ASN.1 của Ứng dụng thông tin mã hóa được liệt kê trong Phụ lục A bên trên được dùng với tất cả các ví dụ. Một bộ biên dịch ASN.1 sẵn có, miễn phí được dùng để tạo ra các bộ mã hóa và giải mã BER từ ASN.1.

E.2 Mã hóa của một khóa riêng

E.2.1 Mô tả ví dụ ứng dụng thông tin mã hóa

Đây là một ví dụ của một khóa riêng RSA của tiêu chuẩn này.

E.2.2 Mã hóa ASN.1 của một khóa riêng RSA

```
privateKeys objects { -- SEQUENCE OF --
        privateRSAKey { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                        label     '4b455931'H -- "KEY1" --,
                        flags     '80'H,
                        authID    '41444d'H -- "ADM" --,
                        userConsent       1
                },
                classAttributes { -- SEQUENCE --
                        ID        '9b'H,
                        usage     '2040'H,
```

```
                native    TRUE,
                accessFlags '98'H,
                keyReference      10
        },
        subClassAttributes { -- SEQUENCE --
                keyIDentifiers { -- SEQUENCE OF --
                        { -- SEQUENCE --
                                IDType 5,
                                        IDValue '3132333435363738'H -- "12345678" --
                                }
                        }
                },
        typeAttributes { -- SEQUENCE --
                value indirect path { -- SEQUENCE --
                        efIDOrPath '3f004041'H
                },
                modulusLength 1024
        }
    }
}
```

E.2.3 Mã hóa và giải mã BER từ ASN.1

```
/*
** Mã hóa một khóa riêng như một Đối tượng dữ liệu trong in EF.OD
*/
voID Part15PrivateKey(const char    *label,
                      unsigned char     objectFlags,
                      unsigned char     *authID,
                      unsigned int      authIDLength,
                      unsigned int      userConsent,
                      unsigned char     native,
                      unsigned char     *ID, unsigned int IDLength,
                      unsigned short    usageFlags,
                      unsigned char     accessFlags,
                      unsigned int      keyReference,
                      unsigned int      IDentifierType,
                      unsigned char     *externalIDentifier,
                      unsigned char     *path, unsigned int pathLength,
                      unsigned int      modulusLength
                      )
{
    unsigned int l;
    CIOChoice *cio;
    PrivateKeyChoice *prk, **prkp;
    CredentialIDentifier *crID, **crIDp;
    PrivateKeyObject_PrivateRSAKeyAttributes pattr      =     { 0 };
    CommonObjectAttributes commonObjAttr                =     { 0 };
    CommonKeyAttributes commonKeyAttr                         =     { 0 };
    CommonPrivateKeyAttributes commonPrivateKeyAttr     =     { 0 };
    PrivateRSAKeyAttributes privateRSAKeyAttr           =     { 0 };
    Path pathOctets                                           =     { 0 };
    AsnOcts issuerHash                                              =     { 0 };

    char commonObjectFlags[1] = { 0 };
    AsnBits commonFlagsAsnBits = { 3, commonObjectFlags };
```

```
char keyUsage[2] = { 0 };
AsnBits keyUsageAsnBits = { 10, keyUsage };
char keyAccessFlags[1] = { 0 };
AsnBits keyAccessFlagsAsnBits = { 5, keyAccessFlags };
/*
** Điều 8.3 Loại CIOChoice
**
** "EF.OD phải bao gồm ... của 0, 1 hoặc nhiều hơn các giá trị DER mã hóa của
**  CIOChoice."
*/
cio = (CIOChoice *)calloc(1, sizeof(PrivateKeyChoice));
cio->choiceID = CIOCHOICE_PRIVATEKEYS;
/*
** "Mong đợi rằng một mục vào EF.OD thường tham chiếu một tệp tin riêng lẻ
** (đường dẫn của việc chọn lựa PathOrObjects) bao gồm các CIO của loại được chỉ định.
** Một mục vào, tuy nhiên ... giữa trực tiếp các CIO
** (việc chọn lựa đối tượng của PathOrObjects),
**  nếu các đối tượng và tệp tin EF.OD có cùng các yêu cầu kiểm soát truy cập."
** PathOrObjects{PrivateKeyChoice}
*/
cio->a.privateKeys = (PrivateKeys *)calloc(1, sizeof(PrivateKeys));
cio->a.privateKeys->choiceID = PATHOROBJECTS_PRIVATEKEYCHOICE_OBJECTS;
cio->a.privateKeys->a.objects = AsnListNew(sizeof (voID*));
/*
** Điều 8.4.1 PrivateKeyChoice
**
** "Loại này bao gồm thông tin ... với một khóa riêng. Mỗi giá trị bao gồm
** các thuộc tính phổ biến với bất kỳ đối tượng, khóa hay khóa riêng nào,
** và các thuộc tính cụ thể với khóa."
*/
prkp = (PrivateKeyChoice **)AsnListAppend(cio->a.privateKeys->a.objects);
*prkp = prk = calloc(1, sizeof(PrivateKeyChoice));
prk->choiceID = PRIVATEKEYCHOICE_PRIVATERSAKEY;
prk->a.privateRSAKey = &pattr;
pattr.commonObjectAttributes = &commonObjAttr;
pattr.classAttributes = &commonKeyAttr;
pattr.subClassAttributes = &commonPrivateKeyAttr;
pattr.typeAttributes = &privateRSAKeyAttr;
/*
** Điều 8.2.8 CommonObjectAttributes
**
** "Loại này là một bộ chứa các thuộc tính phổ biến với tất cả các CIO."
*/
commonObjAttr.label.octs = _strdup(label);
commonObjAttr.label.octetLen = strlen(label);
commonObjectFlags[0] = objectFlags;
commonObjAttr.flags = commonFlagsAsnBits;
commonObjAttr.authID.octetLen=authIDLength;
commonObjAttr.authID.octs = authID;
commonObjAttr.userConsent = &userConsent;
/*
** Điều 8.2.9 CommonKeyAttributes
**
** "Trường ID phải là đơn nhất đối với mỗi đối tượng thông tin khóa,
**  ngoại trừ đối tượng thông tin khóa công khai và đối tượng khóa riêng tương ứng
```

```
** được lưu trữ trên cùng thẻ. Trong trường hợp này, các đối tượng thông tin phải
** được chia sẻ với cùng định danh mà có thể được chia sẻ với một hay nhiều
** đối tượng thông tin chứng chỉ..."
*/
commonKeyAttr.ID.octs = ID;
commonKeyAttr.ID.octetLen = IDLength;
keyUsage[0] = (unsigned char)(usageFlags>>8);
keyUsage[1] = (unsigned char)(usageFlags);
commonKeyAttr.usage = keyUsageAsnBits;
keyAccessFlags[0] = accessFlags;
commonKeyAttr.accessFlags= keyAccessFlagsAsnBits;
commonKeyAttr.native = &native;
commonKeyAttr.keyReference = &keyReference;
/*
** Điều 8.2.10 CommonPrivateKeyAttributes
**
** "Trường tên gọi được trình bày, gọi tên người chủ của khóa, được quy định trong một
** trường nội dung của chứng chỉ tương ứng.
**
** Các giá trị của trường keyIDentifiers có thể được đối chiếu với các định danh
** từ thông điệp hay các giao thức bên ngoài nhằm lựa chọn để chọn ra khóa
* liên quan tới một thao tác đã biết."
*/
commonPrivateKeyAttr.keyIDentifiers =
(CommonPrivateKeyAttributesSeqOf *)AsnListNew(sizeof (voID*));
crIDp = (CredentialIDentifier **)AsnListAppend(commonPrivateKeyAttr.keyIDentifiers);
*crIDp = crID = (CredentialIDentifier *)calloc(1, sizeof(CredentialIDentifier));
issuerHash.octs = _strdup(externalIDentifier);
issuerHash.octetLen = strlen(externalIDentifier);
crID->IDType = IDentifierType;
crID->IDValue.value = &issuerHash;
SetAnyTypeByInt(&(crID->IDValue), IDentifierType);
/*
** Điều 8.4.2 Thuộc tính khóa riêng RSA
**
** "PrivateRSAKeyAttributes.value: Giá trị phải là một đường dẫn tới một tệp tin
** bao gồm một khóa riêng RSA. Nếu giá trị này không cần thiết để quy định một
** đường dẫn tới một tệp tin, giá trị đường dẫn có thể được đặt là
** đường dẫn trắng."
*/
privateRSAKeyAttr.value = (ObjectValue *)calloc(1, sizeof(ObjectValue));
privateRSAKeyAttr.value->choiceID = OBJECTVALUE_INDIRECT;
privateRSAKeyAttr.value->a.indirect =
(ReferencedValue *)calloc(1, sizeof(ReferencedValue));
privateRSAKeyAttr.value->a.indirect->choiceID = REFERENCEDVALUE_PATH;
pathOctets.efIDOrPath.octs = (char *)calloc(1, pathLength);
memcpy(pathOctets.efIDOrPath.octs, path, pathLength);
pathOctets.efIDOrPath.octetLen = pathLength;
privateRSAKeyAttr.value->a.indirect->a.path = &pathOctets;
privateRSAKeyAttr.modulusLength = modulusLength;
/*
** In ra Đối tượng dữ liệu khóa riêng
*/
PrintCIOChoice(stdout, cio, 3);
/*
```

```
        ** Đối tượng dữ liệu khóa riêng mã hóa BER
        */
        BERLength = BEncCIOChoiceContent(gb, cio);
}
/*
** Giải mã một khóa riêng thành một Đối tượng dữ liệu trong EF.OD
*/
PrivateKeyObject_PrivateRSAKeyAttributes *PrivateRSAKey(unsigned char *BER, unsigned int BERLength)
{
        SBuf b;
        GenBuf *gb;
        unsigned int bytesDecoded = 0;
        ENV_TYPE env;
        CIOChoice *cio;
        AsnTag tagID0;
        AsnLen elmtLen0;
        if(setjmp(env)!= 0) exit(0);
        cio = calloc(1, sizeof(CIOChoice));
        SBufInstallData(&b, BER, BERLength);
        SBuftoGenBuf(&b, &gb);
        tagID0 = BDecTag(gb, &bytesDecoded, env);
        elmtLen0 = BDecLen(gb, &bytesDecoded, env);
        /*
        ** Giải mã Đối tượng dữ liệu khóa riêng RSA
        */
        BDecCIOChoiceContent(gb, tagID0, elmtLen0, cio, &bytesDecoded, env);
        return ((PrivateKeyChoice *)(cio->a.privateKeys->a.objects->first->data))->a.privateRSAKey;
}
```

E.2.4 Mã hóa BER

```
<EF_OD>
0xa0,0x51,0xa0,0x4f,0x30,0x4d,0x30,0x12,0x0c,0x04,0x4b,0x45,0x59,0x31,0x03,0x02,
0x05,0x80,0x04,0x03,0x41,0x44,0x4d,0x02,0x01,0x01,0x30,0x12,0x04,0x01,0x9b,0x03,
0x03,0x06,0x20,0x40,0x01,0x01,0xff,0x03,0x02,0x03,0x98,0x02,0x01,0x0a,0xa0,0x13,
0x30,0x11,0xa0,0x0f,0x30,0x0d,0x02,0x01,0x05,0x04,0x08,0x31,0x32,0x33,0x34,0x35,
0x36,0x37,0x38,0xa1,0x0e,0x30,0x0c,0x30,0x06,0x04,0x04,0x3f,0x00,0x40,0x41,0x02,
0x02,0x04,0x00
</EF_OD>
```

Bảng E.1 là một sơ đồ giản lược của mã hóa BER.

