Министерство образования Республики Беларусь
Белорусский государственный университет
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ПРИКЛАДНЫХ ПРОБЛЕМ МАТЕМАТИКИ И ИНФОРМАТИКИ

УТВЕРЖДАЮ
Директор НИИ прикладных проблем
математики и информатики

Ю.С.Харин
«___» _____ 2022 г.

МЕТОДИКА ИСПЫТАНИЙ ПРОГРАММЫ, РЕАЛИЗУЮЩЕЙ УПРАВЛЕНИЕ КРИПТОГРАФИЧЕСКИМИ СООБЩЕНИЯМИ СОГЛАСНО СТБ 34.101.23–2012

МИ.10123.10.01

Листов 61

Минск 2022

Предисловие

Настоящая методика испытаний предназначена для использования в испытательных лабораториях при проведении сертификационных испытаний средств криптографической защиты информации на соответствие требованиям СТБ 34.101.23-2012 «Информационные технологии и безопасность. Синтаксис криптографических сообщений».

Содержание

1 Нормативные ссылки

В настоящем документе использованы ссылки на следующие стандарты:

ГОСТ 19.202-78 «Единая система программной документации. Спецификация. Требования к содержанию и оформлению».

ГОСТ 19.401-2000 «Единая система программной документации. Текст программы. Требования к содержанию, оформлению и контролю качества».

ГОСТ 19.402-2000 «Единая система программной документации. Описание программы. Требования к содержанию, оформлению и контролю качества».

ГОСТ 19.504-79 «Единая система программной документации. Руководство программиста. Требования к содержанию и оформлению».

ГОСТ 34.973-91 (ИСО 8824-87) «Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии 1 (АСН.1)».

СТБ 34.101.19-2012 «Информационные технологии и безопасность. Форматы сертификатов и списков отозванных сертификатов инфраструктуры открытых ключей».

СТБ 34.101.23-2012 «Информационные технологии и безопасность. Синтаксис криптографических сообщений».

СТБ 34.101.27-2022 «Информационные технологии и безопасность. Средства криптографической защиты информации. Требования безопасности».

СТБ 34.101.31-2020 «Информационные технологии и безопасность. Алгоритмы шифрования и контроля целостности».

СТБ 34.101.45-2013 «Информационные технологии и безопасность. Алгоритмы электронной цифровой подписи и транспорта ключа на основе эллиптических кривых».

СТБ 34.101.77-2020 «Информационные технологии и безопасность. Криптографические алгоритмы на основе sponge-функции».

2 Термины, обозначения и сокращения

В настоящем документе применяются термины и обозначения СТБ 34.101.26, а также следующие сокращения:

АСН.1 абстрактно-синтаксическая нотация версии 1;

ЕСПД единая система программной документации;

СКЗИ средство криптографической защиты информации;

ТНПА технический нормативный правовой акт;

ЭЦП электронная цифровая подпись.

3 Объект и цель испытаний

На испытания представляется средство криптографической защиты информации (СКЗИ), реализующее управление криптографическими сообщениями согласно СТБ 34.101.23, и документация на СКЗИ.

Целью испытаний является проверка соответствия процедур формирования и разбора криптографических сообщений, реализованных в объекте испытаний, требованиям СТБ 34.101.23.

4 Требования к объекту испытаний

Программа объекта испытаний может реализовывать следующие процедуры, определенные в СТБ 34.101.23:

– формирование криптографических сообщений с неструктурированными данными;
– разбор криптографических сообщений с неструктурированными данными;
– формирование криптографических сообщений с подписанными данными;
– разбор криптографических сообщений с подписанными данными;
– формирование криптографических сообщений с конвертованными данными;
– разбор криптографических сообщений с конвертованными данными;
– формирование криптографических сообщений с хэшированными данными;
– разбор криптографических сообщений с хэшированными данными;
– формирование криптографических сообщений с шифрованными данными;
– разбор криптографических сообщений с шифрованными данных;
– формирование криптографических сообщений с аутентифицируемыми данными;
– разбор криптографических сообщений с аутентифицируемыми данными;
– формирование криптографических сообщений с аутентифицируемыми конвертованными данными;
– разбор криптографических сообщений с аутентифицируемыми конвертованными данными.

К программе объекта испытаний предъявляются следующие требования, подлежащие проверке во время проведения испытаний:

– в программе должны быть точно и полно реализовываны процедуры СТБ 34.101.23, поддерживаемые объектом испытаний;
– программа, реализующая процедуры СТБ 34.101.23, не должна содержать недокументированные возможности.

Документация на объект испытаний должна включать документы «Спецификация», «Текст программы» и может включать документы «Описание программы», «Руководство программиста» и другие документы. Документация может быть разработана в соответствии с требованиями единой системы программной документации (ЕСПД).

5 Средства и порядок испытаний

5.1 Общие сведения

Испытания программы состоят из трех этапов:

1 Анализ документации.
2 Тестирование программы.
3 Анализ исходных текстов программы.

Выполнение этапа 1 осуществляется экспертами по анализу документации, выполнение этапа 2 — экспертами по тестированию, а выполнение этапа 3 — экспертами по анализу исходных текстов. К проведению испытаний должно быть привлечено не менее двух экспертов по анализу исходных текстов и один или более эксперт по тестированию. К

анализу документации должен быть привлечен, по крайней мере, один эксперт по анализу исходных текстов программ.

По результатам выполнения этапа испытаний эксперт оформляет протокол результатов проверок: протокол анализа документации, протокол тестирования, протокол анализа исходных текстов. В протоколе эксперт делает вывод о соответствии (не соответствии) программы требованиям СТБ 34.101.23. Если программа не поддерживает некоторые процедуры, определенные в СТБ 34.101.23, то в протоколе делается соответствующее примечание. Примеры оформления протоколов приводятся в приложениях А, Б, В. Допускается оформления протоколов в иной форме, но с обязательным указанием результатов по каждой проводимой проверке и вывода о соответствии (не соответствии).

Если в испытуемой программе используются реализации процедур СТБ 34.101.23, которые в составе других программ имеют действующие сертификаты соответствия требованиям СТБ 34.101.23, то проверки по тестированию и анализу исходных текстов для данных реализаций могут не проводиться. При этом для подтверждения соответствия объекта испытаний требованиям СТБ 34.101.23 экспертом оформляется протокол проверки совпадения контрольных характеристик (хэш-значений) файлов реализации испытуемой программы с контрольными характеристиками соответствующих файлов, указанными в сертификатах соответствия.

На основании протоколов результатов проверок оформляется протокол испытаний, обобщающий результаты испытаний программы. В протоколе испытаний вывод о соответствии программы требованиям СТБ 34.101.23 делается тогда и только тогда, когда вывод о соответствии содержится во всех протоколах результатов проверок. Оформление протокола испытаний проводится в соответствии с требованиями технических нормативных правовых актов (ТНПА) в области сертификации продукции, а также документации, применяемой в испытательной лаборатории.

Реализация в программе каждого криптографического алгоритма, используемого в процедурах СТБ 34.101.23, предварительно должна пройти успешные испытания по согласованной с Органом по сертификации методике испытаний.

Испытываемая программа может не поддерживать необязательный функционал, определенный в СТБ 34.101.23 (например, формирование необязательных атрибутов). При этом сужение программой обязательного функционала, определенного в СТБ 34.101.23, не допускается.

5.2 Анализ документации

Эксперт проводит анализ документации путем проверки соответствия документации программе объекта испытаний. Такой анализ состоит в получении экспертных заключений, касающихся проверки следующих документов:

- спецификация (см. п. 6.1.1);
- текст программы (см. п. 6.1.2);
- описание программы (см. п. 6.1.3);
- руководство программиста (см. п. 6.1.4).

Анализ документов «Описание программы» и «Руководство программиста» производится в случае их наличия.

5.3 Тестирование

Эксперт проводит тестирование процедур формирования и разбора криптографических сообщений, реализованных в программе и определенных в СТБ 34.101.23.

Тестирование процедур формирования криптографических сообщений (см. п. 6.2.1) выполняется путем формирования программой соответствующих сообщений с последующим визуальным сравнением текстового представления форматов сформированных сообщений с форматами, определенными в СТБ 34.101.23.

Тестирование процедуры разбора криптографических сообщений (см. п. 6.2.2) проводится путем выполнения программой разбора соответствующих сообщений с последующим сравнением полученных результатов с ожидаемыми.

В тестах используются криптографические сообщения, представленные в виде бинарных файлов, содержащих закодированные значения типов абстрактно-синтаксической нотации версии 1 (АСН.1), спецификация которой приводится в ГОСТ 34.973. Для преобразования бинарных файлов криптографических сообщений в их текстовое представление могут использоваться программы, описанные в приложении Г.1.

При успешном выполнении тест возвращает признак `УСПЕХ`, иначе — `ОШИБКА`. Если при тестировании программы для некоторых входных значений получены результаты отличные от ожидаемых, то эксперт по тестированию должен указать эти входные значения программы и результат ее работы, а также, по требованию, результаты промежуточных вычислений экспертам по анализу исходных текстов.

Для организации тестирования в исходные тексты программы допускается вносить изменения и дополнения, касающиеся:

– способа чтения входных данных;

– способа записи выходных данных.

При внесении модификаций в исходные тексты должен быть проведен анализ корректности внесенных изменений.

5.4 Анализ исходных текстов

Эксперт проводит анализ исходных текстов путем проверки корректности реализации в испытуемой программе процедур СТБ 34.101.26. Такой анализ состоит в получении экспертных заключений, касающихся:

– корректности использования криптографических алгоритмов (см. п. 6.3.1);

– корректности управления секретными данными (см. п. 6.3.2);

– корректности процедуры формирования криптографических сообщений (см. п. 6.3.3);

– корректности процедуры разбора криптографических сообщений (см. п. 6.3.4);

– корректности обработки исключительных ситуаций (см. п. 6.3.5);

– отсутствия недокументированных возможностей (см. п. 6.3.6).

При анализе исходных текстов проверки из п. 6.3.5, 6.3.6 выполняются для всех реализованных процедур формирования и разбора криптографических сообщений. Дополнительно, проверки из п. 6.3.1 выполняются для процедур формирования и разбора криптографических сообщений, в которых используются криптографические алгоритмы, проверки из п. 6.3.3) выполняются для всех процедур формирования криптографических сообщений, проверки из п. 6.3.4 выполняются для всех процедур разбора криптографических сообщений, а проверки из п. 6.3.2 выполняются только для тех процедур, в кото-

рых используются секретные параметры (секретные ключи, личные ключи, одноразовые ключи). При выполнении данных проверок следует учитывать рекомендации по анализу исходных текстов программ, определенные в приложении В СТБ 34.101.27.

6 Методы испытаний

6.1 Анализ документации

6.1.1 Документ «Спецификация»

При анализе документа «Спецификация» эксперт проверяет, что в нем указаны компоненты и документация, представляемые на испытания.

Если документ «Спецификация» разработан в соответствии с требованиями ЕСПД, то эксперт проверяет, что содержание и оформление документа соответствует ГОСТ 19.202.

6.1.2 Документ «Текст программы»

При анализе документа «Текст программы» эксперт проверяет, что исходные тексты программы, реализующие определенные в СТБ 34.101.23 процедуры, представлены полностью и в виде, который использовался при сборке программы.

Если документ «Текст программы» разработан в соответствии с требованиями ЕСПД, то эксперт должен проверяет, что содержание и оформление документа соответствует ГОСТ 19.401.

6.1.3 Документ «Описание программы»

При анализе документа «Описание программы» эксперт проверяет выполнение следующих требований:

– в документе должна быть указана информация, однозначно идентифицирующая вызываемые стандартные функции (версия компилятора, используемые стандартные библиотеки и т.п.);

– документ должен определять программные модули, реализующие определенные в СТБ 34.101.23 процедуры;

– описание программы в терминах программных модулей должно соответствовать исходным текстам программы.

Если документ «Описание программы» разработан в соответствии с требованиями ЕСПД, то эксперт проверяет, что содержание и оформление документа соответствует ГОСТ 19.402.

6.1.4 Документ «Руководство программиста»

При анализе документа «Руководство программиста» эксперт проверяет выполнение следующих требований:

– документ должен содержать описание всех доступных для вызова функций, реализующих определенные в СТБ 34.101.23 процедуры;

– описание функций, реализующих определенные в СТБ 34.101.23 процедуры, и условия их использования должны соответствовать исходным текстам программы.

При описании в документации функций должны выполняться следующие условия:

– каждая функция должна иметь описание назначения;

– каждый параметр функции должен иметь описание назначения, типа и, при необходимости, диапазона допустимых значений;

– каждая функция должна иметь описание возвращаемого результата с указанием типа;

– каждая функция должна иметь описание условий, при выполнении которых в ходе работы функции могут возникать ошибочные ситуации, требующие специальной обработки;

– в случае если при реализации определенной в СТБ 34.101.23 процедуры используется более одной доступной для вызова функции, должны быть указаны порядок и условия вызова данных функций.

Если документ «Руководство программиста» разработан в соответствии с требованиями ЕСПД, то эксперт проверяет, что содержание и оформление документа соответствует ГОСТ 19.504.

6.2 Тестирование

6.2.1 Процедуры формирования криптографических сообщений

6.2.1.1 Общие сведения

Входными данными, задаваемыми при тестировании процедур формирования криптографических сообщений, являются допустимые значения для параметров вызова программы, определяющих состав и содержимое компонент криптографических сообщений.

В тестах для хранения криптографических сообщений используются бинарные файлы, содержащие закодированные значения типов ASN.1, составляющих сообщение.

6.2.1.2 Неструктурированные данные

При тестировании процедуры формирования криптографических сообщений с неструктурированными данными выполняется следующий тест.

Тест CreateUDTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования криптографических сообщений с неструктурированными данными.

2 Средствами испытываемой программы сформировать криптографическое сообщение с неструктурированными данными и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов ASN.1, составляющих сообщение.

3 Преобразовать файл, полученный на шаге 2, в текстовое представление криптографического сообщения.

4 Провести визуальный анализ текстового представления сформированного сообщения на соответствие п. 6 и п. 7 СТБ 34.101.23.

5 Повторить шаги 1 — 4 не менее 4 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).

