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 СРЕДСТВ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ НА СООТВЕТСТВИЕ ТРЕБОВАНИЯМ СТБ 34.101.19-2012

МИ.10119.10.01

Листов 64

Минск 2022

Предисловие

Настоящая методика испытаний предназначена для использования в испытательных лабораториях при проведении сертификационных испытаний средств криптографической защиты информации на соответствие требованиям СТБ 34.101.19-2012 «Информационные технологии и безопасность. Форматы сертификатов и списков отозванных сертификатов инфраструктуры открытых ключей».

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 34.101.27-2022 «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 34.101.19, а также следующие сокращения:

АСН.1	абстрактно-синтаксическая нотация версии 1;
ЕСПД	единая система программной документации;
ИОК	инфраструктура открытых ключей;
СКЗИ	средство криптографической защиты информации;
СОК	сертификат открытого ключа;
СОС	список отозванных сертификатов;
ТНПА	технический нормативный правовой акт;
УЦ	удостоверяющий центр;
ЭЦП	электронная цифровая подпись.

3 Объект и цель испытаний

На испытания представляется средство криптографической защиты информации (СКЗИ), реализующее управление открытыми ключами согласно СТБ 34.101.19, и документация на СКЗИ.

Целью испытаний является проверка соответствия процедур формирования сертификатов открытых ключей (СОК) и списков отозванных сертификатов (СОС), а также процедуры верификация маршрута сертификации, реализованных в объекте испытаний, требованиям СТБ 34.101.19.

4 Требования к объекту испытаний

Программа объекта испытаний может реализовывать следующие процедуры, определенные в СТБ 34.101.19:

– формирование СОК;

– формирование СОС;

– верификация маршрута сертификации.

При этом программа объекта испытаний должна реализовывать, по крайней мере, одну из указанных процедур.

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

– в программе должны быть точно и полно реализовываны процедуры СТБ 34.101.19, поддерживаемые объектом испытаний;

– программа, реализующая процедуры СТБ 34.101.19, 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 34.101.19. Если программа не поддерживает некоторые процедуры, определенные в СТБ 34.101.19, 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 34.101.19, которые в составе других программ имеют действующие сертификаты соответствия требованиям СТБ 34.101.19, то проверки по тестированию и анализу исходных текстов для данных реализаций могут не проводиться. При этом для подтверждения соответствия объекта испытаний требованиям СТБ 34.101.19 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 34.101.19 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 34.101.19, предварительно должна пройти успешные испытания по согласованной с Органом по сертификации методике испытаний.

Испытываемая программа может не поддерживать необязательный функционал, определенный в СТБ 34.101.19 (например, формирование некоторых расширений). При этом сужение программой обязательного функционала, определенного в СТБ 34.101.19, 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, реализованных в программе и определенных в СТБ 34.101.19, включая:

- формирование СОК (см. п. 6.2.1);
- формирование СОС (см. п. 6.2.2);
- верификация маршрута сертификации (см. п. 6.2.3).

Тестирование процедур формирования СОК и СОС выполняется путем формирования программой СОК и СОС соответственно с последующим визуальным сравнением текстового представления форматов сформированных СОК и СОС с форматами, определенными в СТБ 34.101.19.

Тестирование процедуры верификация маршрута сертификации проводится путем выполнения программой верификации маршрутов сертификации с последующим сравнением полученных результатов с ожидаемыми.

В тестах используются СОК и СОС, 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 запросов 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 34.101.19. 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 СОК (см. п. 6.3.3);
– корректности процедуры формирования СОС (см. п. 6.3.4);
– корректности процедуры верификации маршрута сертификации (см. п. 6.3.5);
– корректности обработки исключительных ситуаций (см. п. 6.3.6);
– отсутствия недокументированных возможностей (см. п. 6.3.7).

Для анализа исходных текстов реализации процедуры формирования СОК выполняются проверки из п. 6.3.1 – 6.3.3, 6.3.6, 6.3.7, для реализации процедуры формирования СОС — проверки из п. 6.3.1, 6.3.2, 6.3.4, 6.3.6, 6.3.7, а для реализации процедуры верификации маршрута сертификации — проверки из п. 6.3.1, 6.3.5 – 6.3.7. 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 34.101.19 процедуры, представлены полностью и в виде, который использовался при сборке программы.

Если документ «Текст программы» разработан в соответствии с требованиями ЕСПД, то эксперт 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 34.101.19 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 34.101.19 процедуры;

– описание функций, реализующих определенные в СТБ 34.101.19 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 34.101.19 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 Процедура формирования СОК

Входными данными, задаваемыми при тестировании процедуры формирования СОК, являются допустимые значения для параметров вызова программы, определяющих состав и содержимое СОК.

В тестах для хранения СОК используются бинарные файлы, содержащие закодированных значений типов АСН.1, составляющих СОК.

При тестировании процедуры формирования СОК выполняются следующие тесты.

Тест IssueCertTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования СОК.
2 Средствами испытываемой программы сформировать СОК и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих СОК.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление СОК.
4 Провести визуальный анализ текстового представления сформированного СОК в части базовых компонент на соответствие п. 6.1 СТБ 34.101.19.
5 Повторить шаги 1 — 4 не менее 9 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).
6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе СОК не выявлено несоответствий СТБ 34.101.19, при этом:
1) формат и содержание значения обязательного компонента `tbsCertificate` соответствует типу `TBSCertificate` (см. п. 6.1.1 СТБ 34.101.19):

- компонент `version` (см. п. 6.1.2.1 СТБ 34.101.19), определяющий версию 3 сертификата, содержит значение 2 типа `INTEGER` (если значение компонента отсутствует, то используется значение по умолчанию);
- компонент `serialNumber` (см. п. 6.1.2.2 СТБ 34.101.19), определяющий серийный номер, содержит значение типа `INTEGER` и не превышает 20 октетов;
- компонент `signature` (см. п. 6.1.2.3 СТБ 34.101.19) содержит значение типа `AlgorithmIdentifier` и определяет идентификатор алгоритма, который УЦ использовал для выработки ЭЦП для данного сертификата;
- компонент `issuer` (см. п. 6.1.2.4 СТБ 34.101.19) содержит значение типа `Name` и идентифицирует субъект, который подписал и выдал сертификат;
- компонент `validity` (см. п. 6.1.2.5 СТБ 34.101.19) содержит значение типа `SEQUENCE`, включающее два компонента, которые определяют срок действия сертификата;
- компонент `subject` (см. п. 6.1.2.6 СТБ 34.101.19) содержит значение типа Name и определяет конечного пользователя, связанного с открытым ключом;
- компонент `subjectPublicKeyInfo` (см. п. 6.1.2.7 СТБ 34.101.19) содержит значение типа `SEQUENCE`, включающее два компонента, которые определяют открытый ключ и алгоритм, с которым данный ключ используется;

– необязательные компоненты `issuerUniqueID` и `subjectUniqueID` (см. п. 6.1.2.8 СТБ 34.101.19) отсутствуют (удостоверяющие центры не должны выпускать сертификаты с данными компонентами);
– компонент `extensions` (см. п. 6.1.2.9 СТБ 34.101.19) содержит значение типа `SEQUENCE`, включающее несколько расширений (проверка соответствия форматов и содержания значений расширений СТБ 34.101.19 выполняется в тесте `ExtCertTest`);

2) формат и содержание значения обязательного компонента `signatureAlgorithm` соответствует типу `AlgorithmIdentifier` (см. п. 6.1.1.2 СТБ 34.101.19) и определяет идентификатор алгоритма, который УЦ использовал для выработки ЭЦП для данного сертификата (в данном компоненте должно содержаться то же значение, что и в компоненте `signature` типа `TBSCertificate`);
3) формат и содержание значения обязательного компонента `signatureValue` соответствует типу `BIT STRING` (см. п. 6.1.1.3 СТБ 34.101.19) и содержит значение ЭЦП.
7 Возвратить `ОШИБКА`.

Примечание — Задаваемые в тесте IssueCertTest параметры вызова испытываемой программы в совокупности должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по выпуску СОК в части формирования базовых компонент СОК.

