ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ПРАВИЛА КОДИРОВАНИЯ АСН.1

Часть 1

Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования

Издание официальное

БЗ 1—2003/311

ГОССТАНДАРТ РОССИИ
Москва

Предисловие

1 РАЗРАБОТАН Государственным научно-исследовательским и конструкторско-технологическим институтом «ТЕСТ» Министерства Российской Федерации по связи и информатизации

ВНЕСЕН Министерством Российской Федерации по связи и информатизации

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 13 мая 2003 г. № 140-ст

3 Настоящий стандарт содержит полный аутентичный текст международного стандарта ИСО/МЭК 8825-1—98 «Информационная технология. Правила кодирования АСН.1. Часть 1. Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования» с учетом Поправки № 1 (1999 г.) и Дополнения № 1 (2000 г.)

4 ВВЕДЕН ВПЕРВЫЕ

Содержание

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ПРАВИЛА КОДИРОВАНИЯ АСН.1.

Часть 1

Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования

Information technology. ASN.1 encoding rules. Part 1.
Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER)
and Distinguished Encoding Rules (DER)

Дата введения 2004—07—01

1 Область применения

В настоящем стандарте определен набор базовых правил кодирования, который может использоваться для получения спецификации синтаксиса передачи для значений типов, определенных с использованием абстрактной синтаксической нотации версии 1 (АСН.1), которая установлена в ГОСТ Р ИСО/МЭК 8824-1, ГОСТ Р ИСО/МЭК 8824-2, ГОСТ Р ИСО/МЭК 8824-3 и ГОСТ Р ИСО/МЭК 8824-4. Базовые правила кодирования также применимы для декодирования указанного синтаксиса передачи с целью идентификации передаваемых значений данных. В настоящем стандарте также определен набор канонических и отличительных правил кодирования, которые ограничивают кодирование значений ровно одной из альтернатив, предоставляемых базовыми правилами кодирования.

Эти правила кодирования используются во время передачи (поставщиком услуг уровня представления, когда это требуется контекстом представления).

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ 34.301—91 (ИСО 6429) Информационная технология. 7- и 8-битные кодированные наборы символов. Управляющие функции

ГОСТ Р ИСО/МЭК 7498-1—97 Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель (см. также Рекомендацию МСЭ-Т X.200)

ГОСТ Р ИСО/МЭК 8824-1—2001 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации (см. также Рекомендацию МСЭ-Т X.680)

ГОСТ Р ИСО/МЭК 8824-2—2001 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 2. Спецификация информационного объекта (см. также Рекомендацию МСЭ-Т X.681)

ГОСТ Р ИСО/МЭК 8824-3—2002 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 3. Спецификация ограничения (см. также Рекомендацию МСЭ-Т X.682)

ГОСТ Р ИСО/МЭК 8824-4—2003 Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 4. Параметризация спецификаций АСН.1 (см. также Рекомендацию МСЭ-Т X.683)

ГОСТ Р ИСО/МЭК 9594-8—98 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 8. Основы аутентификации

Издание официальное

ИСО/МЭК 2022—94* Информационная технология. Структура кода символов и методы расширения

ИСО 6093—85* Обработка информации. Представление числовых значений в символьных строках для информационного обмена

ИСО/МЭК 6429—92* Информационная технология. Управляющие функции для кодированных наборов символов

ИСО/МЭК 10646-1—93* Информационная технология. Универсальный, многооктетный кодовый набор символов (UCS). Часть 1. Архитектура и основная многоязычная плоскость

3 Определения

В настоящем стандарте используются определения по ГОСТ Р ИСО/МЭК 7498-1, ГОСТ Р ИСО/МЭК 8824-1, а также следующие определения:

3.1 **динамическое соответствие:** Установление требования к реализации придерживаться при передаче поведения, предписанного настоящим стандартом.

3.2 **статическое соответствие:** Установление требования к реализации обеспечивать допустимое множество возможностей из определенных настоящим стандартом.

3.3 **значение данных:** Информация, заданная как значение типа; тип и значение определены с использованием АСН.1.

3.4 **кодирование** (значений данных): Полная последовательность октетов, используемая для представления значения данных.

3.5 **октеты идентификатора:** Часть кодирования значения данных, которая используется для идентификации типа значения.

Примечание — В некоторых Рекомендациях МСЭ-Т для этой последовательности октетов применяют термин «**элемент данных**», но в настоящем стандарте его не используют, так как в других стандартах он применяется в смысле «**значение данных**».

3.6 **октеты длины:** Часть кодирования значения данных, следующая за октетами идентификатора, которая используется для определения конца кодирования.

3.7 **октеты содержимого:** Часть кодирования значения данных, которая представляет конкретное значение.

3.8 **октеты конца содержимого:** Часть кодирования значения данных, появляющаяся в его конце, которая используется для определения конца кодирования.

Примечание — Не для всех кодирований требуются октеты конца содержимого.

3.9 **простое кодирование:** Кодирование значения данных, в котором октеты содержимого непосредственно представляют это значение.

3.10 **составное кодирование:** Кодирование значения данных, в котором октеты содержимого являются полным кодированием одного или нескольких значений данных.

3.11 **получатель:** реализация декодирования октетов, созданных отправителем, для идентификации значения закодированных данных.

3.12 **отправитель:** Реализация кодирования значения данных для передачи.

3.13 **завершающий 0 бит:** 0 в последней позиции значения «битовая строка» (bitstring).

Примечание — 0 в значении битовой строки, состоящем из единственного бита 0, является завершающим 0 битом. Его удаление порождает пустую битовую строку.

4 Сокращения

В настоящем стандарте использованы следующие сокращения:

АСН.1 — абстрактная синтаксическая нотация версии 1

здп — значение данных (уровня) представления

BER — базовые правила кодирования (Basic Encoding Rules) АСН.1

CER — канонические правила кодирования (Canonical Encoding Rules) АСН.1

DER— отличительные правила кодирования (Distinguished Encoding Rules) АСН.1

---

* Международные стандарты — во ВНИИКИ Госстандарта России.

5 Нотация

В настоящем стандарте использована нотация, определенная в ГОСТ Р ИСО/МЭК 8824-1.

6 Соглашения

6.1 В настоящем стандарте специфицировано значение каждого октета в кодировании с использованием терминов **старший значащий бит** и **младший значащий бит.**

Примечание — В спецификациях нижних уровней используются те же самые обозначения для определения порядка передачи битов в последовательной линии связи или для распределения битов в параллельных каналах.

6.2 В настоящем стандарте биты октета нумеруют от 8 до 1, где бит 8 — **старший значащий бит,** а бит 1 — **младший значащий бит.**

6.3 В настоящем стандарте могут сравниваться две строки октетов. Они равны, если имеют одну и ту же длину и совпадают в каждой позиции октета. Строка октетов S_1 больше строки S_2 только в том случае, если:

а) S_1 и S_2 имеют идентичные октеты в каждой позиции до конечного октета в S_2 включительно, но S_1 длиннее,

или

б) S_1 и S_2 имеют различные октеты в одной или нескольких позициях и в первой такой позиции октет в S_1 больше, чем в S_2, если рассматривать октеты как двоичные числа без знака, бит n которых имеет вес 2^{n-1}.

7 Соответствие

7.1 Динамическое соответствие устанавливается всеми разделами стандарта.

7.2 Статическое соответствие устанавливается теми стандартами, которые определяют применение одного или нескольких из этих правил кодирования.

7.3 Альтернативные кодирования допускаются базовыми правилами кодирования как факультативные возможности отправителя. Получатели, которые заявляют о соответствии базовым правилам кодирования, должны поддержать все альтернативы.

Примечание — Примеры таких альтернативных кодирований показаны в 8.1.3.2б и таблице 3.

7.4 Альтернативные кодирования запрещаются каноническими или отличительными правилами кодирования.

8 Базовые правила кодирования

8.1 Общие правила кодирования

8.1.1 Структура кодирования

8.1.1.1 Кодирование значения данных должно состоять из четырех компонентов, которые должны появляться в следующем порядке:

а) октеты идентификатора (см. 8.1.2);

б) октеты длины (см. 8.1.3);

в) октеты содержимого (см. 8.1.4);

г) октеты конца содержимого (см. 8.1.5).

8.1.1.2 Октеты конца содержимого должны присутствовать только в том случае, если их наличие требуется значением октетов длины (см. 8.1.3).

8.1.1.3 На рисунке 1 показана структура кодирования (простого или составного). На рисунке 2 показан один из вариантов составного кодирования.

8.1.1.4 Структура кодирования не изменяется ни для нотации подтипа АСН.1, ни для нотации расширения типа АСН.1.

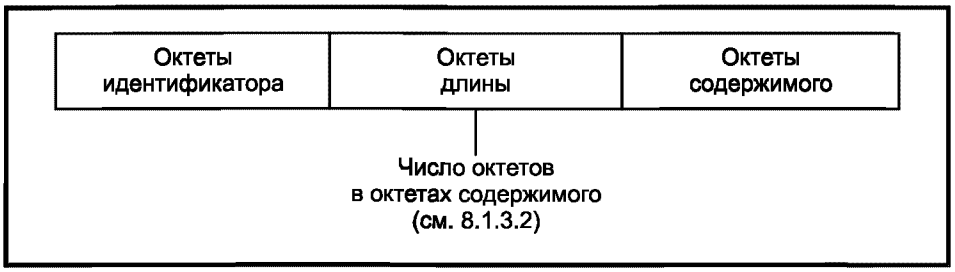

Рисунок 1 — Структура кодирования

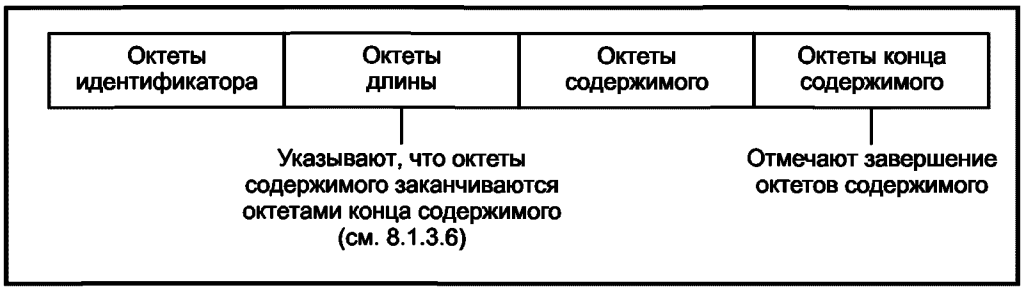

Рисунок 2 — Вариант составного кодирования

8.1.2 Октеты идентификатора

8.1.2.1 Октеты идентификатора должны кодировать тег АСН.1 (класс и номер) типа значения данных.