6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе сформированного сообщения не выявлено несоответствий СТБ 34.101.23, при этом сообщение является значением типа `ContentInfo` (см. п. 6 СТБ 34.101.23), для которого:

– компонент `contentType` содержит значение `id-data` (см. п. 7 СТБ 34.101.23) типа `ContentType`, соответствующего типу `OBJECT IDENTIFIER` (см. приложение А.1 СТБ 34.101.23);

– компоненте `content` содержит данные, которые могут быть интерпретированы согласно документации на испытываемую программу.
7 Возвратить `ОШИБКА`.

6.2.1.3 Подписанные данные

При тестировании процедуры формирования криптографических сообщений с подписываемыми данными выполняется следующий тест.

Тест CreateSDTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования криптографических сообщений с подписанными данными.
2 Средствами испытываемой программы сформировать криптографическое сообщение с подписанными данными и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих сообщение.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление криптографического сообщения.
4 Провести визуальный анализ текстового представления сформированного сообщения на соответствие п. 6 и п. 8 СТБ 34.101.23.
5 Повторить шаги 1 – 4 не менее 4 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).
6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе сформированного сообщения не выявлено несоответствий СТБ 34.101.23, при этом:
1) сообщение является значением типа `ContentInfo` (см. п. 6 СТБ 34.101.23), для которого:

- компонент `contentType` содержит значение `id-signedData` (см. п. 8.1 СТБ 34.101.23) типа `ContentType` (см. приложение А.1 СТБ 34.101.23);
- компонент `content` содержит значение типа `SignedData` (см. п. 8.2 СТБ 34.101.23) и определяет подписанные данные;

2) формат и содержание компонента `content` соответствуют типу `SignedData` (см. п. 8.2 СТБ 34.101.23):

- компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого типа синтаксиса;
- компонент `digestAlgorithms` содержит значение типа `DigestAlgorithmIdentifiers` (см. п. 8.2 и п. 14.1 СТБ 34.101.23) и определяет список использованных алгоритмов хэширования;
- компонент `encapContentInfo` содержит значение типа `EncapsulatedContentInfo` (см. п. 8.3 СТБ 34.101.23) и определяет подписываемые данные, представленные идентификатором типа данных и собственно содержимым;
- необязательный компонент `certificates` (при его наличии) содержит значение типа `CertificateSet` (см. п. 14.3 и 14.4 СТБ 34.101.23) и определяет список сертификатов;

- необязательный компонент `crls` (при его наличии) содержит значение типа `RevocationInfoChoices` (см. п. 14.2 СТБ 34.101.23) и определяет набор данных о статусе отзыва сертификатов, содержащихся в компоненте `certificates`;
- компонент `signerInfos` содержит значение типа `SignerInfos` (см. п. 8.2 и п. 8.4 СТБ 34.101.23) и определяет список данных, связанных с подписывающими сторонами;

3) формат и содержание значения компонента `encapContentInfo` соответствуют типу `EncapsulatedContentInfo` (см. п. 8.3 СТБ 34.101.23):

- компонент `eContentType` содержит значение типа `ContentType` (см. приложение А.1 СТБ 34.101.23) и определяет тип данных;
- необязательный компонент `eContent` (при его наличии) содержит значение типа `OCTET STRING` и определяет собственно данные, закодированные строкой октетов (использование при кодировании отличительных правил не является обязательным);

4) формат и содержание значения каждого элемента компонента `signerInfos` соответствуют типу `SignerInfo` (см. п. 8.4 СТБ 34.101.23):

- компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого синтаксиса в зависимости от значения компонента `sid`;
- компонент `sid` содержит значение типа `SignerIdentifier` и определяет ссылку на сертификат подписывающей стороны, в том числе ссылку на открытый ключ стороны; при этом, если в `sid` выбран компонент типа `IssuerAndSerialNumber` (см. п. 14.5 СТБ 34.101.23), то значение `version` должно принимать значение 1, а если выбран компонент типа `SubjectKeyIdentifier` (см. подпункт 6.2.1.2 СТБ 34.101.19), то – значение 3;
- компонент `digestAlgorithm` содержит значение типа `DigestAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма хэширования, используемого при выработке и проверке ЭЦП;
- необязательный компонент `signedAttrs` (при его наличии) содержит значение типа `SignedAttributes` и определяет подписываемые атрибуты типа `Attribute` (см. п. 15.1 СТБ 34.101.23); при использовании компонента `signedAttrs` он должен содержать атрибуты «тип содержимого» (см. 15.2 СТБ 34.101.23) и «хэш-значение» (см. п. 15.3 СТБ 34.101.23);
- компонент `signatureAlgorithm` содержит значение типа `SignatureAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет алгоритм, используемый для выработки ЭЦП, и параметры алгоритма;
- компонент `signature` содержит значение типа `OCTET STRING` и определяет ЭЦП хэш-значения, выработанную на личном ключе подписывающей стороны (значение компонента определяется с помощью выбранного алгоритма выработки ЭЦП);
- необязательный компонент `unsignedAttrs` (при его наличии) содержит значение типа `UnsignedAttributes` и определяет неподписываемые (не учитываемые при выработке ЭЦП) атрибуты типа `Attribute` (см. п. 15.1 СТБ 34.101.23); при

использовании компонента `unsignedAttrs` он может содержать атрибут «контрподпись» (см. п. 15.5 СТБ 34.101.23).

7 Возвратить `ОШИБКА`.

6.2.1.4 Конвертованные данные

При тестировании процедуры формирования криптографических сообщений с конвертованными данными выполняется следующий тест.

Тест CreateEnvDTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования криптографических сообщений с конвертованными данными.
2 Средствами испытываемой программы сформировать криптографическое сообщение с конвертованными данными и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих сообщение.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление криптографического сообщения.
4 Провести визуальный анализ текстового представления сформированного сообщения на соответствие п. 6 и п. 9 СТБ 34.101.23.
5 Повторить шаги 1 – 4 не менее 4 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).
6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе сформированного сообщения не выявлено несоответствий СТБ 34.101.23, при этом:
1) сообщение является значением типа `ContentInfo` (см. п. 6 СТБ 34.101.23), для которого:

- компонент `contentType` содержит значение `id-envelopedData` (см. п. 9.1 СТБ 34.101.23) типа `ContentType` (см. приложение А.1 СТБ 34.101.23);
- компонент `content` содержит значение типа `EnvelopedData` (см. п. 9.2 СТБ 34.101.23) и определяет конвертованные данные;

2) формат и содержание компонента content соответствуют типу `EnvelopedData` (см. п. 9.2 СТБ 34.101.23):

- компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого типа синтаксиса;
- необязательный компонент `originatorInfo` (при его наличии) содержит значение типа `OriginatorInfo` (см. п. 9.2 СТБ 34.101.23) и определяет информацию об отправителе; компонент присутствует, только если его требуется использовать в алгоритме управления ключами; компонент может содержать сертификаты, представленные в компоненте типа `CertificateSet` (см. п. 14.3 и 14.4 СТБ 34.101.23), и списки отозванных сертификатов, представленные в компоненте типа `RevocationInfoChoices` (см. п. 14.2 СТБ 34.101.23);
- компонент `recipientInfos` содержит значение типа `RecipientInfos` (см. п. 9.2 и п. 9.3 СТБ 34.101.23) и определяет информацию, связанную с получателями и

представленную значениями типа `RecipientInfo`; в данном компоненте должна быть представлена информация по крайней мере для одного получателя;

- компонент `encryptedContentInfo` содержит значение типа `EncryptedContentInfo` (см. п. 9.2 СТБ 34.101.23) и определяет зашифрованные данные;
- необязательный компонент `unprotectedAttrs` (при его наличии) содержит значение типа `UnprotectedAttributes` (см. п. 9.2 СТБ 34.101.23) и определяет дополнительные незашифрованные атрибуты типа `Attribute` (см. п. 15.1 СТБ 34.101.23);

3) формат и содержание значения компонента `encryptedContentInfo` соответствуют типу `EncryptedContentInfo` (см. п. 9.2 СТБ 34.101.23):

- компонент `contentType` содержит значение типа `ContentType` (см. приложение А.1 СТБ 34.101.23) и определяет идентификатор типа данных;
- компонент `contentEncryptionAlgorithm` содержит значение типа `ContentEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры используемого алгоритма шифрования;
- необязательный компонент `encryptedContent` (при его наличии) содержит значение типа `EncryptedContent` (см. п. 9.2 СТБ 34.101.23) и определяет зашифрованные данные (при отсутствии компонента зашифрованные данные должны быть получены из внешнего источника);

4) формат и содержание значения каждого элемента компонента `recipientInfos` соответствуют типу `RecipientInfo` (см. п. 9.3 СТБ 34.101.23), который является выбором одного из следующих компонентов:

- компонент `ktri` содержит значение типа `KeyTransRecipientInfo` (см. п. 9.4 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для управления ключами шифрования данных используется транспорт ключа;
- компонент `kari` содержит значение типа `KeyAgreeRecipientInfo` (см. п. 9.4.1 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для управления ключами шифрования данных используется согласование ключа;
- компонент `kekri` содержит значение типа `KEKRecipientInfo` (см. п. 9.4.2 СТБ 34.101.23) и определяет информацию, касающуюся получателей при использовании заранее распределенных ключей шифрования данных;
- компонент `pwri` содержит значение типа `PasswordRecipientinfo` (см. п. 9.4.3 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для формирования ключей шифрования данных используются пароли;
- компонент `ori` содержит значение типа `OtherRecipientInfo` (см. п. 9.4.4 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда применяются дополнительные методы управления ключами шифрования данных;

5) формат и содержание значения компонента `ktri` (при его наличии) соответствуют типу `KeyTransRecipientInfo` (см. п. 9.4 СТБ 34.101.23):

- компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого синтаксиса в зависимости от значения компонента `rid`;
- компонент `rid` содержит значение типа `RecipientIdentifier` (см. п. 9.4 СТБ 34.101.23) и определяет ссылку на сертификат получателя, в том числе ссылку на открытый ключ получателя; при этом, если в `rid` выбран компонент типа `IssuerAndSerialNumber` (см. п. 14.5 СТБ 34.101.23), то значение `version` должно принимать значение 0, а если выбран компонент типа `SubjectKeyIdentifier` (см. подпункт 6.2.1.2 СТБ 34.101.19), то — значение 2;
- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет для получателя идентификатор и параметры алгоритма, используемые для шифрования ключа шифрования данных;
- компонент `encryptedKey` содержит значение типа `EncryptedKey` (см. п. 9.3 СТБ 34.101.23) и определяет для получателя зашифрованный ключ шифрования данных;

6) формат и содержание значения компонента `kari` (при его наличии) соответствуют типу `KeyAgreeRecipientInfo` (см. п. 9.4.1 СТБ 34.101.23):

- компонент `version` содержит значение 3 типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23);
- компонент `originator` содержит значение типа `OriginatorIdentifierOrKey` (см. п. 9.4.1 СТБ 34.101.23) и определяет открытый ключ отправителя одним из трех способов: с использованием значения типа `IssuerAndSerialNumber`, с использованием значения типа `SubjectKeyIdentifier` (см. подпункт 6.2.1.2 СТБ 34.101.19), с использованием значения типа `OriginatorPublicKey` (см. п. 9.4.1 СТБ 34.101.23);
- необязательный компонент `ukm` (при его наличии) содержит значение типа `UserKeyingMaterial` (см. п. 14.7 СТБ 34.101.23) и определяет дополнительные данные алгоритма согласования общего ключа;
- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма, который используется для шифрования ключа шифрования данных;
- компонент `recipientEncryptedKeys` содержит значение типа `RecipientEncryptedKeys` (см. п. 9.4.1 СТБ 34.101.23) и определяет список идентификаторов получателей и предназначенных им зашифрованных ключей шифрования данных;

7) формат и содержание значения компонента `kekri` (при его наличии) соответствуют типу `KEKRecipientInfo` (см. п. 9.4.2 СТБ 34.101.23):

- компонент `version` содержит значение 4 типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23);
- компонент `kekid` содержит значение типа `KEKIdentifier` (см. п. 9.4.2 СТБ 34.101.23) и указывает на секретный ключ шифрования ключей, который был предварительно распределен отправителю и одному или нескольким получателям;

- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма шифрования ключа шифрования данных;
- компонент `encryptedKey` содержит значение типа `EncryptedKey` (см. п. 9.3 СТБ 34.101.23) и определяет ключ шифрования данных, зашифрованный на ключе шифрования ключей;

8) формат и содержание значения компонента `pwri` (при его наличии) соответствуют типу `PasswordRecipientinfo` (см. п. 9.4.3 СТБ 34.101.23):

- компонент `version` содержит значение 0 типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23);
- компонент `kekid` содержит значение типа `KEKIdentifier` (см. п. 9.4.2 СТБ 34.101.23) и указывает на секретный ключ шифрования ключей, который был предварительно распределен отправителю и одному или нескольким получателям;
- необязательный компонент `keyDerivationAlgorithm` (при его наличии) содержит значение типа `KeyDerivationAlgorithm Identifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма выработки ключа шифрования ключей по паролю или другому общему секрету;
- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма шифрования ключа шифрования данных;
- компонент `encryptedKey` содержит значение типа `EncryptedKey` (см. п. 9.3 СТБ 34.101.23) и определяет ключ шифрования данных, зашифрованный на ключе шифрования ключей;

9) формат и содержание значения компонента `ori` (при его наличии) соответствуют типу `OtherRecipientInfo` (см. п. 9.4.4 СТБ 34.101.23):

- компонент `oriType` содержит значение типа `OBJECT IDENTIFIER` и определяет метод управления ключами;
- компонент `oriValue` содержит значение, тип которого определяется значением компонента oriType, и определяет данные, необходимые для применения выбранного метода управления.

7 Возвратить `ОШИБКА`.

6.2.1.5 Хэшированные данные

При тестировании процедуры формирования криптографических сообщений с хэшированными данными выполняется следующий тест.

Тест CreateDDTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования криптографических сообщений с хэшированными данными.