Bảng E.1 - EF.PrKD của Khóa riêng RSA

																	Loại dữ liệu
A0	51	ClOChoice: đối tượng dữ liệu khóa riêng															
		A0	4F	PrivateKeyChoice: Khóa RSA riêng													
				30	4D	Đối tượng khóa RSA riêng											
						30	12	Thuộc tính đối tượng phổ thông									
								0C	04	label	4D, 45, 59, 31						Chuỗi UTF-8
								30	02	flags	05, 81						BIT STRING
								04	03	auth id	41, 44, 44						OCTET STRING
								02	01	userConsest	01						INTERGER
						A0	13	Thuộc tính khóa riêng phổ thông									
								30	11	Chuỗi							
										A0	0f	keyIdentifier					
												30	0D	Chuỗi			
														02	01	IdType	05	INTERGER
														60	08	IdValue	31, 32, 33, 34, 35, 36, 37 38	OpenType
						A1	0e	Thuộc tính khóa RSA riêng									
								30	0C	Chuỗi							
										30	06	Đường dẫn					
												04	04	efidOrPath		3F, 00, 40, 41	OCTET STRING
										02	02	ModulusLength				04, 00	INTERGER

E.3 Mã hóa một Bộ chứa dữ liệu được bảo vệ

E.3.1 Mô tả ví dụ Ứng dụng thông tin mã hóa

Một đối tượng bộ chứa dữ liệu với hai điều kiện an ninh, một với READ và một với UPDATE. Dữ liệu trong các bộ chứa dữ liệu là một BER-TLV. Khóa bí mật SK-1 phải được xác nhận nhằm thay đổi mật khẩu AO-1.

E.3.2 Mã hóa ASN.1 của Đối tượng Bộ chứa dữ liệu được bảo vệ

```
dataContainerObjects objects { -- SEQUENCE OF --
        iso7816DO { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                        label '444f2d31'H -- "DO-1" --,
                        flags '40'H,
                        accessControlRules { -- SEQUENCE OF --
                                { -- SEQUENCE --
                                        accessMode '80'H,
                                        securityCondition or { -- SEQUENCE OF --
                                                authID '414f2d31'H -- "AO-1" --,
                                                authID '414f2d32'H -- "AO-2" --
                                                }
                                },
                                { -- SEQUENCE --
                                        accessMode '40'H,
                                        securityCondition and { -- SEQUENCE OF --
                                                authID '414f2d31'H -- "AO-1" --,
                                                authID '414f2d32'H -- "AO-2" --
                                                }
                                        }
                                }
                        },
        classAttributes { -- SEQUENCE --
        },
                typeAttributes direct '80020102'H
                }
        }
authObjects objects { -- SEQUENCE OF --
        pwd { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                        label '414f2d31'H -- "AO-1" --,
                        flags '40'H
                },
                classAttributes { -- SEQUENCE --
                        authID '414f2d31'H -- "AO-1" --,
                        authReference 1,
                        selDentifier 2
                },
                typeAttributes { -- SEQUENCE --
                        pwdFlags '0400'H,
```

```
                        pwdType 1,
                        minLength 4,
                        storedLength 12,
                        maxLength 8,
                        padChar 'ff'H -- " " --,
                        path { -- SEQUENCE --
                                efIDOrPath '3f004045'H
                        }
                        }
                }
        }
secretKeys objects { -- SEQUENCE OF --
        genericSecretKey { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                        label '534b2d31'H -- "SK-1" --,
                        flags '40'H,
                        authID '414f2d31'H -- "AO-1" --
                        },
                classAttributes { -- SEQUENCE --
                        ID '534b2d31'H -- "SK-1" --,
                        usage '0200'H,
                        native TRUE,
                        accessFlags '10'H,
                        keyReference 10
                        },
                subClassAttributes { -- SEQUENCE --
                        keyLen 64
                },
                typeAttributes { -- SEQUENCE --
                        keyType {2 8},
                        keyAttr '58'H
                }
        }
}
```