Тест ExtCertTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования СОК.
2 Средствами испытываемой программы сформировать СОК и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих СОК.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление СОК.
4 Провести визуальный анализ текстового представления сформированного СОК в части расширений на соответствие п. 6.2 СТБ 34.101.19.
5 Повторить шаги 1 — 4 не менее 9 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).
6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе СОК не выявлено несоответствий СТБ 34.101.19, при этом:
1) компонент `extensions` (см. п. 6.1.2.9 СТБ 34.101.19) содержит значение типа `SEQUENCE`, включающее несколько расширений, каждое из которых содержит идентификатор и закодированное значение заданного типа, а также булевую переменную, которая по умолчанию принимает значение `FALSE` и определяет критичность расширения (см. п. 6.2 СТБ 34.101.19);
2) в компоненте `extensions` содержатся:

– некритическое расширение `authorityKeyIdentifier` с идентификатором 2.5.29.35 (см. п. 6.2.1.1 СТБ 34.101.19), идентифицирующее открытый ключ УЦ, соответствующий личному ключу, использующемуся для подписи сертификата (для самоподписанного сертификата расширение может быть опущено);

- некритическое расширение `SubjectKeyIdentifier` с идентификатором 2.5.29.14 (см. п. 6.2.1.2 СТБ 34.101.19), идентифицирующее сертификат, содержащий определенный открытый ключ;
- расширение `KeyUsage` с идентификатором 2.5.29.15 (см. п. 6.2.1.3 СТБ 34.101.19), определяющее назначение открытого ключа, содержащегося в СОК;
- расширение `certificatePolicies` с идентификатором 2.5.29.32 (см. п. 6.2.1.4 СТБ 34.101.19), определяющее политики сертификата;

3) в компоненте `extensions` могут содержаться:

- некритическое расширение `PolicyMappings` с идентификатором 2.5.29.33 (см. п. 6.2.1.5 СТБ 34.101.19), определяющее отображение политики;
- расширение `SubjectAltName` с идентификатором 2.5.29.17 (см. п. 6.2.1.6 СТБ 34.101.19), определяющее альтернативное имя субъекта;
- расширение `IssuerAltName` с идентификатором 2.5.29.18 (см. п. 6.2.1.7 СТБ 34.101.19), определяющее альтернативное имя эмитента;
- некритическое расширение `SubjectDirectoryAttributes` с идентификатором 2.5.29.9 (см. п. 6.2.1.8 СТБ 34.101.19), определяющее идентификационные атрибуты субъекта;
- расширение `BasicConstraints` с идентификатором 2.5.29.19 (см. п. 6.2.1.9 СТБ 34.101.19), определяющее является ли УЦ субъектом сертификата, а также максимальную глубину действительных маршрутов сертификации, которые содержат данный сертификат (должно включаться во все сертификаты УЦ, которые содержат открытые ключи для проверки ЭЦП сертификатов);
- критическое расширение `NameConstraints` с идентификатором 2.5.29.30 (см. п. 6.2.1.10 СТБ 34.101.19), определяющее пространство имен, которому должны принадлежать все имена субъектов в последующих сертификатах маршрута сертификации;
- критическое расширение `PolicyConstraints` с идентификатором 2.5.29.36 (см. п. 6.2.1.11 СТБ 34.101.19), определяющее ограничения для верификации маршрута сертификации;
- расширение `ExtKeyUsageSyntax` с идентификатором 2.5.29.37 (см. п. 6.2.1.12 СТБ 34.101.19), определяющее расширенную область применения ключей;
- расширение `CRLDistributionPoints` с идентификатором 2.5.29.31 (см. п. 6.2.1.13 СТБ 34.101.19), определяющее пункты распространения списка отозванных сертификатов;
- критическое расширение `InhibitAnyPolicy` с идентификатором 2.5.29.54 (см. п. 6.2.1.14 СТБ 34.101.19), определяющее запрещение политики `anyPolicy`;
- некритическое расширение `InhibitAnyPolicy` с идентификатором 2.5.29.46 (см. п. 6.2.1.15 СТБ 34.101.19), определяющее способы получения приращений СОС;
- некритическое расширение `AuthorityInfoAccessSyntax` с идентификатором 1.3.6.1.5. 5.7.1.1 (см. п. 6.2.2.1 СТБ 34.101.19), определяющее каким образом можно получить доступ к информации об УЦ и сервисам УЦ;
- некритическое расширение `SubjectInfoAccessSyntax` с идентификатором 1.3.6.1.5.5.7.1.11 (см. п. 6.2.2.2 СТБ 34.101.19), определяющее каким образом

можно получить доступ к информации и сервисам субъекта сертификата, в котором указано данное расширение;

– нестандартные (пользовательские) расширения.

7 Возвратить `ОШИБКА`.

Примечание 1 — Задаваемые в тесте ExtCertTest параметры вызова испытываемой программы в совокупности должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по выпуску СОК в части формирования расширений.

Примечание 2 — Корректность формирования нестандартных расширений должна проверяться по документам, определяющим данное расширение.

6.2.2 Процедура формирования СОС

Входными данными, задаваемыми при тестировании процедуры формирования СОС, являются допустимые значения для параметров вызова программы, определяющих состав и содержимое СОС.

В тестах для хранения СОС используются бинарные файлы, содержащие закодированных значений типов АСН.1, составляющих СОС.

При тестировании процедуры формирования СОС выполняются следующие тесты.

Тест IssueCrlTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования СОС.

2 Средствами испытываемой программы сформировать СОС и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих СОС.

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

4 Провести визуальный анализ текстового представления сформированного СОС в части основных компонент на соответствие п. 7.1 СТБ 34.101.19.

5 Повторить шаги 1 — 4 не менее 9 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).

6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе СОС не выявлено несоответствий СТБ 34.101.19, при этом:

1) формат и содержание значения обязательного компонента `tbsCertList` соответствует типу `TBSCertList` (см. п. 7.1.1.1 и п. 7.1.2 СТБ 34.101.19):

– компонент version (см. п. 7.1.2.1 СТБ 34.101.19), определяющий версию 2 СОС, содержит значение 1 типа `INTEGER` (если значение компонента отсутствует, то используется значение по умолчанию);

– компонент `signature` (см. п. 7.1.2.2 СТБ 34.101.19) содержит значение типа `AlgorithmIdentifier` и определяет идентификатор алгоритма, который УЦ использовал для выработки ЭЦП для данного СОС;

– компонент `issuer` (см. п. 7.1.2.3 СТБ 34.101.19) содержит значение типа `Name` и идентифицирует субъект, который подписал и выпустил СОС;

– компонент `thisUpdate` (см. п. 7.1.2.4 СТБ 34.101.19) содержит значение типа `Time` и определяет дату выпуска СОС;

– компонент `nextUpdate` (см. п. 7.1.2.5 СТБ 34.101.19) содержит значение типа `Time` и определяет дату выпуска следующего СОС;
– необязательный компонент `revokedCertificates` (см. п. 7.1.2.6 СТБ 34.101.19) содержит значение типа `SEQUENCE`, включающее два обязательных компонента, которые определяют серийный номер и дату отзыва СОК, и один необязательный компонент, который определяет расширения записей СОС (проверка соответствия форматов и содержания значений расширений СТБ 34.101.19 выполняется в тесте `EntryExtCrlTest`);
– компонент `crlExtensions` (см. п. 7.1.2.7 СТБ 34.101.19) содержит значение типа `Extensions` и определяет расширения СОС (проверка соответствия форматов и содержания значений расширений СТБ 34.101.19 выполняется в тесте `ExtCrlTest`);

2) формат и содержание значения обязательного компонента `signatureAlgorithm` соответствует типу `AlgorithmIdentifier` (см. п. 7.1.1.2 СТБ 34.101.19) и определяет идентификатор алгоритма, который УЦ использовал для выработки ЭЦП для данного СОС (в данном компоненте должно содержаться то же значение, что и в компоненте `signature` типа `TBSCertList`);
3) формат и содержание значения обязательного компонента `signatureValue` соответствует типу `BIT STRING` (см. п. 7.1.1.3 СТБ 34.101.19) и содержит значение ЭЦП.
7 Возвратить `ОШИБКА`.

Примечание — Задаваемые в тесте IssueCrlTest параметры вызова испытываемой программы в совокупности должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по выпуску СОС в части формирования основных компонент СОС.

Тест ExtCrlTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования СОС.
2 Средствами испытываемой программы сформировать СОС и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих СОС.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление СОС.
4 Провести визуальный анализ текстового представления сформированного СОС в части расширений на соответствие п. 7.2 СТБ 34.101.19.
5 Повторить шаги 1 — 4 не менее 9 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).
6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе СОС не выявлено несоответствий СТБ 34.101.19, при этом:
1) необязательный компонент `crlExtensions` (см. п. 7.1.2.7 СТБ 34.101.19) содержит значение типа `SEQUENCE`, включающее несколько расширений, каждое из которых содержит идентификатор и закодированное значение заданного типа, а также булевую переменную, которая по умолчанию принимает значение FALSE и определяет критичность расширения (см. п. 7.2 СТБ 34.101.19);
2) в компоненте `crlExtensions` содержатся:

– некритическое расширение `authorityKeyIdentifier` с идентификатором 2.5.29.35 (см. п.7.2.1 и п. 6.2.1.1 СТБ 34.101.19), идентифицирующее открытый ключ УЦ, соответствующий личному ключу, использующемуся для подписи СОС;

– некритическое расширение `CRLNumber` с идентификатором 2.5.29.20 (см. п. 7.2.3 СТБ 34.101.19), определяющее СОС;

3) в компоненте `crlExtensions` могут содержаться:

– расширение `IssuerAltName` с идентификатором 2.5.29.18 (см. п. 7.2.2 и п. 6.2.1.7 СТБ 34.101.19), определяющее альтернативное имя эмитента;

– критическое расширение `deltaCRLIndicator` с идентификатором 2.5.29.27 (см. п. 7.2.4 СТБ 34.101.19), определяющее, что СОС является приращением;

– критическое расширение `issuingDistributionPoint` с идентификатором 2.5.29.46 (см. п. 7.2.5 СТБ 34.101.19), которое указывает выпускающий пункт распределения и область действия определенного СОС, а также говорит о том, распространяется ли СОС на сертификаты конечных пользователей, только на сертификаты УЦ, только на атрибутные сертификаты или на ограниченный набор кодов причин;

– некритическое расширение `FreshestCRL` с идентификатором 2.5.29.46 (см. п. 7.2.6 СТБ 34.101.19), определяющее способ получения информации приращения СОС для полного СОС;

– некритическое расширение `AuthorityInfoAccessSyntax` с идентификатором 1.3.6.1.5. 5.7.1.1 (см. п. 7.2.7 СТБ 34.101.19), определяющее каким образом можно получить доступ к информации об УЦ и сервисам УЦ.