8.1.2.2 Для тегов с номером от 0 до 30 (включительно) октеты идентификатора должны содержать единственный октет, закодированный следующим образом:

а) биты 8 и 7 представляют класс тега и должны кодироваться так, как определено в таблице 1;

б) бит 6 должен быть нулем или единицей согласно правилам 8.1.2.5;

в) биты с 5 по 1 должны кодировать номер тега как двоичное целое число с битом 5 в качестве старшего значащего бита.

Таблица 1 — Кодирование класса тега

Класс	Бит 8	Бит 7
Универсальный	0	0
Прикладной	0	1
Контекстно зависимый	1	0
Пользовательский	1	1

8.1.2.3 На рисунке 3 показан вид октета идентификатора для типа с номером тега от 0 до 30 (включительно).

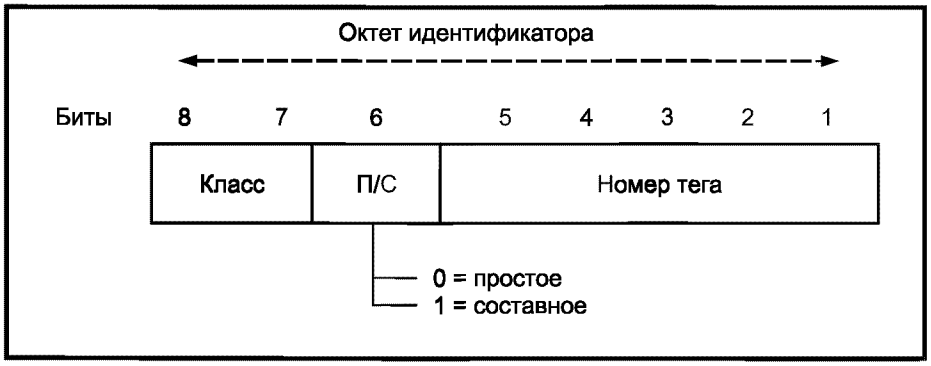

Рисунок 3 — Октет идентификатора (небольшой номер тега)

8.1.2.4 Для тегов с номерами, большими или равными 31, идентификатор должен состоять из головного октета, за которым следуют один или несколько октетов продолжения.

8.1.2.4.1 Головной октет должен быть закодирован следующим образом:

а) биты 8 и 7 представляют класс тега и должны быть закодированы так, как определено в таблице 1;

б) бит 6 должен быть нулем или единицей согласно правилам 8.1.2.5;

в) биты от 5 до 1 должны быть закодированы как 11111_2.

8.1.2.4.2 Последующие октеты должны кодировать номер тега следующим образом:

а) бит 8 каждого октета должен иметь значение 1, если он не является последним октетом идентификатора;

б) биты с 7 по 1 первого октета продолжения, за которыми следуют биты с 7 по 1 второго октета продолжения, за которыми, в свою очередь, следуют биты с 7 по 1 каждого следующего октета продолжения, до последнего, включительно, должны быть кодированием двоичного целого числа без знака, равного номеру тега, с битом 7 первого октета продолжения в качестве старшего значащего бита;

в) биты с 7 по 1 первого октета продолжения не должны быть все равны нулю.

8.1.2.4.3 На рисунке 4 показана форма октетов идентификатора для типа с тегом, номер которого больше 30.

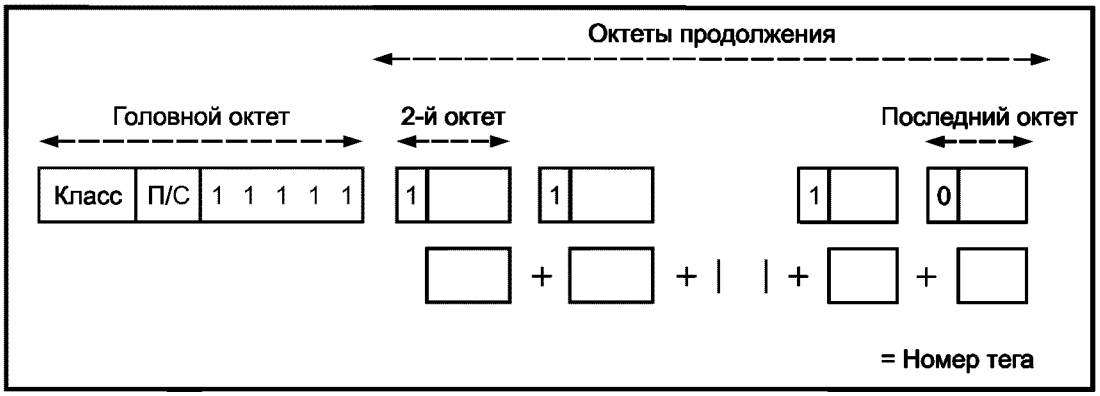

Рисунок 4 — Октет идентификатора (большой номер тега)

8.1.2.5 Бит 6 должен быть равен нулю, если кодирование простое, и единице, если кодирование составное.

Примечание — В последующих разделах для каждого типа определено, является его кодирование простым или составным.

8.1.2.6 В ГОСТ Р ИСО/МЭК 8824-1 установлено, что тег типа, определенного с использованием ключевого слова «CHOICE», принимает значение тега того типа, значение данных которого выбрано.

8.1.2.7 В ГОСТ Р ИСО/МЭК 8824-2, 14.2 и 14.4 установлено, что тег типа, определенного с использованием конструкции «ObjectClassFieldType», неопределен, если он является полем типа, полем значения переменного типа или полем множества значений переменного типа. Этот тип впоследствии определяется как тип АСН.1, и его полное кодирование идентично кодированию значения присвоенного типа (включая октеты идентификатора).

8.1.3 Октеты длины

8.1.3.1 Определены две формы октетов длины, а именно:

а) определенная форма (см. 8.1.3.3);

б) неопределенная форма (см. 8.1.3.6).

8.1.3.2 Отправитель должен использовать:

а) определенную форму (см. 8.1.3.3), если кодирование простое;

б) либо определенную (см. 8.1.3.3), либо неопределенные формы (см. 8.1.3.6), по своему выбору, если кодирование составное и непосредственно доступно целиком;

в) неопределенную форму (см. 8.1.3.6), если кодирование составное и непосредственно доступно не полностью.

8.1.3.3 Для определенной формы октеты длины должны состоять из одного или нескольких октетов и представлять число октетов содержимого, используя короткую (см. 8.1.3.4) или длинную форму (см. 8.1.3.5), по выбору отправителя.

Примечание — Короткая форма может быть использована только в том случае, если число октетов содержимого меньше или равно 127.

8.1.3.4 В короткой форме октеты длины должны состоять из одного октета, в котором бит 8 является нулевым, а биты с 7 по 1 кодируют число октетов содержимого (которое может быть нулевым) как двоичное целое число без знака с битом 7 в качестве старшего значащего бита.

Пример

L= 38 может быть закодирована как 00100110_2.

8.1.3.5 В длинной форме октеты длины должны состоять из начального октета и одного или нескольких последующих октетов. Начальный октет должен быть закодирован следующим образом:

а) 8 бит должен быть равен единице;

б) биты с 7 по 1 должны кодировать число последующих октетов длины как двоичное целое число без знака с битом 7 в качестве старшего значащего бита;

в) значение 11111111_2 не должно использоваться.

Примечание 1 — Это ограничение введено для возможного последующего расширения.

Биты с 8 по 1 первого октета продолжения с последующими битами с 8 по 1 второго октета продолжения, с последующими битами с 8 по 1 каждого следующего октета, включая последний октет продолжения, должны быть кодированием двоичного целого числа без знака, равного числу октетов содержимого, с битом 8 первого октета продолжения в качестве старшего значащего бита.

Пример

L = 201 может быть закодирована как:

$$10000001_2$$
$$11001001_2$$

Примечание 2 — В длинной форме отправитель может выбирать, использовать ли октетов длины больше, чем минимально необходимо.

8.1.3.6 Для неопределенной формы октеты длины указывают, что октеты содержимого заканчиваются октетами конец-содержимого (см. 8.1.5) и должны содержать единственный октет.

8.1.3.6.1 Единственный октет должен иметь 8 бит, равный единице, и биты с 7 по 1, равные нулю.

8.1.3.6.2 Если используется неопределенная форма длины, то октеты конец-содержимого (см. 8.1.5) должны присутствовать в кодировании после октетов содержимого.

8.1.4 Октеты содержимого

Октеты содержимого должны состоять из нуля, одного или нескольких октетов и кодировать значение данных так, как определено в последующих разделах.

Примечание — Октеты содержимого зависят от типа значения данных; последующие разделы расположены в том же порядке, что и определения типов в АСН.1.

8.1.5 Октеты конец-содержимого

Октеты конец-содержимого должны присутствовать, если длина закодирована так, как определено в 8.1.3.6, в противном случае они присутствовать не должны.

Октеты конец-содержимого должны состоять из двух нулевых октетов.

Примечание — Октеты конец-содержимого могут рассматриваться как кодирование значения с тегом универсального класса, форма которого является простой, номер тега — нулевой, а содержимое отсутствует. Таким образом:

Конец-содержимого	Длина	Содержимое
00_{16}	0016	Отсутствует

8.2 Кодирование булевского значения

8.2.1 Кодирование булевского значения должно быть простым. Октеты содержимого должны состоять из одного октета.

8.2.2 Если булевское значение есть FALSE («ложь»), то октет должен быть нулем. Если булевское значение TRUE («истина»), то октет должен иметь любое ненулевое значение по выбору отправителя.

Пример — Если тип BOOLEAN, то значение TRUE может быть закодировано как:

Булевский тип	Длина	Содержимое
01_{16}	01_{16}	FF_{16}

8.3 Кодирование целочисленного значения

8.3.1 Кодирование целочисленного значения должно быть простым. Октеты содержимого должны состоять из одного или нескольких октетов.

8.3.2 Если октеты содержимого кодирования целочисленного значения содержат более одного октета, то биты первого октета и 8 бит второго октета:

а) не должны все быть единицами;

б) не должны все быть нулевыми.