2 Средствами испытываемой программы сформировать криптографическое сообщение с хэшированными данными и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих сообщение.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление криптографического сообщения.
4 Провести визуальный анализ текстового представления сформированного сообщения на соответствие п. 6 и п. 10 СТБ 34.101.23.
5 Повторить шаги 1 – 4 не менее 4 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).
6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе сформированного сообщения не выявлено несоответствий СТБ 34.101.23, при этом:
1) сообщение является значением типа `ContentInfo` (см. п. 6 СТБ 34.101.23), для которого:

- компонент `contentType` содержит значение `id-digestedData` (см. п. 10.1 СТБ 34.101.23) типа `ContentType` (см. приложение А.1 СТБ 34.101.23);
- компонент `content` содержит значение типа `DigestedData` (см. п. 10.2 СТБ 34.101.23) и определяет хэшированные данные;

2) формат и содержание компонента `content` соответствуют типу `DigestedData` (см. п. 10.2 СТБ 34.101.23):

- компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого типа синтаксиса; если вложенные данные являются неструктурированными и имеют тип `id-data`, то номер версии ДОЛЖЕН быть равен 0, в остальных случаях номер версии должен быть равен 2;
- компонент `digestAlgorithms` содержит значение типа `DigestAlgorithmIdentifiers` (см. п. 8.2 и п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма хэширования;
- компонент `encapContentInfo` содержит значение типа `EncapsulatedContentInfo` (см. п. 8.3 СТБ 34.101.23) и определяет данные, для которых вычисляется хэш-значение;
- компонент `digest` содержит значение типа `Digest` (см. п. 10.2 СТБ 34.101.23) и определяет хэш-значение данных;

3) формат и содержание значения компонента `encapContentInfo` соответствуют типу `EncapsulatedContentInfo` (см. п. 8.3 СТБ 34.101.23):

- компонент `eContentType` содержит значение типа `ContentType` (см. приложение А.1 СТБ 34.101.23) и определяет тип данных;
- необязательный компонент `eContent` (при его наличии) содержит значение типа `OCTET STRING` и определяет собственно данные, закодированные строкой октетов (использование при кодировании отличительных правил не является обязательным).

7 Возвратить `ОШИБКА`.

6.2.1.6 Шифрованные данные

При тестировании процедуры формирования криптографических сообщений с шифрованными данными выполняется следующий тест.

Тест CreateEncrDTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования криптографических сообщений с шифрованными данными.
2 Средствами испытываемой программы сформировать криптографическое сообщение с шифрованными данными и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих сообщение.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление криптографического сообщения.
4 Провести визуальный анализ текстового представления сформированного сообщения на соответствие п. 6 и п. 11 СТБ 34.101.23.
5 Повторить шаги 1 – 4 не менее 4 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).
6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе сформированного сообщения не выявлено несоответствий СТБ 34.101.23, при этом:
1) сообщение является значением типа `ContentInfo` (см. п. 6 СТБ 34.101.23), для которого:

- компонент `contentType` содержит значение `id-encryptedData` (см. п. 11.1 СТБ 34.101.23) типа `ContentType` (см. приложение А.1 СТБ 34.101.23);
- компонент `content` содержит значение типа `EncryptedData` (см. п. 11.2 СТБ 34.101.23) и определяет шифрованные данные;

2) формат и содержание компонента `content` соответствуют типу `EncryptedData` (см. п. 11.2 СТБ 34.101.23):

- компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого типа синтаксиса; если в `content` включен компонент `unprotectedAttrs`, то номер версии должен быть равен 2, а если компонент `unprotectedAttrs` отсутствует, то номер версии должен быть равен 0;
- компонент `encryptedContentInfo` содержит значение типа `EncryptedContentInfo` (см. п. 9.2 СТБ 34.101.23) и определяет зашифрованные данные;
- необязательный компонент `unprotectedAttrs` (при его наличии) содержит значение типа `UnprotectedAttributes` (см. п. 9.2 СТБ 34.101.23) и определяет дополнительные незашифрованные атрибуты типа `Attribute` (см. п. 15.1 СТБ 34.101.23);

3) формат и содержание значения компонента `encryptedContentInfo` соответствуют типу `EncryptedContentInfo` (см. п. 9.2 СТБ 34.101.23):

- компонент `contentType` содержит значение типа `ContentType` (см. приложение А.1 СТБ 34.101.23) и определяет идентификатор типа данных;

– компонент `contentEncryptionAlgorithm` содержит значение типа `ContentEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры используемого алгоритма шифрования;
– необязательный компонент `encryptedContent` (при его наличии) содержит значение типа `EncryptedContent` (см. п. 9.2 СТБ 34.101.23) и определяет зашифрованные данные (при отсутствии компонента зашифрованные данные должны быть получены из внешнего источника).

7 Возвратить `ОШИБКА`.

6.2.1.7 Аутентифицируемые данные

При тестировании процедуры формирования криптографических сообщений с аутентифицируемыми данными выполняется следующий тест.

Тест CreateADTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования криптографических сообщений с аутентифицируемыми данными.
2 Средствами испытываемой программы сформировать криптографическое сообщение с аутентифицируемыми данными и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих сообщение.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление криптографического сообщения.
4 Провести визуальный анализ текстового представления сформированного сообщения на соответствие п. 6 и п. 12 СТБ 34.101.23.
5 Повторить шаги 1 – 4 не менее 4 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).
6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе сформированного сообщения не выявлено несоответствий СТБ 34.101.23, при этом:
1) сообщение является значением типа `ContentInfo` (см. п. 6 СТБ 34.101.23), для которого:

– компонент `contentType` содержит значение `id-ct-authData` (см. п. 12.1 СТБ 34.101.23) типа `ContentType` (см. приложение А.1 СТБ 34.101.23);
– компонент `content` содержит значение типа `AuthenticatedData` (см. п. 12.2 СТБ 34.101.23) и определяет аутентифицируемые данные;

2) формат и содержание компонента `content` соответствуют типу `AuthenticatedData` (см. п. 12.2 СТБ 34.101.23):

– компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого типа синтаксиса;
– необязательный компонент `originatorInfo` (при его наличии) содержит значение типа `OriginatorInfo` (см. п. 9.2 СТБ 34.101.23) и определяет информацию об отправителе; компонент присутствует, только если его требуется использовать в алгоритме управления ключами; компонент может содержать сертификаты, представленные в компоненте типа `CertificateSet` (см. п. 14.3 и 14.4

СТБ 34.101.23), и списки отозванных сертификатов, представленные в компоненте типа `RevocationInfoChoices` (см. п. 14.2 СТБ 34.101.23);

- компонент `recipientInfos` содержит значение типа `RecipientInfos` (см. п. 9.2 и п. 9.3 СТБ 34.101.23) и определяет информацию, связанную с получателями и представленную значениями типа `RecipientInfo`; в данном компоненте должна быть представлена информация по крайней мере для одного получателя;
- компонент `macAlgorithm` содержит значение типа `MessageAuthenticationCodeAlgorithm` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры используемого алгоритма выработки имитовставки;
- необязательный компонент `digestAlgorith` (при его наличии) содержит значение типа `DigestAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма хэширования, используемого для обработки вложенных данных при наличии аутентифицируемых атрибутов; если в `content` включен компонент `digestAlgorithm`, то в `content` также должен быть включен и компонент `authAttrs`;
- компонент `encapContentInfo` содержит значение типа `EncapsulatedContentInfo` (см. п. 8.3 СТБ 34.101.23) и определяет вложенные данные, целостность и подлинность которых контролируется;
- необязательный компонент `authAttrs` (при его наличии) содержит значение типа `AuthAttributes` (см. п. 12.2 СТБ 34.101.23) и определяет набор аутентифицируемых атрибутов типа `Attribute` (см. п. 15.1 СТБ 34.101.23); компонент должен присутствовать, если значение типа `EncapsulatedContentInfo` содержит данные, тип которых отличается от `id-data`; если присутствует компонент `authAttrs`, то также должен присутствовать компонент `digestAlgorithm`; значение типа `AuthAttributes` должно быть закодировано с помощью отличительных правил, даже если все остальные значения закодированы с помощью базовых правил; если присутствует компонент `authAttrs`, то он должен содержать по крайней мере атрибуты «тип содержимого» (см. 15.2 СТБ 34.101.23) и «хэш-значение» (см. 15.3 СТБ 34.101.23);
- компонент `mac` содержит значение типа `MessageAuthenticationCode` (см. п. 12.2 СТБ 34.101.23) и определяет имитовставку.
- необязательный компонент `unauthAttrs` (при его наличии) содержит значение типа `UnauthAttributes` (см. п. 12.2 СТБ 34.101.23) и определяет неаутентифицируемые атрибуты типа `Attribute` (см. п. 15.1 СТБ 34.101.23);

3) формат и содержание значения компонента `encapContentInfo` соответствуют типу `EncapsulatedContentInfo` (см. п. 8.3 СТБ 34.101.23):

- компонент `eContentType` содержит значение типа `ContentType` (см. приложение А.1 СТБ 34.101.23) и определяет тип данных;
- необязательный компонент `eContent` (при его наличии) содержит значение типа `OCTET STRING` и определяет собственно данные, закодированные строкой октетов (использование при кодировании отличительных правил не является обязательным);

4) формат и содержание значения каждого элемента компонента `recipientInfos` соответствуют типу `RecipientInfo`(см. п. 9.3 СТБ 34.101.23), который является выбором одного из следующих компонентов:

- компонент `ktri` содержит значение типа `KeyTransRecipientInfo` (см. п. 9.4 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для управления ключами выработки имитовставки используется транспорт ключа;
- компонент `kari` содержит значение типа `KeyAgreeRecipientInfo` (см. п. 9.4.1 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для управления ключами выработки имитовставки используется согласование ключа;
- компонент `kekri` содержит значение типа `KEKRecipientInfo` (см. п. 9.4.2 СТБ 34.101.23) и определяет информацию, касающуюся получателей при использовании заранее распределенных ключей выработки имитовставки;
- компонент `pwri` содержит значение типа `PasswordRecipientinfo` (см. п. 9.4.3 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для формирования ключей выработки имитовставки используются пароли;
- компонент `ori` содержит значение типа `OtherRecipientInfo` (см. п. 9.4.4 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда применяются дополнительные методы управления ключами выработки имитовставки;

5) формат и содержание значения компонента `ktri` (при его наличии) соответствуют типу `KeyTransRecipientInfo` (см. п. 9.4 СТБ 34.101.23):

- компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого синтаксиса в зависимости от значения компонента `rid`;
- компонент `rid` содержит значение типа `RecipientIdentifier` (см. п. 9.4 СТБ 34.101.23) и определяет ссылку на сертификат получателя, в том числе ссылку на открытый ключ получателя; при этом, если в `rid` выбран компонент типа `IssuerAndSerialNumber` (см. п. 14.5 СТБ 34.101.23), то значение version должно принимать значение 0, а если выбран компонент типа `SubjectKeyIdentifier` (см. подпункт 6.2.1.2 СТБ 34.101.19), то — значение 2;
- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет для получателя идентификатор и параметры алгоритма, используемые для шифрования ключа выработки имитовставки;
- компонент `encryptedKey` содержит значение типа `EncryptedKey` (см. п. 9.3 СТБ 34.101.23), и определяет для получателя зашифрованный ключ выработки имитовставки;

6) формат и содержание значения компонента `kari` (при его наличии) соответствуют типу `KeyAgreeRecipientInfo` (см. п. 9.4.1 СТБ 34.101.23):

- компонент `version` содержит значение 3 типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23);

- компонент originator содержит значение типа OriginatorIdentifierOrKey (см. п. 9.4.1 СТБ 34.101.23) и определяет открытый ключ отправителя одним из трех способов: с использованием значения типа IssuerAndSerialNumber, с использованием значения типа SubjectKeyIdentifier (см. подпункт 6.2.1.2 СТБ 34.101.19), с использованием значения типа OriginatorPublicKey (см. п. 9.4.1 СТБ 34.101.23);
- необязательный компонент ukm (при его наличии) содержит значение типа UserKeyingMaterial (см. п. 14.7 СТБ 34.101.23) и определяет дополнительные данные алгоритма согласования общего ключа;
- компонент keyEncryptionAlgorithm содержит значение типа KeyEncryptionAlgorithmIdentifier (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма, который используется для шифрования ключа выработки имитовставки;
- компонент recipientEncryptedKeys содержит значение типа RecipientEncryptedKeys (см. п. 9.4.1 СТБ 34.101.23) и определяет список идентификаторов получателей и предназначенных им зашифрованных ключей выработки имитовставки;

7) формат и содержание значения компонента kekri (при его наличии) соответствуют типу KEKRecipientInfo (см. п. 9.4.2 СТБ 34.101.23):

- компонент version содержит значение 4 типа CMSVersion, соответствующего типу INTEGER (см. п. 14.6 СТБ 34.101.23);
- компонент kekid содержит значение типа KEKIdentifier (см. п. 9.4.2 СТБ 34.101.23) и указывает на секретный ключ шифрования ключей, который был предварительно распределен отправителю и одному или нескольким получателям;
- компонент keyEncryptionAlgorithm содержит значение типа KeyEncryptionAlgorithmIdentifier (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма шифрования ключа выработки имитовставки;
- компонент encryptedKey содержит значение типа EncryptedKey (см. п. 9.3 СТБ 34.101.23) и определяет ключ выработки имитовставки, зашифрованный на ключе шифрования ключей;

8) формат и содержание значения компонента pwri (при его наличии) соответствуют типу PasswordRecipientinfo (см. п. 9.4.3 СТБ 34.101.23):

- компонент version содержит значение 0 типа CMSVersion, соответствующего типу INTEGER (см. п. 14.6 СТБ 34.101.23);
- компонент kekid содержит значение типа KEKIdentifier (см. п. 9.4.2 СТБ 34.101.23) и указывает на секретный ключ выработки имитовставки, который был предварительно распределен отправителю и одному или нескольким получателям;
- необязательный компонент keyDerivationAlgorithm (при его наличии) содержит значение типа KeyDerivationAlgorithmIdentifier (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма выработки ключа шифрования ключей по паролю или другому общему секрету;

– компонент keyEncryptionAlgorithm содержит значение типа KeyEncryptionAlgorithmIdentifier (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма шифрования ключа выработки имитовставки;

– компонент encryptedKey содержит значение типа EncryptedKey (см. п. 9.3 СТБ 34.101.23) и определяет ключ выработки имитовставки, зашифрованный на ключе шифрования ключей;

9) формат и содержание значения компонента ori (при его наличии) соответствуют типу OtherRecipientInfo (см. п. 9.4.4 СТБ 34.101.23):

– компонент oriType содержит значение типа OBJECT IDENTIFIER и определяет метод управления ключами;

– компонент oriValue содержит значение, тип которого определяется значением компонента oriType, и определяет данные, необходимые для применения выбранного метода управления.