7 Возвратить `ОШИБКА`.

Примечание — Задаваемые в тесте ExtCrlTest параметры вызова испытываемой программы в совокупности должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по выпуску СОС в части формирования расширений СОС.

Тест EntryExtCrlTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования СОС.

2 Средствами испытываемой программы сформировать СОС и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих СОС.

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

4 Провести визуальный анализ текстового представления сформированного СОС в части расширений записей на соответствие п. 7.3 СТБ 34.101.19.

5 Повторить шаги 1 — 4 не менее 9 раз, задавая различные параметры вызова испытываемой программы (если имеется такая возможность).

6 Возвратить `УСПЕХ`, если на шаге 4 при визуальном анализе СОС не выявлено несоответствий СТБ 34.101.19, при этом:

1) необязательный компонент `crlEntryExtensions` (см. п. 7.3 СТБ 34.101.19) содержит значение типа `SEQUENCE`, включающее несколько расширений, каждое из кото-

рых содержит идентификатор и закодированное значение заданного типа, а также булевую переменную, которая по умолчанию принимает значение FALSE и определяет критичность расширения (см. п. 7.2 СТБ 34.101.19);

2) в компоненте `crlEntryExtensions` могут содержаться:

- некритическое расширение `reasonCode` с идентификатором 2.5.29.21 (см. п. 7.3.1), определяющее причину отзыва сертификата;
- некритическое расширение `holdInstructionCode` с идентификатором 2.5.29.23 (см. п. 7.3.2 СТБ 34.101.19), представляющее собой зарегистрированный идентификатор инструкции, который указывает, какое действие следует предпринять после обнаружения сертификата, действие которого было приостановлено;
- некритическое расширение `invalidityDate` с идентификатором 2.5.29.24 (см. п. 7.3.3 СТБ 34.101.19), которое содержит дату, когда произошла достоверно известная или предполагаемая компрометация личного ключа или когда сертификат стал недействительным при иных обстоятельствах;
- критическое расширение `certificateIssuer` с идентификатором 2.5.29.29 (см. п. 7.3.4 СТБ 34.101.19), определяющее эмитента сертификата, связанного с записью в косвенном СОС.

7 Возвратить `ОШИБКА`.

Примечание — Задаваемые в тесте EntryExtCrlTest параметры вызова испытываемой программы в совокупности должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по выпуску СОС в части формирования расширений записи СОС.

6.2.3 Процедура верификации маршрута сертификации

Входными данными, задаваемыми при тестировании процедуры верификация маршрута сертификации, являются маршруты сертификации, включающие СОК и СОС, представленные в виде бинарных файлов, содержащих закодированные значения типов АСН.1.

При тестировании процедуры верификации маршрута сертификации выполняются базовые тесты и тесты известного ответа. Базовые тесты являются обязательными. Тесты известного ответа являются дополнительными и предназначены для верификации маршрута сертификации с СОК и СОС, которые подписаны алгоритмом, определенным в СТБ 34.101.45-2013. Если интерфейсы испытуемой программы не позволяют выполнять верификацию произвольных маршрутов сертификации (с различными алгоритмами ЭЦП, длинами маршрутов и др.), то тесты известного ответа не выполняются, при этом в протоколе тестирование делается соответствующее примечание.

Для базовых тестов СОК и СОС, входящие в маршрут сертификации, предоставляются совместно с испытываемой программой или формируются экспертом (при наличии необходимого программного обеспечения). Перед тестированием процедуры верификация маршрута сертификации эксперт проводит визуальное сравнение текстового представления форматов СОК и СОС, входящих в маршрут сертификации, с форматами, определенными в Стандарте. Для изменения СОК и СОС, которое выполняется в некоторых базовых тестах, могут использоваться программы, описанные в приложении Г.2.

Для тестов известного ответа в качестве входных данных используются СОК и СОС, определенные в приложении Д (закодированные АСН.1-файлы с данными СОК и СОС являются неотъемлемой частью настоящей методики).

Базовые тесты. В ходе тестирования процедуры верификация маршрута сертификации выполняются следующие базовые тесты.

Тест ValidPathTest

1 Задать корректный маршрут сертификации.
2 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
3 Повторить шаги 1 – 2 не менее 9 раз, задавая различные корректные маршруты сертификации.
4 Возвратить `УСПЕХ`, если на шаге 2 маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
5 Возвратить `ОШИБКА`.

Примечание — Задаваемые в тесте ValidPathCertTest маршруты должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по верификации маршрута сертификации.

Тест InvalidPathCertTest

1 Изменить СОК, входящий в корректный маршрут сертификации, таким образом, чтобы СОК был признан некорректным.
2 Задать маршрут сертификации с измененным СОК.
3 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
4 Повторить шаги 1 – 3 не менее 9 раз, задавая различные маршруты сертификации с измененным СОК.
5 Возвратить `УСПЕХ`, если на шаге 3 маршрут отклонен и возвращена ошибка, соответствующая внесенным изменениям.
6 Возвратить `ОШИБКА`.

Примечание 1 — Изменения СОК, входящих в маршрут сертификации, могут состоять, например, в изменении значения компонента `signatureValue` или компонента `tbsCertificate`.
Примечание 2 — Задаваемые в тесте InvalidPathCertTest маршруты должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по верификации маршрута сертификации.

Тест InvalidPathCrlTest

1 Изменить СОС, входящий в корректный маршрут сертификации, таким образом, чтобы СОС был признан некорректным.
2 Задать маршрут сертификации с измененным СОС.
3 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
4 Повторить шаги 1 – 3 не менее 9 раз, задавая различные маршруты сертификации с измененным СОС.

5 Возвратить `УСПЕХ`, если на шаге 3 маршрут отклонен и возвращена ошибка, соответствующая внесенным изменениям.
6 Возвратить `ОШИБКА`.

Примечание 1 — Изменения СОС, входящих в маршрут сертификации, могут состоять, например, в изменении значения компонента `signatureValue` или компонента `tbsCertList`.
Примечание 2 — Задаваемые в тесте InvalidPathCertTest маршруты должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по верификации маршрута сертификации.

Тесты известного ответа. Тесты известного ответа основаны на тестах из документов «Public Key Interoperability Test Suite (PKITS) Certification Path Validation» (PKITS) и «NIST Recommendation for X.509 Path Validation».

Входные параметры процедуры верификации маршрута сертификации, используемые в тестах известного ответа, являются уточнением параметров, определенных в Стандарте. Большинство тестов известного ответа выполняется при следующих условиях:

- доверенной стороной является корневой УЦ;
- параметр `user-initial-policy-set` принимает специальное значение `anyPolicy`;
- параметр `initial-explicit-policy` установлен в `FALSE`;
- параметр `initial-policy-mapping-inhibit` установлен в `FALSE`;
- параметр `initial-any-policy-inhibit` установлен в `FALSE`.

Перечисленные выше значения параметров считаются параметрами по умолчанию.

Во всех тестах известного ответа, не связанных с обработкой политик применения сертификатов, используется политика по умолчанию 2.16.840.1.101.3.2.1.48.1. Если параметр `user-initial-policy-set` содержит указанное значение или параметр `initial-explicit-policy` установлен в `FALSE`, то значения этих параметров не влияют на результаты тестов, которые не связаны с обработкой политик. Значение параметра `initial-policy-mapping-inhibit` не влияет на результаты тестов, в которых отсутствуют сертификаты, содержащие расширение `PolicyMappings`, а параметр `initial-any-policy-inhibit` — на результаты тестов, в которых отсутствуют сертификаты с расширением `CertificatePolicy`, содержащим политику `anyPolicy`.

Если реализация процедуры верификации маршрута сертификации не позволяет установить значение параметра `user-initial-policy-set` в `anyPolicy`, то во всех случаях, когда необходимо установить данное значение параметра, следует использовать множество политик, перечисленных в таблице 1.