E.3.3 Mã từ ASN.1 đối với mã hóa và giải mã BER

```
/*
** Mã hóa một Đối tượng dữ liệu được bảo vệ
*/
volD DataObject(unsigned char      *label,
                unsigned char      objectFlags,
                unsigned char      *password1,
                unsigned char      *password2
                )
{
        CIOChoice *cio;
        DataContainerObjectChoice *dco, **dcop;
        AccessControlRule *acr, **acrp;
```

```
SecurityCondition *sc, **scp;
SecurityCondition securityCondition1;
SecurityCondition securityCondition2;
AsnOcts authID1;
AsnOcts authID2;
CommonObjectAttributes commonObjectAttr = { 0 };
CommonDataContainerObjectAttributes commonDataContainerObjectAttributes = { 0 };
ISO7816DOAttributes iso7816DOAttributes = { 0 };
DataContainerObject_ISO7816DOAttributes pattr = { 0 };
CredentialIDentifier credentialIDentifier = { 0 };
Path pathOctets    = { 0 };
AsnOcts dataObjectValue1 = { sizeof(doValue1), doValue1 };
char commonObjectFlags[1] = { 0 };
AsnBits commonFlagsAsnBits = { 2, commonObjectFlags };
char acessControlRuleFlags1[1] = { 0 };
AsnBits acessControlRuleAsnBits1 = { 4, acessControlRuleFlags1 };
char acessControlRuleFlags2[1] = { 0 };
AsnBits acessControlRuleAsnBits2 = { 4, acessControlRuleFlags2 };
char usageFlagBits[2] = { 0 };
AsnBits usageFlagsAsnBits = = { 10, usageFlagBits };
authID1.octetLen = strlen(password1);
authID1.octs = strdup(password1);
authID2.octetLen = strlen(password2);
authID2.octs = strdup(password2);
/*
** Chọn lựa Đối tượng dữ liệu
*/
cio = (CIOChoice *)calloc(1, sizeof(DataContainerObjectChoice));
cio->choiceID = CIOCHOICE_DATACONTAINEROBJECTS;
cio->a.dataContainerObjects = (DataContainerObjects *)calloc(1, sizeof(DataContainerObjects));
cio->a.dataContainerObjects->a.objects = AsnListNew(sizeof (voID*));
cio→a.dataContainerObjects→choiceID = PATHOROBJECTS_DATACONTAINEROBJECTCHOICE_OBJECTS;
dcop = (DataContainerObjectChoice **)AsnListAppend(cio->a.dataContainerObjects->a.objects);
*dcop = dco = calloc(1, sizeof(DataContainerObjectChoice));
dco->choiceID = DATACONTAINEROBJECTCHOICE_ISO7816DO;
dco->a.iso7816DO = &pattr;
pattr.commonObjectAttributes = &commonObjectAttr;
pattr.classAttributes = &commonDataContainerObjectAttributes;
pattr.subClassAttributes = NULL;
pattr.typeAttributes = &iso7816DOAttributes;
/*
** Thuộc tính Đối tượng phổ biến
*/
commonObjectAttr.label.octetLen = strlen(label);
commonObjectAttr.label.octs = label;
commonObjectFlags[0] = objectFlags;
commonObjectAttr.flags = commonFlagsAsnBits;
```

```
commonObjectAttr.accessControlRules = AsnListNew(sizeof (void*));
acrp = (AccessControlRule **)AsnListAppend(commonObjectAttr.accessControlRules);
*acrp = acr = calloc(1, sizeof(AccessControlRule));
acessControlRuleFlags1[0] = (unsigned char)(READ_FLAG);
acr->accessMode = acessControlRuleAsnBits1;
securityCondition1.choiceID = SECURITYCONDITION_SEACOS_OR;
securityCondition1.a.seacos_or = AsnListNew(sizeof (void*));
scp = (SecurityCondition **)AsnListAppend(securityCondition1.a.seacos_or);
*scp = sc = calloc(1, sizeof(SecurityCondition));
sc->choiceID = SECURITYCONDITION_AUTHID;
sc->a.authID = &authID1;
scp = (SecurityCondition **)AsnListAppend(securityCondition1.a.seacos_or);
*scp = sc = calloc(1, sizeof(SecurityCondition));
sc->choiceID = SECURITYCONDITION_AUTHID;
sc->a.authID = &authID2;
acr->securityCondition = &securityCondition1;
acrp = (AccessControlRule **)AsnListAppend(commonObjectAttr.accessControlRules);
*acrp = acr = calloc(1, sizeof(AccessControlRule));
acessControlRuleFlags2[0] = (unsigned char)(UPDATE_FLAG);
acr->accessMode = acessControlRuleAsnBits2;
securityCondition2.choiceID = SECURITYCONDITION_SEACOS_AND;
securityCondition2.a.seacos_and = AsnListNew(sizeof (void*));
scp = (SecurityCondition **)AsnListAppend(securityCondition2.a.seacos_and);
*scp = sc = calloc(1, sizeof(SecurityCondition));
sc->choiceID = SECURITYCONDITION_AUTHID;
sc->a.authID = &authID1;
scp = (SecurityCondition **)AsnListAppend(securityCondition2.a.seacos_and);
*scp = sc = calloc(1, sizeof(SecurityCondition));
sc->choiceID = SECURITYCONDITION_AUTHID;
sc->a.authID = &authID2;
acr->securityCondition = &securityCondition2;
/*
** Thuộc tính Đối tượng bộ chứa dữ liệu phổ biến
*/
/*
** Thuộc tính Đối tượng dữ liệu của bộ TCVN ISO/IEC 7816
*/
iso7816DOAttributes.choiceID = OBJECTVALUE_DIRECT;
iso7816DOAttributes.a.direct.value = &dataObjectValue1;
SetAnyTypeByInt(&(iso7816DOAttributes.a.direct), issuerKeyHash);
/*
** In ra Đối tượng dữ liệu
*/
PrintCIOChoice(stdout, cio, 3);
/*
** Đối tượng dữ liệu mã hóa BER
*/
```

```
        BERLength += BEncCIOChoiceContent(gb,cio);
}
voID Password(const char  *label,
                unsigned char           objectFlags,
                unsigned char           *authID,
                unsigned int            authReference,
                unsigned int            seReference,
                unsigned char           *ID,
                unsigned char           *path, unsigned int pathLength,
                unsigned short  pwdFlags,
                unsigned int            pwdType,
                unsigned int            minimumLength,
                unsigned int            storedLength,
                unsigned int            maximumLength,
                unsigned char           paddingCharacter
                )
{
CIOChoice *cio;
AuthenticationObjectChoice *auth, **authp;
AuthenticationObject_PasswordAttributes pattr                                   = { 0 };
CommonObjectAttributes commonObjAttr                                            = { 0 };
CommonAuthenticationObjectAttributes commonAuthenticationObjectAttr     = { 0 };
PasswordAttributes passwordAttributes                                           = { 0 };
Path pathOctets                 = { 0 };
AsnOcts padChar                 = { 0 };
char commonObjectFlags[1]               = { 0 };
AsnBits commonFlagsAsnBits              = { 2, commonObjectFlags };
char passwordFlags[2]           = { 0 };
AsnBits passwordFlagsBits               = { 12, passwordFlags };
/*
** Chứng thực Chọn lựa đối tượng
*/
cio = (CIOChoice *)calloc(1, sizeof(AuthenticationObjectChoice));
cio->choiceID = CIOCHOICE_AUTHOBJECTS;
cio->a.authObjects = (AuthObjects *)calloc(1, sizeof(AuthObjects));
cio->a.authObjects->choiceID = PATHOROBJECTS_AUTHENTICATIONOBJECTCHOICE_OBJECTS;
cio->a.authObjects->a.objects = AsnListNew(sizeof (voID*));
authp = (AuthenticationObjectChoice **)AsnListAppend(cio->a.authObjects->a.objects);
*authp = auth = calloc(1, sizeof(AuthenticationObjectChoice));
auth->choiceID = AUTHENTICATIONOBJECTCHOICE_PWD;
auth->a.pwd = &pattr;
pattr.commonObjectAttributes    = &commonObjAttr;
pattr.classAttributes           = &commonAuthenticationObjectAttr;
pattr.subClassAttributes        = (AsnNull *)NULL;
pattr.typeAttributes            = &passwordAttributes;
/*
** Thuộc tính Đối tượng phổ biến
```

```
*/
commonObjAttr.label.octs          = _strdup(label);
commonObjAttr.label.octetLen      = strlen(label);
commonObjectFlags[0]              = objectFlags;
commonObjAttr.flags               = commonFlagsAsnBits;
/*
** Thuộc tính Đối tượng chứng thực phổ biến
*/
commonAuthenticationObjectAttr.authID.octs          = ID;
commonAuthenticationObjectAttr.authID.octetLen      = strlen(ID);
commonAuthenticationObjectAttr.authReference        = &authReference;
commonAuthenticationObjectAttr.seIDentifier         = &seReference;
/*
** Thuộc tính Mật khẩu
*/
passwordFlags[0] = (unsigned char)(pwdFlags>>8);
passwordFlags[1] = (unsigned char)(pwdFlags);
passwordAttributes.pwdFlags       = passwordFlagsBits;
passwordAttributes.pwdType              = pwdType;
passwordAttributes.minLength      = minimumLength;
passwordAttributes.storedLength   = storedLength;
passwordAttributes.maxLength      = (AsnInt *)calloc(1, sizeof(AsnInt));
*passwordAttributes.maxLength     = maximumLength;
padChar.octetLen = 1;
padChar.octs = (char *)calloc(1,1);
padChar.octs[0] = paddingCharacter;
passwordAttributes.padChar = padChar;
pathOctets.efIDOrPath.octs = path;
pathOctets.efIDOrPath.octetLen = pathLength;
passwordAttributes.path = &pathOctets;
/*
** In ra Đối tượng dữ liệu chứng thực
*/
fprintf(stdout, "\n\n");
PrintCIOChoice(stdout, cio, 3);
/*
**  Đối tượng dữ liệu chứng thực mã hóa BER
*/
BERLength += BEncCIOChoiceContent(gb,cio);
}
voID SecretKey(const char  *label,
               unsigned char          objectFlags,
               unsigned char          *authID,
               unsigned short   usageFlags,
               unsigned int           keyReference,
               unsigned char          *ID,
               unsigned int           keyLength
```

```
        )

CIOChoice *cio;
SecretKeyChoice *sk, **skp;
SecretKeyObject_GenericKeyAttributes pattr = { 0 };
CommonObjectAttributes commonObjAttr            = { 0 };
CommonKeyAttributes commonKeyAttr                       = { 0 };
CommonSecretKeyAttributes commonSecretKeyAttr   = { 0 };
GenericKeyAttributes genericKeyAttr             = { 0 };
Path pathOctets                                 = { 0 };
AsnOcts keyOIDOcts                                      = { 0 };
AsnOcts keyAttrOcts                                     = { 0 };
AsnOID keyOID                                           = { 0 };
char commonObjectFlags[1]                               = { 0 };
AsnBits commonFlagsAsnBits                              = { 2, commonObjectFlags };
char keyUsage[2]                                = { 0 };
AsnBits keyUsageAsnBits                         = { 9, keyUsage };
char keyNativeAsnBool                           = FALSE;
char keyAccessFlags[1]                          = { 0 };
AsnBits keyAccessFlagsAsnBits                   = { 4, keyAccessFlags };
/*
** Chọn lựa Khóa bí mật
*/
cio = (CIOChoice *)calloc(1, sizeof(SecretKeyChoice));
cio->choiceID = CIOCHOICE_SECRETKEYS;
cio->a.secretKeys = (SecretKeys *)calloc(1, sizeof(SecretKeys));
cio->a.secretKeys->choiceID = PATHOROBJECTS_SECRETKEYCHOICE_OBJECTS;
cio->a.secretKeys->a.objects = AsnListNew(sizeof (voID*));
skp = (SecretKeyChoice **)AsnListAppend(cio->a.secretKeys->a.objects);
*skp = sk = calloc(1, sizeof(SecretKeyChoice));
sk->choiceID = SECRETKEYCHOICE_GENERICSECRETKEY;
sk->a.genericSecretKey = &pattr;
pattr.commonObjectAttributes        = &commonObjAttr;
pattr.classAttributes               = &commonKeyAttr;
pattr.subClassAttributes            = &commonSecretKeyAttr;
pattr.typeAttributes                = &genericKeyAttr;
/*
** Thuộc tính Đối tượng phổ biến
*/
commonObjAttr.label.octs = label;
commonObjAttr.label.octetLen = strlen(label);
commonObjAttr.authID.octetLen = strlen(authID);
commonObjAttr.authID.octs = authID;
commonObjectFlags[0] = objectFlags;
commonObjAttr.flags = commonFlagsAsnBits;
/*
** Thuộc tính khóa công khai
```

```
        */
        commonKeyAttr.ID.octs = ID;
        commonKeyAttr.ID.octetLen = strlen(ID);
        keyUsage[0] = (unsigned char)(usageFlags>>8);
        keyUsage[1] = (unsigned char)(usageFlags);
        commonKeyAttr.usage = keyUsageAsnBits;
        keyNativeAsnBool = TRUE;
        commonKeyAttr.native = &keyNativeAsnBool;
        keyAccessFlags[0] = NEVEREXTRACTABLE_FLAG;
        commonKeyAttr.accessFlags= keyAccessFlagsAsnBits;
        commonKeyAttr.keyReference = &keyReference;
        /*
        ** Thuộc tính Khóa bí mật phổ biến
        */
        commonSecretKeyAttr.keyLen = (AsnInt *)calloc(1, sizeof(AsnInt));
        *commonSecretKeyAttr.keyLen = keyLength;
        /*
        ** Loại Khóa bí mật chung
        */
        keyOIDOcts.octetLen = 1;
        keyOIDOcts.octs = (char *)calloc(1,1);
        keyOIDOcts.octs[0] = 88;
        genericKeyAttr.keyType = keyOIDOcts;
        SetAnyTypeByInt(&genericKeyAttr.keyAttr, subjectKeyID);
        keyAttrOcts.octetLen = 1;
        keyAttrOcts.octs = (char *)calloc(1,1);
        keyAttrOcts.octs[0] = 88;
        genericKeyAttr.keyAttr.value = &keyAttrOcts;
        /*
        ** In ra Đối tượng dữ liệu khóa bí mật
        */
        fprintf(stdout, "\n\n");
        PrintCIOChoice(stdout, cio, 3);
        /*
        ** Đối tượng dữ liệu khóa bí mật mã hóa BER
        */
        BERLength += BEncCIOChoiceContent(gb,cio);
}
/*
** Tìm Đối tượng chứng thực mà bảo vệ một Đối tượng dữ liệu
*/
voID Access_Condition_for_Data_Object(unsigned char *DOBER, unsigned int DOBERLength,
                                      unsigned char *AOBER, unsigned int AOBERLength)
{
        ENV_TYPE env;
        CIOChoice *cioDO, *cioAO;
        SBuf dob, aob;
```

```
GenBuf *dogb, *aogb;
unsigned int bytesDecoded = 0;
AsnTag tagID0;
AsnLen elmtLen0;
DataContainerObjectChoice *dataObject;
AuthenticationObjectChoice *authenticationObject;
AuthenticationObject_PasswordAttributes* pwd;
AccessControlRule *accessControlRule;
SecurityCondition *securityCondition, *securityConditionAuthID;
AsnOcts *authID1;
if(setjmp(env)!=0) exit(0);
/*
** Lấy quy tắt truy cập đối với việc đọc Đối tượng dữ liệu
*/
cioDO = (CIOChoice *)calloc(1, sizeof(DataContainerObjectChoice));
SBufInstallData(&dob, DOBER, DOBERLength);
SBuftoGenBuf(&dob, &dogb);
tagID0 = BDecTag(dogb, &bytesDecoded, env);
elmtLen0 = BDecLen(dogb, &bytesDecoded, env);
BDecCIOChoiceContent(dogb, tagID0, elmtLen0, cioDO, &bytesDecoded, env);
dataObject = (DataContainerObjectChoice *)(cioDO->a.dataContainerObjects->a.objects->first->data);
FOR_EACH_LIST_ELMT(accessControlRule, dataObject->a.iso7816DO->commonObjectAttributes-
>accessControlRules)
{
        if(accessControlRule->accessMode.bits && READ_FLAG)
                break;
}
if(accessControlRule == NULL)
        exit(0);
securityCondition = accessControlRule->securityCondition;
if(securityCondition->choiceID != SECURITYCONDITION_SEACOS_OR)
        exit(0);
securityConditionAuthID = (SecurityCondition *)(securityCondition->a.seacos_or->first->data);
        if(securityConditionAuthID->choiceID != SECURITYCONDITION_AUTHID)
                exit(0);
authID1 = (AsnOcts *)securityConditionAuthID->a.authID;
/*
** Tìm Đối tượng chứng thực liên quan tới thuật ngữ đầu tiên trong điều kiện OR
*/
cioAO = (CIOChoice *)calloc(1, sizeof(AuthenticationObjectChoice));
SBufInstallData(&aob, AOBER, AOBERLength);
SBuftoGenBuf(&aob, &aogb);
tagID0 = BDecTag(aogb, &bytesDecoded, env);
elmtLen0 = BDecLen(aogb, &bytesDecoded, env);
BDecCIOChoiceContent(aogb, tagID0, elmtLen0, cioAO, &bytesDecoded, env);
FOR_EACH_LIST_ELMT(authenticationObject, cioAO->a.dataContainerObjects->a.objects)
{
```

```
            if(authenticationObject->choiceID == AUTHENTICATIONOBJECTCHOICE_PWD)
            {
                pwd = authenticationObject->a.pwd;
                if((authID1->octetLen == pwd->commonObjectAttributes->label.octetLen) &&
                    memcmp(authID1->octs, pwd->commonObjectAttributes->label.octs, authID1->octetLen) ==
                    0)
                {
                    /*
                    ** Tìm AO đi kèm với thuật ngữ đầu tiên trong điều kiện OR
                    ** liên quan tới chế độ truy cập ĐỌC của DO
                    */
                    break;
                }
            }
        }
}
```