7 Возвратить ОШИБКА.

6.2.1.8 Аутентифицируемые конвертованные данные

При тестировании процедуры формирования криптографических сообщений с аутентифицируемыми конвертованными данными выполняется следующий тест.

Тест CreateAEDTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования криптографических сообщений с аутентифицируемыми конвертованными данными.

2 Средствами испытываемой программы сформировать криптографическое сообщение с аутентифицируемыми конвертованными данными и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих сообщение.

3 Преобразовать файл, полученный на шаге 2, в текстовое представление криптографического сообщения.

4 Провести визуальный анализ текстового представления сформированного сообщения на соответствие п. 6 и п. 13 СТБ 34.101.23.

5 Повторить шаги 1 – 4 не менее 4 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).

6 Возвратить УСПЕХ, если на шаге 4 при визуальном анализе сформированного сообщения не выявлено несоответствий СТБ 34.101.23, при этом:

1) сообщение является значением типа ContentInfo (см. п. 6 СТБ 34.101.23), для которого:

– компонент contentType содержит значение id-ct-authEnvelopedData (см. п. 13.1 СТБ 34.101.23) типа ContentType (см. приложение А.1 СТБ 34.101.23);

– компонент content содержит значение типа AuthEnvelopedData (см. п. 13.2 СТБ 34.101.23) и определяет аутентифицируемые конвертованные данные;

2) формат и содержание компонента content соответствуют типу AuthEnvelopedData (см. п. 13.2 СТБ 34.101.23):

- компонент `version` содержит значение 0 типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23);
- необязательный компонент `originatorInfo` (при его наличии) содержит значение типа `OriginatorInfo` (см. п. 9.2 СТБ 34.101.23) и определяет информацию об отправителе; компонент присутствует, только если его требуется использовать в алгоритме управления ключами; компонент может содержать сертификаты, представленные в компоненте типа `CertificateSet` (см. п. 14.3 и 14.4 СТБ 34.101.23), и списки отозванных сертификатов, представленные в компоненте типа `RevocationInfoChoices` (см. п. 14.2 СТБ 34.101.23);
- компонент `recipientInfos` содержит значение типа `RecipientInfos` (см. п. 9.2 и п. 9.3 СТБ 34.101.23) и определяет информацию, связанную с получателями и представленную значениями типа `RecipientInfo`; в данном компоненте должна быть представлена информация по крайней мере для одного получателя;
- компонент `authEncryptedContentInfo` содержит значение типа `EncryptedContentInfo` (см. п. 9.2 СТБ 34.101.23) и определяет зашифрованные данные;
- необязательный компонент `authAttrs` (при его наличии) содержит значение типа `AuthAttributes` (см. п. 12.2 СТБ 34.101.23) и определяет набор аутентифицируемых атрибутов типа `Attribute` (см. п. 15.1 СТБ 34.101.23); компонент должен присутствовать, если значение типа `EncryptedContentInfo` содержит данные, тип которых отличается от `id-data`; значение типа `AuthAttributes` должно быть закодировано с помощью отличительных правил, даже если все остальные значения типа `AuthEnvelopedData` закодированы с помощью базовых правил; данный компонент не должен содержать атрибут «хэш-значение» (см. 15.3 СТБ 34.101.23);
- компонент `mac` содержит значение типа `MessageAuthenticationCode` (см. п. 12.2 СТБ 34.101.23) и определяет имитовставку, вычисленную с помощью выбранного алгоритма одновременного шифрования и имитозащиты;
- необязательный компонент `unauthAttrs` (при его наличии) содержит значение типа `UnauthAttributes` (см. п. 12.2 СТБ 34.101.23) и определяет неаутентифицируемые атрибуты типа `Attribute` (см. п. 15.1 СТБ 34.101.23);

3) формат и содержание значения компонента `authEncryptedContentInfo` соответствуют типу `EncryptedContentInfo` (см. п. 9.2 СТБ 34.101.23):

- компонент `contentType` содержит значение типа `ContentType` (см. приложение А.1 СТБ 34.101.23) и определяет идентификатор типа данных;
- компонент `contentEncryptionAlgorithm` содержит значение типа `ContentEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры используемого алгоритма одновременного шифрования и имитозащиты данных;
- необязательный компонент `encryptedContent` (при его наличии) содержит значение типа `EncryptedContent` (см. п. 9.2 СТБ 34.101.23) и определяет зашифрованные данные (при отсутствии компонента зашифрованные данные должны быть получены из внешнего источника);

4) формат и содержание значения каждого элемента компонента `recipientInfos` соответствуют типу `RecipientInfo` (см. п. 9.3 СТБ 34.101.23), который является выбором одного из следующих компонентов:

- компонент `ktri` содержит значение типа `KeyTransRecipientInfo` (см. п. 9.4 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для управления ключами одновременного шифрования и имитозащиты данных используется транспорт ключа;
- компонент `kari` содержит значение типа `KeyAgreeRecipientInfo` (см. п. 9.4.1 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для управления ключами одновременного шифрования и имитозащиты данных используется согласование ключа;
- компонент `kekri` содержит значение типа `KEKRecipientInfo` (см. п. 9.4.2 СТБ 34.101.23) и определяет информацию, касающуюся получателей при использовании заранее распределенных ключей одновременного шифрования и имитозащиты данных;
- компонент `pwri` содержит значение типа `PasswordRecipientinfo` (см. п. 9.4.3 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда для формирования ключей одновременного шифрования и имитозащиты данных используются пароли;
- компонент `ori` содержит значение типа `OtherRecipientInfo` (см. п. 9.4.4 СТБ 34.101.23) и определяет информацию, касающуюся получателей в случае, когда применяются дополнительные методы управления ключами одновременного шифрования и имитозащиты данных;

5) формат и содержание значения компонента `ktri` (при его наличии) соответствуют типу `KeyTransRecipientInfo` (см. п. 9.4 СТБ 34.101.23):

- компонент `version` содержит значение типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23), и определяет номер версии применяемого синтаксиса в зависимости от значения компонента `rid`;
- компонент `rid` содержит значение типа `RecipientIdentifier` (см. п. 9.4 СТБ 34.101.23) и определяет ссылку на сертификат получателя, в том числе ссылку на открытый ключ получателя; при этом, если в `rid` выбран компонент типа `IssuerAndSerialNumber` (см. п. 14.5 СТБ 34.101.23), то значение `version` должно принимать значение 0, а если выбран компонент типа `SubjectKeyIdentifier` (см. подпункт 6.2.1.2 СТБ 34.101.19), то – значение 2;
- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет для получателя идентификатор и параметры алгоритма, используемые для шифрования ключа одновременного шифрования и имитозащиты данных;
- компонент `encryptedKey` содержит значение типа `EncryptedKey` (см. п. 9.3 СТБ 34.101.23), и определяет для получателя зашифрованный ключ одновременного шифрования и имитозащиты данных;

6) формат и содержание значения компонента `kari` (при его наличии) соответствуют типу `KeyAgreeRecipientInfo` (см. п. 9.4.1 СТБ 34.101.23):

- компонент `version` содержит значение 3 типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23);
- компонент `originator` содержит значение типа `OriginatorIdentifierOrKey` (см. п. 9.4.1 СТБ 34.101.23) и определяет открытый ключ отправителя одним из трех способов: с использованием значения типа `IssuerAndSerialNumber`, с использованием значения типа `SubjectKeyIdentifier` (см. подпункт 6.2.1.2 СТБ 34.101.19), с использованием значения типа `OriginatorPublicKey` (см. п. 9.4.1 СТБ 34.101.23);
- необязательный компонент `ukm` (при его наличии) содержит значение типа `UserKeyingMaterial` (см. п. 14.7 СТБ 34.101.23) и определяет дополнительные данные алгоритма согласования общего ключа;
- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма, который используется для шифрования ключа одновременного шифрования и имитозащиты данных;
- компонент `recipientEncryptedKeys` содержит значение типа `RecipientEncryptedKeys` (см. п. 9.4.1 СТБ 34.101.23) и определяет список идентификаторов получателей и предназначенных им зашифрованных ключей одновременного шифрования и имитозащиты данных;

7) формат и содержание значения компонента `kekri` (при его наличии) соответствуют типу `KEKRecipientInfo` (см. п. 9.4.2 СТБ 34.101.23):

- компонент `version` содержит значение 4 типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23);
- компонент `kekid` содержит значение типа `KEKIdentifier` (см. п. 9.4.2 СТБ 34.101.23) и указывает на секретный ключ шифрования ключей, который был предварительно распределен отправителю и одному или нескольким получателям;
- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма шифрования ключа одновременного шифрования и имитозащиты данных;
- компонент `encryptedKey` содержит значение типа `EncryptedKey` (см. п. 9.3 СТБ 34.101.23) и определяет ключ одновременного шифрования и имитозащиты данных, зашифрованный на ключе шифрования ключей;

8) формат и содержание значения компонента `pwri` (при его наличии) соответствуют типу `PasswordRecipientinfo` (см. п. 9.4.3 СТБ 34.101.23):

- компонент `version` содержит значение 0 типа `CMSVersion`, соответствующего типу `INTEGER` (см. п. 14.6 СТБ 34.101.23);
- компонент `kekid` содержит значение типа `KEKIdentifier` (см. п. 9.4.2 СТБ 34.101.23) и указывает на секретный ключ одновременного шифрования и имитозащиты данных, который был предварительно распределен отправителю и одному или нескольким получателям;
- необязательный компонент `keyDerivationAlgorithm` (при его наличии) содержит значение типа `KeyDerivationAlgorithm Identifier` (см. п. 14.1

СТБ 34.101.23) и определяет идентификатор и параметры алгоритма выработки ключа шифрования ключей по паролю или другому общему секрету;

- компонент `keyEncryptionAlgorithm` содержит значение типа `KeyEncryptionAlgorithmIdentifier` (см. п. 14.1 СТБ 34.101.23) и определяет идентификатор и параметры алгоритма шифрования ключа одновременного шифрования и имитозащиты данных;
- компонент `encryptedKey` содержит значение типа `EncryptedKey` (см. п. 9.3 СТБ 34.101.23) и определяет ключ одновременного шифрования и имитозащиты данных, зашифрованный на ключе шифрования ключей;

9) формат и содержание значения компонента `ori` (при его наличии) соответствуют типу `OtherRecipientInfo` (см. п. 9.4.4 СТБ 34.101.23):

- компонент `oriType` содержит значение типа `OBJECT IDENTIFIER` и определяет метод управления ключами;
- компонент `oriValue` содержит значение, тип которого определяется значением компонента `oriType`, и определяет данные, необходимые для применения выбранного метода управления.

7 Возвратить `ОШИБКА`.

6.2.2 Процедуры разбора криптографических сообщений

6.2.2.1 Общие сведения

Входными данными, задаваемыми в процессе тестирования процедур разбора криптографических сообщений, являются криптографические сообщения, представленные в виде бинарных файлов, содержащих закодированные значения типов АСН.1.

При тестировании процедур разбора криптографических сообщений выполняются базовые тесты и тесты известного ответа. Базовые тесты являются обязательными для выполнения. Тесты известного ответа являются дополнительными и предназначены для разбора криптографических сообщений с неструктурированными данными, а также криптографических сообщений, сформированных с использованием криптографических алгоритмов, определенных в СТБ 34.101.31 и СТБ 34.101.45.

Для базовых тестов файлы с криптографическими сообщениями предоставляются совместно с испытываемой программой или формируются экспертом (при наличии необходимого программного обеспечения). Перед тестированием процедур разбора криптографических сообщений эксперт проводит визуальное сравнение текстового представления сообщений с форматами, определенными в СТБ 34.101.23. Для изменения криптографических сообщений, которое выполняется в некоторых базовых тестах, могут использоваться программы, описанные в приложении Г.2. Задаваемые в базовых тестах криптографические сообщения должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по разбору сообщений.

Для тестов известного ответа в качестве входных данных используются криптографические сообщения, определенные в приложении Д (закодированные АСН.1-файлы с криптографическими сообщениями являются неотъемлемой частью настоящей методики).

6.2.2.2 Неструктурированные данные

Базовые тесты. При тестировании процедуры разбора криптографических сообщений с неструктурированными данными выполняются следующие базовые тесты.

Тест VProcUDTest

1 Задать корректное криптографическое сообщение с неструктурированными данными.
2 Средствами испытываемой программы выполнить разбор криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные корректные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения успешно обработаны и признаны корректными.
5 Возвратить `ОШИБКА`.

Тест IProcUDTest

1 Изменить криптографическое сообщение с неструктурированными данными таким образом, чтобы в соответствии со СТБ 34.101.23 сообщение стало некорректным.
2 Средствами испытываемой программы выполнить разбор измененного криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные измененные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения признаны некорректными.
5 Возвратить `ОШИБКА`.

Примечание — Изменение криптографического сообщения с неструктурированными данными может состоять, например, в изменении значения компонента `contentType` (см. п. 6 СТБ 34.101.23).

Тесты известного ответа. При тестировании процедуры разбора криптографических сообщений с неструктурированными данными выполняются следующие тесты известного ответа.

Тест VUDTest

1 Задать в качестве криптографического сообщения файл «VUDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение успешно обработано и признано корректным.
4 Возвратить `ОШИБКА`.

Примечание 1 — Цель теста VUDTest — проверка способности реализации обрабатывать корректные криптографические сообщения с неструктурированными данными.
Примечание 2 — Испытуемая программа может не распознавать данные контейнера, так как их структура может не поддерживаться испытываемой программой.

Тест IContentTypeTest

1 Задать в качестве криптографического сообщения файл «IContentTypeTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `contentType` (см. п. 6 СТБ 34.101.23). Цель теста `IContentTypeTest` — проверка способности реализации выявлять некорректные идентификаторы, определяющие тип данных контейнера.

6.2.2.3 Подписанные данные

Базовые тесты. При тестировании процедуры разбора криптографических сообщений с подписанными данными выполняются следующие базовые тесты.

Тест VProcSDTest

1 Задать корректное криптографическое сообщение с подписанными данными.
2 Средствами испытываемой программы выполнить разбор криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные корректные криптографические сообщения.
4 Тест выполнен успешно, если на шаге 2 криптографические сообщения успешно обработаны и признаны корректными.
5 Возвратить `ОШИБКА`.