Примечание — Эти правила гарантируют, что целочисленное значение всегда кодируется в наименьшем возможном числе октетов.

8.3.3 Октеты содержимого должны быть дополнительным кодом двоичного числа, равного кодируемому целочисленному значению, образованным битами с 8 по 1 первого октета, с последующими битами с 8 по 1 второго октета, с последующими битами с 8 по 1 каждого очередного октета, включая последний октет содержимого.

Примечание — Значение дополнительного кода двоичного числа получается путем нумерации битов в октетах содержимого, начиная с бита 1 последнего октета как бита 0 и заканчивая нумерацию битом 8 первого октета. Каждому биту присваивается числовое значение 2^N, где N — номер бита в описанной выше нумерации. Значение дополнительного кода двоичного числа получается суммированием числовых значений, присвоенных каждому биту, тех битов, которые равны единице, исключая бит 8 первого октета, и последующего уменьшения этой суммы на числовое значение, присвоенное биту 8 первого октета, если тот бит равен единице.

8.4 Кодирование перечислимого значения

Кодирование перечислимого значения должно быть кодированием связанного с ним целочисленного значения.

Примечание — Кодирование является простым.

8.5 Кодирование действительного значения

8.5.1 Кодирование действительного значения должно быть простым.

8.5.2 Если действительное значение является нулевым, то в кодировании не должно быть никаких октетов содержимого.

8.5.3 Если действительное значение ненулевое, то используемое для кодирования основание B' выбирается отправителем. Если B' равно 2, 8 или 16, то должно использоваться двоичное кодирование, определенное в 8.5.5. Если B' равно 10, то должно использоваться символьное кодирование, определенное в 8.5.6.

Примечание — Форма хранения, создания или обработки отправителями и получателями и форма, используемая в нотации значения АСН.1, полностью не зависят от основания, используемого при передаче.

8.5.4 Бит 8 первого октета содержимого должен быть установлен следующим образом:

а) если бит 8 = 1, то применяется двоичное кодирование, определенное в 8.5.5;

б) если бит 8 = 0 и бит 7 = 0, то применяется десятичное кодирование, определенное в 8.5.6;

в) если бит 8 = 0, а бит 7 = 1, то «SpecialRealValue» (см. ГОСТ Р ИСО/МЭК 8824-1) кодируется так, как определено в 8.5.7.

8.5.5 При использовании двоичного кодирования (бит 8 = 1), если мантисса M ненулевая, то она должна быть представлена знаком S, неотрицательным целочисленным значением N и двоичным масштабным коэффициентом F, как то:

$$M = S \times N \times 2^F;$$

$$0 \le F < 4;$$

$S = +1$ или -1.

Примечание — Коэффициент F требуется в некоторых случаях для выравнивания подразумеваемой точки мантиссы к позиции, требуемой правилами кодирования настоящего раздела. Это выравнивание не всегда может быть достигнуто модификацией экспоненты E. Если основание B', используемое для кодирования, равно 8 или 16, то изменениями компонента E подразумеваемая точка может быть сдвинута только на 3 или 4 бита соответственно. Следовательно, для перемещения подразумеваемой точки в нужное положение могут потребоваться отличные от нуля значения коэффициента F.

8.5.5.1 Бит 7 первого октета содержимого должен быть равен 1, если S равен -1, в противном случае он равен 0.

8.5.5.2 Биты с 6 по 5 первого октета содержимого должны кодировать значение основания *B'* следующим образом:

Биты с 6 по 5	Основание
00	основание 2
01	основание 8
10	основание 16
11	зарезервировано для последующих редакций настоящего стандарта.

8.5.5.3 Биты с 4 по 3 первого октета содержимого должны кодировать значение двоичного масштабного коэффициента *F* как двоичное целое число без знака.

8.5.5.4 Биты со 2 по 1 первого октета содержимого должны кодировать формат экспоненты следующим образом: если биты со 2 по 1 равны:

а) 00, то второй октет содержимого кодирует значение экспоненты в виде дополнительного кода двоичного числа;

б) 01, то второй и третий октеты содержимого кодируют значение экспоненты в виде дополнительного кода двоичного числа;

в) 10, то второй, третий и четвертый октеты содержимого кодируют значение экспоненты в виде дополнительного кода двоичного числа;

г) 11, то второй октет содержимого кодирует (как двоичное число без знака) число октетов *X*, используемых для кодирования значения экспоненты, а октеты содержимого с третьего по $(X + 3)$-й включительно кодируют значение экспоненты в виде дополнительного кода двоичного числа; значение *X* должно быть не меньше единицы; первые девять битов переданной экспоненты не все должны быть нулевыми или единичными.

8.5.5.5 Остающиеся октеты содержимого кодируют значение целого числа *N* (см. 8.5.5) в виде двоичного числа без знака.

Примечания

1 Для неканонических BER не требуется нормализация мантиссы с «плавающей» точкой. Это позволяет реализатору передавать октеты, содержащие мантиссу, без выполнения функций сдвига мантиссы в памяти. В канонических и отличительных правилах кодирования нормализация определена, и мантисса (если она не 0) должна сдвигаться до тех пор, пока наименьший значащий бит не станет равным 1.

2 Это представление действительных чисел сильно отличается от форматов, обычно используемых в аппаратуре с «плавающей» точкой, но оно предназначено для легкого преобразования в такие форматы и из таких форматов (см. приложение С).

8.5.6 Когда используется десятичное кодирование (биты с 8 по 7 = 00), все октеты содержимого после первого образуют поле, в смысле ИСО 6093, выбранной отправителем длины, закодированное в соответствии с ИСО 6093. Выбор представления числа по ИСО 6093 определяется битами с 6 по 1 первого октета содержимого следующим образом:

Биты с 6 по 1	Представление числа
000001	ИСО 6093, формат NR1
000010	ИСО 6093, формат NR2
000011	ИСО 6093, формат NR3

Остальные значения битов с 6 по 1 зарезервированы для настоящего стандарта.

Не должны использоваться масштабные коэффициенты, определенные в сопровождающей документации (см. ИСО 6093).

Примечания

1 Рекомендации ИСО 6093 относительно использования, по крайней мере, одной цифры слева от десятичного знака сохраняются и в настоящем стандарте, но не являются обязательными.

2 Использование нормированной формы (см. ИСО 6093) не существенно и остается на усмотрение отправителя.

8.5.7 Когда должны быть закодированы «SpecialRealValues» (биты с 8 по 7 = 01), то должен быть только один октет содержимого со следующими значениями:

01000000 значение равно PLUS-INFINITY;

01000001 значение равно MINUS-INFINITY.

Все другие значения, имеющие биты 8 и 7, равные 0 и 1 соответственно, зарезервированы для дополнений к настоящему стандарту.

8.6 Кодирование значения «битовая строка»

8.6.1 Кодирование значения «битовая строка» должно быть простым или составным, по усмотрению отправителя.

Примечание — Если необходимо передать часть битовой строки до того, как она вся станет доступной, то используется составное кодирование.

8.6.2 Октеты содержимого для простого кодирования должны содержать начальный октет, с последующим нулем, одним или несколькими октетами продолжения.

8.6.2.1 Биты в битовой строке, с первого до последнего, должны размещаться в битах с 8 до 1 первого октета продолжения, далее — в битах с 8 до 1 второго октета продолжения, затем — в битах с 8 до 1 каждого следующего октета и завершаться тем количеством битов, которое необходимо в конечном октете продолжения, начиная с бита 8.

Примечание — Термины «первый бит» и «завершающий бит» определены в ГОСТ Р ИСО/МЭК 8824-1.

8.6.2.2 Начальный октет должен кодировать, в виде двоичного целого числа без знака с битом 1 в качестве наименьшего значащего бита, число неиспользованных битов в конечном октете продолжения. Число должно находиться в диапазоне от нуля до семи.

8.6.2.3 Если битовая строка пуста, то не должно быть никаких октетов продолжения, а начальный октет должен быть нулевым.

8.6.2.4 При применении подраздела 21.7 ГОСТ Р ИСО/МЭК 8824-1 кодировщики/декодировщики BER могут добавлять или убирать завершающие нулевые биты значения.

Примечание — Если значение битовой строки не имеет единичных битов, то кодировщик (по усмотрению отправителя) может кодировать значение с начальным октетом, равным 0, или может кодировать его как битовую строку с одним или несколькими нулевыми битами после начального октета.

8.6.3 Октеты содержимого для составного кодирования должны состоять из нуля, одного или нескольких вложенных кодирований.

Примечание — Каждое такое кодирование включает в себя октеты идентификатора, длины, содержимого и может включать октеты конец-содержимого, если оно является составным.

8.6.4 Для кодирования значения битовой строки таким способом это значение должно быть сегментировано. Каждый сегмент должен состоять из ряда последовательных битов значения и должен, возможно, за исключением последнего, содержать число битов, кратное восьми. Каждый бит в общем значении должен присутствовать ровно в одном сегменте, а размещение границ сегментов значения не имеет.

Примечание — Сегмент может иметь нулевой размер, то есть не содержать биты.

8.6.4.1 Каждое кодирование в октетах содержимого должно представлять сегмент общей битовой строки, кодовое представление которой является результатом рекурсивного применения настоящего раздела. При этом применении каждый сегмент рассматривается как значение битовой строки. Кодовые представления сегментов должны появляться в октетах содержимого в том порядке, в каком их биты появляются в общем значении.

Примечания

1 Как следствие этой рекурсии, каждое кодирование в октетах содержимого само может быть простым или составным. Однако обычно такие кодирования являются простыми.

2 В частности, теги в октетах содержимого всегда универсального класса номер 3.

8.6.4.2 Пример. Если тип BIT STRING, то его значение '0A3B5F291CD'H может быть закодировано так, как показано ниже. В данном примере BitString представлена как примитив:

BitString	Длина	Содержимое
03_{16}	07_{16}	$040A3B5F291CD0_{16}$

Это же значение может быть закодировано так, как показано ниже. В данном примере BitString представлена как конструкция:

BitString	Длина	Содержимое		
23_{16}	80_{16}	BitString	Длина	Содержимое
E0C	Длина	03_{16}	03_{16}	$000A3B_{16}$
00_{16}	00_{16}	03_{16}	05_{16}	$045F291CD0_{16}$

8.7 Кодирование значения «строка октетов»

8.7.1 Кодирование значения «строка октетов» должно быть простым или составным, по усмотрению отправителя.