Таблица 1 — Политики применения сертификатов

Идентификатор	Зарегистрированное имя	Имя в методике
2.5.29.32.0	`anyPolicy`	`anyPolicy`
2.16.840.1.101.3.2.1.48.1	`NIST test-policy-1`	`NIST-test-policy-1`
2.16.840.1.101.3.2.1.48.2	`NIST test-policy-2`	`NIST-test-policy-2`
2.16.840.1.101.3.2.1.48.3	`NIST test-policy-3`	`NIST-test-policy-3`

Тест VSEETest1

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `ValidCertificatePathTest1EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — Цель теста VSEETest1 — проверка способности реализации обрабатывать последовательности имен и ЭЦП, а также проверять срок действия сертификатов маршрута сертификации. Кроме того, при тестировании проверяется корректность обработки расширений `BasicConstraints` и `KeyUsage` в промежуточных сертификатах.

Тест ICASTest2

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `BadSignedCACert`.
5 Задать в качестве СОС подчиненного УЦ `BadSignedCACRL`.
6 Задать в качестве СОК конечного участника `InvalidCASignatureTest2EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. ЭЦП в промежуточном сертификате некорректная.
9 Возвратить `ОШИБКА`.

Примечание — Цель теста ICASTest2 — проверка способности реализации выявлять некорректные ЭЦП в промежуточных сертификатах маршрута сертификации.

Тест IEESTest3

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `InvalidEESignatureTest3EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. ЭЦП в сертификате конечного участника не корректная.
9 Возвратить `ОШИБКА`.

Примечание — Цель теста IEESTest3 — проверка способности реализации выявлять некорректные ЭЦП в сертификате конечного участника.

Тест ICAnotBeforeDateTest1

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `BadnotBeforeDateCACert`.
5 Задать в качестве СОС подчиненного УЦ `BadnotBeforeDateCACRL`.
6 Задать в качестве СОК конечного участника `InvalidCAnotBeforeDateTest1EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. срок действия промежуточного сертификата еще не наступил.
9 Возвратить `ОШИБКА`.

Примечание — В тесте ICAnotBeforeDateTest1 компонент `notBefore` промежуточного сертификата указывает на дату, которая еще не наступила.

Тест IEEnotBeforeDateTest2

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `InvalidEEnotBeforeDateTest2EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. срок действия сертификата еще не наступил.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IEEnotBeforeDateTest2 компонент `notBefore` сертификата конечного участника указывает на дату, которая еще не наступила.

Тест Vpre2000UTCnotBeforeDateTest3

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `Validpre2000UTCnotBeforeDateTest3EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.

8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте Vpre2000UTCnotBeforeDateTest3 компонент `notBefore` сертификата конечного участника указывает на 1950 год и закодирован типом `UTCTime`.

Тест VGeneralizedTimenotBeforeDateTest4

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `ValidGeneralizedTimenotBeforeDateTest4EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте VGeneralizedTimenotBeforeDateTest4 компонент `notBefore` сертификата конечного участника закодирован типом `GeneralizedTime`.

Тест ICAnotAfterDateTest5

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `BadnotAfterDateCACert`.
5 Задать в качестве СОС подчиненного УЦ `BadnotAfterDateCACRL`.
6 Задать в качестве СОК конечного участника `InvalidCAnotAfterDateTest5EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. время действия промежуточного сертификата истекло.
9 Возвратить `ОШИБКА`.

Примечание — В тесте ICAnotAfterDateTest5 компонент `notAfter` промежуточного сертификата указывает на время, предшествующее текущему.

Тест IEEnotAfterDateTest6

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.

6 Задать в качестве СОК конечного участника `InvalidEEnotAfterDateTest6EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. время действия сертификата истекло.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IEEnotAfterDateTest6 компонент `notAfter` сертификата конечного участника указывает на время, предшествующее текущему.

Тест INCEETest1

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `InvalidNameChainingTest1EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. нарушена последовательность имен.
9 Возвратить `ОШИБКА`.

Примечание — В тесте INCEETest1 общее имя (атрибут `cn`) эмитента сертификата конечного участника не соответствует общему имени субъекта промежуточного сертификата.

Тест INCOTest2

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `NameOrderingCACert`.
5 Задать в качестве СОС подчиненного УЦ `NameOrderCACRL`.
6 Задать в качестве СОК конечного участника `InvalidNameChainingOrderTest2EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. нарушена последовательность имен.
9 Возвратить `ОШИБКА`.

Примечание — В тесте INCOTest2 имя эмитента сертификата конечного участника и имя субъекта промежуточного сертификата содержат одинаковые иерархические имена (RDNs), но порядок их следования отличается.

Тест VNCWTest3

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `ValidNameChainingWhitespaceTest3EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте VNCWTest3 имя эмитента сертификата конечного участника и имя субъекта промежуточного сертификата отличаются лишь промежутками между словами (количеством пробелов).

Тест VNCWTest4

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `ValidNameChainingWhitespaceTest4EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте VNCWTest4 имя эмитента сертификата конечного участника и имя субъекта промежуточного сертификата отличаются лишь пробелами в начале и в конце.

Тест VNCCTest5

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `ValidNameChainingCapitalizationTest5EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.

8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте VNCCTest5 имя эмитента сертификата конечного участника и имя субъекта промежуточного сертификата отличаются лишь регистром. Тест должен быть пройден успешно, т.к. сравнение не должно учитывать регистр символов.

Тест IRCATest2

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК первого подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС первого подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК второго подчиненного УЦ `RevokedsubCACert`.
7 Задать в качестве СОС второго подчиненного УЦ `RevokedsubCACRL`.
8 Задать в качестве СОК конечного участника `InvalidRevokedCATest2EE`.
9 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
10 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, либо выдано сообщение о невозможности определения его статуса, так как один из промежуточных сертификатов маршрута отозван.
11 Возвратить `ОШИБКА`.

Примечание — В тесте IRCATest2 СОК второго промежуточного УЦ содержится в СОС, выпущенном первым промежуточным УЦ.

Тест IREETest3

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК конечного участника `InvalidRevokedEETest3EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, т.к. он отозван.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IREETest3 СОК конечного участника содержится в СОС, выпущенном промежуточным УЦ.

Тест IBCRLSTest4

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `BadCRLSignatureCACert`.
5 Задать в качестве СОС подчиненного УЦ `BadCRLSignatureCACRL`.
6 Задать в качестве СОК конечного участника `InvalidBadCRLSignaturesTest4EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, либо выдано сообщение о невозможности определения его статуса, так как ЭЦП промежуточного СОС некорректная.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IBCRLSTest4 подпись промежуточного СОС некорректная.

Тест IBCRLINTest5

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `BadCRLIssuerNameCACert`.
5 Задать в качестве СОС подчиненного УЦ `BadCRLIssuerNameCACRL`.
6 Задать в качестве СОК конечного участника `InvalidBadCRLIssuerNameTest5EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, либо выдано сообщение о невозможности определения его статуса.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IBCRLINTest5 имя эмитента СОС, выпущенного промежуточным УЦ, не соответствует имени эмитента СОК конечного участника.

Тест VLSNTest16

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `LongSerialNumberCACert`.
5 Задать в качестве СОС подчиненного УЦ `LongSerialNumberCACRL`.
6 Задать в качестве СОК конечного участника `ValidLongSerialNumberTest16EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте VLSNTest16 серийный номер СОК конечного участника состоит из 20-ти октетов, и он не содержится в СОС. Однако в СОС содержится серийный номер, отличающийся от данного лишь последним значимым октетом.

Тест IMbasicConstraintsTest1

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `MissingbasicConstraintsCACert`.
5 Задать в качестве СОС подчиненного УЦ `MissingbasicConstraintsCACRL`.
6 Задать в качестве СОК конечного участника `InvalidMissingbasicConstraintsTest1EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IMbasicConstraintsTest1 в сертификате промежуточного УЦ отсутствует расширение `BasicConstraints`.

Тест IcAFalseTest3

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `basicConstraintsNotCriticalcAFalseCACert`.
5 Задать в качестве СОС подчиненного УЦ `basicConstraintsNotCriticalcAFalseCACRL`.
6 Задать в качестве СОК конечного участника `InvalidcAFalseTest3EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IcAFalseTest3 в сертификате промежуточного УЦ присутствует некритическое расширение `BasicConstraints`, в котором компонент `cA` установлен в `FALSE`.

Тест IpathLenConstraintTest5

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК первого подчиненного УЦ `pathLenConstraint0CACert`.
5 Задать в качестве СОС первого подчиненного УЦ `pathLenConstraint0CACRL`.
6 Задать в качестве СОК второго подчиненного УЦ `pathLenConstraint0subCACert`.
7 Задать в качестве СОС второго подчиненного УЦ `pathLenConstraint0subCACRL`.
8 Задать в качестве СОК конечного участника `InvalidpathLenConstraintTest5EE`.