Тест IProcSDTest

1 Изменить криптографическое сообщение с подписанными данными таким образом, чтобы в соответствии со СТБ 34.101.23 сообщение стало некорректным.
2 Средствами испытываемой программы выполнить разбор измененного криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные измененные криптографические сообщения.
4 Тест выполнен успешно, если на шаге 2 криптографические сообщения признаны некорректными.
5 Возвратить `ОШИБКА`.

Примечание — Изменение криптографического сообщения с подписанными данными может состоять, например, в изменении значения компонента `signature` (см. п. 8.4 СТБ 34.101.23).

Тесты известного ответа. При тестировании процедуры разбора криптографических сообщений с подписанными данными выполняется тест известного ответа `IContentTypeTest` (см. п. 6.2.2.2), а также следующие тесты известного ответа.

Тест VSDTest

1 Задать в качестве криптографического сообщения файл «VSDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение успешно обработано и признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Цель теста VSDTest — проверка способности реализации обрабатывать корректные криптографические сообщения с подписываемыми данными.

Тест ISignatureSDTest

1 Задать в качестве криптографического сообщения файл «ISignatureSDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `signature` (см. п. 8.4 СТБ 34.101.23). Цель теста ISignatureSDTest — проверка способности реализации выявлять некорректные ЭЦП.

Тест IVersionSDTest

1 Задать в качестве криптографического сообщения файл «IVersionSDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `SignedData.version` (см. п. 8.2 СТБ 34.101.23). Цель теста IVersionSDTest — проверка способности реализации выявлять некорректные значения для версии применяемого синтаксиса.

Тест IDigAlgIdSDTest

1 Задать в качестве криптографического сообщения файл «IDigAlgIdSDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `SignedData.digestAlgorithms.algorithm` (см. п. 8.2 СТБ 34.101.23). Цель теста IDigAlgIdSDTest — проверка способности реализации выявлять некорректные значения идентификатора алгоритма хэширования.

Тест IContentSDTest

1 Задать в качестве криптографического сообщения файл «IContentSDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит измененное значение подписываемых данных. Цель теста IContentSDTest — проверка способности реализации выявлять модификацию подписываемых данных.

Тест VSidSDTest

1 Задать в качестве криптографического сообщения файл «VSidSDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение успешно обработано и признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Цель теста VSidSDTest — проверка способности реализации обрабатывать корректные криптографические сообщения с подписываемыми данными, для которого в компоненте `sid` ссылка на сертификат подписывающей стороны представлена значением типа `SubjectKeyIdentifier` (см. п. 6.2.1.2 СТБ 34.101.19).

Тест ISignAttrSDTest

1 Задать в качестве криптографического сообщения файл «ISignAttrSDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит измененное значение подписываемых атрибутов. Цель теста ISignAttrSDTest — проверка способности реализации выявлять модификацию подписываемых атрибутов.

Тест IDigAlgIdSISDTest

1 Задать в качестве криптографического сообщения файл «IDigAlgIdSISDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `SignedData.signerInfos.digestAlgorithms.algorithm` (см. п. 8.4 СТБ 34.101.23). Цель теста IDigAlgIdSISDTest — проверка способности реализации выявлять некорректные значе-

ния для идентификатора алгоритма хэширования, используемого при выработке и проверке ЭЦП.

Тест ISignAlgIdSISDTest

1 Задать в качестве криптографического сообщения файл «ISignAlgIdSISDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `SignedData.signerInfos.signatureAlgorithms.algorithm` (см. п. 8.4 СТБ 34.101.23). Цель теста ISignAlgIdSISDTest — проверка способности реализации выявлять некорректные значения идентификатора алгоритмов ЭЦП.

Тест ISignAlgPrmSISDTest

1 Задать в качестве криптографического сообщения файл «ISignAlgPrmSISDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `SignedData.signerInfos.signatureAlgorithms.parameters` (см. приложение Д.2 СТБ 34.101.45). Цель теста ISignAlgPrmSISDTest — проверка способности реализации выявлять некорректные значения параметров алгоритмов ЭЦП.

6.2.2.4 Конвертованные данные

Базовые тесты. При тестировании процедуры разбора криптографических сообщений с конвертованными данными выполняются следующие базовые тесты.

Тест VProcEDTest

1 Задать корректное криптографическое сообщение с конвертованными данными.
2 Средствами испытываемой программы выполнить разбор криптографического сообщения.
3 Повторить шаги 1 — 2 не менее 4 раз, задавая различные корректные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения успешно обработаны и признаны корректными.
5 Возвратить `ОШИБКА`.

Тест IProcEDTest

1 Изменить криптографическое сообщение с конвертованными данными таким образом, чтобы в соответствии со СТБ 34.101.23 сообщение стало некорректным.

2 Средствами испытываемой программы выполнить разбор измененного криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные измененные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения признаны некорректными.
5 Возвратить `ОШИБКА`.

Примечание — Изменение криптографического сообщения с конвертованными данными может состоять, например, в изменении значения компонента `signature` (см. п. 8.4 СТБ 34.101.23).

Тесты известного ответа. При тестировании процедуры разбора криптографических сообщений с конвертованными данными выполняется тест известного ответа `IContentTypeTest` (см. п. 6.2.2.2), а также следующие тесты известного ответа.

Тест VEDTest

1 Задать в качестве криптографического сообщения файл «VEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение успешно обработано и признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Цель теста VEDTest — проверка способности реализации обрабатывать корректные криптографические сообщения с вложенными данными.

Тест IEncrKeyEDTest

1 Задать в качестве криптографического сообщения файл «IEncrKeyEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение защищенного ключа, представленное в компоненте `encryptedKey` (см. п. 9.4 СТБ 34.101.23), вложенной в компоненту `recipientInfos` (см. п. 9.2 СТБ 34.101.23). Цель теста IEncrKeyEDTest — проверка способности реализации выявлять некорректную имитозащиту ключа.

Тест IVersionEDTest

1 Задать в качестве криптографического сообщения файл «IVersionEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `EnvelopedData.version` (см. п. 9.2 СТБ 34.101.23). Цель теста IVersionEDTest — проверка

способности реализации выявлять некорректные значения для версии применяемого синтаксиса.

Тест IKEAlgIdEDTest

1 Задать в качестве криптографического сообщения файл «IKEAlgIdEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `EnvelopedData.recipientInfos.ktri.keyEncryptionAlgorithm.algorithm` (см. п. 9.4 СТБ 34.101.23). Цель теста IKEAlgIdEDTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма шифрования ключа шифрования данных.

Тест ICEAlgIdEDTest

1 Задать в качестве криптографического сообщения файл «ICEAlgIdEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `EnvelopedData`. `encryptedContentInfo.contentEncryptionAlgorithm.algorithm` (см. п. 9.2 СТБ 34.101.23). Цель теста ICEAlgIdEDTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма шифрования данных.

6.2.2.5 Хэшированные данные

Базовые тесты. При тестировании процедуры разбора криптографических сообщений с хэшированными данными выполняются следующие базовые тесты.

Тест VProcDDTest

1 Задать корректное криптографическое сообщение с хэшированными данными.
2 Средствами испытываемой программы выполнить разбор криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные корректные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения успешно обработаны и признаны корректными.
5 Возвратить `ОШИБКА`.

Тест IProcDDTest

1 Изменить криптографическое сообщение с хэшированными данными таким образом, чтобы в соответствии со СТБ 34.101.23 сообщение стало некорректным.

2 Средствами испытываемой программы выполнить разбор измененного криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные измененные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения признаны некорректными.
5 Возвратить `ОШИБКА`.

Примечание — Изменение криптографического сообщения с хэшированными данными может состоять, например, в изменении значения компонента `digest` (см. п. 10.2 СТБ 34.101.23).

Тесты известного ответа. При тестировании процедуры разбора криптографических сообщений с хэшированными данными выполняется тест известного ответа `IContentTypeTest` (см. п. 6.2.2.2), а также следующие тесты известного ответа.

Тест VDDTest

1 Задать в качестве криптографического сообщения файл «VDDTest.bin».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение успешно обработано и признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Цель теста VDDTest — проверка способности реализации обрабатывать корректные криптографические сообщения с хэшированными данными.

Тест IDigestDDTest

1 Задать в качестве криптографического сообщения файл «IDigestDDTest.bin».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное хэш-значение, представленное в компоненте `digest` (см. п. 10.2 СТБ 34.101.23). Цель теста IDigestDDTest — проверка способности реализации выявлять некорректные хэш-значения данных.

Тест IVersionDDTest

1 Задать в качестве криптографического сообщения файл «IVersionDDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `DigestedData.version` (см. п. 10.2 СТБ 34.101.23). Цель теста IVersionDDTest — проверка способности реализации выявлять некорректные значения для версии применяемого синтаксиса.

Тест IDigAlgIdDDTest

1 Задать в качестве криптографического сообщения файл «IDigAlgIdDDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `DigestedData.digestAlgorithm.algorithm` (см. п. 10.2 СТБ 34.101.23). Цель теста IDigAlgIdDDTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма хэширования.

6.2.2.6 Шифрованные данные

Базовые тесты. При тестировании процедуры разбора криптографических сообщений с шифрованными данными выполняются следующие базовые тесты.

Тест VProcEncrDTest

1 Задать корректное криптографическое сообщение с шифрованными данными.
2 Средствами испытываемой программы выполнить разбор криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные корректные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения успешно обработаны и признаны корректными.
5 Возвратить `ОШИБКА`.

Тест IProcEncrDTest

1 Изменить криптографическое сообщение с шифрованными данными таким образом, чтобы в соответствии со СТБ 34.101.23 сообщение стало некорректным.
2 Средствами испытываемой программы выполнить разбор измененного криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные измененные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения признаны некорректными.
5 Возвратить `ОШИБКА`.

Примечание — Изменение криптографического сообщения с шифрованными данными может состоять, например, в изменении значения компонента `version` (см. п. 11.2 СТБ 34.101.23).

Тесты известного ответа. При тестировании процедуры разбора криптографических сообщений с шифрованными данными выполняется тест известного ответа `IContentTypeTest` (см. п. 6.2.2.2), а также следующие тесты известного ответа.

Тест VEncrDTest

1 Задать в качестве криптографического сообщения файл «VEncrDTest.bin».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение успешно обработано и признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Цель теста VEncrDTest — проверка способности реализации обрабатывать корректные криптографические сообщения с шифрованными данными.

Тест IVerEncrDTest

1 Задать в качестве криптографического сообщения файл «IVerEncrDTest.bi»».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `version` (см. п. 11.2 СТБ 34.101.23). Цель теста IVerEncrDTest — проверка способности реализации выявлять некорректные значения версии, применяемого синтаксиса.

Тест IEAlgIdEncrDTest

1 Задать в качестве криптографического сообщения файл «IEAlgIdEncrDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение `EncryptedData.encryptedContentInfo.contentEncryptionAlgorithm.algorithm` (см. п. 11.2 СТБ 34.101.23). Цель теста IEAlgIdEncrDTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма шифрования данных.

6.2.2.7 Аутентифицируемые данные

Базовые тесты. При тестировании процедуры разбора криптографических сообщений с аутентифицируемыми данными выполняются следующие базовые тесты.

Тест VProcADTest

1 Задать корректное криптографическое сообщение с аутентифицируемыми данными.
2 Средствами испытываемой программы выполнить разбор криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные корректные криптографические сообщения.

4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения успешно обработаны и признаны корректными.
5 Возвратить `ОШИБКА`.

Тест IProcADTest

1 Изменить криптографическое сообщение с аутентифицируемыми данными таким образом, чтобы в соответствии со СТБ 34.101.23 сообщение стало некорректным.
2 Средствами испытываемой программы выполнить разбор измененного криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные измененные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения признаны некорректными.
5 Возвратить `ОШИБКА`.

Примечание — Изменение криптографического сообщения с аутентифицируемыми данными может состоять, например, в изменении значения компонента `mac` (см. п. 12.2 СТБ 34.101.23).

Тесты известного ответа. При тестировании процедуры разбора криптографических сообщений с аутентифицируемыми данными выполняется тест известного ответа `IContentTypeTest` (см. п. 6.2.2.2), а также следующие тесты известного ответа.

Тест VADTest

1 Задать в качестве криптографического сообщения файл «VADTest.bin».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение успешно обработано и признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Цель теста VADTest — проверка способности реализации обрабатывать корректные криптографические сообщения с аутентифицируемыми данными.

Тест IEncrKeyADTest

1 Задать в качестве криптографического сообщения файл «IEncrKeyADTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение защищенного ключа, представленное в компоненте `encryptedKey` (см. п. 9.4 СТБ 34.101.23), вложенной в компоненту `recipientInfos` (см. п. 12.2 СТБ 34.101.23). Цель теста IEncrKeyADTest — проверка способности реализации выявлять некорректную имитозащиту ключа.

Тест IMacADTest

1 Задать в качестве криптографического сообщения файл «IMacADTest.bin».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание – Криптографическое сообщение содержит некорректное значение компонента `AuthenticatedData.mac` (см. п. 12.2 СТБ 34.101.23). Цель теста IMacADTest — проверка способности реализации выявлять некорректные значения имитовставки.

Тест IVersionADTest

1 Задать в качестве криптографического сообщения файл «IVersionADTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `AuthenticatedData.version` (см. п. 12.2 СТБ 34.101.23). Цель теста IVersionADTest — проверка способности реализации выявлять некорректные значения для версии применяемого синтаксиса.

Тест IKEAlgIdADTest

1 Задать в качестве криптографического сообщения файл «IKEAlgIdADTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `AuthenticatedData.recipientInfos.ktri.keyEncryptionAlgorithm.algorithm` (см. п. 9.4 СТБ 34.101.23). Цель теста IKEAlgIdADTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма шифрования ключа имитозащиты данных.