Примечание — Если необходимо передать часть строки октетов до того, как станет доступна вся строка, то используется составное кодирование.

8.7.2 Простое кодирование содержит нуль, один или несколько октетов содержимого, равных по значению октетам в значении данных, в порядке их появления в значении данных, и со старшим значащим битом октета значения данных, выровненным по старшему значащему биту октета содержимого.

8.7.3 Октеты содержимого для составного кодирования должны состоять из нуля, одного или нескольких кодирований.

Примечание — Каждое такое кодирование включает в себя октеты идентификатора, длины, содержимого и может включать октеты конец-содержимого, если оно является составным.

8.7.3.1 Для кодирования значения «строка октетов» таким способом это значение должно быть сегментировано. Каждый сегмент должен состоять из ряда последовательных октетов значения. Расположение границ сегментов значения не имеет.

Примечание — Сегмент может иметь нулевой размер, то есть не содержать октеты.

8.7.3.2 Каждое кодирование в октетах содержимого должно представлять сегмент общей строки октетов, кодовое представление которой является результатом рекурсивного применения настоящего раздела. При этом применении каждый сегмент рассматривается как значение строки октетов. Кодовые представления сегментов должны появляться в октетах содержимого в том порядке, в каком их октеты появляются в общем значении.

Примечания

1 Как следствие этой рекурсии, каждое кодирование в октетах содержимого само может быть простым или составным. Однако обычно такие кодирования являются простыми.

2 В частности, теги в октетах содержимого всегда универсального класса номер 4.

8.8 Кодирование вырожденного значения

8.8.1 Кодирование вырожденного значения должно быть простым.

8.8.2 Октеты содержимого не должны содержать октетов.

Примечание — Октет длины равен нулю.

Пример. Если тип NULL, то NULL может быть закодирован как:

NULL	Длина
05_{16}	00_{16}

8.9 Кодирование значения «последовательность»

8.9.1 Кодирование значения «последовательность» должно быть составным.

8.9.2 Октеты содержимого должны состоять из полного кодирования одного значения данных для каждого из типов, перечисленных в определении АСН.1 типа «последовательность», в порядке их появления в определении, если только тип не был указан с ключевым словом «OPTIONAL» или «DEFAULT».

8.9.3 Кодирование значения данных может, но не обязательно, присутствовать для типа, указанного с ключевым словом «OPTIONAL» или «DEFAULT». Если оно присутствует, то должно появляться в кодировании в точке, соответствующей появлению типа в определении АСН.1.

Пример. Если тип

SEQUENCE {name IA5String, ok BOOLEAN},

то значение

{name «Smith», ok TRUE}

может быть закодировано как:

Последовательность	Длина	Содержимое		
30_{16}	$0A_{16}$			
		IA5String	Длина	Содержимое
		16_{16}	05_{16}	«Smith»
		BOOLEAN	Длина	Содержимое
		01_{16}	01_{16}	FF_{16}

8.10 Кодирование значения «последовательность-из»

8.10.1 Кодирование значения «последовательность-из» должно быть составным.

8.10.2 Октеты содержимого должны состоять из нуля, одного или нескольких полных кодирований значений данных для типа, перечисленного в определении АСН.1.

8.10.3 Порядок кодированных значений данных должен быть таким же, что и порядок значений данных в кодируемом значении «последовательность-из».

8.11 Кодирование значения «множество»

8.11.1 Кодирование значения «множество» должно быть составным.

8.11.2 Октеты содержимого должны состоять из полного кодирования значения данных для каждого из типов, перечисленных в определении АСН.1 типа «множество», в порядке, выбранном отправителем, если тип не был указан с ключевым словом «OPTIONAL» или «DEFAULT».

8.11.3 Кодирование значения данных может, но не обязательно, присутствовать для типа, который был указан с ключевым словом «OPTIONAL» или «DEFAULT».

Примечание — Порядок значений данных в значении «множество» не существенен и не устанавливает ограничений на порядок во время передачи.

8.12 Кодирование значения «множество-из»

8.12.1 Кодирование «множество-из» должно быть составным.

8.12.2 Применяется правило 8.10.2.

8.12.3 Порядок значений данных не должен сохраняться при кодировании и последующем декодировании.

8.13 Кодирование выборочного значения

Кодирование выборочного значения должно быть таким же, как кодирование значения выбранного типа.

Примечания

1 Кодирование может быть простым или составным в зависимости от выбранного типа.

2 Тег, используемый в октетах идентификатора, является тегом выбранного типа, специфицированного в определении выборочного типа АСН.1

8.14 Кодирование тегированного значения

8.14.1 Кодирование тегированного значения должно быть получено из полного кодирования соответствующего значения данных типа, появляющегося в нотации «TaggedType» (называемого базовым кодированием), как определено в 8.14.2 и 8.14.3.

8.14.2 Если в определении типа не используется неявное тегирование (см. ГОСТ Р ИСО/МЭК 8824-1, 28.6), то кодирование должно быть составным, и октеты содержимого должны быть полным базовым кодированием.

8.14.3 Если в определении типа использовалось неявное тегирование, то:

а) кодирование должно быть составным, если базовое кодирование составное, в противном случае оно должно быть простым;

б) октеты содержимого должны быть такими же, как октеты содержимого базового кодирования.

Пример. С определениями типов АСН.1 (в среде явного тегирования)

```
Type1 :: = VisibleString
Type2 :: = [APPLICATION 3] IMPLICIT Type1
Type3 :: = [2] Type2
Type4 :: = [APPLICATION 7] IMPLICIT Type3
Type5 :: = [2] IMPLICIT Type2
```

значение

«Jones»

кодируется следующим образом:

Для Type1:		
VisibleString	Длина	Содержимое
$1A_{16}$	05_{16}	$4A6F6E6573_{16}$
Для Type2:		
[APPLICATION 3]	Длина	Содержимое
43_{16}	05_{16}	$4A6F6E6573_{16}$

Для Type3:

[2]	Длина	Содержимое		
$A2_{16}$	07_{16}			
		[APPLICATION 3]	Длина	Содержимое
		43_{16}	05_{16}	$4A6F6E6573_{16}$

Для Type4:

[APPLICATION 7]	Длина	Содержимое		
67_{16}	07_{16}			
		[APPLICATION 3]	Длина	Содержимое
		43_{16}	05_{16}	$4A6F6E6573_{16}$

Для Type5:

[2]	Длина	Содержимое
82_{16}	05_{16}	$4A6F6E6573_{16}$

8.15 Кодирование открытого типа

Значение открытого типа является также значением некоторого (другого) типа АСН.1. Кодирование такого значения должно быть полным кодированием, специфицированным для этого другого типа.

8.16 Кодирование значения «экземпляр-из»

8.16.1 Кодирование типа «экземпляр-из» должно быть кодированием BER следующего типа «последовательность» со значением, определенным в 8.16.2:

```
[UNIVERSAL 8] IMPLICIT SEQUENCE
{
        type-id <DefinedObjectClass>.&id,
        value    [0] EXPLICIT <DefinedObjectClass>. &Type
}
```

где «<DefinedObjectClass>» замещается конкретным классом «DefinedObjectClass», использованным в нотации «InstanceOfType».

Примечание — Когда значение является значением единственного типа АСН.1 и для него используется кодирование BER, то кодирование этого данного типа идентично кодированию соответствующего значения внешнего типа, где для представления абстрактного значения используется альтернатива «syntax».

8.16.2 Значение компонентов типа «последовательность» в 8.16.1 должно быть таким же, как значения соответствующих компонентов ассоциированного типа в ГОСТ Р ИСО/МЭК 8824-2, приложение С.7.

8.17 Кодирование значения типа «встроенное-здп»

8.17.1 Кодирование значения типа «встроенное-здп» должно быть кодированием BER типа, определенного в ГОСТ Р ИСО/МЭК 8824-1.

8.17.2 Содержимое «data-value» OCTET STRING должно быть кодированием значения абстрактных данных типа «встроенное-здп» (см. ГОСТ Р ИСО/МЭК 8824-1, 32.3а), использующим идентифицированный синтаксис передачи, значения всех других полей должны быть теми же, что и значения в абстрактном значении.

8.18 Кодирование значения внешнего типа

8.18.1 Кодирование значения внешнего типа должно быть кодированием BER следующего типа «последовательность», принимаемого как определенного в контексте EXPLICIT TAGS (явные теги), со значением, определенным в последующих подразделах:

```
[UNIVERSAL 8] IMPLICIT SEQUENCE {
        direct-reference            OBJECT IDENTIFIER OPTIONAL,
        indirect-reference          INTEGER OPTIONAL,
        data-value-descriptor       ObjectDescriptor OPTIONAL,
        encoding                    CHOICE {
            single-ASN1-type            [0] ABSTRACT-SYNTAX. &Type,
            octet-aligned               [1] IMPLICIT OCTET STRING,
            arbitrary                   [2] IMPLICIT BIT STRING} }
```

Примечание — Этот тип «последовательность» является тем же, который был определен в ГОСТ Р ИСО/МЭК 8824—93, и результирующее кодирование значения внешнего типа не изменяется.

8.18.2 Значение полей зависит от передаваемых абстрактных значений, и является значением типа, определенного в 32.5 ГОСТ Р ИСО/МЭК 8824-1.

8.18.3 Компонент «data-value-descriptor» должен присутствовать только в том случае, если «data-value-descriptor» присутствует в абстрактном значении, и должен иметь то же значение.

8.18.4 Значения «direct-reference» и «indirect-reference» должны присутствовать или отсутствовать согласно таблице 2. Таблица 2 отображает альтернативы «identification» внешнего типа, приведенные в ГОСТ Р ИСО/МЭК 8824-1, 32.5, в определенные в 8.18.1 компоненты внешнего типа «direct-reference» и «indirect-reference».

8.18.5 Значение данных должно быть закодировано в соответствии с синтаксисом передачи, идентифицированного кодированием, и должно быть размещено в альтернативе «encoding», выбранной, как определено ниже.