9 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
10 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным.
11 Возвратить `ОШИБКА`.

Примечание — В тесте IpathLenConstraintTest5 в сертификате первого промежуточного УЦ присутствует расширение `BasicConstraints`, в котором компонент `pathLenConstraint` установлен в 0. Это означает, что больше промежуточных сертификатов быть не должно, но в маршруте присутствует второй промежуточный сертификат.

Тест VpathLenConstraintTest7

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `pathLenConstraint0CACert`.
5 Задать в качестве СОС подчиненного УЦ `pathLenConstraint0CACRL`.
6 Задать в качестве СОК конечного участника `ValidpathLenConstraintTest7EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте VpathLenConstraintTest7 в сертификате промежуточного УЦ присутствует расширение `BasicConstraints`, в котором компонент `pathLenConstraint` установлен в 0. Это означает, что больше промежуточных сертификатов быть не должно. Следующим сертификатом является сертификат конечного участника.

Тест IkeyUsageCkeyCertSignFalseTest1

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `keyUsageCriticalkeyCertSignFalseCACert`.
5 Задать в качестве СОС подчиненного УЦ `keyUsageCriticalkeyCertSignFalseCACRL`.
6 Задать в качестве СОК конечного участника `InvalidkeyUsageCriticalkeyCertSignFalseTest1EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, либо выдано сообщение о невозможности определения его статуса.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IkeyUsageCkeyCertSignFalseTest1 промежуточный сертификат содержит критическое расширение `KeyUsage`, в котором компонент `keyCertSign` установлен в `FALSE`.

Тест VkeyUsageNCTest3

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `keyUsageNotCriticalCACert`.
5 Задать в качестве СОС подчиненного УЦ `keyUsageNotCriticalCACRL`.
6 Задать в качестве СОК конечного участника `ValidkeyUsageNotCriticalTest3EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте VkeyUsageNCTest3 промежуточный сертификат содержит некритическое расширение `KeyUsage`.

Тест IkeyUsageCcRLSignFalseTest4

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `keyUsageCriticalcRLSignFalseCACert`.
5 Задать в качестве СОС подчиненного УЦ `keyUsageCriticalcRLSignFalseCACRL`.
6 Задать в качестве СОК конечного участника `InvalidkeyUsageCriticalcRLSignFalseTest4EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если СОК конечного участника признан недействительным, либо выдано сообщение о невозможности определения его статуса.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IkeyUsageCcRLSignFalseTest4 промежуточный сертификат содержит критическое расширение `KeyUsage`, в котором компонент `cRLSign` установлен в `FALSE`.

Тест DiffPTest4

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК первого подчиненного УЦ `GoodCACert`.
5 Задать в качестве СОС первого подчиненного УЦ `GoodCACRL`.
6 Задать в качестве СОК второго подчиненного УЦ `GoodsubCACert`.
7 Задать в качестве СОС второго подчиненного УЦ `GoodsubCACRL`.
8 Задать в качестве СОК конечного участника `DifferentPoliciesTest4EE`.
9 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
10 Возвратить `УСПЕХ`, если маршрут отклонен и возвращена ошибка.

11 Возвратить ОШИБКА.

Примечание — В тесте DiffPTest4 все сертификаты маршрута, кроме сертификата конечного участника, подчиняются одной и той же политике.

Тест DiffPTest5

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ TrustAnchorRootCertificate.
3 Задать в качестве СОС корневого УЦ TrustAnchorRootCRL.
4 Задать в качестве СОК первого подчиненного УЦ GoodCACert.
5 Задать в качестве СОС первого подчиненного УЦ GoodCACRL.
6 Задать в качестве СОК второго подчиненного УЦ PoliciecP2subCA2Cert.
7 Задать в качестве СОС второго подчиненного УЦ PoliciesP2subCA2CRL.
8 Задать в качестве СОК конечного участника DifferentPoliciesTest5EE.
9 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
10 Возвратить УСПЕХ, если маршрут отклонен и возвращена ошибка.
11 Возвратить ОШИБКА.

Примечание — В тесте DiffPTest5 все сертификаты маршрута, кроме второго промежуточного сертификата, подчиняются одной и той же политике.

Тест DiffPTest12

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ TrustAnchorRootCertificate.
3 Задать в качестве СОС корневого УЦ TrustAnchorRootCRL.
4 Задать в качестве СОК подчиненного УЦ PoliciesP3CACert.
5 Задать в качестве СОС подчиненного УЦ PoliciesP3CACRL.
6 Задать в качестве СОК конечного участника DifferentPoliciesTest12EE.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить УСПЕХ, если маршрут отклонен и возвращена ошибка.
9 Возвратить ОШИБКА.

Примечание — В тесте DiffPTest12 маршрут состоит из двух сертификатов, которые подчиняются различным политикам.

Тест VdistributionPointTest1

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ TrustAnchorRootCertificate.
3 Задать в качестве СОС корневого УЦ TrustAnchorRootCRL.
4 Задать в качестве СОК подчиненного УЦ distributionPoint1CACert.
5 Задать в качестве СОС подчиненного УЦ distributionPoint1CACRL.
6 Задать в качестве СОК конечного участника ValiddistributionPointTest1EE.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.

8 Возвратить УСПЕХ, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить ОШИБКА.

Примечание — В тесте VdistributionPointTest1 СОК конечного участника содержит расширение CRLDistributionPoints, в котором определен один пункт распространения СОС, заданный компонентом distributionPoint. СОС, область действия которого распространяется на данный сертификат конечного участника, содержит расширение IssuingDistributionPoint, в котором определен тот же пункт распространения СОС.

Тест IdistributionPointTest2

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ TrustAnchorRootCertificate.
3 Задать в качестве СОС корневого УЦ TrustAnchorRootCRL.
4 Задать в качестве СОК подчиненного УЦ distributionPoint1CACert.
5 Задать в качестве СОС подчиненного УЦ distributionPoint1CACRL.
6 Задать в качестве СОК конечного участника InvaliddistributionPointTest2EE.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить УСПЕХ, если маршрут отклонен, т.к. сертификат конечного участника отозван.
9 Возвратить ОШИБКА.

Примечание — В тесте IdistributionPointTest2 СОК конечного участника содержит расширение CRLDistributionPoints, в котором определен один пункт распространения СОС, заданный компонентом distributionPoint. СОС, область действия которого распространяется на данный сертификат конечного участника, содержит расширение IssuingDistributionPoint, в котором определен тот же пункт распространения СОС. СОК конечного участника отозван.

Тест IdistributionPointTest3

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ TrustAnchorRootCertificate.
3 Задать в качестве СОС корневого УЦ TrustAnchorRootCRL.
4 Задать в качестве СОК подчиненного УЦ distributionPoint1CACert.
5 Задать в качестве СОС подчиненного УЦ distributionPoint1CACRL.
6 Задать в качестве СОК конечного участника InvaliddistributionPointTest3EE.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить УСПЕХ, если маршрут отклонен, т.к. статус сертификата конечного участника не может быть определен.
9 Возвратить ОШИБКА.

Примечание — В тесте IdistributionPointTest3 СОК конечного участника содержит расширение CRLDistributionPoints, в котором определен один пункт распространения СОС, заданный компонентом distributionPoint. СОС, область действия которого распространяется на

данный сертификат конечного участника, содержит расширение `IssuingDistributionPoint`, в котором определен другой пункт распространения СОС.

Тест VNoIssuingDistributionPointTest10

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `NoissuingDistributionPointCACert`.
5 Задать в качестве СОС подчиненного УЦ `NoissuingDistributionPointCACRL`.
6 Задать в качестве СОК конечного участника `ValidNoissuingDistributionPointTest10EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
9 Возвратить `ОШИБКА`.

Примечание — В тесте VNoIssuingDistributionPointTest10 СОС, область действия которого распространяется на сертификат конечного участника, не содержит расширение `IssuingDistributionPoint`. СОК конечного участника содержит расширение `CRLDistributionPoints`, в котором определен пункт распространения СОС через `distributionPoint`.

Тест IonlyContainsUCCRLTest11

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `onlyContainsUserCertsCACert`.
5 Задать в качестве СОС подчиненного УЦ `onlyContainsUserCertsCACRL`.
6 Задать в качестве СОК конечного участника `InvalidonlyContainsUserCertsTest11EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут отклонен по причине невозможности определения статуса сертификата конечного участника.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IonlyContainsUCCRLTest11 СОС, область действия которого распространяется на сертификат конечного участника, содержит расширение `IssuingDistributionPoint`, в котором компонент `onlyContainsUserCerts` установлен в TRUE. Конечным участником является УЦ.

Тест IonlyContainsACTest14

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.

4 Задать в качестве СОК подчиненного УЦ `onlyContainsAttributeCertsCACert`.
5 Задать в качестве СОС подчиненного УЦ `onlyContainsAttributeCertsCACRL`.
6 Задать в качестве СОК конечного участника `InvalidonlyContainsAttributeCertsTest14EE`.
7 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
8 Возвратить `УСПЕХ`, если маршрут отклонен по причине невозможности определения статуса сертификата конечного участника.
9 Возвратить `ОШИБКА`.