Тест IMacAlgIdADTest

1 Задать в качестве криптографического сообщения файл «IMacAlgIdADTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение для компонента `AuthenticatedData.macAlgorithm.algorithm` (см. п. 12.2 СТБ 34.101.23). Цель теста

IMacAlgIdADTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма имитозащиты данных

Тест VAttrADTest

1 Задать в качестве криптографического сообщения файл «VAttrADTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит необязательные компоненты `AuthenticatedData.digestAlgorithm`, определяющий алгоритм хэширования, и `AuthenticatedData.authAttrs`, определяющий аутентифицируемые атрибуты «тип-содержимого» и «хэш-значение» (см. п. 12.2 СТБ 34.101.23). Цель теста VAttrADTest — проверка способности реализации обрабатывать аутентифицируемые атрибуты «тип содержимого» и «хэш-значение» (см. п. 15 СТБ 34.101.23).

Тест IDigAlgADTest

1 Задать в качестве криптографического сообщения файл «IDigAlgADTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит необязательные компоненты `AuthenticatedData.digestAlgorithm`, определяющий алгоритм хэширования, и `AuthenticatedData.authAttrs`, определяющий аутентифицируемые атрибуты «тип-содержимого» и «хэш-значение» (см. п. 12.2 СТБ 34.101.23). Компонент `AuthenticatedData.digestAlgorithm` содержит некорректный идентификатор алгоритма хэширования. Цель теста IDigAlgADTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма хэширования.

Тест IAttrADTest

1 Задать в качестве криптографического сообщения файл «IAttrADTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит необязательные компоненты `AuthenticatedData.digestAlgorithm`, определяющий алгоритм хэширования, и `AuthenticatedData.authAttrs`, определяющий аутентифицируемые атрибуты «тип-содержимого» и «хэш-значение» (см. п. 12.2 СТБ 34.101.23). Компонент `AuthenticatedData.authAttrs` содержит измененное значение атрибута «хэш-значение»» (см. п. 15 СТБ 34.101.23). Цель теста IAttrADTest — проверка способности реализации выявлять измененное значение аутентифицируемого атрибута.

6.2.2.8 Аутентифицируемые конвертованные данные

Базовые тесты. При тестировании процедуры разбора криптографических сообщений с аутентифицируемыми конвертованными данными выполняются следующие базовые тесты.

Тест VProcAEDTest

1 Задать корректное криптографическое сообщение с аутентифицируемыми конвертованными данными.
2 Средствами испытываемой программы выполнить разбор криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные корректные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения успешно обработаны и признаны корректными.
5 Возвратить `ОШИБКА`.

Тест IProcAEDTest

1 Изменить криптографическое сообщение с аутентифицируемыми конвертованными данными таким образом, чтобы в соответствии со СТБ 34.101.23 сообщение стало некорректным.
2 Средствами испытываемой программы выполнить разбор измененного криптографического сообщения.
3 Повторить шаги 1 – 2 не менее 4 раз, задавая различные измененные криптографические сообщения.
4 Возвратить `УСПЕХ`, если на шаге 2 криптографические сообщения признаны некорректными.
5 Возвратить `ОШИБКА`.

Примечание — Изменение криптографического сообщения с аутентифицируемыми конвертованными данными может состоять, например, в изменении значения компонента `mac` (см. п. 12.2 СТБ 34.101.23).

Тесты известного ответа. При тестировании процедуры разбора криптографических сообщений с аутентифицируемыми конвертованными данными выполняется тест известного ответа `IContentTypeTest` (см. п. 6.2.2.2), а также следующие тесты известного ответа.

Тест VAEDTest

1 Задать в качестве криптографического сообщения файл «VAEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение успешно обработано и признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Цель теста VAEDTest — проверка способности реализации обрабатывать корректные криптографические сообщения с аутентифицируемыми конвертованными данными.

Тест IEncrKeyAEDTest

1 Задать в качестве криптографического сообщения файл «IEncrKeyAEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение защищенного ключа, представленное в компоненте `encryptedKey` (см. п. 9.4 СТБ 34.101.23), вложенной в компоненту `recipientInfos` (см. п. 12.2 СТБ 34.101.23). Цель теста IEncrKeyAEDTest — проверка способности реализации выявлять некорректную имитозащиту ключа.

Тест IMacAEDTest

1 Задать в качестве криптографического сообщения файл «IMacAEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `AuthEnvelopedData.mac` (см. п. 12.2 СТБ 34.101.23). Цель теста IMacAEDTest — проверка способности реализации выявлять некорректные значения имитовставки.

Тест IVersionAEDTest

1 Задать в качестве криптографического сообщения файл «IVersionAEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение `AuthEnvelopedData.version` (см. п. 13.2 СТБ 34.101.23). Цель теста AuthEnvelopedData — проверка способности реализации выявлять некорректные значения для версии применяемого синтаксиса.

Тест IKEAlgIdAEDTest

1 Задать в качестве криптографического сообщения файл «IKEAlgIdAEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.
5 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение компонента `AuthEnvelopedData.recipientInfos.ktri.keyEncryptionAlgorithm.algorithm` (см. п. 9.4 СТБ 34.101.23). Цель теста IKEAlgIdAEDTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма шифрования ключа шифрования и имитозащиты данных.

Тест IAEAlgIdAEDTest

1 Задать в качестве криптографического сообщения файл «IAEAlgIdAEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит некорректное значение `AuthEnvelopedData.encryptedContentInfo.contentEncryptionAlgorithm.algorithm` (см. п. 13.2 СТБ 34.101.23). Цель теста IAEAlgIdAEDTest — проверка способности реализации выявлять некорректные значения для идентификатора алгоритма одновременного шифрования и имитозащиты данных.

Тест VAttrAEDTest

1 Задать в качестве криптографического сообщения файл «VAttrAEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано корректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит необязательный компонент `AuthEnvelopedData.authAttrs`, определяющий аутентифицируемый атрибут «тип-содержимого» (см. п. 13.2 СТБ 34.101.23). Цель теста VAttrAEDTest — проверка способности реализации обрабатывать аутентифицируемый атрибут «тип содержимого» (см. п. 15 СТБ 34.101.23).

Тест IAttrAEDTest

1 Задать в качестве криптографического сообщения файл «IAttrAEDTest.der».
2 Средствами испытываемой программы выполнить процедуру разбора криптографического сообщения.
3 Возвратить `УСПЕХ`, если криптографическое сообщение признано некорректным.
4 Возвратить `ОШИБКА`.

Примечание — Криптографическое сообщение содержит необязательный компонент `AuthenticatedData.authAttrs`, определяющий аутентифицируемый атрибут «тип-содержимого» (см. п. 13.2 СТБ 34.101.23). Компонент `AuthenticatedData.authAttrs` содержит измененное значение атрибута «тип содержимого» (см. п. 15 СТБ 34.101.23). Цель теста IAttrADTest — проверка способности реализации выявлять измененное значение аутентифицируемого атрибута.

6.3 Анализ исходных текстов

6.3.1 Корректность использования криптографических алгоритмов

Криптографические алгоритмы используются при формировании и разборе всех криптографических форматов за исключением неструктурированных данных.

Использование функций, реализующих криптографический алгоритм, должно выполняться в соответствии с СТБ 34.101.23, ТНПА на криптографический алгоритм и документацией на испытываемую программу. Для каждого вызова в программе функций, реализующих криптографический алгоритм, эксперт выполняет следующие проверки:

1 Типы и значения параметров, фактически переданных в функцию, соответствуют типам и допустимым значениям параметров функции (с учетом стандартных правил преобразования типов языка программирования).

2 Если функция возвращает значение, то проводится анализ корректности использования возвращаемого значения, например, корректность использования в операторе присваивания, допустимость игнорирования возвращаемого значения и т.п.

3 Если вызов функции может привести к возникновению исключительной ситуации или ошибки, проверяется наличие и корректность обработки исключительной ситуации.

4 Если до и после вызова функции должны выполняться определенные действия, то проверяется наличие и корректность выполнения требуемых действий.

5 Если функция использует глобальные переменные, то проверяется наличие инициализации данных переменных.

Примечание — Под функцией понимается часть программы, которая выполняет специфические действия и описывается типом возвращаемого значения, именем функции, формальными параметрами. Выполнение функции осуществляется посредством вызова из программы или другой функции. Данному термину в языках программирования соответствуют такие понятия как «функция», «процедура», «метод» и т.п.

6.3.2 Корректность управления секретными данными

Секретные данные — это ключи, параметры и другие данные криптографических алгоритмов, значения которых в соответствии со стандартом или документацией на СКЗИ должны быть защищены от раскрытия, т.е. должны храниться в секрете.

Эксперт проверяет, что секретные данные используются в строгом соответствии с криптографическим алгоритмом. Другие операции с секретными данными не допускаются.

Эксперт проверяет, что все копии секретных данных в открытом виде уничтожаются при завершении работы с ними, при этом:

– значение секретных данных, размещенное в области памяти глобальной переменной, уничтожается перед каждым выходом из программы;

– значение секретных данных, размещенное в области памяти локальной переменной функции, уничтожается перед каждым выходом из данной функции;

– значение секретных данных, размещенное в динамической памяти, уничтожается перед каждым освобождением динамической памяти.

Примечание – Под уничтожением понимается такое изменение данных, хранящихся в электронных устройствах (оперативная память, память на магнитных носителях и др.), которое предотвращает их последующее восстановление. Например, уничтожение может состоять в записи в области памяти, занимаемой значениями секретных данных, фиксированных или случайно выбранных значений.

6.3.3 Корректность процедур формирования криптографических сообщений

При анализе корректности реализации процедур формирования криптографических сообщений исходные тексты программы оцениваются частично, по выбору эксперта. Корректность реализации процедуры означает, что реализация функционально соответствует СТБ 34.101.23 и что реализация не содержит ошибок и уязвимостей.

Эксперт должен проверить по крайней мере следующие аспекты реализации процедуры формирования запроса на получение сертификата:

1 Реализация процедуры формирования криптографических сообщений с подписанными данными должна хэшировать и вырабатывать ЭЦП в соответствии с п. 8.5, 8.6 СТБ 34.101.23 и с учетом приложения Б СТБ 34.101.23.

2 Реализация процедуры формирования криптографических сообщений с конвертованными данными должна зашифровывать данные и ключи в соответствии с п. 9.5, 9.6 СТБ 34.101.23 и с учетом приложения Б СТБ 34.101.23.

3 Реализация процедуры формирования криптографических сообщений с хэшированными данными должна хэшировать в соответствии с п. 8.5 СТБ 34.101.23 (для случая, когда подписываемые атрибуты отсутствуют) и с учетом приложения Б СТБ 34.101.23.

4 Реализация процедуры формирования криптографических сообщений с шифрованными данными должна зашифровывать с учетом приложения Б СТБ 34.101.23.

5 Реализация процедуры формирования криптографических сообщений с аутентифицируемыми данными должна вычислять имитовставку в соответствии с п. 12.3 СТБ 34.101.23 и с учетом приложения Б СТБ 34.101.23.

6 Реализация процедуры формирования криптографических сообщений с аутентифицируемыми конвертованными данными должна выполнять зашифрование и имитозащиту входных данных и зашифрование ключей в соответствии с п. 13.3 и 13.4 СТБ 34.101.23 и с учетом приложения Б СТБ 34.101.23.

6.3.4 Корректность процедур разбора криптографических сообщений

При анализе корректности реализации процедур разбора криптографических сообщений исходные тексты программы оцениваются частично, по выбору эксперта. Корректность реализации процедуры означает, что реализация функционально соответствует СТБ 34.101.23 и что реализация не содержит ошибок и уязвимостей.

Эксперт должен проверить по крайней мере следующие аспекты реализации процедур разбора криптографических сообщений:

1 Реализация процедуры разбора криптографических сообщений с подписанными данными должна хэшировать и проверять ЭЦП в соответствии с п. 8.5, 8.7 СТБ 34.101.23 и с учетом приложения Б СТБ 34.101.23;

2 Реализация процедуры разбора криптографических сообщений с конвертованными данными должна расшифровывать данные и ключи в соответствии с п. 9.5, 9.6 СТБ 34.101.23 и с учетом приложения Б СТБ 34.101.23;

3 Реализация процедуры формирования криптографических сообщений с хэшированными данными должна хэшировать в соответствии с п. 8.5 СТБ 34.101.23 (для случая, когда подписываемые атрибуты отсутствуют) и с учетом приложения Б СТБ 34.101.23;

4 Реализация процедуры разбора криптографических сообщений с шифрованными данными должна расшифровывать с учетом приложения Б СТБ 34.101.23;

5 Реализация процедуры разбора криптографических сообщений с аутентифицируемыми данными должна проверять имитовставку в соответствии с п. 12.4 СТБ 34.101.23 и с учетом приложения Б СТБ 34.101.23;

6 Реализация процедуры разбора криптографических сообщений с аутентифицируемыми конвертованными данными должна выполнять расшифрование и контроль целостности входных данных и расшифрование ключей с учетом п. 13.3, 13.4 и приложения Б СТБ 34.101.23.

6.3.5 Корректность обработки исключительных ситуаций

Под исключительной ситуацией понимается ошибочная ситуация, возникающая при выполнении программы и требующая специальной обработки. Данному термину в языках программирования соответствует такие понятия как «ошибка», «исключение» и т.п.

Эксперт проверяет корректность обработки исключительных ситуаций при выполнении проверок, проводимых в п. 6.3.1 – 6.3.4.

Для анализа корректности обработки исключительных ситуаций эксперт проверяет, что:

1 После каждого вызова функции, выполнение которой может приводить к возникновению исключительной ситуации, имеются проверка на случай возникновения исключительной ситуации и соответствующая обработка исключительной ситуации.

2 При проверке и обработке исключительной ситуации учтены все возможные виды исключительных ситуаций, возникновение которых возможно согласно документации на вызываемую функцию.

3 Исключительные ситуации обрабатываются адекватно (возвращаются верные коды ошибок и сообщения об ошибках и т.п.).

6.3.6 Отсутствие недокументированных возможностей

Эксперт определяет отсутствие недокументированных возможностей по результатам проверок, выполненных в п. 6.3.1 – 6.3.5.

Обнаруженные недокументированные возможности отражаются в протоколе анализа исходных текстов или в приложении к нему.