Таблица 2 — Альтернативные кодирования для «identification»

identification	direct-reference	indirect-reference
syntaxes	***НЕДОПУСТИМО***	***НЕДОПУСТИМО***
syntax	syntax	ОТСУТСТВУЕТ
presentation-context-id	ОТСУТСТВУЕТ	presentation-context-id
context-negotiation	transfer-syntax	presentation-context-id
transfer-syntax	***НЕДОПУСТИМО***	***НЕДОПУСТИМО***
fixed	***НЕДОПУСТИМО***	***НЕДОПУСТИМО***

8.18.6 Если значение данных является значением единственного типа данных АСН.1 и если правила кодирования для этого значения данных определены в настоящем стандарте, то реализация отправителя должна использовать, по своему усмотрению, любой из следующих выборов для «Encoding»:

- single-ASN.1-type;
- octet-aligned;
- arbitrary.

8.18.7 Если кодированное значение данных, использующее согласованное кодирование, содержит целое число октетов, то реализация отправителя должна использовать, по своему усмотрению, любой из следующих выборов для «Encoding»:

- octet-aligned;
- arbitrary.

Примечание — Значение данных, которое является последовательностью типов АСН.1 и для которого синтаксис передачи специфицирует простое сцепление строк октетов, созданных применением базовых правил кодирования АСН.1 для каждого типа АСН.1, попадает в эту категорию, а не в категорию 8.18.6.

8.18.8 Если кодированное значение данных, использующее согласованное кодирование, содержит не целое число октетов, то для «Encoding» должно быть выбрано.

- arbitrary.

8.18.9 Если для «Encoding» выбрано «single-ASN1-type», то тип АСН.1 должен заменить открытый тип со значением, равным значению кодируемых данных.

Примечание — Диапазон значений, которые могут встретиться в открытом типе, определяется регистрацией значения идентификатора объекта, ассоциированного с «direct-reference», и/или значением целого числа, ассоциированного с «indirect-reference».

8.18.10 Если для «Encoding» выбрано «octet-aligned», то значение данных должно кодироваться в соответствии с согласованным синтаксисом передачи, а получающиеся октеты должны образовывать значение «строка октетов».

8.18.11 Если для «Encoding» выбрано «arbitrary», то значение данных должно кодироваться в соответствии с согласованным синтаксисом передачи, а получающиеся октеты должны образовывать значение «битовая строка».

8.19 Кодирование значения «идентификатор объекта»

8.19.1 Кодирование значения «идентификатор объекта» должно быть простым.

8.19.2 Содержимое октетов должно быть (упорядоченным) списком кодирований подидентификаторов (см. 8.19.3 и 8.19.4), соединенных вместе.

Каждый подидентификатор представляется как последовательность (одного или нескольких) октетов. Бит 8 каждого октета указывает, является ли он последним в последовательности: бит 8 в последнем октете — нулевой; бит 8 в каждом предшествующем октете — единица. Биты 7—1 всех октетов в последовательности вместе кодируют подидентификатор. Сцепляясь, эти группы бит образуют двоичное число без знака, наиболее значащий бит которого является битом 7 первого октета, а наименее значащий бит — битом 1 последнего октета. Подидентификатор должен быть закодирован в минимально возможном количестве октетов, то есть головной октет подидентификатора не должен иметь значение 80_{16}.

8.19.3 Число подидентификаторов (N) должно быть на единицу меньше, чем число компонентов идентификатора объекта в кодируемом значении.

8.19.4 Числовое значение первого подидентификатора получается из значений первых двух компонентов идентификатора объекта в кодируемом значении по формуле

$$(X*40) + Y,$$

где X — значение первого компонента идентификатора объекта, а Y — второго.

Примечание — Эта упаковка первых двух компонентов идентификатора объекта учитывает, что из корневого узла присвоены только три значения.

8.19.5 Числовое значение i-го подидентификатора ($2 \le i \le N$) является ($i + 1$)-м компонентом идентификатора объекта.

Пример. Значение OBJECT IDENTIFIER

{joint-iso-ccitt 100 3}

или, что тоже,

{2 100 3}

имеет первый подидентификатор 180 и второй подидентификатор 3. Получается следующее кодирование:

OBJECT IDENTIFIER

	Длина	Содержимое
06_{16}	03_{16}	813403_{16}

8.19bis Кодирование значения относительного идентификатора объекта

Примечание — Кодирование компонентов идентификатора объекта в относительном идентификаторе объекта такое же, как и кодирование компонентов (после второго) в идентификаторе объекта.

8.19bis.1 Кодирование значения относительного идентификатора объекта должно быть простым.

8.19bis.2 Октеты содержимого должны быть упорядоченным списком сцепленных кодирований подидентификаторов (см. 8.19bis.3, 8.19bis.4). Каждый подидентификатор представляется как серия (из одного или нескольких) октетов. Бит 8 каждого октета указывает, является ли этот октет последним в серии: бит 8 последнего октета равен нулю, бит 8 каждого предшествующего октета равен единице. Биты 7—1 октетов в серии кодируют подидентификатор. Концептуально эти группы битов сцеплены так, что образуют целое двоичное число без знака, старшим значащим битом которого является бит 7 первого октета, а младшим — бит 1 последнего октета. Подидентификатор должен быть закодирован в минимально возможном количестве октетов, т. е. первый октет подидентификатора не должен иметь значение 80^{16}.

8.19bis.3 Количество подидентификаторов (N) должно быть равно количеству дуг идентификатора объекта в кодируемом значении относительного идентификатора объекта.

8.19bis.4 Численное значение i-го подидентификатора ($1 \le i \le N$) должно быть значением i-й дуги идентификатора объекта в кодируемом значении относительного идентификатора объекта.

8.19bis.5 Пример. Значение относительного идентификатора объекта

{8571 3 2}

имеет подидентификаторы 8571, 3 и 2. Для него получается следующее кодирование:

Относительный идентификатор объекта	Длина	Содержимое
OD^{16}	04^{16}	$C27B0302^{16}$

8.20 Кодирование значений ограниченных типов символьных строк

8.20.1 Значение данных состоит из строки символов из набора символов, специфицированного в определении типа ACH.1.

8.20.2 Каждое значение данных должно быть закодировано независимо от других значений данных того же типа.

8.20.3 Каждая символьная строка должна быть закодирована так, как если бы она была объявлена

[UNIVERSALx] IMPLICIT OCTET STRING,

где x — номер тега универсального класса, присвоенный типу символьной строки в ГОСТ Р ИСО/МЭК 8824-1. Значение строки октетов специфицируется в 8.20.4 и 8.20.5.

8.20.4 Когда символьная строка специфицирована в ГОСТ Р ИСО/МЭК 8824-1 прямой ссылкой на таблицу перечислений (NumericString и PrintableString), значение строки октетов должно быть таким, как установлено в 8.20.5 для типа VisibleString с тем же значением символьной строки.

8.20.5 Для ограниченных символьных строк, кроме UniversalString и BMPString, строка содержит октеты, определенные в ИСО/МЭК 2022 для кодирования в 8-битном контексте, используя управляющую последовательность и кодирования символов, зарегистрированные в соответствии с ИСО 2375.

8.20.5.1 Может использоваться только та управляющая последовательность, которая специфицирована одним из регистрационных номеров, используемых для определения типа строки символов в ГОСТ Р ИСО/МЭК 8824-1.

8.20.5.2 В начале каждой строки должны быть присвоены определенные регистрационные номера для обозначения и вызова G0, и/или C0, и/или C1 (используя терминологию ИСО/МЭК 2022). Для каждого типа эти номера установлены в таблице 3 вместе с подразумеваемыми ими управляющими последовательностями.

8.20.5.3 Некоторые типы символьных строк не должны содержать явных управляющих последовательностей в своих кодированиях; во всех остальных случаях любая управляющая последовательность, разрешенная 8.20.5.1, может появиться в любое время, включая начало кодирования. В таблице 3 перечислены типы, для которых допускаются явные управляющие последовательности.

8.20.5.4 Объявления не должны использоваться, если они явно не разрешены пользователем ACH.1.

Примечание — Выбор типа ACH.1 предоставляет ограниченные возможности функций объявлений. Конкретные прикладные протоколы могут передавать объявления в других элементах протокола или подробно указывать способ использования объявлений.

Таблица 3 — Использование управляющих последовательностей

Тип	Присвоенный G0 (регистрационный номер)	Присвоенный C0 или C1 (регистрационный номер)	Присвоенные управляющие последовательности и фиксация регистра (где применимо)	Явные последовательности разрешены?
NumericString	6	Нет	ESC 2/8 4/2 LS0	Нет
PrintableString	6	Нет	ESC 2/8 4/2 LS0	Нет
TeletexString (T61String)	102	106 (C0) 107 (C1)	ESC 2/8 7/5 LS0 ESC 2/1 4/5 ESC 2/2 4/8	Да
VideotexString	2	1 (C0) 73 (C1)	ESC 2/8 7/5 LS0 ESC 2/1 4/0 ESC 2/2 4/1	Да
VisibleString (IS0646String)	6	Нет	ESC 2/8 4/2 LS0	Нет
IA5String	6	1 (C0)	ESC 2/8 4/2 LS0 ESC 2/1 4/0	Нет
GraphicString	6	Нет	ESC 2/8 4/2 LS0	Да
GeneralString	6	1 (C0)	ESC 2/8 4/2 LS0 ESC 2/1 4/0	Да

Примечание — Многие из обычно используемых символов (например, A—Z) появляются в ряде символьных репертуаров с индивидуальными регистрационными номерами и управляющими последовательностями. Когда типы ACH.1 допускают управляющие последовательности, возможно несколько кодирований для конкретной символьной строки (см. также 7.3).

Пример. С определением типа ACH.1

Name :: = VisibleString

значение

«Jones»

может быть закодировано (простая форма) как

VisibleString	Длина	Содержимое
$1A_{16}$	05_{16}	$4A6F6E6573_{16}$

или (составная форма, определенная длина) как

VisibleString	Длина	Содержимое		
$3A_{16}$	09_{16}			
		OctetString	Длина	Содержимое
		04_{16}	03_{16}	$4A6F6E_{16}$
		OctetString	Длина	Содержимое
		04_{16}	023_{16}	6573_{16}

или (составная форма, неопределенная длина) как

VisibleString	Длина	Содержимое		
$3A_{16}$	80_{16}			
		OctetString	Длина	Содержимое
		04_{16}	03_{16}	$4A6F6E_{16}$
		OctetString	Длина	Содержимое
		04_{16}	023_{16}	6573_{16}
		EOC	Длина	
		00_{16}	00_{16}	

8.20.6 Приведенный пример иллюстрирует три из (многих) возможных форм (по усмотрению отправителя). Получатели обязаны обрабатывать все допустимые формы (см. 7.3).