E.3.4 Mã hóa BER

```
<EF_DO>
0xa7,0x46,0xa0,0x44,0xa0,0x42,0x30,0x34,0x0c,0x04,0x44,0x4f,0x2d,0x31,0x03,
0x02,0x06,0x40,0x30,0x28,0x30,0x12,0x03,0x02,0x04,0x80,0xa2,0x0c,0x04,0x04,
0x41,0x4f,0x2d,0x31,0x04,0x04,0x41,0x4f,0x2d,0x32,0x30,0x12,0x03,0x02,0x04,
0x40,0xa1,0x0c,0x04,0x04,0x41,0x4f,0x2d,0x31,0x04,0x04,0x41,0x4f,0x2d,0x32,
0x30,0x00,0xa1,0x08,0xa0,0x06,0x60,0x04,0x80,0x02,0x01,0x02
</EF_DO>
<EF_AO>
0xa8,0x3e,0xa0,0x3c,0x30,0x3a,0x30,0x0a,0x0c,0x04,0x41,0x4f,0x2d,0x31,0x03,0x02,
0x06,0x40,0x30,0x0c,0x04,0x04,0x41,0x4f,0x2d,0x31,0x02,0x01,0x01,0x80,0x01,0x02,
0xa1,0x1e,0x30,0x1c,0x03,0x03,0x04,0x04,0x00,0x0a,0x01,0x01,0x02,0x01,0x04,0x02,
0x01,0x0c,0x02,0x01,0x08,0x04,0x01,0xff,0x30,0x06,0x04,0x04,0x3f,0x00,0x40,0x45
</EF_AO>
<EF_SK>
0xa3,0x3b,0xa0,0x39,0xaf,0x37,0x30,0x10,0x0c,0x04,0x53,0x4b,0x2d,0x31,0x03,0x02,
0x06,0x40,0x04,0x04,0x41,0x4f,0x2d,0x31,0x30,0x14,0x04,0x04,0x53,0x4b,0x2d,0x31,
0x03,0x02,0x02,0x00,0x01,0x01,0xff,0x03,0x02,0x04,0x10,0x02,0x01,0x0a,0xa0,0x05,
0x30,0x03,0x02,0x01,0x40,0xa1,0x06,0x30,0x04,0x06,0x01,0x58,0x58
</EF_SK>
```

E.4 Mã hóa một Chứng nhận

E.4.1 Mô tả ví dụ Ứng dụng thông tin mã hóa

Một mô tả của một chứng nhận X.509.

E.4.2 Mã hóa ASN.1 của một Chứng nhận X.509

```
certificates objects { -- SEQUENCE OF --
        x509Certificate { -- SEQUENCE --
```

```
commonObjectAttributes { -- SEQUENCE --
        label '436572746966696361746520 31'H -- "Certificate 1" --,
        flags '40'H,
        authID '17'H,
        userConsent 5
},
classAttributes { -- SEQUENCE --
        ID '41444d'H -- "ADM" --,
        authority FALSE,
        IDentifier { -- SEQUENCE --
                IDType 0,
                IDValue '3132333435363738'H -- "12345678" --
                },
                certHash { -- SEQUENCE --
                        hashAlg { -- SEQUENCE --
                                algorithm {0 17 34 51},
                                parameters '332211'H
                        },
                        certID { -- SEQUENCE --
                                issuer iPAddress 'c0a82d01'H,
                                serialNumber 13107
                        },
                        hashVal '998877'H
                        },
                        trustedUsage { -- SEQUENCE --
                                keyUsage '2000'H
                        },
                        IDentifiers { -- SEQUENCE OF --
                                { -- SEQUENCE --
                                        IDType 5,
                                        IDValue '616263'H -- "abc" --
                                },
                                { -- SEQUENCE --
                                        IDType 5,
                                        IDValue '78797a'H -- "xyz" --
                                }
                        }
                },
                typeAttributes { -- SEQUENCE --
                        value indirect path { -- SEQUENCE --
                                efIDOrPath '3f004042'H -- "? @B" --
                                },
                        subject rdnSequence { -- SEQUENCE OF --
                                { -- SET OF --
                                { -- SEQUENCE --
                                        type {1 11 68},
                                        value NULL,
```

```
                                                    valuesWithContext { -- SET OF --
                                                        { -- SEQUENCE --
                                                            distingAttrValue '5577'H,
                                                            contextList { -- SET OF --
                                                                { -- SEQUENCE --
                                                                    contextType {2 40 86},
                                                                    contextValues { -- SET OF
                                                                    --
                                                                        '876543'H
                                                                },
                                                                fallback TRUE
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    },
                issuer rdnSequence { -- SEQUENCE OF --
                        { -- SET OF --
                            { -- SEQUENCE --
                                type {2 5 102},
                                value NULL,
                                valuesWithContext { -- SET OF --
                                    { -- SEQUENCE --
                                        distingAttrValue '8899'H,
                                        contextList { -- SET OF --
                                            { -- SEQUENCE --
                                                contextType {2 40 86},
                                                contextValues { -- SET OF --
                                                    '785634'H
                                                },
                                                fallback TRUE
                                            }
                                        }
                                    }
                                }
                            }
                        },
                serialNumber 22376
                }
        }
}
```