Примечание — В тесте IonlyContainsACTest14 СОС, область действия которого распространяется на сертификат конечного участника, содержит расширение `IssuingDistributionPoint`, в котором компонент `onlyContainsAttributeCerts` установлен в TRUE.

Тест VonlySomeReasonsTest15

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК подчиненного УЦ `onlySomeReasonsCA3Cert`.
5 Задать в качестве первого СОС подчиненного УЦ `onlySomeReasonsCA3compromiseCRL`.
6 Задать в качестве второго СОС подчиненного УЦ `onlySomeReasonsCA3otherreasonsCRL`.
7 Задать в качестве СОК конечного участника `ValidonlySomeReasonsTest15EE`.
8 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
9 Возвратить `УСПЕХ`, если:
– маршрут отклонен и выдано предупреждение о невозможности определения статуса сертификата для реализации, которая не может обработать компонент `onlySomeReasons` расширения `IssuingDistributionPoint`,
– маршрут сертификации успешно проверен и СОК конечного участника признан действительным для реализации, которая может обработать компонент `onlySomeReasons` расширения `IssuingDistributionPoint`.
10 Возвратить `ОШИБКА`.

Примечание — В тесте VonlySomeReasonsTest15 промежуточным УЦ выпущены два СОС: область действия первого из СОС распространяется лишь на СОК, причина отзыва которых — компрометация ключа (`keyCompromise`) и компрометация УЦ (`cACompromise`); второй СОС содержит все сертификаты, отозванные по иной причине. Оба СОС имеют расширение `IssuingDistributionPoint`, в котором указан один и тот же пункт распространения СОС. СОК конечного участника содержит расширение `CRLDistributionPoints` с тем же пунктом распространения СОС.

Тест VUNCCETest1

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК конечного участника `ValidUnknownNotCriticalCertificateExtension-Test1EE`.
5 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
6 Возвратить `УСПЕХ`, если маршрут сертификации успешно проверен и СОК конечного участника признан действительным.
7 Возвратить `ОШИБКА`.

Примечание — В тесте VUNCCETest1 СОК конечного участника содержит некритическое пользовательское расширение.

Тест IUCCETest2

1 Задать параметры по умолчанию.
2 Задать в качестве СОК корневого УЦ `TrustAnchorRootCertificate`.
3 Задать в качестве СОС корневого УЦ `TrustAnchorRootCRL`.
4 Задать в качестве СОК конечного участника `InvalidUnknownCriticalCertificateExtension-Test2EE`.
5 Средствами испытываемой программы выполнить процедуру верификация маршрута сертификации.
6 Возвратить `УСПЕХ`, если маршрут отклонен, т.к. СОК конечного участника содержит критическое нераспознанное расширение.
7 Возвратить `ОШИБКА`.

Примечание — В тесте IUCCETest2 СОК конечного участника содержит критическое пользовательское расширение.

6.3 Анализ исходных текстов

6.3.1 Корректность использования криптографических алгоритмов

В программе для обеспечения и проверки подлинности СОК и СОС используются алгоритмы хэширования и алгоритмы электронной цифровой подписи. Дополнительно алгоритмы хэширования могут использоваться для определения идентификаторов ключа (см., например, п. 6.2.1.2 СТБ 34.101.19).

Использование функций, реализующих криптографический алгоритм, должно выполняться в соответствии с СТБ 34.101.19, 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.

В процедурах формирования СОК и СОС используются следующие секретные данные:

– личный ключ подписи;

– одноразовый ключ подписи.

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 процедуры формирования СОК

При анализе корректности реализации процедуры формирования СОК исходные тексты программы оцениваются частично, по выбору эксперта. Корректность реализации процедуры означает, что реализация функционально соответствуют п. 6 СТБ 34.101.19 и что реализация не содержит ошибок и уязвимостей.

Эксперт должен проверить по крайней мере следующие аспекты реализации процедуры формирования СОК:

1 Реализация должна подписывать СОК на личном ключе эмитента.

2 Реализация должна включать в СОК идентификатор криптографического алгоритма, который используется для подписи СОК.

3 Реализация должна включать в СОК информацию, которая определяет долговременные параметры алгоритмов ЭЦП, используемые для подписи СОК.

4 Реализация должна включать в СОК информацию, которая определяет алгоритм, с которым должен использоваться открытый ключ из СОК.

5 Реализация должна поддерживать включение в СОК по крайней мере расширений `authorityKeyIdentifier` (см. п. 6.2.1.1 СТБ 34.101.19), `SubjectKeyIdentifier` (см. п. 6.2.1.2 СТБ 34.101.19), `KeyUsage` (см. п. 6.2.1.3 СТБ 34.101.19), `certificatePolicies` (см. п. 6.2.1.4 СТБ 34.101.19), `BasicConstraints` (см. п. 6.2.1.9 СТБ 34.101.19).

6.3.4 Корректность процедуры формирования СОС

При анализе корректности реализации процедуры формирования СОС исходные тексты программы оцениваются частично, по выбору эксперта. Корректность реализации процедуры означает, что реализация функционально соответствуют п. 7 СТБ 34.101.19 и что реализация не содержит ошибок и уязвимостей.

Эксперт должен проверить по крайней мере следующие аспекты реализации процедуры формирования СОС:

1 Реализация должна подписывать СОС на личном ключе эмитента.

2 Реализация должна включать в СОС идентификатор криптографического алгоритма, который используется для подписи СОС.

3 Реализация должна включать в СОС информацию, которая определяет долговременные параметры алгоритмов ЭЦП, используемые при подписи СОС.

4 Реализация должна включать в СОС всю необходимую информацию о статусе отзыва сертификата.

5 Реализация должна поддерживать включение в СОС по крайней мере расширений `authorityKeyIdentifier` (см. п.7.2.1 и п. 6.2.1.1 СТБ 34.101.19), `CRLNumber` (см. п. 7.2.3 СТБ 34.101.19).

6.3.5 Корректность процедуры верификации маршрута сертификации

При анализе корректности реализации процедуры верификации маршрута сертификации исходные тексты программы оцениваются частично, по выбору эксперта. Корректность реализации процедуры означает, что реализация функционально соответствуют п. 8 СТБ 34.101.19 и что реализация не содержит ошибок и уязвимостей.

Эксперт должен проверить по крайней мере следующие аспекты реализации процедуры верификации маршрута сертификации:

1 Реализация должна проверять ЭЦП всех сертификатов маршрута сертификации.

2 Реализация должна проверять, что компонент `notBefore` каждого сертификата в маршруте указывает на дату проверки или более раннюю дату, а компонент `notAfter` содержит дату проверки или более позднюю дату.

3 Реализация должна обрабатывать последовательность имен согласно СТБ 34.101.19: в маршруте сертификации имя эмитента промежуточного либо конечного сертификата должно соответствовать имени субъекта предшествующего сертификата цепочки, а имя эмитента первого сертификата цепочки должно соответствовать имени корневого УЦ (при проверке соответствия имена могут быть закодированы различными типами; для

типа `PrintableString` перед сравнением необходимо сначала удалить лишние пробелы в начале и в конце строки, а промежутки между словами заменить одним пробелом).

4 Реализация должна распознавать в СОК по крайней мере расширения `authorityKeyIdentifier` (см. п. 6.2.1.1 СТБ 34.101.19), `SubjectKeyIdentifier` (см. п. 6.2.1.2 СТБ 34.101.19), `KeyUsage` (см. п. 6.2.1.3 СТБ 34.101.19), `certificatePolicies` (см. п. 6.2.1.4 СТБ 34.101.19), `SubjectAltName` (см. п. 6.2.1.6 СТБ 34.101.19), `BasicConstraints` (см. п. 6.2.1.9 СТБ 34.101.19), `NameConstraints` (см. п. 6.2.1.10 СТБ 34.101.19), `PolicyConstraints` (см. п. 6.2.1.11 СТБ 34.101.19), `ExtKeyUsageSyntax` (см. п. 6.2.1.12 СТБ 34.101.19), `InhibitAnyPolicy` (см. п. 6.2.1.14 СТБ 34.101.19).

5 Реализация должна извлекать из СОС необходимую информацию о статусе отзыва каждого сертификата маршрута.

6 Реализация должна проверять для СОС подпись эмитента, при этом эмитент должен иметь полномочия на подпись, а область действия СОС должна покрывать проверяемый сертификат.

7 Реализация должна распознавать в СОС по крайней мере расширения `authorityKeyIdentifier` (см. п.7.2.1 и п. 6.2.1.1 СТБ 34.101.19), `CRLNumber` (см. п. 7.2.3 СТБ 34.101.19).

6.3.6 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 – 6.3.5.

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.).