8.20.7 Для типа «UniversalString» строка октетов должна содержать октеты, определенные в ИСО/МЭК 10646-1, используя четырехоктетную каноническую форму (см. 14.2 ИСО/МЭК 10646-1). Сигнатуры использоваться не должны. Управляющие функции могут использоваться при условии их соответствия ограничениям 8.20.9.

8.20.8 Для типа «BMPString» строка октетов должна содержать октеты, определенные в ИСО/МЭК 10646-1, используя двухоктетную BMP форму (см. 14.2 ИСО/МЭК 10646-1). Сигнатуры использоваться не должны. Управляющие функции могут использоваться при условии их соответствия ограничениям 8.20.9.

8.20.9 Управляющие функции C0 и C1 по ГОСТ 34.301 могут использоваться за следующими исключениями.

Примечания

1 Цель настоящего подраздела — разрешить полезные управляющие функции, такие как LF, CR, TAB и т. д., но запретить использование переходов к другим наборам символов.

2 Управляющие функции C0 и C1 кодируются в два октета для BMPString и в четыре — для UniversalString.

а) Не должны использоваться управляющие последовательности объявления, определенные в ИСО/МЭК 2022.

Примечание 3 — Принят контекст кодирования символов по ИСО/МЭК 10646-1.

б) Не должны использоваться назначающие и идентифицирующие управляющие последовательности, определенные в ИСО/МЭК 2022, включая идентифицирующие управляющие последовательности, разрешенные ИСО/МЭК 10646-1, 17.2 17.4.

Примечание 4 — ACH.1 позволяет использовать нотацию подтипа PermittedAlphabet для выбора набора разрешенных символов. PermittedAlphabet используется также для выбора уровня реализации по ИСО/МЭК 10646-1. Строка BMPString всегда использует двухоктетную форму, а UniversalString — четырехоктетную.

в) Не должны использоваться вызывающие или контролирующие управляющие последовательности ИСО/МЭК 2022, такие как SHIFT IN (S1), SHIFT OUT (SO) или LOCKING SHIFT FOR G3 (SS3).

г) Кодирование должно соответствовать ИСО/МЭК 106460-1 и оставаться в том же кодовом наборе.

д) Не должны использоваться управляющие последовательности для идентификации графических символов по ИСО/МЭК 10646-1, 17.3.

Примечание 5 — Приложения ACH.1 используют подтипы для указания подмножеств графических символов по ИСО/МЭК 10646-1 и выбора ячеек по ИСО/МЭК 10646-1, которые соответствуют управляющим символам ГОСТ 34.301.

е) Не должны использоваться управляющие последовательности по ИСО/МЭК 10646-1, 17.5 для переключения на коды ИСО/МЭК 2022.

8.21 Кодирование значений неограниченного типа символьных строк

8.21.1 Кодирование значения неограниченного типа символьных строк должно быть кодированием BER типа, определенного в ГОСТ Р ИСО/МЭК 8824-1, 39.5.

8.21.2 Содержимое «string-value» OCTET STRING должно быть кодированием значения абстрактных данных неограниченного типа символьных строк (см. ГОСТ Р ИСО/МЭК 8824-1, 39.3а), использующим идентифицированный синтаксис передачи, значения всех других полей должны быть теми же, что и в абстрактном значении.

8.22 Следующие «полезные типы» должны кодироваться так, как если бы они были заменены определениями, приведенными в разделах 41—43 ГОСТ Р ИСО/МЭК 8824-1:

- обобщенное время,
- универсальное время,
- описатель объекта.

9 Канонические правила кодирования

Кодирование значений данных по каноническим правилам кодирования является базовым кодированием, описанным в разделе 8, с ограничениями, перечисленными в настоящем разделе.

9.1 Формы длины

Если кодирование составное, то в нем используется неопределенная форма длины. Если кодирование простое, то оно должно включать в себя наименьшие октеты длины (см. для сравнения 8.1.3.2б).

9.2 Формы кодирования строк

Значения строк битов, октетов и ограниченных символов должны кодироваться простым кодированием, если они требуют не более 1000 октетов содержимого, и составным кодированием — в противном случае. Фрагменты строк, содержащиеся в составном кодировании, должны кодироваться простым кодированием. Кодирование каждого фрагмента, за исключением, может быть, последнего, должно иметь 1000 октетов содержимого (см. для сравнения 8.20.6).

9.3 Компоненты множества

Кодирования значений компонентов множества должны появляться в порядке, определяемом их тегами, как установлено в ГОСТ Р ИСО/МЭК 8824-1, 6.4. Кроме того, для установления порядка кодирования компонентов, когда один или несколько из них являются нетегированными выборочными типами, каждый нетегированный выборочный тип упорядочивается так, как если бы имел тег, равный наименьшему тегу в этом выборочном типе или в любом вложенном нетегированном выборочном типе.

Пример принятия контекста тегирования IMPLICIT TAGS:

```
A :: = SET
{
    a [3] INTRGER,
    b [1] CHOICE
    {
      c [2] INTRGER,
      d [4] INTRGER
    },
    c CHOICE
    {
      f CHOICE
      {
```

```
            g [5] INTRGER,
            h [6] INTRGER
        },
        i CHOICE
        {
            j [0] INTEGER
        }
    }
}
```

Порядок, в котором кодируются компоненты множества, всегда следующий: e, b, a, так как тег [0] меньше, чем [1] и [3].

10 Отличительные правила кодирования

Кодирование значений данных по отличительным правилам кодирования является базовым, описанным в разделе 8, с ограничениями, перечисленными в настоящем разделе.

10.1 Формы длины

Должна использоваться определенная форма длины кодирования, закодированная в минимальном числе октетов (см. для сравнения 8.1.3.2б).

10.2 Формы кодирования строк

Для типов строк битов, октетов и ограниченных символов не должно использоваться составное кодирование (см. для сравнения 8.1.3.2б).

10.3 Набор компонентов

Кодирования значений компонентов множества должны появляться в порядке, определяемом их тегами, как установлено в ГОСТ Р ИСО/МЭК 8824-1, 6.4.

Примечание — Когда компонент множества является нетегированным выборочным типом, положение компонента в этом порядке зависит от тега кодируемого выбранного компонента.

11 Ограничения на BER, использующие CER и DER

Указание раздела 8 «должно быть кодирование BER» интерпретируется как «должно быть кодирование CER или DER». (См. 8.16.1 и 8.18.1)

11.1 Булевские значения

Если кодирование представляет булевское значение TRUE, то его единственный октет содержимого должен иметь все восемь битов, равные единице (см. для сравнения 8.2.2).

11.2 Неиспользованные биты

11.2.1 Каждый неиспользованный бит в последнем октете кодирования значения битовой строки должен быть равен нулю.

11.2.2 При применении ГОСТ Р ИСО/МЭК 8824-1, 21.7 из битовой строки перед ее кодированием следует удалить все завершающие 0.

Примечания

1 В случае, когда применяется ограничение размера, абстрактное значение, доставляемое декодером приложению, должно удовлетворять ограничению размера и может отличаться от переданного значения только количеством завершающих битов, равных 0.

2 Если битовая строка не имеет битов, равных 1, то кодировщик должен закодировать значение длиной 1 и начальным октетом, равным 0.

11.3 Действительные значения

11.3.1 Если кодирование представляет значение с основанием B, равным 2, то должно использоваться двоичное кодирование по основанию 2. Перед кодированием мантисса M и экспонента E выбираются так, что M равна 0 или является четной.

Примечание — Это необходимо, так как одно и тоже действительное значение может рассматриваться и как $\{M, 2, E\}$, и как $\{M', 2, E'\}$ с $M \neq M'$, если для некоторого ненулевого целого n:

$$M' = M \times 2^{-n},$$
$$E' = E + n.$$

При кодировании двоичный масштабирующий фактор F должен быть равен нулю, а M и E должны быть представлены наименьшим возможным числом октетов.

11.3.2 Если кодирование представляет значение с основанием *B*, равным 10, то должно использоваться десятичное кодирование. При этом применяются следующие правила.

11.3.2.1 Должна использоваться форма ИСО 6093 NR3 (см. 8.5.6).

11.3.2.2 В кодировании не должен использоваться символ SPACE.

11.3.2.3 Если действительное значение отрицательно, то оно должно начинаться с символа MINUS SIGN (—), в противном случае — с цифры.

11.3.2.4 Первая и последняя цифры мантиссы не должны быть равны 0.

11.3.2.5 Непосредственно за последней цифрой мантиссы должен следовать символ FULL STOP (.) и знак экспоненты *E*.

11.3.2.6 Если экспонента имеет значение 0, то оно должно быть записано как «+0», в противном случае первая цифра экспоненты не должна быть нулем и знак PLUS SING не должен использоваться.

11.4 Значения GeneralString

Кодирование значений типа GeneralString (и его подтипов) должно порождать управляющие последовательности для назначения и вызова нового регистра только тогда, когда регистр отличается от назначенного в текущий момент как G0, C0 или C1. Все назначения и вызовы должны относиться к набору G0 или C0.

Примечание — Принято, что каждый символ в значении символьной строки ассоциирован с конкретной записью в Международном регистре кодированных наборов символов.

11.5 Компоненты множества и последовательности с принимаемыми по умолчанию значениями

Кодирование значения множества или последовательности не должно содержать кодирования компонента, значение которого равно значению, принимаемому по умолчанию.

11.6 Компоненты «множество-из»

Кодирования значений компонентов значения «множество-из» должны появляться в возрастающем порядке, если рассматривать эти кодирования как строки октетов с короткими компонентами, дополненными нулевыми октетами на завершающем конце.

Примечание — Дополняющие октеты не должны появляться в кодированиях.

11.7 Обобщенное время

11.7.1 Кодирование должно завершаться символом «Z», как описано в ГОСТ Р ИСО/МЭК 8824-1.

11.7.2 Элемент секунд всегда должен присутствовать.

11.7.3 При указании долей секунд (если они есть) должны опускаться завершающие нули; если доли секунд равны нулю, то они должны полностью опускаться вместе с десятичной точкой.

Пример. Элемент секунд «26.000» должен быть представлен как «26»; элемент секунд «26.5200» — как «26.52».

11.7.4 Элемент десятичная точка, если он присутствует, должен быть символом точка «.».

11.7.5 Полночь (GMT) должна быть представлена в форме

«YYYYMMDD000000Z»

где «YYYYMMDD» — день, следующий за рассматриваемой полночью.