E.4.3 Mã từ ASN.1 đối với Mã hóa và Giải mã BER

```
** Ví dụ về mã của Mã hóa BER
*/
voID X509Certificate(unsigned char *label,
                     unsigned char    objectFlags,
                     unsigned char    *ID, unsigned int IDLength,
                     unsigned char    authority,
                     unsigned short   usageFlags,
                     unsigned int     externalIDentifierType,
                     unsigned char    *externalIDentifier,
                     unsigned char    *path, unsigned int pathLength,
                     unsigned char    *BER, unsigned int *BERLength
                     )
{
        unsigned int l;
        SBuf b;
        GenBuf *gb;
        unsigned char buffer[1024];
        // Đầu ra
        CIOChoice *cio;
        CertificateChoice *prk, **prkp;
        AccessControlRule *acr, **acrp;
        SecurityCondition *sc, **scp;
        CredentialIDentifier *cID, **cIDp;
        RelativeDistinguishedName *rdn, **rdnp;
        AttributeTypeAndDistinguishedValue *atadv, **atadvp;
        AttributeTypeAndDistinguishedValueSetOfSeq *atadvsos, **atadvsosp;
        Context *atadvsosso, **atadvsossop;
        AsnAny *any, **anyp;
        SecurityCondition securityCondition1;
        SecurityCondition securityCondition2;
        AsnOcts authID11 = { sizeof(AuthID11), AuthID11 };
        AsnOcts authID12 = { sizeof(AuthID12), AuthID12 };
        AsnOcts authID21 = { sizeof(AuthID21), AuthID21 };
        AsnOcts authID22 = { sizeof(AuthID22), AuthID22 };
        CertificateObject_X509CertificateAttributes pattr      = { 0 };
        CommonObjectAttributes commonObjAttr                   = { 0 };
        CommonCertificateAttributes commonCertificateAttr  = { 0 };
        X509CertificateAttributes x509CertificateAttr          = { 0 };
        CredentialIDentifier credentialIDentifier              = { 0 };
        Path pathOctets                                        = { 0 };
        AsnOcts issuerHash                                             = { 0 };
        AsnOcts issuerHash1                                    = { sizeof(IDentifier1), IDentifier1 };
        AsnOcts issuerHash2                                    = { sizeof(IDentifier2), IDentifier2 };
        AsnOcts asnATADVvalue          = { sizeof(ATADVvalue), ATADVvalue };
        AsnOcts asnATADVdistvalue              = { sizeof(ATADVdistvalue), ATADVdistvalue };
        AsnOcts asnATADVvalueIssuer    = { sizeof(ATADVvalueIssuer ), ATADVvalueIssuer };
        AsnOcts asnATADVdistvalueIssuer = { sizeof(ATADVdistvalueIssuer ), ATADVdistvalueIssuer };
```

```
char commonObjectFlags[1]                    = { 0 };
AsnBits commonFlagsAsnBits                   = { 2, commonObjectFlags };
char acessControlRuleFlags1[1]     = { 0 };
AsnBits acessControlRuleAsnBits1   = { 4, acessControlRuleFlags1 };
char acessControlRuleFlags2[1]     = { 0 };
AsnBits acessControlRuleAsnBits2   = { 4, acessControlRuleFlags2 };
char usageFlagBits[2]              = { 0 };
AsnBits usageFlagsAsnBit{ 10, usageFlagBits };
CertHash certHash                            = { 0 };
AlgorithmIDentifier algoID         = { 0 };
AsnOcts parameters                           = { 0 };
CertID certID                      = { 0 };
GeneralName issuer;
Usage usage = { 0 };
AsnAny contextValue1;
AsnOcts contextValue1Octs = { sizeof(ContextValue1Octs), ContextValue1Octs };
AsnAny contextValue1Issuer;
AsnOcts contextValue1OctsIssuer = { sizeof(ContextValue1OctsIssuer), ContextValue1OctsIssuer };
InitAnyCryptographicInformationFramework();
InitAnyInformationFramework2();
SBufInit(&b, buffer, sizeof(buffer));
SBufResetInWriteRvsMode(&b);
SBuftoGenBuf(&b, &gb);
/*
** Điều 8.3 Loại CIOChoice
**
** "EF.OD phải bao gồm tổ hợp của 0, 1 hay nhiều giá trị DER mã hóa của CIOChoice."
**
*/
cio = (CIOChoice *)calloc(1, sizeof(CertificateChoice));
cio->choiceID = CIOCHOICE_CERTIFICATES;
/*
** "Mong đợi rằng một mục vào EF.OD thường tham chiếu một tệp tin riêng lẻ
** (chọn lựa đường dẫn của PathOrObjects) bao gồm các CIO của loại được chỉ định.
** Một mục vào có thể, tuy nhiên giữ trực tiếp các CIO (chọn lựa các đối tượng của
** PathOrObjects), nếu các đối tượng và tệp tin EF.OD có cùng các yêu cầu
**  kiểm soát truy cập."
**
** PathOrObjects{CertificateChoice}
*/
cio->a.certificates = (P15Certificates *)calloc(1, sizeof(P15Certificates));
cio->a.certificates->choiceID = PATHOROBJECTS_CERTIFICATECHOICE_OBJECTS;
cio->a.certificates->a.objects = AsnListNew(sizeof (voID*));
/*
** Điều 8.4.1 CertificateChoice
**
** "Loại này bao gồm thông tin với một khóa riêng. Mỗi giá trị bao gồm
```

```
** các thuộc tính phổ biến với bất kỳ đối tượng, khóa, khóa riêng nào
** và các thuộc tính cụ thể với khóa."
*/
prkp = (CertificateChoice **)AsnListAppend(cio->a.certificates->a.objects);
*prkp = prk = calloc(1, sizeof(CertificateChoice));
prk->choiceID = CERTIFICATECHOICE_X509CERTIFICATE;
prk->a.x509Certificate = &pattr;
pattr.commonObjectAttributes      = &commonObjAttr;
pattr.classAttributes             = &commonCertificateAttr;
pattr.subClassAttributes          = NULL;
pattr.typeAttributes              = &x509CertificateAttr;
/*
** Điều 8.2.8 CommonObjectAttributes
**
** "Loại này là một bộ chứa với các thuộc tính phổ biến với tất cả các CIO."
**
*/
commonObjAttr.label.octs = label;
commonObjAttr.label.octetLen = strlen(label);
commonObjectFlags[0] = objectFlags;
commonObjAttr.flags = commonFlagsAsnBits;
commonObjAttr.authID.octetLen = sizeof(authID);
commonObjAttr.authID.octs = authID;
commonObjAttr.userConsent = &one;
/*
** Điều 8.2.15 CommonCertificateAttributes
**
** "Khi một khóa công khai trong một chứng nhận tham chiếu tới một chứng nhận
** tham chiếu bởi một đối tượng thông tin chứng nhân liên quan tới một khóa riêng
** tham chiếu bởi đối tượng thông tin khóa riêng, thì các đối tượng thông tin
** phải chia sẻ giá trị giống nhau với trường ID. Yêu cầu này tìm kiếm một
** khóa riêng đơn giản liên quan tới một chứng nhận cụ thể và ngược lại.
** Nhiều chứng nhận có cùng khóa phải chia sẻ giá trị giống nhau cho ID."
*/
commonCertificateAttr.ID.octetLen = IDLength;
commonCertificateAttr.ID.octs = ID;
commonCertificateAttr.authority = &authority;
issuerHash.octs = externalIDentifier;
issuerHash.octetLen = strlen(externalIDentifier);
credentialIDentifier.IDValue.value = &issuerHash;
SetAnyTypeByInt(&(credentialIDentifier.IDValue), externalIDentifierType);
commonCertificateAttr.IDentifier = &credentialIDentifier;
/* Băm chứng nhận */
commonCertificateAttr.certHash = &certHash;
certHash.hashAlg = &algoID;
algoID.algorithm.octetLen = sizeof(algo1ID);
algoID.algorithm.octs = algo1ID;
```

```
parameters.octetLen = sizeof(algo1Pm);
parameters.octs = algo1Pm;
algoID.parameters.value = &parameters;
SetAnyTypeByInt(&algoID.parameters, externalIDentifierType);
/* Bộ định danh chứng nhận */
certHash.certID = &certID;
certID.issuer = &issuer;
issuer.choiceID = GENERALNAME_IPADDRESS;
issuer.a.iPAddress = (AsnOcts *)calloc(1, sizeof(AsnOcts));
issuer.a.iPAddress->octetLen = sizeof(issuerIPAddress);
issuer.a.iPAddress->octs = issuerIPAddress;
certID.serialNumber = 0x3333;
/* Băm chứng nhận */
certHash.hashVal.bitLen = 8*sizeof(hashbits);
certHash.hashVal.bits = hashbits;
/* Cách dùng */
commonCertificateAttr.trustedUsage = &usage;
usageFlagBits[0] = (unsigned char)(usageFlags>>8);
usageFlagBits[1] = (unsigned char)(usageFlags);
usage.keyUsage = usageFlagsAsnBits;
/* Bộ định danh */
commonCertificateAttr.IDentifiers = AsnListNew(sizeof (voID*));
cIDp = (CredentialIDentifier **)AsnListAppend(commonCertificateAttr.IDentifiers);
*cIDp = cID = calloc(1, sizeof(CredentialIDentifier));
cID->IDType = externalIDentifierType;
cID->IDValue.value = &issuerHash1;
SetAnyTypeByInt(&(cID->IDValue), externalIDentifierType);
cIDp = (CredentialIDentifier **)AsnListAppend(commonCertificateAttr.IDentifiers);
*cIDp = cID = calloc(1, sizeof(CredentialIDentifier));
cID->IDType = externalIDentifierType;
cID->IDValue.value = &issuerHash2;
SetAnyTypeByInt(&(cID->IDValue), externalIDentifierType);

/*
** Điều 8.7.2 Thuộc tính chứng nhận X509
**
** "X509CertificateAttributes.value: Giá trị phải là một ReferencedValue
** hoặc nhận dạng một tệp tin gồm một chứng nhận mã hóa DER tại địa điểm có sẵn,
** hoặc một URL trỏ tới vài địa điểm mà chứng nhận có thể tìm thấy.
**
** "X509CertificateAttributes.subject, X509CertificateAttributes.issuer và
** X509CertificateAttributes.serialNumber: các trường này tương tự như với
** các trường tương ứng trong ISO/IEC 9594-8:1998. Các giá trị của các trường này
** phải giống hệt các trường tương ứng trong chính chứng nhận.
** Lí do tạo ra tùy chọn là nhằm cung cấp nhiều tiện ích về không gian, khi chúng
** sẵn sàng xem xét trong chính chứng nhận."
*/
```

```
x509CertificateAttr.value = (ObjectValue *)calloc(1, sizeof(ObjectValue));
x509CertificateAttr.value->choiceID = OBJECTVALUE_INDIRECT;
x509CertificateAttr.value->a.indirect = (ReferencedValue *)calloc(1, sizeof(ReferencedValue));
x509CertificateAttr.value->a.indirect->choiceID = REFERENCEDVALUE_PATH;
pathOctets.efIDOrPath.octs = (char *)calloc(1, pathLength);
memcpy(pathOctets.efIDOrPath.octs, path, pathLength);
pathOctets.efIDOrPath.octetLen = pathLength;
x509CertificateAttr.value->a.indirect->a.path = &pathOctets;
/* Nội dung */
x509CertificateAttr.subject = (Name *)calloc(1, sizeof(Name));
x509CertificateAttr.subject->choiceID = NAME_RDNSEQUENCE;
x509CertificateAttr.subject->a.rdnSequence = AsnListNew(sizeof (voID*));
rdnp = (RelativeDistinguishedName **)AsnListAppend(x509CertificateAttr.subject->a.rdnSequence);
*rdnp = rdn = AsnListNew(sizeof (voID*));
atadvp = (AttributeTypeAndDistinguishedValue **)AsnListAppend(rdn);
*atadvp=atadv=
        (AttributeTypeAndDistinguishedValue*)
                calloc(1,sizeof(AttributeTypeAndDistinguishedValue));
atadv->type.octetLen = sizeof(ATADVtype);
atadv->type.octs = ATADVtype;
atadv->value.value = &asnATADVvalue;
SetAnyTypeByOID(&(atadv->value), &noRevAvail);
atadv->valuesWithContext = AsnListNew(sizeof (voID*));
atadvsosp= (AttributeTypeAndDistinguishedValueSetOfSeq **)AsnListAppend(atadv->valuesWithContext);
*atadvsosp=atadvsos=
        (AttributeTypeAndDistinguishedValueSetOfSeq *)
                calloc(1,sizeof(AttributeTypeAndDistinguishedValueSetOfSeq));
atadvsos->distingAttrValue.value = &asnATADVdistvalue;
SetAnyTypeByInt(&(atadvsos->distingAttrValue), externalIDentifierType);
atadvsos->contextList = AsnListNew(sizeof (voID*));
atadvsossop = (Context **)AsnListAppend(atadvsos->contextList);
*atadvsossop = atadvsosso = (Context *)calloc(1, sizeof(Context));
atadvsosso->contextType.octetLen = sizeof(contextTypeIssuer);
atadvsosso->contextType.octs = contextTypeIssuer;
atadvsosso->contextValues = AsnListNew(sizeof (voID*));
anyp = (AsnAny **)AsnListAppend(atadvsosso->contextValues);
contextValue1.value = &contextValue1Octs;
SetAnyTypeByInt(&(contextValue1), externalIDentifierType);
*anyp = any = &contextValue1;
atadvsosso->fallback = &True;
atadv->primaryDistinguished = FALSE;
/* Tổ chức phát hành */
x509CertificateAttr.issuer = (Name *)calloc(1, sizeof(Name));
x509CertificateAttr.issuer->choiceID = NAME_RDNSEQUENCE;
x509CertificateAttr.issuer->a.rdnSequence = AsnListNew(sizeof (voID*));
rdnp = (RelativeDistinguishedName **)AsnListAppend(x509CertificateAttr.issuer->a.rdnSequence);
*rdnp = rdn = AsnListNew(sizeof (voID*));
```

```
        atadvp = (AttributeTypeAndDistinguishedValue **)AsnListAppend(rdn);
        *atadvp=atadv=
                (AttributeTypeAndDistinguishedValue *)
                        calloc(1, sizeof(AttributeTypeAndDistinguishedValue));
        atadv->type.octetLen = sizeof(ATADVtypeIssuer);
        atadv->type.octs = ATADVtypeIssuer;
        atadv->value.value = &asnATADVvalueIssuer;
        SetAnyTypeByOID(&(atadv->value), &noRevAvail);
        atadv->valuesWithContext = AsnListNew(sizeof (voID*));
        atadvsosp= (AttributeTypeAndDistinguishedValueSetOfSeq **)AsnListAppend(atadv->valuesWithContext);
        *atadvsosp=atadvsos=
                (AttributeTypeAndDistinguishedValueSetOfSeq *)
                        calloc(1, sizeof(AttributeTypeAndDistinguishedValueSetOfSeq));
        atadvsos->distingAttrValue.value = &asnATADVdistvalueIssuer;
        SetAnyTypeByInt(&(atadvsos->distingAttrValue), externalIDentifierType);
        atadvsos->contextList = AsnListNew(sizeof (voID*));
        atadvsossop = (Context **)AsnListAppend(atadvsos->contextList);
        *atadvsossop = atadvsosso = (Context *)calloc(1, sizeof(Context));
        atadvsosso->contextType.octetLen = sizeof(contextTypeIssuer);
        atadvsosso->contextType.octs = contextTypeIssuer;
        atadvsosso->contextValues = AsnListNew(sizeof (voID*));
        anyp = (AsnAny **)AsnListAppend(atadvsosso->contextValues);
        contextValue1Issuer.value = &contextValue1OctsIssuer;
        SetAnyTypeByInt(&(contextValue1Issuer), externalIDentifierType);
        *anyp = any = &contextValue1Issuer;
        atadvsosso->fallback = &True;
        atadv->primaryDistinguished = FALSE;
        x509CertificateAttr.serialNumber = &certSerialNumber;
        /*
        ** In ra Đối tượng dữ liệu chứng nhận
        */
        PrintCIOChoice(stdout, cio, 3);
        /*
        ** Đối tượng dữ liệu chứng nhận mã hóa BER
        */
        *BERLength = BEncCIOChoiceContent(gb,cio);
        GenBufResetInReadMode(gb);
        l = 0;
        memcpy(BER, GenBufGetSeg(gb, &l), *BERLength);
}
/*
** Ví dụ về Mã đối với Mã hóa BER
*/
Path_to_X509_Certificate(unsigned char *BER, unsigned int BERLength)
{
        SBuf b;
        GenBuf *gb;
```

```
    unsigned int bytesDecoded = 0;
    ENV_TYPE env;
    AsnTag tagID0;
    AsnLen elmtLen0;
    CIOChoice *cio;
    CertificateChoice *certificate;
    CertificateObject_X509CertificateAttributes* x509Certificate;
    X509CertificateAttributes* typeAttributes;
    ObjectValue* value;
    unsigned int i, pathLength;
    unsigned char *path;
    if(setjmp(env)!= 0) exit(0);
    cio = calloc(1, sizeof(CIOChoice));
    SBufInstallData(&b, BER, BERLength);
    SBuftoGenBuf(&b, &gb);
    tagID0 = BDecTag(gb, &bytesDecoded, env);
    elmtLen0 = BDecLen(gb, &bytesDecoded, env);
    /*
    ** Giả mã Chứng nhận X.509
    */
    BDecCIOChoiceContent(gb, tagID0, elmtLen0, cio, &bytesDecoded, env);
    /*
    ** Tìm đường dẫn tới chứng nhận
    */
    certificate = (CertificateChoice *)(cio->a.certificates->a.objects->first->data);
    x509Certificate = certificate->a.x509Certificate;
    typeAttributes = x509Certificate->typeAttributes;
    value = typeAttributes->value;
    printf("Path to Certificate: ");
    pathLength = value->a.indirect->a.path->efIDOrPath.octetLen;
    path = value->a.indirect->a.path->efIDOrPath.octs;
    for(i = 0; i < pathLength; i+=2)
    printf("0x%02x%02x ", path[i], path[i+1]);
    printf("\n");
}
```