6.3.7 Отсутствие недокументированных возможностей

Эксперт определяет отсутствие недокументированных возможностей по результатам проверок, выполненных в п. 6.3.1 – 6.3.6.

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 открытыми ключами согласно СТБ 34.101.19-2012

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 открытыми ключами согласно СТБ 34.101.19-2012.

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

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 открытыми ключами согласно СТБ 34.101.19. 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 открытыми ключами согласно СТБ 34.101.19-2012

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	IssueCertTest	{Поле 8}
2	ExtCertTest	{Поле 8}
3	IssueCrlTest	{Поле 8}
4	ExtCrlTest	{Поле 8}
5	EntryExtCrlTest	{Поле 8}
6	ValidPathTest	{Поле 8}
7	InvalidPathCertTest	{Поле 8}
8	InvalidPathCrlTest	{Поле 8}
...	...	...

3. Заключение по результатам тестирования: программа {Поле 4} соответствует (не соответствует) требованиям, установленным в СТБ 34.101.19-2012.

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 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 34.101.19. 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 открытыми ключами согласно СТБ 34.101.19-2012

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 процедуры формирования СОК	{Поле 8}
5	Корректность процедуры формирования СОС	{Поле 8}
6	Корректность процедуры верификации маршрута сертификации	{Поле 8}
7	Корректность обработки исключительных ситуаций	{Поле 8}
8	Отсутствие недокументированных возможностей	{Поле 8}

3. Заключение по результатам анализа исходных текстов программ: программа {Поле 4} соответствует требованиям, установленным в СТБ 34.101.19–2012.

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 34.101.19. 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 запросов на получение сертификата, используемых в тестах известного ответа. Каждый запрос в тестах основан на базовом запросе. Этот базовый запрос содержит типичные для всех запросов значения основных компонентов. При описании тестовых запросов приводятся лишь значения компонентов, которые отличаются от базовых.

Д.1 Базовый СОК корневого УЦ

Компонент сертификата или тип ASN.1	Флаг	Значение компонента	Пояснения
Certificate			
tbsCertificate			Значение данного компонента подписывается ЭЦП
version		2	Версия 3 СОК
serialNumber			
CertificateSerialNumber		{Определенное значение}	Значение всегда строго задано
signature			
AlgorithmIdentifier			Должен совпадать с идентификатором, указанным в компоненте `signatureAlgorithm`
algorithm		`bign-with-hbelt`	Алгоритмы ЭЦП СТБ 34.101.45-2013 с функцией хэширования СТБ 34.101.31-2011
parameters		`NULL`	Параметры по умолчанию `bign-curve256v1`
issuer			
Name		{Определенное имя}	Использует формат RFC2253
validity			
notBefore			
Time			
UTCTime		100101083000Z	01 января 2010, 08:30:00 GMT
notAfter			
Time			
UTCTime		301231083000Z	31 декабря 2030, 08:30:00 GMT
subject			
Name		{Определенное имя}	Использует формат RFC2253
subjectPublicKeyInfo			
algorithm			
AlgorithmIdentifier			
algorithm		`bign-pubkey`	Открытый ключ СТБ 34.101.45-2013
parameters		`bign-curve256v1`	Стандартные параметры для уровня стойкости $l = 128$
subjectPublicKey			
BIT STRING		Значение открытого ключа	Длина 512 бит
extensions			
authorityKeyIdentifier	`FALSE`		
keyIdentifier		`OCTET STRING`	Хэш-значение СТБ 34.101.31-2011 от кодированного значения открытого ключа эмитента
subjectKeyIdentifier	`FALSE`		
keyIdentifier		`OCTET STRING`	Хэш-значение СТБ 34.101.31-2011 от кодированного значения открытого ключа субъекта. Совпадает с идентификатором, указанным в компоненте `authorityKeyIdentifier`.
basicConstraints	`TRUE`		
cA		`TRUE`	
pathLenConstraint		`INTEGER`	Отсутствует
keyUsage	`TRUE`		
digitalSignature		0	
nonRepudiation		0	
keyEncipherment		0	
dataEncipherment		0	
keyAgreement		0	
keyCertSign		1	
cRLSign		1	
encipherOnly		0	
decipherOnly		0	
signatureAlgorithm			
AlgorithmIdentifier			
algorithm		`bign-with-hbelt`	Алгоритмы ЭЦП СТБ 34.101.45-2013 с функцией хэширования СТБ 34.101.31-2011
parameters		`NULL`	Параметры по умолчанию `bign-curve256v1`
signature		`BIT STRING`	ЭЦП от `tbsCertificate`

Д.2 Базовый СОК подчиненного УЦ

Компонент сертификата или тип ASN.1	Флаг	Значение компонента	Пояснения
Certificate			
tbsCertificate			Значение данного компонента подписывается ЭЦП
version		2	Версия 3 СОК
serialNumber			
CertificateSerialNumber		{Определенное значение}	Значение всегда строго задано
signature			
AlgorithmIdentifier			Должен совпадать с идентификатором, указанным в компоненте `signatureAlgorithm`
algorithm		`bign-with-hbelt`	Алгоритмы ЭЦП СТБ 34.101.45-2013 с функцией хэширования СТБ 34.101.31-2011
parameters		`NULL`	Параметры по умолчанию `bign-curve256v1`
issuer			
Name		{Определенное имя}	Использует формат X.500
validity			
notBefore			
Time			
UTCTime		100101083000Z	01 января 2010, 08:30:00 GMT
notAfter			
Time			
UTCTime		301231083000Z	31 декабря 2030, 08:30:00 GMT
subject			
Name		{Определенное имя}	Использует формат X.500
subjectPublicKeyInfo			
algorithm			
AlgorithmIdentifier			
algorithm		`bign-pubkey`	Открытый ключ СТБ 34.101.45-2013
parameters		`bign-curve256v1`	Стандартные параметры
subjectPublicKey			
BIT STRING		Значение открытого ключа	Длина 512 бит
extensions			
authorityKeyIdentifier	FALSE		
keyIdentifier		`OCTET STRING`	Хэш-значение СТБ 34.101.31-2011 от кодированного значения открытого ключа эмитента
subjectKeyIdentifier	FALSE		
keyIdentifier		`OCTET STRING`	Хэш-значение СТБ 34.101.31-2011 от кодированного значения открытого ключа субъекта.
basicConstraints	TRUE		
cA		`TRUE`	
pathLenConstraint		`INTEGER`	Отсутствует
keyUsage	TRUE		
digitalSignature		0	
nonRepudiation		0	
keyEncipherment		0	
dataEncipherment		0	
keyAgreement		0	
keyCertSign		1	
cRLSign		1	
encipherOnly		0	
decipherOnly		0	
certificatePolicies	FALSE		Не содержит классификаторов политик
PolicyInformation			
policyIdentifier			
CertPolicyId		`NIST-test-policy-1`	Тестовая политика по умолчанию
signatureAlgorithm			
AlgorithmIdentifier			
algorithm		`bign-with-hbelt`	Алгоритмы ЭЦП СТБ 34.101.45-2013 с функцией хэширования СТБ 34.101.31-2011
parameters		`NULL`	Параметры по умолчанию `bign-curve256v1`
signature		`BIT STRING`	ЭЦП от `tbsCertificate`

Д.3 Базовый СОК конечного участника

Компонент сертификата или тип ASN.1	Флаг	Значение компонента	Пояснения
Certificate			
tbsCertificate			Значение данного компонента подписывается ЭЦП
version		2	Версия 3 СОК
serialNumber			
CertificateSerialNumber		{Определенное значение}	Значение всегда строго задано
signature			
AlgorithmIdentifier			Должен совпадать с идентификатором, указанным в компоненте signatureAlgorithm
algorithm		bign-with-hbelt	Алгоритмы ЭЦП СТБ 34.101.45-2013 с функцией хэширования СТБ 34.101.31-2011
parameters		NULL	Параметры по умолчанию bign-curve256v1
issuer			
Name		{Определенное имя}	Использует формат X.500
validity			
notBefore			
Time			
UTCTime		100101083000Z	01 января 2010, 08:30:00 GMT
notAfter			
Time			
UTCTime		301231083000Z	31 декабря 2030, 08:30:00 GMT
subject			
Name		{Определенное имя}	Использует формат X.500
subjectPublicKeyInfo			
algorithm			
AlgorithmIdentifier			
algorithm		bign-pubkey	Открытый ключ СТБ 34.101.45-2013
parameters		bign-curve256v1	Стандартные параметры
subjectPublicKey			
BIT STRING		Значение открытого ключа	Длина 512 бит
extensions			
authorityKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Хэш-значение СТБ 34.101.31-2011 от кодированного значения открытого ключа эмитента
subjectKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Хэш-значение СТБ 34.101.31-2011 от кодированного значения открытого ключа субъекта.
basicConstraints	FALSE		
cA		FALSE	
pathLenConstraint		Отсутствует	
keyUsage	TRUE		
digitalSignature		1	
nonRepudiation		1	
keyEncipherment		1	
dataEncipherment		0	
keyAgreement		0	
keyCertSign		0	
cRLSign		0	
encipherOnly		0	
decipherOnly		0	
certificatePolicies	FALSE		Не содержит классификаторов политик
PolicyInformation			
policyIdentifier			
CertPolicyId		NIST-test-policy-1	Тестовая политика по умолчанию
signatureAlgorithm			
AlgorithmIdentifier			
algorithm		bign-with-hbelt	Алгоритмы ЭЦП СТБ 34.101.45-2013 с функцией хэширования СТБ 34.101.31-2011
parameters		NULL	Параметры по умолчанию bign-curve256v1
signature		BIT STRING	ЭЦП от tbsCertificate