11.7.6 Примеры правильных представлений:

«19920521000000Z»

«19920622123421Z»

«19920722132100.3Z»

11.7.7 Примеры неправильных представлений:

«19920520240000Z» (неверно представлена полночь)

«19920622123421.0Z» (ошибочный завершающий 0)

«19920722132100.30Z» (ошибочный завершающий 0)

11.8 UTCTime

11.8.1 Кодирование должно завершаться «Z», как описано в ГОСТ Р ИСО/МЭК 8824-1.

11.8.2 Элемент секунд всегда должен присутствовать.

11.8.3 Полночь (GMT) должна быть представлена в форме

«YYMMDD000000Z»

где «YYMMDD» — день, следующий за рассматриваемой полночью.

11.8.4 Примеры правильных представлений:

«920521000000Z«

«920622123421Z»

«920722132100Z»

11.8.5 Примеры неправильных представлений:

«920520240000Z» (неверно представлена полночь)

«9207221321Z» (опущено 00 секунд)

12 Использование BER, CER и DER в определении синтаксиса передачи

12.1 Правила кодирования, определенные в настоящем стандарте, могут быть указаны и применены в любом месте, где необходимо специфицировать недвусмысленное, неделимое и самовыделяющее представление строк октетов для всех значений единственного типа АСН.1.

Примечание — Все такие строки октетов являются недвусмысленными в пределах единственного типа АСН.1. Они не обязательно остаются недвусмысленными, если перемешаны с кодированиями разных типов АСН.1.

12.2 Следующие значения идентификаторов и описателей объектов присвоены для идентификации и описания базовых правил кодирования:

{joint-iso-itu-t asn1 (1) basic-encoding (1)}

и

«Basic Encoding of a single ASN.1 type»

12.3 Следующие значения идентификаторов и описателей объектов присвоены для идентификации и описания канонических правил кодирования:

{joint-iso-itu-t asn1 (1) ber-derived(2) canonical-encoding(0)}

и

«Canonical Encoding of a single ASN.1 type»

12.4 Следующие значения идентификаторов и описателей объектов присвоены для идентификации и описания отличительных правил кодирования:

{joint-iso-itu-t asn1 (1) ber-derived(2) distinguished-encoding(0)}

и

«Distinguished Encoding of a single ASN.1 type»

12.5 Когда недвусмысленная спецификация определяет абстрактный синтаксис как множество значений данных уровня представления, каждое из которых является значением некоторого конкретно названного типа АСН.1, обычно (но не обязательно) выборочного типа, тогда с именем абстрактного синтаксиса может использоваться одно из значений идентификатора объекта, определенного в 12.2, 12.3 или 12.4, для этого конкретно названного типа АСН.1, используемого в определении абстрактного синтаксиса.

12.6 Имена, установленные в 12.2 — 12.4, не должны использоваться с именем абстрактного синтаксиса для идентификации синтаксиса передачи, если для определения абстрактного синтаксиса не выполнено условие 12.5.

ПРИЛОЖЕНИЕ А
(справочное)

Пример кодирования

В настоящем приложении иллюстрируются базовые правила кодирования, определенные в настоящем стандарте, на примере представления в октетах (гипотетической) персональной записи, определенной с использованием АСН.1.

А.1 Описание АСН.1 структуры записи

Ниже формально описана структура гипотетической персональной записи с использованием АСН.1, определенной в ГОСТ Р ИСО/МЭК 8824-1.

```
PersonnelRecord :: = [APPLICATION 0] IMPLICIT SET {
        name            Name,
        title           [0] VisibleString,
        number          EmployeeNumber,
        dateOfHire      [1] Date,
        name0fSpouse    [2] Name,
        children        [3] IMPLICIT
            SEQUENCE OF ChildInformation DEFAULT {} }

ChildInformation : : = SET
{name                   Name,
date0fBirth             [0] Date}

Name :: = [APPLICATION 1] IMPLICIT SEQUENCE
{givenName              VisibleString,
initial                 VisibleString,
familyName              VisibleString}

EmployeeNumber :: = [APPLICATION 2] IMPLICIT INTEGER

Date :: = [APPLICATION 3] IMPLICIT VisibleString — YYYYMMDD
```

А.2 Описание АСН.1 значения записи

Значение персональной записи для John Smith ниже описано формально с использованием АСН.1.

```
{name {givenName    «John», initial «P», familyName «Smith»},
 title              «Director»,
 number             51,
 date0fHire         «19710917»,
 name0fSpouse {givenName «Mary», initial «T», familyName «Smith»},
 children
   {{{givenName «Ralph», initial «T», familyName «Smith»},
           date0fBirth «19571111»},
   {{givenName «Susan», initial «B», familyName «Jones»},
           date0fBirth «19590717»}}}
```

А.3 Представление этого значения записи

Ниже показано представление в октетах приведенного выше значения записи (после применения определенных в настоящем стандарте базовых правил кодирования). Значения идентификаторов, длин и содержимого целых чисел — шестнадцатеричные, по две шестнадцатеричные цифры на октет. Значения содержимого символьных строк приведены как текст, по одному символу на октет.

Personnel Record	Длина	Содержимое		
60	8185			
	Name	Длина	Содержимое	
	61	10		
		VisibleString	Длина	Содержимое
		1A	04	«John»
		VisibleString	Длина	Содержимое
		1A	01	«P»
		VisibleString	Длина	Содержимое
		1A	05	«Smith »

title A0	Длина 0A	Содержимое						
		VisibleString 1A	Длина 08	Содержимое «Director»				
Employee Number 42	Длина 01	Содержимое 33						
date0f Hire A1	Длина 0A	Содержимое						
		Date 43	Длина 08	Содержимое «19710917»				
name0f Spouse A2	Длина 12	Содержимое						
		Name 61	Длина 10	Содержимое				
				VisibleString 1A	Длина 04	Содержимое «Mary»		
				VisibleString 1A	Длина 01	Содержимое «T»		
				VisibleString 1A	Длина 01	Содержимое «Smith»		
[3] A3	Длина 42	Содержимое						
		Множество 31	Длина 1F	Содержимое				
				Name 61	Длина 11	Содержимое		
					VisibleString 1A	Длина 05	Содержимое «Ralph»	
					VisibleString 1A	Длина 01	Содержимое «T»	
					VisibleString 1A	Длина 05	Содержимое «Smith»	
				date0f Birth A0	Длина 0A	Содержимое		
					Date 43	Длина 08	Содержимое «19571111»	
		Множество 31	Длина 1F	Содержимое				
				Name 61	Длина 11	Содержимое		
					VisibleString 1A	Длина 05	Содержимое «Susan»	
					VisibleString 1A	Длина 01	Содержимое «B»	
					VisibleString 1A	Длина 05	Содержимое «Jones»	
				date0f Birth A0	Длина 0A	Содержимое		
					Date 43	Длина 08	Содержимое «19590717»	

ПРИЛОЖЕНИЕ В
(справочное)

Присвоение значений идентификаторов объектов

В настоящем стандарте присвоены следующие значения:

Раздел Значение идентификатора объекта

12.2 {joint-iso-itu-t asn1 (1) basic-encoding (1)}

Значение описателя объекта

«Basic Encoding of a single ASN.1 type»

Раздел Значение идентификатора объекта

12.3 {joint-iso-itu-t asn1 (1) ber-derived(2) canonical-encoding(0)}

Значение описателя объекта

«Canonical Encoding of a single ASN.1 type»

Раздел Значение идентификатора объекта

12.4 {joint-iso-itu-t asn1 (1) ber-derived(2) distinguished-encoding(0)}

Значение описателя объекта

«Distinguished Encoding of a single ASN.1 type»

ПРИЛОЖЕНИЕ С
(справочное)

Пример кодирования значения действительного числа

С.1 Отправитель, обычно, проверяет свое машинное представление чисел с плавающей точкой для определения (независящих от значения) алгоритмов, которые должны использоваться для преобразования значений между этим представлением и октетами длины и содержимого в кодировании вещественного значения АСН.1. В настоящем приложении описаны шаги, которые должны быть сделаны в таком процессе, исходя из (искусственного) машинного представления мантиссы чисел с плавающей точкой, показанного на рисунке С.1.

Принято, что экспонента может быть легко получена в виде целого числа *E* из машинного представления чисел с плавающей точкой.

Рисунок С.1 — Представление чисел с плавающей точкой

С.2 Октеты содержимого, которые должны быть созданы для отправки ненулевого значения с использованием двоичного кодирования (как определено в настоящем стандарте), суть:

1 *S bb ff ee* Октеты для *S* Октеты для *N*

где *S* (знак мантиссы) зависит от преобразуемого значения, *bb* — фиксированное значение (например, 10) для основания представления (в данном случае принято основание 16), *ff* — фиксированное значение *F*, вычисленное, как описано в С.3, *ee* — фиксированная длина значения экспоненты, вычисленная, как описано в С.4. (В настоящем приложении не рассматривается случай, когда *E* превышает три октета.)

С.3 Алгоритм будет передавать октеты 1—5 машинного представления как значение *N* после установки битов 8—3 октета 1 и битов 4—1 октета 5 равными нулю. Принято, что подразумеваемая десятичная точка в машинном представлении находится между битами 2 и 1 октета 1. Это подразумеваемое положение может быть смещено к ближайшей точке после конца октета 5 путем уменьшения значения Е до преобразования. Во взятой в качестве примера системе можно смещать на четыре бита при уменьшении экспоненты на единицу (т. к. принято основание 16), так что уменьшение на 9 поместит подразумеваемую точку между битами 6 и 5 октета 6.

Следовательно, для правильного размещения точки в M значением M должно быть N, умноженное на 2^3. (Подразумеваемое положение в N— после бита 1 октета 5). Таким образом, получаем основные параметры:

$F = 3$ (так что $ff = 11$),
декремент экспоненты равен 9.

С.4 Длина, необходимая для экспоненты, вычисляется через максимальное число октетов, необходимых для представления значений:

E_{min} — избыток — декремент экспоненты;
E_{max} — избыток — декремент экспоненты,

где E_{min} и E_{max} — минимальное и максимальное целые значения представления экспоненты, избыток — любое значение, которое нужно вычесть для получения правильного значения экспоненты, а декремент экспоненты вычислен в С.3. Пусть это вычисление дает длину 3 октета. Тогда *ee* равно 10. Примем также, что избыток равен нулю.