E.4.4 Mã hóa BER

```
<BER>
0xa4,0x81,0xe1,0xa0,0x81,0xde,0x30,0x81,0xdb,0x30,0x19,0x0c,0x0d,0x43,0x65,0x72,
0x74,0x69,0x66,0x69,0x63,0x61,0x74,0x65,0x20,0x31,0x03,0x02,0x06,0x40,0x04,0x01,
0x17,0x02,0x01,0x05,0x30,0x58,0x04,0x03,0x41,0x44,0x4d,0x01,0x01,0x00,0x30,0x0d,
0x02,0x01,0x00,0x60,0x08,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0xa0,0x22,0xa0,
0x0c,0x30,0x0a,0x06,0x03,0x11,0x22,0x33,0x60,0x03,0x33,0x22,0x11,0xa1,0x0c,0x30,
0x0a,0x87,0x04,0xc0,0xa8,0x2d,0x01,0x02,0x02,0x33,0x33,0x03,0x04,0x00,0x99,0x88,
0x77,0xa1,0x05,0x03,0x03,0x06,0x20,0x00,0xa2,0x14,0x30,0x08,0x02,0x01,0x05,0x60,
0x03,0x61,0x62,0x63,0x30,0x08,0x02,0x01,0x05,0x60,0x03,0x78,0x79,0x7a,0xa1,0x64,
0x30,0x62,0x30,0x06,0x04,0x04,0x3f,0x00,0x40,0x42,0x30,0x28,0x31,0x26,0x30,0x24,
0x06,0x02,0x33,0x44,0x60,0x02,0x44,0x33,0x31,0x1a,0x30,0x18,0xa0,0x04,0x60,0x02,
```

0x55,0x77,0x31,0x10,0x30,0x0e,0x06,0x02,0x78,0x56,0x31,0x05,0x60,0x03,0x87,0x65,
0x43,0x01,0x01,0xff,0xa0,0x2a,0x30,0x28,0x31,0x26,0x30,0x24,0x06,0x02,0x55,0x66,
0x60,0x02,0x66,0x77,0x31,0x1a,0x30,0x18,0xa0,0x04,0x60,0x02,0x88,0x99,0x31,0x10,
0x30,0x0e,0x06,0x02,0x78,0x56,0x31,0x05,0x60,0x03,0x78,0x56,0x34,0x01,0x01,0xff,
0x02,0x02,0x57,0x68
</BER>

Bảng E.2 là một mô tả giản đồ của việc mã hóa BER.

Bảng E.2 - EF.CD của Chứng nhận X.059

Tag	Độ dài	Trường	Giá trị	Loại dữ liệu
A4	81 E1	Chọn CIO: Chứng nhận		
A0	81 DE	Chọn chứng nhận: chứng nhận X.509		
30	81 DB	Đối tượng chứng nhận X.509		
30	19	Thuộc tính đối tượng phổ dụng		
0C	0D	label	43, 65, 72, 74, 69, 66, 69, 63, 61, 74, 65, 20, 31	UTF-8 String
03	02	flags	05, 40	BIT STRING
04	01	authId	17	OCTET STRING
02	01	userContent	05	INTERGER
30	58	Thuộc tính chứng nhận phổ dụng		
04	03	ID	41, 44, 4D	OCTET STRING
01	01	Authority	00	BOOLEAN
30	0D	Identifier		
02	01	idType	00	INTERGER
60	08	idValue	31, 32, 33, 34, 35, 36, 37, 38	OpenType
A0	22	certHash		
A0	0C	hashAlg		
30	0A			
06	03	Thuật toán	11, 22, 33	OID
60	03	Thông số	33, 22, 11	OpenType
A1	0C	certID		
30	0A			
87	04	Issuer (iPAddress)	C0, A8, 2D, 01	OECTET STRING
02	02	serialNumber	33, 33	INTERGER

Bảng E.2 - EF.CD của Chứng nhận X.059 (tiếp theo)

														Loại dữ liệu
						03	04	hashVal				00, 99, 88, 77		INTERGER
			A1	05	trustUsage									
						03	03	keyUsage				06, 20, 00		BIT STRING
			A2	14	bộ định danh									
						30	08	CredentialIdentifier						
								02	01	idType		05		INTERGER
								60	03	idValue		61, 62, 63		OpenType
						30	08	CredentialIdentifier						
								02	01	idType		05		INTERGER
								60	03	idValue		78, 79, 7A		OpenType

Bảng E.2 - EF.CD của Chứng nhận X.059 (tiếp theo)

				Loại dữ liệu
A1	64	Thuộc tính chứng nhận X.509		
30	62	Chọn:		
30	06	Value		
04	04	EfidOrPath	3F, 00, 40, 42	OCTET STRING
30	28	Nội dung (RDNSequence)		
31	26	rdnSequence		
30	24	AttributeTypeAndDistinguishedValue		
06	02	Type	33, 44	OID
60	02	Value	44, 33	OpenType
31	1A	Tập valuesWithContext		
30	18	Chuỗi		
A0	04	SupportedAttributes		
60	02		55, 77	OpenType
31	10	contextList		
30	0E	Chuỗi		
06	02	contextType		OID
31	05	Tập		
60	03	Ngữ cảnh	9, 65, 43	OpenType
01	01	fallback	FF	BOOLEAN

Bảng E.2 - EF.CD của Chứng nhận X.059 (kết thúc)

Thẻ	Độ dài	Trường	Giá trị	Loại dữ liệu
A0	2A	bên phát hành (RDNSequence)		
30	28	Chọn:		
31	26	Tập AttributeTypeAndValue		
30	24			
06	02	Type	55, 66	OID
60	02	Value	66, 77	OpenType
31	1A	Tập valuesWithContext		
30	18	Chuỗi		
A0	04	SupportedAttributes		
60	02		88, 99	OpenType
31	10	Tập contextList		
30	0E	Chuỗi		
06	02	contextType	78, 56	OID
31	05	Tập Context		
60	03	Context	78, 56, 34	
01	01	fallback	FF	BOOLEAN
02	02	SerialNumber	57, 68	INTERGER

E.5 Mã hóa ứng dụng thông tin mã hóa ESIGN

E.5.1 Mô tả ví dụ ứng dụng thông tin mã hóa

Việc mã hóa của một ví dụ của Ứng dụng thông tin mã hóa với ESIGN được mô tả trong Điều 16 của CWA 14890-1:2004.