Приложение А
Форма протокола анализа документации

Экз. {Поле 1}

Протокол № {Поле 2} **от** {Поле 3}
результатов анализа документации
программы {Поле 4}, реализующей управление криптографическими сообщениями согласно СТБ 34.101.23-2012

1. Документы:

№	Название документа	Номер
1	{Поле 5}	{Поле 6}
2	{Поле 7}	{Поле 8}
3	{Поле 9}	{Поле 10}
4	{Поле 11}	{Поле 12}

2. При анализе документации были выполнены следующие проверки:

№	Название проверки	Отметка о выполнении
1	Проверка документа «Спецификация»	{Поле 13}
2	Проверка документа «Текст программы»	{Поле 13}
3	Проверка документа «Описание программы»	{Поле 13}
4	Проверка документа «Руководство программиста»	{Поле 13}

3. Заключение по результатам анализа документации: документация {Поле 6}, {Поле 8}, {Поле 10}, {Поле 12} соответствует (не соответствует) программе объекта испытаний в части управления криптографическими сообщениями согласно СТБ 34.101.23-2012.

Эксперт, {Поле 15} {Поле 16}
{Поле 14}

В поле 1 указывается номер экземпляра протокола.

В поле 2 указывается номер, однозначно идентифицирующий протокол.

В поле 3 указывается дата составления протокола.

В поле 4 указывается название объекта испытаний в соответствии с представленной к испытаниям документацией.

В полях 5 и 6 указываются соответственно полное название документа «Спецификация» и его идентификационный/децимальный номер.

В полях 7 и 8 указываются соответственно полное название документа «Текст программы» и его идентификационный/децимальный номер.

В полях 9 и 10 указываются соответственно полное название документа «Описание программы» и его идентификационный/децимальный номер.

В полях 11 и 12 указываются соответственно полное название документа «Руководство программиста» и его идентификационный/децимальный номер.

В поле 13 указывается результат выполнения проверки: «положительно» — результат проверки положительный, «отрицательно» — результат проверки отрицательный. После завершения анализа документации и заполнения таблицы делается вывод о соответствии (не соответствии) документации программе объекта испытаний в части управления криптографическими сообщениями согласно СТБ 34.101.23. Вывод о соответствии делается только тогда, когда результаты всех проверок являются положительными.

В полях 14 и 16 указываются соответственно должность и Ф. И. О. эксперта.

В поле 15 ставится собственноручная подпись эксперта.

Информация об обнаруженных несоответствиях приводится в протоколе или приложении к протоколу в произвольной форме.

Приложение Б
Форма протокола тестирования

Экз. {Поле 1}

Протокол № {Поле 2} **от** {Поле 3}
результатов тестирования
программы {Поле 4}, реализующей управление криптографическими сообщениями согласно СТБ 34.101.23-2012

1. Файлы исходных текстов программ:

№	Имя файла	Хэш-значение
1	{Поле 5}	{Поле 6}
2	{Поле 5}	{Поле 6}
...	...	...

Хэш-значения для файлов вычислены согласно {Поле 7}.

2. В ходе тестирования объекта испытаний были выполнены следующие тесты:

№	Название теста	Отметка о выполнении
1	CreateUDTest	{Поле 8}
2	CreateSDTest	{Поле 8}
3	CreateEnvDTest	{Поле 8}
4	CreateDDTest	{Поле 8}
...	...	...

3. Заключение по результатам тестирования: программа {Поле 4} соответствует (не соответствует) требованиям, установленным в СТБ 34.101.23-2012.

Эксперт, {Поле 10} {Поле 11}
{Поле 9}

В поле 1 указывается номер экземпляра протокола.

В поле 2 указывается номер, однозначно идентифицирующий протокол.

В поле 3 указывается дата составления протокола.

В поле 4 указывается название объекта испытаний в соответствии с представленной к испытаниям документацией.

В поле 5 указываются имена исходных файлов программ объекта испытаний.

В поле 6 указывается значение функции хэширования для тестируемых файлов, вычисленное в соответствии со стандартом, указанным в поле 7. Разрешается использовать функции хэширования, определенные в СТБ 34.101.31 или СТБ 34.101.77.

В поле 8 указывается результат выполнения теста: «положительно» — тест завершен успешно, «отрицательно» — тест завершен с ошибкой; «не проводился» — тест не проводился, так как программа не поддерживает алгоритм или режим, определенный в тесте.

После завершения тестирования и заполнения таблицы делается вывод о соответствии (не соответствии) программной реализации объекта испытаний СТБ 34.101.23. Вывод о соответствии делается только тогда, когда все проводимые тесты выполнены успешно.

В полях 9, 11 указываются соответственно должность и Ф. И. О. эксперта.

В поле 10 ставится собственноручная подпись эксперта.

Приложение В
Форма протокола анализа исходных текстов

Экз. {Поле 1}

Протокол № {Поле 2} **от** {Поле 3}
результатов анализа исходных текстов
программы {Поле 4}, реализующей управление криптографическими сообщениями согласно СТБ 34.101.23-2012

1. Файлы исходных текстов программ:

№	Имя файла	Хэш-значение
1	{Поле 5}	{Поле 6}
2	{Поле 5}	{Поле 6}
	...	...

Хэш-значения для файлов вычислены согласно {Поле 7}.

2. В ходе анализа исходных текстов программ были выполнены следующие проверки:

№	Название проверки	Результат проверки
1	Корректность использования криптографических алгоритмов	{Поле 8}
2	Корректность использования секретных параметров	{Поле 8}
3	Корректность уничтожения значений секретных параметров	{Поле 8}
4	Корректность процедур формирования криптографических сообщений	{Поле 8}
5	Корректность процедур разбора криптографических сообщений	{Поле 8}
6	Корректность обработки исключительных ситуаций	{Поле 8}
7	Отсутствие недокументированных возможностей	{Поле 8}

3. Заключение по результатам анализа исходных текстов программ: программа {Поле 4} соответствует требованиям, установленным в СТБ 34.101.23–2012.

Эксперт, {Поле 9} {Поле 10} {Поле 11}

В поле 1 указывается номер экземпляра протокола.

В поле 2 указывается номер, однозначно идентифицирующий протокол.

В поле 3 указывается дата составления протокола.

В поле 4 указывается название объекта испытаний в соответствии с представленной к испытаниям документацией.

В поле 5 указываются имена исходных файлов программ объекта испытаний.

В поле 6 указывается значение функции хэширования для исходных файлов программ, вычисленное в соответствии со стандартом, указанным в поле 7. Разрешается использовать функции хэширования, определенные в СТБ 34.101.31 или СТБ 34.101.77.

В поле 8 указывается результат выполнения проверки: «положительно» — результат проверки положительный, «отрицательно» — результат проверки отрицательный, «не проводилась» — проверка не требуется по причине специфики реализации программ объекта испытаний (например, в программе не используются глобальные переменные). После завершения анализа исходных текстов программ и заполнения таблицы делается вывод о соответствии (не соответствии) объекта испытаний СТБ 34.101.23. Вывод о соответствии делается только тогда, когда результаты всех проводимых проверок являются положительными.

В полях 9, 11 указываются соответственно должность и Ф. И. О. эксперта.

В поле 10 ставится собственноручная подпись эксперта.

Информация об обнаруженных ошибках и недокументированных возможностях приводится в протоколе или приложении к протоколу в произвольной форме и должна включать:

1) описание ошибки или недокументированной возможности;
2) имя файла и номера строк программы, содержащих ошибку.

Приложение Г
Тестовое программное обеспечение

Г.1 Программы преобразования АСН.1-файлов в текстовое представление

Программы, описанные в настоящем подразделе, является свободно распространяемыми программами, которые предназначены для анализа содержимого двоичных файлов, содержащих закодированные значения типов АСН.1. Они позволяют преобразовывать закодированные бинарные файлы в их текстовое представление.

Г.1.1 Программа `dumpasn1.exe`

Программа `dumpasn1.exe` является консольным приложением. Тексты программы располагаются по адресу: `https://www.cs.auckland.ac.nz/~pgut001/dumpasn1.c`.

Для преобразования файла запроса на получение сертификата в файл с текстовым представлением может использоваться, например, следующая команда:

```
dumpasn1.exe -t -a -z src.bin > dst.txt
```

В команде параметры имеют следующие значения: `src.bin` — закодированный исходный файл; `dst.txt` — текстовое представление исходного файла; `t` — отображение значений компонент в текстовом виде; `a` — отображение целиком блоков данных, длина которых больше 128 байтов; `z` — допущение полей нулевой длины.

Распространенные идентификаторы объектов могут передаваться в программу через конфигурационный файл `dumpasn1.cfg`. Стандартный конфигурационный файл периодически обновляется и размещается по адресу `https://www.cs.auckland.ac.nz/~pgut001/dumpasn1.cfg`. Для распознавания дополнительных идентификаторов, определенных в отечественных криптографических стандартах, может использоваться расширение конфигурационного файла, размещенное по адресу `https://github.com/agievich/bee2/blob/master/doc/dumpasn1by.cfg`.

Г.1.2 Программа `ASN.1 Editor`

Программа `ASN.1 Editor` является приложением операционной системы Windows с графическим пользовательским интерфейсом. Исполняемый файл программы располагается по адресу: `http://www.codeproject.com/Articles/4910/ASN-Editor`.

Для преобразования файла запроса на получение сертификата в текстовое представление необходимо открыть файл с помощью пункта основного меню программы: File → Open.

Г.2 Программы автоматической генерации тестовых наборов

Г.2.1 Технология Fuzzing

Для автоматизации тестирования программ, обрабатывающих сложные форматы данных, часто применяется технология Fuzzing. Специальная программа `fuzzer` обрабатывает испытываемую программу `prg`, которая в свою очередь обрабатывает файл `file`. Программа `fuzzer` выполняет многочисленные модификации `file`, подает эти модификации на вход `prg` и оценивает реакцию. Интерес для `fuzzer` представляют всевозможные

исключительные ситуации: зависания (hungs), утечки памяти (leaks), нарушение утверждений времени компиляции (asserts) и т. д. Любое из найденных исключений для испытуемой криптографической программы недопустимо.

Программа `fuzzer` выполняет модификации `file` разными способами. В большинстве случаев модификации формируются случайно, и тогда тестирование проходит по принципу «черный ящик». Намного больше ошибок можно выявить тогда, когда `fuzzer` учитывает (частично или полностью) формат `file`, т.е. поддерживает принцип «серый ящик» или даже «белый ящик».

Г.2.2 Программа AFL

Программа `AFL` (American Fuzzy Lop) — это свободно распространяемый `fuzzer`, с помощью которого найдено большое число уязвимостей в криптографических продуктах. Вся необходимая информация об `AFL`, в том числе документация и исходные файлы, размещена по адресу `http://lcamtuf.coredump.cx/afl`.

Испытуемая программа `prg` должна компилироваться средствами `AFL`, в свою очередь основанных на инструментах GCC или CLANG. Примерный make-файл для сборки испытуемой программы:

```
CC = path_to_afl/afl-gcc
CXX = path_to_afl/afl-g++
LDFLAGS = ...
CFLAGS = ...
PROGS = prg

all: $(PROGS)
prg: prg.c
  $(CC) $(CFLAGS) $@.c -o $@ $(LDFLAGS)
clean:
  rm -f $(PROGS) *.o ...
```

Испытуемая программа должна принимать на вход файл `file`. Тестовый файл, в окрестности которого будет организовано тестирование, помещается в специальный каталог `tests`. В это каталог могут быть добавлены любые другие тестовые файлы. Модификации `file`, которые привели к исключениям в `prg`, помещаются в каталог `findings`.

Тестирование запускается по команде

```
path_to_afl/afl-fuzz -i tests -o findings path_to_prg/prg @@
```

Информацию по дополнительным опциям команды, а также дополнительным возможностям `AFL` можно найти на упомянутом сайте.

Перед сборкой `AFL` в виртуальной среде на платформе Windows следует включить директиву `SIMPLE_FILES` в заголовочном файле `config.h` и в функции `trim_case()` модуля `afl_fuzz.c` строку

```
fd = open(q->fname, O_WRONLY | O_CREAT | O_EXCL, 0600);
```

изменить на

```
fd = open(q->fname, O_WRONLY | O_CREAT, 0600);
```

Приложение Д
Описание тестовых данных

В данном приложении приводится описание криптографических сообщений, используемых в тестах известного ответа. Каждое криптографическое сообщение определенного типа в тестах основано на базовом криптографическом сообщении соответствующего типа. Эти базовые криптографические сообщения содержат типичные для всех криптографических сообщений соответствующего типа значения основных компонентов. При описании тестовых криптографических сообщений приводятся описание базового криптографического сообщения, а также значения компонентов, которые отличаются от указанных в базовом сообщении.

Д.1 Криптографические сообщения с неструктурированными данными

Д.1.1 Криптографическое сообщение VUDTest

Является базовым криптографическим сообщением.

Имеет следующие значения основных компонент:

Компонент сообщения или тип ASN.1	Значение компонента	Пояснения
`ContentInfo`		
`contentType`	1.2.840.113549.1.7.1	Идентификатор определяет неструктурированные данные
`content`	0123456789	Данные, представленные типом `OCTET STRING`

Д.1.2 Криптографическое сообщение IContentTypeTest

Основано на базовом сообщении.

Значение компонента `ContentInfo.contentType`: 1.2.112.0.2.0.34.101.31.

Д.2 Криптографические сообщения с подписанными данными

Д.2.1 Криптографическое сообщение VSDTest

Является базовым криптографическим сообщением.

Имеет следующие значения основных компонент:

Компонент сообщения или тип ASN.1	Значение компонента	Пояснения
ContentInfo		
contentType	1.2.840.113549.1.7.2	Идентификатор определяет подписанные данные
SignedData		Данные контейнера, определяющие подписанные данные
version	1	Версия синтаксиса
digestAlgorithms		
algorithm	1.2.112.0.2.0.34.101.31.81	Соответствует алгоритму belt-hash
parameters	NULL	Параметры не задаются
encapContentInfo		
eContentType	1.2.840.113549.1.7.1	Идентификатор определяет неструктурированные данные
eContent	0123456789	Данные, закодированные типом OCTET STRING
certificates	{Определенное значение}	Содержит сертификат подписавшей стороны
signerInfos		
version	1	Номер версии ситаксиса
sid	{Определенное значение}	Содержит значение компонента issuerAndSerialNumber
digestAlgorithms		
algorithm	1.2.112.0.2.0.34.101.31.81	Соответствует алгоритму belt-hash
parameters	NULL	Параметры не задаются
signedAttrs		Подписываемые атрибуты
Attribute		
attrType	1.2.840.113549.1.9.3	Атрибут «тип содержимого»
attrValues	1.2.840.113549.1.7.1	Неструктурированные данные
Attribute		
attrType	1.2.840.113549.1.9.5	Атрибут «время подписания»
attrValues	10/11/2016 07:32:01	GMT Значение типа UTCTime
Attribute		
attrType	1.2.840.113549.1.9.4	Атрибут «хэш-значение»
attrValues	{Определенное значение}	Значение типа OCTET STRING из 32 октетов
signatureAlgorithm		
algorithm	1.2.112.0.2.0.34.101.45.12	Соответствует алгоритму bign-with-hbelt
parameters	NULL	Параметры не задаются
signature	{Определенное значение}	ЭЦП, представленное значением типа OCTET STRING

Д.2.2 Криптографическое сообщение ISignatureSDTest

Основано на базовом сообщении.