С.5 Алгоритм пересылки теперь такой:

а) передать поле октетов идентификатора объекта базовых правил кодирования с тегом АСН.1 для действительного типа;

б) проверить на равенство нулю и, если это так, передать поле длины базовых правил кодирования АСН.1 со значением 0 (нет октетов содержимого) и завершить алгоритм;

в) проверить и запомнить знак мантиссы;

г) передать поле длины базовых правил кодирования АСН.1 со значением 9, имеющее вид:
11101110, если отрицательное, или
10101110, если положительное;

д) создать и передать 4 октета экспоненты со значением
$E - 9$;

е) обнулить биты 8—3 октета 1 и биты 4—1 октета 5, а затем передать 5 октетов мантиссы.

С.6 Должен быть подготовлен алгоритм получения для обработки любого базового кодирования АСН.1, но здесь может быть непосредственно использована единица с плавающей точкой. Алгоритм такой:

а) проверить октет 1 содержимого; если он равен 1×101110, то отправление совместимо с получением и можно просто обратить алгоритм отправления;

б) в противном случае, для символьного кодирования, вызвать стандартную процедуру преобразования десятичных символов в число с плавающей точкой и работать со «SpecialRealValue» в соответствии с прикладной семантикой (возможно, потребуется установка наибольшего и наименьшего обрабатываемого числа с плавающей точкой);

с) для двоичной передачи — поместить N в единицы с плавающей точкой, отбросить, если необходимо, октеты с менее значащего конца, умножить на 2^F и на B^E и, если нужно, сделать отрицательным. Реализаторы могут найти возможные в специальных случаях оптимизации, но может оказаться (кроме оптимизаций, относящихся к преобразованиям на совместимых машинах), что потери от них будут больше выигрыша.

С.7 Приведенный выше алгоритм служит лишь иллюстрацией. Реализаторы должны определять собственные наилучшие стратегии.

ПРИЛОЖЕНИЕ D
(справочное)

Использование DER и CER в аутентификации источника данных

D.1 Решаемая проблема

D.1.1 Отличающие и канонические правила кодирования предназначены для помощи при обеспечении методов защиты целостности, использующих аутентификацию передаваемого материала.

Примечание — В оставшейся части настоящего приложения, для простоты, упоминается только DER. Однако все сказанное применимо и для CER.

D.1.2 Понятие аутентификатора включает в себя получение битового шаблона, который должен передаваться, применение к нему какой-либо функции хеширования для уменьшения до нескольких октетов, шифрование этих октетов для аутентификации аутентификатора и передачу аутентификатора вместе с исходным материалом (исходный материал передается открыто). При получении аутентификатор повторно вычисляется на основе полученного открытого текста и сравнивается с полученным аутентификатором. Если они равны, то текст не поддельный, в противном случае — поддельный.

D.1.3 Это простое понятие становится более сложным в модели ВОС, в частности, для уровня представления.

D.1.4 Возникают две проблемы, одна из которых — вопрос о моделировании и так называемой независимости уровней, вторая — относится к использованию ретрансляторов прикладного уровня (таких как в Рекомендации МСЭ-Т X.400).

D.1.5 Проблемы моделирования связаны с тем, что функция хеширования и алгоритм шифрования являются частью прикладной операции, но приложение не знает и не управляет фактическим кодированием, используемым уровнем представления. Аналогично, при получении декодирование и, следовательно, разбор битовой строки являются вопросом уровня представления. Имеются четыре решения, предложенные в обход этой проблемы:

а) управлять порядком использования фактических октетов, создаваемых на уровне представления для использования в аутентификаторе (этот подход одобрен экспертами по архитектуре верхнего уровня и уровня представления);

б) опустить методы хеширования и аутентификации на уровень представления (это решение было отклонено как часть более широкого вопроса об обеспечении шифрования в АСН.1. На момент отклонения доводы сводились к тому, что работы по безопасности еще полностью не окончены и не желательно навязывать случайные результаты);

в) модель комплексного взаимодействия с уровнем представления, когда при передаче значение представляется для кодирования, кодируется и возвращается прикладному уровню, который вычисляет аутентификатор, а затем все вместе передается. При получении кодированное значение передается прикладному уровню для проверки аутентификатора (эта модель была отклонена группой по архитектуре верхнего уровня);

г) полностью осуществлять кодирование на прикладном уровне и не использовать услуг уровня представления для согласования синтаксиса передачи (фактически, это отказ от базовой модели ВОС, и не может быть принято в качестве общего решения).

D.1.6 Можно говорить о том, что неудача при согласовании модели, описывающей очевидно простой и работоспособный процесс (создание кодирования, потом — аутентификатора, их передача, проверка аутентификатора при получении), не является чем-то, что может рассматриваться как долговременная проблема. Это было бы справедливо, если бы не было второй проблемы прикладной ретрансляции и если бы не было других работоспособных решений. (В настоящем приложении описывается альтернативное решение, использованное в ГОСТ Р ИСО/МЭК 9594-8, которое представляется свободным от проблем моделирования и ретрансляции).

D.1.7 Вторая проблема состоит в том, что если имеется прикладная ретрансляция, то синтаксис передачи, используемый для второй пересылки, может отличаться от того, который согласован для первой (например, использование блочных правил кодирования для одной и базовых — для другой). Это приведет к нарушению аутентификатора, если его не открывать и не вычислять повторно при ретрансляции, что подразумевается защищенным обменом и требуется сквозной защитой.

Примечание — Были предложения ввести признак уровня представления «не перекодировать при прикладной ретрансляции», но это тоже порождает проблемы моделирования и пр.

D.1.8 Таким образом, мы приходим к попытке работать с моделью, в которой уровень представления (вместе с прикладными ретрансляциями) обеспечивает передачу абстрактного синтаксиса и семантики информации, но не гарантирует, что фактическое кодирование битового шаблона (синтаксис передачи) будет сохранен от начала и до конца.

D.1.9 Требуется обеспечить методы аутентификации, который может работать с абстрактным типом данных, а не с передаваемой битовой строкой.

D.1.10 Рабочая группа справочника первой пыталась найти решение этой проблемы, и их модель описана ниже.

D.2 Подход к решению

D.2.1 Ниже описана концептуальная модель того, что нужно сделать, а затем — оптимизация реализации, исключающая двойное кодирование/декодирование, подразумеваемое концептуальной моделью.

D.2.2 Концептуальная модель работает следующим образом:

а) отправитель, на прикладном уровне, преобразует абстрактное значение в битовую строку, используя DER, и создает аутентификатор из этой битовой строки, который добавляется к абстрактному значению, и оба значения передаются с помощью обычных методов уровня представления и любого синтаксиса передачи. Концептуально, отправление кодируется дважды: один раз — для аутентификатора (используя DER) на прикладном уровне, второй раз — для фактической передачи (используя согласованный синтаксис передачи) на уровне представления.

Примечание — Важным свойством битовой строки, создаваемой DER, является ее однозначное соответствие абстрактному значению. Таким образом, сквозная передача без потери информации на уровне абстрактного синтаксиса эквивалентна сквозной передаче битовой строки, на которой основан аутентификатор;

б) получатель декодирует полученную битовую строку на уровне представления, используя согласованный синтаксис передачи (который может отличаться от использованного отправителем, если имела место прикладная ретрансляция), и передает абстрактное значение приложению. На прикладном уровне абстрактное значение кодируется повторно с использованием DER для создания битовой строки для аутентификации.

D.2.3 Таким образом, концептуально, кодирование осуществляется дважды при отправке, а при получении осуществляется одно декодирование и одно кодирование. Реализаторы могут фактически следовать этой процедуре, если код, обеспечиваемый операцией уровня представления и поставщиком, отличается от кода для поддержки приложения. Пока не ясно, насколько это существенно. При использовании интегрированных реализаций имеется возможность оптимизации, описанная ниже. Следует заметить, что DER не являются более жесткими по отношению к приложению, чем BER, за исключением типа «множество-из». Если обрабатывается большое «множество-из», то реализации может потребоваться вызов процедуры дисковой сортировки. Проектировщики приложений должны стараться этого избегать и использовать «последовательность-из» вместо «множество-из», когда предвидится использование DER.

D.3 Оптимизация реализации

D.3.1 Модель ВОС и стандарты протоколов специфицируют требуемое поведение, но они ни коим образом не устанавливают ограничений на архитектуру и структуру фактического кода реализации. Таким образом, реализатор может добиваться желаемого результата выбранным им путем.

D.3.2 На конце отправителя битовая строка, которая создается (концептуально — на прикладном уровне) может быть сохранена и использована для обеспечения кодирования, которое, концептуально, осуществляется на уровне представления. Это подходит для отправителя, если согласованным синтаксисом передачи является BER или DER. Если это не так, то двойное кодирование необходимо.

D.3.3 Аналогично, на конце получателя полученная битовая строка может быть сохранена (для любого синтаксиса передачи), и реализация может использовать ее для проверки аутентификатора. Если все согласуется — нет проблем. Если не согласуется, то это может быть проблемой синтаксиса передачи и тогда необходимо повторное кодирование абстрактного значения для выявления подделки.

D.3.4 Для того чтобы повысить шансы не иметь двойного кодирования/декодирования, системам, использующим этот метод, рекомендуется пытаться согласовать в первую очередь в качестве синтаксиса передачи DER (используя соответствующий идентификатор объекта), а затем — BER и другие правила кодирования.

УДК 681.324:006.354 ОКС 35.100.60 П85 ОКСТУ 4002

Ключевые слова: обработка данных, информационный обмен, сетевое взаимодействие, взаимосвязь открытых систем, коммуникационная процедура, преобразование данных, кодирование, правила (инструкции)

Редактор *В.П. Огурцов*
Технический редактор *В.Н. Прусакова*
Корректор *В.С. Черная*
Компьютерная верстка *Л.А. Круговой*

Изд. лиц. № 02354 от 14.07.2000. Сдано в набор 22.05.2003. Подписано в печать 19.06.2003. Усл. печ. л. 3,72.
Уч.-изд. л. 3,30. Тираж 262 экз. С 10865. Зак. 525.

ИПК Издательство стандартов, 107076 Москва, Колодезный пер., 14.
http://www.standards.ru e-mail: info@standards.ru
Набрано в Издательстве на ПЭВМ
Филиал ИПК Издательство стандартов — тип. «Московский печатник», 105062 Москва, Лялин пер., 6.
Плр № 080102