E.5.2 Mã hóa ASN.1 của ứng dụng thông tin mã hóa IAS

```
{ -- SEQUENCE OF --
        privateKeys path { -- SEQUENCE --
                efIDOrPath '4001'H
                },
        publicKeys path { -- SEQUENCE --
                efIDOrPath '4002'H
                },
        certificates path { -- SEQUENCE --
                efIDOrPath '4005'H
                },
        trustedCertificates path { -- SEQUENCE --
                efIDOrPath '4004'H
                },
        dataContainerObjects path { -- SEQUENCE --
                efIDOrPath '4006'H
                },
        authObjects path { -- SEQUENCE --
                efIDOrPath '4003'H
                }
        }
cardInfo { -- SEQUENCE --
        version 2,
        serialNumber '0102030405060708'H,
        manufacturerID '41434d45'H -- "ACME" --,
        label '5369676e617475726520417070c69636174696f6e'H
        cardflags '60'H,
        seInfo { -- SEQUENCE OF --
                { -- SEQUENCE --
                        se 1,
                        aID 'a000000167455349474e'H
                },
                { -- SEQUENCE --
                        se 2,
                        aID 'a000000167455349474e'H
                }
        },
        supportedAlgorithms { -- SEQUENCE OF --
                { -- SEQUENCE --
                        reference 1,
                        algorithm 544,
                        parameters "H -- "" --,
                        supportedOperations '02'H,
                        objID {0 1 3 14 3 2 26},
                        algRef 16
                },
                { -- SEQUENCE --
```

```
                        reference 2,
                        algorithm -2147483648,
                        parameters "H -- "" --,
                        supportedOperations '40'H,
                        objID {0 1 3 36 3 4 3 2 1},
                        algRef 17
                },
                { -- SEQUENCE --
                        reference 3,
                        algorithm 544,
                        parameters "H -- "" --,
                        supportedOperations '40'H,
                        objID {0 1 2 72 113 37 1 1 5},
                        algRef 18
                },
                { -- SEQUENCE --
                        reference 4,
                        algorithm -2147483647,
                        parameters "H -- "" --,
                        supportedOperations '50'H,
                        objID {0 1 3 36 7 2 1 1},
                        algRef 23
                },
                { -- SEQUENCE --
                        reference 5,
                        algorithm -2147483646,
                        parameters "H -- "" --,
                        supportedOperations '10'H,
                        objID {0 1 3 36 3 4 3 2 1}
                }
        },
        issuerID '4d61696e20537472656574204261 6e6b'H -- "Main Street Bank" --,
        holderID '53616c6c7920477265656e'H -- "Sally Green" --,
        lastUpdate generalizedTime '313938353131303632313036323 72e335a'H -- "19851106210627.3Z" --,
        preferredLanguage '4573706572616e746f'H -- "Esperanto" --
        }
AuthenticationObjects { -- SEQUENCE OF --
        pwd { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                        label '476c6f62616c20506173737 76f7264'H -- "Mật khẩu toàn cầu" --,
                        flags '40'H,
                        authID '03'H,
                        accessControlRules { -- SEQUENCE OF --
                                { -- SEQUENCE --
                                        accessMode '20'H,
                                        securityCondition authReference { -- SEQUENCE --
                                                authMethod 'c0'H,
                                                selDentifier 2
                                        }
                                }
                        }
                },
                classAttributes { -- SEQUENCE --
                        authID '01'H,
                        authReference 1,
```

```
                    selDentifier 2
                },
                typeAttributes { -- SEQUENCE --
                    pwdFlags '08'H,
                    pwdType 0,
                    minLength 4,
                    storedLength 0,
                    maxLength 8,
                    pwdReference 1,
                    padChar '00'H -- " " --,
                    path { -- SEQUENCE --
                        efIDOrPath "H -- "" --
                    }
                }
            },
            pwd { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                    label '5369676e617475726520506173737776f7264'H -- "Mật khẩu chữ ký" --,
                    flags '40'H,
                    authID '00'H,
                    accessControlRules { -- SEQUENCE OF --
                    { -- SEQUENCE --
                        accessMode '20'H,
                        securityCondition authReference { -- SEQUENCE --
                            authMethod 'c0'H,
                            selDentifier 2
                            }
                    }
                }
            },
            classAttributes { -- SEQUENCE --
                authID '02'H,
                authReference 129,
                selDentifier 2
            },
            typeAttributes { -- SEQUENCE --
                pwdFlags '48'H,
                pwdType 0,
                minLength 6,
                storedLength 0,
                maxLength 8,
                pwdReference 129,
                padChar '00'H -- " " --,
                path { -- SEQUENCE --
                    efIDOrPath '3f003f01'H
                }
            }
},
            pwd { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                    label
'526573657474696e6720436f646520666f722074686520476c6f62616c20506173737776f7264'H
-- "Resetting Code for the Global Password" --,
                    flags '40'H,
                    authID '03'H,
```

```
                accessControlRules { -- SEQUENCE OF --
                        { -- SEQUENCE --
                                accessMode '20'H,
                                securityCondition authReference { -- SEQUENCE --
                                        authMethod 'c0'H,
                                        selDentifier 2
                                        }
                        }
                },
                classAttributes { -- SEQUENCE --
                        authID '03'H,
                        authReference 129,
                        selDentifier 2
                },
                typeAttributes { -- SEQUENCE --
                        pwdFlags '4a'H,
                        pwdType 0,
                        minLength 8,
                        storedLength 0,
                        maxLength 8,
                        pwdReference 129,
                        padChar '00'H -- " " --,
                        path { -- SEQUENCE --
                                efIDOrPath "H -- "" --
                                }
                        }
                }
        }
PrivateKeyObjects { -- SEQUENCE OF --
        privateRSAKey { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                        label '5369676e6174757265204b6579'H -- "Khóa chữ ký" --,
                        flags '00'H,
                        authID '02'H,
                        userConsent 1,
                        accessControlRules { -- SEQUENCE OF --
                                { -- SEQUENCE --
                                        accessMode '20'H,
                                        securityCondition or { -- SEQUENCE OF --
                                                and { -- SEQUENCE OF --
                                                        authID '02'H,
                                                        authReference { -- SEQUENCE --
                                                                authMethod 'a0'H,
                                                                selDentifier 1
                                                        }
                                                },
                                                and { -- SEQUENCE OF --
                                                        authID '01'H,
                                                        authReference { -- SEQUENCE --
                                                                authMethod "H,
                                                                selDentifier 2
                                                        }
                                                }
                                        }
```

```
                        }
                    }
                },
                classAttributes { -- SEQUENCE --
                    ID '01'H,
                    usage '30'H,
                    native TRUE,
                    accessFlags 'b8'H,
                    keyReference 132,
                    algReference { -- SEQUENCE OF --
                        2,
                        3
                    }
                },
                subClassAttributes { -- SEQUENCE -
                },
                typeAttributes { -- SEQUENCE --
                    value indirect path { -- SEQUENCE --
                        efIDOrPath "H -- "" --
                    },
                    modulusLength 1024
                }
            },
            privateRSAKey { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                    label '534b2e4943432e415554'H -- "SK.ICC.AUT" --,
                    flags '00'H,
                    authID "H -- "" --,
                    userConsent 1
                },
                classAttributes { -- SEQUENCE --
                    ID '02'H,
                    usage '50'H,
                    native TRUE,
                    accessFlags 'b0'H,
                    keyReference 17,
                    algReference { -- SEQUENCE OF --
                        4
                    }
                },
                subClassAttributes { -- SEQUENCE --
                },
                typeAttributes { -- SEQUENCE --
                    value indirect path { -- SEQUENCE --
                        efIDOrPath "H -- "" --
                    },
                    modulusLength 1024
                }
            }
}
PublicKeyOjects { -- SEQUENCE OF --
            publicRSAKey { -- SEQUENCE --
                commonObjectAttributes { -- SEQUENCE --
                    label '504b2e5243412e43532d415554'H -- "PK.RCA.CS-AUT" --,
                    flags '40'H,
```

```
                authID '1'H -- "" --
            },
            classAttributes { -- SEQUENCE --
                ID '03'H,
                usage '01'H,
                native TRUE,
                keyReference 11
            },
            subClassAttributes { -- SEQUENCE --
            },
            typeAttributes { -- SEQUENCE --
                value indirect path { -- SEQUENCE --
                    efIDOrPath '1'H -- "" --
                },
                modulusLength 1024
            }
        }
}
CertificateOjects { -- SEQUENCE OF --
        x509Certificate { -- SEQUENCE --
            commonObjectAttributes { -- SEQUENCE --
                label '436572746966696361746520666f72205369676e617475726520536572766963652'H
                    -- "Chứng chỉ với Dịch vụ chữ ký" --
            },
            classAttributes { -- SEQUENCE --
                ID '01'H,
                authority FALSE
            },
            typeAttributes { -- SEQUENCE --
                value indirect path { -- SEQUENCE --
                    efIDOrPath '3f003f01c000'H
                }
            }
        },
        x509Certificate { -- SEQUENCE --
            commonObjectAttributes { -- SEQUENCE --
                label '434120436572746966696361746520666f72205369676e617475726520536572766963652'H
                    -- "Chứng chỉ CA với Dịch vụ chữ ký" --
            },
            classAttributes { -- SEQUENCE --
                ID '01'H,
                authority TRUE
            },
            typeAttributes { -- SEQUENCE --
                value indirect path { -- SEQUENCE --
                    efIDOrPath '3f003f01c608'H
                }
        }
},
        x509Certificate { -- SEQUENCE --
            commonObjectAttributes { -- SEQUENCE --
                label '435f43562e4943432e415554'H -- "C_CV.ICC.AUT" --
            },
            classAttributes { -- SEQUENCE --
                ID '02'H,
```

```
                authority FALSE
            },
            typeAttributes { -- SEQUENCE --
                value indirect path { -- SEQUENCE --
                    efIDOrPath '3f002f03'H
                }
            }
        }
}
DataContainerObjects { -- SEQUENCE OF --
        iso7816DO { -- SEQUENCE --
            commonObjectAttributes { -- SEQUENCE --
                label '446973706c6179204d657373616765'H -- "Display Message" --,
                flags 'c0'H,
                accessControlRules { -- SEQUENCE OF --
                    { -- SEQUENCE --
                        accessMode '40'H,
                        securityCondition or { -- SEQUENCE OF -
                            and { -- SEQUENCE OF --
                                authID '01'H,
                                authReference { -- SEQUENCE --
                                    authMethod '20'H,
                                    selDentifier 1
                                }
                            },
                            and { -- SEQUENCE OF --
                                authID '01'H,
                                authReference { -- SEQUENCE --
                                    authMethod 'e0'H,
                                    selDentifier 2
                                }
                            }
                        }
                    },
                    { -- SEQUENCE --
                        accessMode '80'H,
                        securityCondition authReference { -- SEQUENCE --
                            authMethod 'c0'H,
                            selDentifier 2
                        }
                    }
                }
            },
            classAttributes { -- SEQUENCE --
                applicationName 'a000000167455349474e'H
            },
            typeAttributes indirect path { -- SEQUENCE --
                efIDOrPath '3f003f01d000'H
            }
        }
}
```

E.5.3 Mã từ ASN.1 đối với mã hóa và giải mã BER

Không được cung cấp.