Содержит измененное значение компонента `SignedData.signerInfos.signature`.

Д.2.3 Криптографическое сообщение IVersionSDTest

Основано на базовом сообщении.

Значение компонента `SignedData.version`: 0.

Д.2.4 Криптографическое сообщение IDigAlgIdSDTest

Основано на базовом сообщении.

Значение компонента `SignedData.digestAlgorithms.algorithm`: 1.2.112.0.2.0.34.101.31.91.

Д.2.5 Криптографическое сообщение IContentSDTest

Основано на базовом сообщении.

Значение компонента `SignedData.encapContentInfo.eContent`: 1123456789.

Д.2.6 Криптографическое сообщение VSidSDTest

Основано на базовом сообщении.

Значение компонента `SignedData.signerInfos.sid` содержит ссылку на сертификат подписывающей стороны, представленную значением типа `SubjectKeyIdentifier` (см. п. 6.2.1.2 СТБ 34.101.19).

Д.2.7 Криптографическое сообщение ISignAttrSDTest

Основано на базовом сообщении.

Значение подписываемого атрибута «время подписания»: 09/11/2016 07:32:01 GMT.

Д.2.8 Криптографическое сообщение IDigAlgIdSISDTest

Основано на базовом сообщении.

Значение компонента `SignedData.signerInfos.digestAlgorithms.algorithm`: 1.2.112.0.2.0.34.101.31.91.

Д.2.9 Криптографическое сообщение ISignAlgIdSISDTest

Основано на базовом сообщении.

Значение компонента `SignedData.signerInfos.signatureAlgorithms.algorithm`: 1.2.112.0.2.0.34.101.45.21.

Д.2.10 Криптографическое сообщение ISignAlgPrmSISDTest

Основано на базовом сообщении.

Значение компонента `SignedData.signerInfos.signatureAlgorithms.parameters`: 1.2.112.0.2.0.34.101.31.81.

Д.3 Криптографические сообщения с конвертованными данными

Д.3.1 Криптографическое сообщение VEDTest

Является базовым криптографическим сообщением.

Имеет следующие значения основных компонент:

Компонент сообщения или тип ASN.1	Значение компонента	Пояснения
`ContentInfo`		
`contentType`	1.2.840.113549.1.7.3	Идентификатор определяет конвертованные данные
`EnvelopedData`		Данные контейнера, определяющие конвертованные данные
`version`	0	Версия синтаксиса
`recipientInfos`		Информация, связанная с получателем
`ktri`		Используется транспорт ключа
`version`	0	Номер версии ситаксиса
`rid`	{Определенное значение}	Содержит значение компонента `issuerAndSerialNumber`
`keyEncryptionAlgorithm`		
`algorithm`	1.2.112.0.2.0.34.101.45.41	Соответствует алгоритму `bign-keytransport`
`parameters`	`NULL`	Параметры не задаются
`encryptedKey`	{Определенное значение}	Защищенный ключ, представленный значением типа `OCTET STRING`
`encryptedContentInfo`		
`contentType`	1.2.840.113549.1.7.1	Идентификатор определяет неструктурированные данные
`contentEncryptionAlgorithm`		
`algorithm`	1.2.112.0.2.0.34.101.31.43	Соответствует алгоритму `belt-ctr256`
`parameters`	{Определенное значение}	Синхропосылка, представленная типом `OCTET STRING`
`encryptedContent`	{Определенное значение}	Зашифрованные данные 0123456789, представленные типом `OCTET STRING`

Д.3.2 Криптографическое сообщение IEncrKeyEDTest

Основано на базовом сообщении.

Содержит измененное значение компонента `EnvelopedData.recipientInfos.ktri.encryptedKey`.

Д.3.3 Криптографическое сообщение IVersionEDTest

Основано на базовом сообщении.

Значение компонента `EnvelopedData.version`: 9.

Д.3.4 Криптографическое сообщение IKEAlgIdEDTest

Основано на базовом сообщении.

Значение компонента `EnvelopedData.recipientInfos.ktri.keyEncryptionAlgorithm.algorithm`: 1.2.112.0.2.0.34.101.45.21.

Д.3.5 Криптографическое сообщение ICEAlgIdEDTest

Основано на базовом сообщении.

Значение компонента `EnvelopedData.encryptedContentInfo.contentEncryptionAlgorithm.algorithm`: 1.2.112.0.2.0.34.101.31.81.

Д.4 Криптографические сообщения с хэшированными данными

Д.4.1 Криптографическое сообщение VDDTest

Является базовым криптографическим сообщением.

Имеет следующие значения основных компонент:

Компонент сообщения или тип ASN.1	Значение компонента	Пояснения
`ContentInfo`		
`contentType`	1.2.840.113549.1.7.5	Идентификатор определяет хэшированные данные
`DigestedData`		Данные контейнера, определяющие хэшированные данные
`version`	0	Версия синтаксиса
`digestAlgorithms`		
`algorithm`	1.2.112.0.2.0.34.101.31.81	Соответствует алгоритму `belt-hash`
`parameters`	`NULL`	Параметры не задаются
`encapContentInfo`		
`eContentType`	1.2.840.113549.1.7.1	Идентификатор определяет неструктурированные данные
`eContent`	0123456789	Данные, закодированные типом `OCTET STRING`
`digest`	{Определенное значение}	Хэш-значение, представленное значением типа `OCTET STRING`

Д.4.2 Криптографическое сообщение IDigestDDTest

Основано на базовом сообщении.

Содержит измененное значение компонента `DigestedData.digest`.

Д.4.3 Криптографическое сообщение IVersionDDTest

Основано на базовом сообщении.

Значение компонента `DigestedData.version`: 3.

Д.4.4 Криптографическое сообщение IDigAlgIdDDTest

Основано на базовом сообщении.

Значение компонента `DigestedData.digestAlgorithm.algorithm`: 1.2.112.0.2.0.34.101.31.91.

Д.5 Криптографические сообщения с шифрованными данными

Д.5.1 Криптографическое сообщение VEncrDTest

Является базовым криптографическим сообщением.

Имеет следующие значения основных компонент:

Компонент сообщения или тип ASN.1	Значение компонента	Пояснения
ContentInfo		
contentType	1.2.840.113549.1.7.6	Идентификатор определяет шифрованные данные
EncryptedData		Данные контейнера, определяющие конвертованные данные
version	0	Версия синтаксиса
encryptedContentInfo		
contentType	1.2.840.113549.1.7.1	Идентификатор определяет неструктурированные данные
contentEncryptionAlgorithm		
algorithm	1.2.112.0.2.0.34.101.31.43	Соответствует алгоритму belt-ctr256
parameters	{Определенное значение}	Синхропосылка, представленная типом OCTET STRING
encryptedContent	{Определенное значение}	Зашифрованные данные 0123456789, представленные типом OCTET STRING

Д.5.2 Криптографическое сообщение IVerEncrDTest

Основано на базовом сообщении.

Значение компонента EncryptedData.version: 3.

Д.5.3 Криптографическое сообщение IEAlgIdEncrDTest

Основано на базовом сообщении.

Значение компонента EncryptedData.encryptedContentInfo.contentEncryptionAlgorithm.algorithm: 1.2.112.0.2.0.34.101.31.81.

Д.6 Криптографические сообщения с аутентифицированными данными

Д.6.1 Криптографическое сообщение VADTest

Является базовым криптографическим сообщением.

Имеет следующие значения основных компонент:

Компонент сообщения или тип ASN.1	Значение компонента	Пояснения
ContentInfo		
contentType	1.2.840.113549.1.9.16.1.2	Идентификатор определяет аутентифицированные данные
AuthenticatedData		Данные контейнера, определяющие аутентифицированные данные
version	0	Версия синтаксиса
recipientInfos		Информация, связанная с получателем
ktri		Используется транспорт ключа
version	0	Номер версии ситаксиса
rid	{Определенное значение}	Содержит значение компонента issuerAndSerialNumber
keyEncryptionAlgorithm		
algorithm	1.2.112.0.2.0.34.101.45.41	Соответствует алгоритму bign-keytransport
parameters	NULL	Параметры не задаются
encryptedKey	{Определенное значение}	Защищенный ключ, представленный значением типа OCTET STRING
macAlgorithms		
algorithm	1.2.112.0.2.0.34.101.31.53	Соответствует алгоритму belt-mac256
parameters	NULL	Параметры не задаются
encapContentInfo		
eContentType	1.2.840.113549.1.7.1	Идентификатор определяет неструктурированные данные
eContent	content	Данные, закодированные типом OCTET STRING
mac	{Определенное значение}	Имитовставка, представленная значением типа OCTET STRING

Д.6.2 Криптографическое сообщение IEncrKeyADTest

Основано на базовом сообщении.

Содержит измененное значение компонента `AuthenticatedData.recipientInfos.ktri.encryptedKey`.

Д.6.3 Криптографическое сообщение IMacADTest

Основано на базовом сообщении.

Содержит измененное значение компонента `AuthenticatedData.mac`.

Д.6.4 Криптографическое сообщение IVersionADTest

Основано на базовом сообщении.

Значение компонента `AuthenticatedData.version`: 5.

Д.6.5 Криптографическое сообщение IKEAlgIdADTest

Основано на базовом сообщении.

Значение компонента `AuthenticatedData.recipientInfos.ktri.keyEncryptionAlgorithm.algorithm`: 1.2.112.0.2.0.34.101.45.21.

Д.6.6 Криптографическое сообщение IMacAlgIdADTest

Основано на базовом сообщении.

Значение компонента `AuthenticatedData.macAlgorithm.algorithm`: 1.2.112.0.2.0.34.101.31.43.

Д.6.7 Криптографическое сообщение VAttrADTest

Основано на базовом сообщении.

Содержит необязательный компонент `AuthenticatedData.digestAlgorithm`, компоненты которого имеют значения:

`AuthenticatedData.digestAlgorithm.algorithm` = 1.2.112.0.2.0.34.101.31.81;

`AuthenticatedData.digestAlgorithm.parameters` = `NULL`.

Содержит необязательный компонент `AuthenticatedData.authAttrs` с атрибутами «тип содержимого» и «хэш-значение».

Д.6.8 Криптографическое сообщение IDigAlgADTest

Основано на сообщении `VAttrADTest`.

Значение компонента `AuthenticatedData.digestAlgorithm.algorithm`: 1.2.112.0.2.0.34.101.31.53

Д.6.9 Криптографическое сообщение IAttrADTest

Основано на сообщении `VAttrADTest`.

Содержит измененное значение атрибута «хэш-значение» компонента `AuthenticatedData.authAttrs`.

Д.7 Криптографические сообщения с аутентифицированными конвертованными данными

Д.7.1 Криптографическое сообщение VAEDTest

Является базовым криптографическим сообщением.

Имеет следующие значения основных компонент:

Компонент сообщения или тип ASN.1	Значение компонента	Пояснения
ContentInfo		
contentType	1.2.840.113549.1.9.16.1.23	Идентификатор определяет аутентифицированные конвертованные данные
AuthenticatedData		Данные контейнера, определяющие аутентифицированные конвертованные данные
version	0	Версия 1 синтаксиса
recipientInfos		Информация, связанная с получателем
ktri		Используется транспорт ключа
version	0	Номер версии ситаксиса
rid	{Определенное значение}	Содержит значение компонента issuerAndSerialNumber
keyEncryptionAlgorithm		
algorithm	1.2.112.0.2.0.34.101.45.41	Соответствует алгоритму bign-keytransport
parameters	NULL	Параметры не задаются
encryptedKey	{Определенное значение}	Защищенный ключ, представленный значением типа OCTET STRING
authEncryptedContentInfo		
contentType	1.2.840.113549.1.7.1	Идентификатор определяет неструктурированные данные
contentEncryptionAlgorithm		
algorithm	1.2.112.0.2.0.34.101.31.63	Соответствует алгоритму belt-datawrap256
parameters	0123456789abcdef	Синхропосылка, представленная типом OCTET STRING
encryptedContent	{Определенное значение}	Зашифрованные данные «content», представленные типом OCTET STRING
mac	{Определенное значение}	Имитовставка, представленная значением типа OCTET STRING

Д.7.2 Криптографическое сообщение IEncrKeyAEDTest

Основано на базовом сообщении.

Содержит измененное значение компонента AuthEnvelopedData.recipientInfos.ktri.encryptedKey.

Д.7.3 Криптографическое сообщение IMacAEDTest

Основано на базовом сообщении.

Содержит измененное значение компонента AuthEnvelopedData.mac.

Д.7.4 Криптографическое сообщение IVersionAEDTest

Основано на базовом сообщении.

Значение компонента AuthEnvelopedData.version: 5.

Д.7.5 Криптографическое сообщение IKEAlgIdAEDTest

Основано на базовом сообщении.

Значение компонента AuthEnvelopedData.recipientInfos.ktri.keyEncryptionAlgorithm.algorithm: 1.2.112.0.2.0.34.101.31.81.

Д.7.6 Криптографическое сообщение IAEAlgIdAEDTest

Основано на базовом сообщении.

Значение компонента AuthEnvelopedData.encryptedContentInfo.contentEncryptionAlgorithm.algorithm: 1.2.112.0.2.0.34.101.31.81.

Д.7.7 Криптографическое сообщение VAttrAEDTest

Основано на базовом сообщении.

Содержит необязательный компонент `AuthEnvelopedData.authAttrs` с атрибутом «тип содержимого».

Д.7.8 Криптографическое сообщение IAttrAEDTest

Основано на сообщении `VAttrADTest`.

Содержит измененное значение атрибута «тип содержимого» компонента `AuthenticatedData.authAttrs`.