E.5.4 Mã hóa BER

<ESIGN_EF_OD>
0xa0,0x3c,0x30,0x08,0xa0,0x06,0x30,0x04,0x04,0x02,0x40,0x01,0x30,0x08,0xa1,0x06,
0x30,0x04,0x04,0x02,0x40,0x02,0x30,0x08,0xa4,0x06,0x30,0x04,0x04,0x02,0x40,0x05,
0x30,0x08,0xa5,0x06,0x30,0x04,0x04,0x02,0x40,0x04,0x30,0x08,0xa7,0x06,0x30,0x04,
0x04,0x02,0x40,0x06,0x30,0x08,0xa8,0x06,0x30,0x04,0x04,0x02,0x40,0x03
</ESIGN_EF_OD>
<ESIGN_EF_CardInfo>
0x02,0x01,0x02,0x04,0x08,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x0c,0x04,0x41,
0x43,0x4d,0x45,0x80,0x15,0x53,0x69,0x67,0x6e,0x61,0x74,0x75,0x72,0x65,0x20,0x41,
0x70,0x70,0x6c,0x69,0x63,0x61,0x74,0x69,0x6f,0x6e,0x03,0x02,0x05,0x60,0x30,0x22,
0x30,0x0f,0x02,0x01,0x01,0x04,0x0a,0xa0,0x00,0x00,0x01,0x67,0x45,0x53,0x49,0x47,
0x4e,0x30,0x0f,0x02,0x01,0x02,0x04,0x0a,0xa0,0x00,0x00,0x01,0x67,0x45,0x53,0x49,
0x47,0x4e,0xa2,0x7e,0x30,0x17,0x02,0x01,0x01,0x02,0x02,0x02,0x20,0x60,0x00,0x60,
0x01,0x02,0x06,0x06,0x01,0x03,0x0e,0x03,0x02,0x1a,0x02,0x01,0x10,0x30,0x18,0x02,
0x01,0x02,0x02,0x01,0x00,0x60,0x00,0x60,0x01,0x40,0x06,0x08,0x01,0x03,0x24,0x03,
0x04,0x03,0x02,0x01,0x02,0x01,0x11,0x30,0x19,0x02,0x01,0x03,0x02,0x02,0x02,0x20,
0x60,0x00,0x60,0x01,0x40,0x06,0x08,0x01,0x02,0x48,0x71,0x25,0x01,0x01,0x05,0x02,
0x01,0x12,0x30,0x17,0x02,0x01,0x04,0x02,0x01,0x01,0x60,0x00,0x60,0x01,0x50,0x06,
0x07,0x01,0x03,0x24,0x07,0x02,0x01,0x01,0x02,0x01,0x17,0x30,0x15,0x02,0x01,0x05,
0x02,0x01,0x02,0x60,0x00,0x60,0x01,0x10,0x06,0x08,0x01,0x03,0x24,0x03,0x04,0x03,
0x02,0x01,0x83,0x10,0x4d,0x61,0x69,0x6e,0x20,0x53,0x74,0x72,0x65,0x65,0x74,0x20,
0x42,0x61,0x6e,0x6b,0x84,0x0b,0x53,0x61,0x6c,0x6c,0x79,0x20,0x47,0x72,0x65,0x65,
0x6e,0xa5,0x13,0x18,0x11,0x31,0x39,0x38,0x35,0x31,0x31,0x30,0x36,0x32,0x31,0x30,
0x36,0x32,0x37,0x2e,0x33,0x5a,0x13,0x09,0x45,0x73,0x70,0x65,0x72,0x61,0x6e,0x74,
0x6f
</ESIGN_EF_CardInfo>
<ESIGN_EF_AOD>
0xa0,0x82,0x01,0x1a,0x30,0x50,0x30,0x26,0x0c,0x0f,0x47,0x6c,0x6f,0x62,0x61,0x6c,
0x20,0x50,0x61,0x73,0x73,0x77,0x6f,0x72,0x64,0x03,0x01,0x00,0x04,0x01,0x03,0x30,
0x0d,0x30,0x0b,0x03,0x01,0x00,0x30,0x06,0x03,0x01,0x80,0x02,0x01,0x02,0x30,0x09,
0x04,0x01,0x01,0x02,0x01,0x01,0x80,0x01,0x02,0xa1,0x1b,0x30,0x19,0x03,0x01,0x00,
0x0a,0x01,0x00,0x02,0x01,0x04,0x02,0x01,0x00,0x02,0x01,0x08,0x80,0x01,0x01,0x04,
0x01,0x00,0x30,0x02,0x04,0x00,0x30,0x5a,0x30,0x29,0x0c,0x12,0x53,0x69,0x67,0x6e,
0x61,0x74,0x75,0x72,0x65,0x20,0x50,0x61,0x73,0x73,0x77,0x6f,0x72,0x64,0x03,0x01,
0x00,0x04,0x01,0x00,0x30,0x0d,0x30,0x0b,0x03,0x01,0x00,0x30,0x06,0x03,0x01,0x80,
0x02,0x01,0x02,0x30,0x0a,0x04,0x01,0x02,0x02,0x02,0x00,0x81,0x80,0x01,0x02,0xa1,
0x21,0x30,0x1f,0x03,0x02,0x06,0x40,0x0a,0x01,0x00,0x02,0x01,0x06,0x02,0x01,0x00,
0x02,0x01,0x08,0x80,0x02,0x00,0x81,0x04,0x01,0x00,0x30,0x06,0x04,0x04,0x3f,0x00,
0x3f,0x01,0x30,0x6a,0x30,0x3d,0x0c,0x26,0x52,0x65,0x73,0x65,0x74,0x74,0x69,0x6e,
0x67,0x20,0x43,0x6f,0x64,0x65,0x20,0x66,0x6f,0x72,0x20,0x74,0x68,0x65,0x20,0x47,
0x6c,0x6f,0x62,0x61,0x6c,0x20,0x50,0x61,0x73,0x73,0x77,0x6f,0x72,0x64,0x03,0x01,
0x00,0x04,0x01,0x03,0x30,0x0d,0x30,0x0b,0x03,0x01,0x00,0x30,0x06,0x03,0x01,0x80,
0x02,0x01,0x02,0x30,0x0a,0x04,0x01,0x03,0x02,0x02,0x00,0x81,0x80,0x01,0x02,0xa1,
0x1d,0x30,0x1b,0x03,0x02,0x06,0x40,0x0a,0x01,0x00,0x02,0x01,0x08,0x02,0x01,0x00,
0x02,0x01,0x08,0x80,0x02,0x00,0x81,0x04,0x01,0x00,0x30,0x02,0x04,0x00
</ESIGN_EF_AOD>

<ESIGN_EF_PrKD>
0xa0,0x81,0xa8,0x30,0x67,0x30,0x3b,0x0c,0x0d,0x53,0x69,0x67,0x6e,0x61,0x74,0x75,
0x72,0x65,0x20,0x4b,0x65,0x79,0x03,0x01,0x00,0x04,0x01,0x02,0x02,0x01,0x01,0x30,
0x21,0x30,0x1f,0x03,0x01,0x00,0xa2,0x1a,0xa1,0x0b,0x04,0x01,0x02,0x30,0x06,0x03,
0x01,0x80,0x02,0x01,0x01,0xa1,0x0b,0x04,0x01,0x01,0x30,0x06,0x03,0x01,0x00,0x02,
0x01,0x02,0x30,0x18,0x04,0x01,0x01,0x03,0x01,0x00,0x01,0x01,0xff,0x03,0x01,0x80,
0x02,0x02,0x00,0x84,0xa1,0x06,0x02,0x01,0x02,0x02,0x01,0x03,0xa0,0x02,0x30,0x00,
0xa1,0x0a,0x30,0x08,0x30,0x02,0x04,0x00,0x02,0x02,0x04,0x00,0x30,0x3d,0x30,0x14,
0x0c,0x0a,0x53,0x4b,0x2e,0x49,0x43,0x43,0x2e,0x41,0x55,0x54,0x03,0x01,0x00,0x04,
0x00,0x02,0x01,0x01,0x30,0x15,0x04,0x01,0x02,0x03,0x02,0x06,0x40,0x01,0x01,0xff,
0x03,0x01,0x80,0x02,0x01,0x11,0xa1,0x03,0x02,0x01,0x04,0xa0,0x02,0x30,0x00,0xa1,
0x0a,0x30,0x08,0x30,0x02,0x04,0x00,0x02,0x02,0x04,0x00
</ESIGN_EF_PrKD>
<ESIGN_EF_PuKD>
0xa0,0x36,0x30,0x34,0x30,0x14,0x0c,0x0d,0x50,0x4b,0x2e,0x52,0x43,0x41,0x2e,0x43,
0x53,0x2d,0x41,0x55,0x54,0x03,0x01,0x00,0x04,0x00,0x30,0x0c,0x04,0x01,0x03,0x03,
0x01,0x00,0x01,0x01,0xff,0x02,0x01,0x0b,0xa0,0x02,0x30,0x00,0xa1,0x0a,0x30,0x08,
0x30,0x02,0x04,0x00,0x02,0x02,0x04,0x00
</ESIGN_EF_PuKD>
<ESIGN_EF_CD>
0xa0,0x81,0xa3,0x30,0x3b,0x30,0x23,0x0c,0x21,0x43,0x65,0x72,0x74,0x69,0x66,0x69,
0x63,0x61,0x74,0x65,0x20,0x66,0x6f,0x72,0x20,0x53,0x69,0x67,0x6e,0x61,0x74,0x75,
0x72,0x65,0x20,0x53,0x65,0x72,0x76,0x69,0x63,0x65,0x30,0x06,0x04,0x01,0x01,0x01,
0x01,0x00,0xa1,0x0c,0x30,0x0a,0x30,0x08,0x04,0x06,0x3f,0x00,0x3f,0x01,0xc0,0x00,
0x30,0x3e,0x30,0x26,0x0c,0x24,0x43,0x41,0x20,0x43,0x65,0x72,0x74,0x69,0x66,0x69,
0x63,0x61,0x74,0x65,0x20,0x66,0x6f,0x72,0x20,0x53,0x69,0x67,0x6e,0x61,0x74,0x75,
0x72,0x65,0x20,0x53,0x65,0x72,0x76,0x69,0x63,0x65,0x30,0x06,0x04,0x01,0x01,0x01,
0x01,0xff,0xa1,0x0c,0x30,0x0a,0x30,0x08,0x04,0x06,0x3f,0x00,0x3f,0x01,0xc6,0x08,
0x30,0x24,0x30,0x0e,0x0c,0x0c,0x43,0x5f,0x43,0x56,0x2e,0x49,0x43,0x43,0x2e,0x41,
0x55,0x54,0x30,0x06,0x04,0x01,0x02,0x01,0x01,0x00,0xa1,0x0a,0x30,0x08,0x30,0x06,
0x04,0x04,0x3f,0x00,0x2f,0x03
</ESIGN_EF_CD>
<ESIGN_EF_DO>
0xa0,0x62,0xa0,0x60,0x30,0x44,0x0c,0x0f,0x44,0x69,0x73,0x70,0x6c,0x61,0x79,0x20,
0x4d,0x65,0x73,0x73,0x61,0x67,0x65,0x03,0x01,0x80,0x30,0x2e,0x30,0x1f,0x03,0x01,
0x00,0xa2,0x1a,0xa1,0x0b,0x04,0x01,0x01,0x30,0x06,0x03,0x01,0x00,0x02,0x01,0x01,
0xa1,0x0b,0x04,0x01,0x01,0x30,0x06,0x03,0x01,0x80,0x02,0x01,0x02,0x30,0x0b,0x03,
0x01,0x80,0x30,0x06,0x03,0x01,0x80,0x02,0x01,0x02,0x30,0x0c,0x0c,0x0a,0xa0,0x00,
0x00,0x01,0x67,0x45,0x53,0x49,0x47,0x4e,0xa1,0x0a,0x30,0x08,0x04,0x06,0x3f,0x00,
0x3f,0x01,0xd0,0x00
</ESIGN_EF_DO>

Thư mục tài liệu tham khảo

[1] H. A LVESTRAND, "Tags for the IDentification of Languages," IETF RFC 1766.

[2] ANSI X3.4-1986, Coded Character Sets - 7-Bit American National Standard Code for Information Interchange.

[3] ANSI X9.68:2-2001, Digital Certificates for Mobile/Wireless and High Transaction Volume Financial Systems: Part 2: Domain Certificate Syntax.

[4] T. BERNERS -LEE, R. FIELDING, L. MASINTER, "Uniform Resource IDentifiers (URI): Generic Syntax", IETF RFC 2396, tháng 8 1998.

[5] J. CALLAS, L. DONNERHACKE, H. FINNEY, R. THAYER, "OpenPGP Message", IETF RFC 2440, tháng 11 1998.

[6] C. ELLISON, B. FRANTZ, B. LAMPSON, R. RIVEST, B. THOMAS, T. Y LONEN, "SPKI Certificate Theory", IETF RFC 2693, tháng 9 1999.

[7] ISO/IEC 9594-6:1998 | ITU-T Recommendation X.520 (1997), Information technology - Open Systems Interconnection - The Directory: Selected attribute types.

[8] ISO/IEC 8825-2:1998 | ITU-T Recommendation X.691 (1997), Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER).

[9] RSA Laboratories, PKCS #11 v2.11: Cryptographic Token Interface Standard.

[10] RSA Laboratories, PKCS #15 v1.1: Cryptographic Token Information Syntax Standard.

[11] WAP Forum, Wireless Application Protocol - Wireless Transport Layer Security Protocol Specification, phiên bản 06.