Д.4 Базовый СОС

Компонент сертификата или тип ASN.1	Флаг	Значение компонента	Пояснения
Certificate			
tbsCertificate			Значение данного компонента подписывается ЭЦП
version		1	Версия 2 СОС
signature			
AlgorithmIdentifier			Должен совпадать с идентификатором, указанным в компоненте signatureAlgorithm
algorithm		bign-with-hbelt	Алгоритмы ЭЦП СТБ 34.101.45-2013 с функцией хэширования СТБ 34.101.31-2011
parameters		NULL	Параметры по умолчанию bign-curve256v1
issuer			
Name		{Определенное имя}	Имя эмитента СОС в формате X.500
thisUpdate			
Time			
UTCTime		100101083000Z	01 января 2010, 08:30:00 GMT
nextUpdate			
Time			
UTCTime		301231083000Z	31 декабря 2030, 08:30:00 GMT
revokedCertificates			Отсутствуют
userCertificate			
CertificateSerialNumber			
revocationDate			
Time			
UTCTime		100101083000Z	01 января 2010, 08:30:00 GMT
crlEntryExtensions			
reasonCode	FALSE		
CRLReason		keyCompromise	
BIT STRING		Значение открытого ключа	Длина 512 бит
crlExtensions			
crlNumber	FALSE	1	
authorityKeyIdentifier	FALSE		
keyIdentifier		OCTET STRING	Хэш-значение СТБ 34.101.31-2011 от кодированного значения открытого ключа эмитента
signatureAlgorithm			
AlgorithmIdentifier			
algorithm		bign-with-hbelt	Алгоритмы ЭЦП СТБ 34.101.45-2013 с функцией хэширования СТБ 34.101.31-2011
parameters		NULL	Параметры по умолчанию bign-curve256v1
signature		BIT STRING	ЭЦП от tbsCertificate

Д.5 Тестовые СОК и СОС

Д.5.1 TrustAnchorRootCertificate

Основан на: базовый СОК корневого УЦ
Серийный номер: 1
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Самоподписанный сертификат

Д.5.2 TrustAnchorRootCRL

Основан на: базовый СОС
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Отозванные сертификаты:
serialNumber: 104
crlEntryExtensions:
  reasonCodeExtension: не критическое
    reasons: keyCompromise
Кем подписан: TrustAnchorRootCertificate

Д.5.3 GoodCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 2
Эмитент: `cn` = Trust Anchor, `o` = Test Certificates 2011, `c` = BY
Субъект: `cn` = Good CA, `o` = Test Certificates 2011, `c` = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.4 GoodCACRL

Основан на: базовый СОС
Эмитент: `cn` = Good CA, `o` = Test Certificates 2011, `c` = BY
Отозванные сертификаты:
`serialNumber`: 14
`crlEntryExtensions`:
  `reasonCodeExtension`: не критическое
    `reasons`: keyCompromise
`serialNumber`: 15
`crlEntryExtensions`:
  `reasonCodeExtension`: не критическое
    `reasons`: keyCompromise
Кем подписан: GoodCACert

Д.5.5 ValidCertificatePathTest1EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: `cn` = Good CA, `o` = Test Certificates 2011, `c` = BY
Субъект: `cn` = Valid EE Certificate Test1, `o` = Test Certificates 2011, `c` = BY
Кем подписан: GoodCACert

Д.5.6 BadSignedCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 3
Эмитент: `cn` = Trust Anchor, `o` = Test Certificates 2011, `c` = BY
Субъект: `cn` = Bad Signed CA, `o` = Test Certificates 2011, `c` = BY
Кем подписан: TrustAnchorRootCertificate (подпись изменена)

Д.5.7 BadSignedCACRL

Основан на: базовый СОС
Эмитент: `cn` = Bad Signed CA, `o` = Test Certificates 2011, `c` = BY
Кем подписан: BadSignedCACert

Д.5.8 InvalidCASignatureTest2EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: `cn` = Bad Signed CA, `o` = Test Certificates 2011, `c` = BY
Субъект: `cn` = Invalid CA Signature Test2, `o` = Test Certificates 2011, `c` = BY
Кем подписан: BadSignedCACert

Д.5.9 InvalidEESignatureTest3EE

Основан на: базовый СОК конечного участника
Серийный номер: 2
Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid EE Signature Test3, o = Test Certificates 2011, c = BY
Кем подписан: GoodCACert (подпись изменена)

Д.5.10 BadnotBeforeDateCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 4
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
notBefore: UTC: "470101120100Z"
notAfter: UTC: "490101120100Z"
Субъект: cn = Bad notBefore Date CA, o = Test Certificates 2011, c = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.11 BadnotBeforeDateCACRL

Основан на: базовый СОС
Эмитент: cn = Bad notBefore Date CA, o = Test Certificates 2011, c = BY
Кем подписан: BadnotBeforeDateCACert

Д.5.12 InvalidCAnotBeforeDateTest1EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = Bad notBefore Date CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid CA notBefore Date EE Certificate Test1, o = Test Certificates 2011, c = BY
Кем подписан: BadnotBeforeDateCACert

Д.5.13 InvalidEEnotBeforeDateTest2EE

Основан на: базовый СОК конечного участника
Серийный номер: 3
Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY
notBefore: UTC: "470101120100Z"
notAfter: UTC: "490101120100Z"
Субъект: cn = Invalid EE notBefore Date EE Certificate Test2, o = Test Certificates 2011, c = BY
Кем подписан: GoodCACert

Д.5.14 Validpre2000UTCnotBeforeDateTest3EE

Основан на: базовый СОК конечного участника
Серийный номер: 4
Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY
notBefore: UTC: "500101120100Z"
Субъект: cn = Valid pre2000 UTC notBefore Date EE Certificate Test3, o = Test Certificates 2011, c = BY
Кем подписан: GoodCACert

Д.5.15 ValidGeneralizedTimenotBeforeDateTest4EE

Основан на: базовый СОК конечного участника
Серийный номер: 5
Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY
notBefore: GT: "20020101120100Z"
Субъект: cn = Valid GeneralizedTime notBefore Date EE Certificate Test4, o = Test Certificates 2011, c = BY
Кем подписан: GoodCACert

Д.5.16 BadnotAfterDateCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 5
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
notAfter: UTC: "110101083000Z"
Субъект: cn = Bad notAfter Date CA, o = Test Certificates 2011, c = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.17 BadnotAfterDateCACRL

Основан на: базовый СОС
Эмитент: cn = Bad notAfter Date CA, o = Test Certificates 2011, c = BY
Кем подписан: BadnotAfterDateCACert

Д.5.18 InvalidCAnotAfterDateTest5EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = Bad notAfter Date CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid CA notAfter Date EE Certificate Test5, o = Test Certificates 2011, c = BY
Кем подписан: BadnotAfterDateCACert

Д.5.19 InvalidEEnotAfterDateTest6EE

Основан на: базовый СОК конечного участника
Серийный номер: 6
Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY
notAfter: UTC: "110101083000Z"
Субъект: cn = Invalid EE notAfter Date EE Certificate Test6, o = Test Certificates 2011, c = BY
Кем подписан: GoodCACert

Д.5.20 InvalidNameChainingTest1EE

Основан на: базовый СОК конечного участника
Серийный номер: 9
Эмитент: cn = Good CA Root, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid Name Chaining EE Certificate Test1, o = Test Certificates 2011, c = BY
Кем подписан: GoodCACert

Д.5.21 NameOrderingCACert

Основан на: базовый СОК подчиненного УЦ

Серийный номер: 6

Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY

Субъект: cn = Name Ordering CA, ou = Organizational Unit Name 2, ou = Organizational Unit Name 1, o = Test Certificates 2011, c = BY

Кем подписан: TrustAnchorRootCertificate

Д.5.22 NameOrderCACRL

Основан на: базовый СОС

Эмитент: cn = Name Ordering CA, ou = Organizational Unit Name 2, ou = Organizational Unit Name 1, o = Test Certificates 2011, c = BY

Кем подписан: NameOrderingCACert

Д.5.23 InvalidNameChainingOrderTest2EE

Основан на: базовый СОК конечного участника

Серийный номер: 1

Эмитент: cn = Name Ordering CA, ou = Organizational Unit Name 1, ou = Organizational Unit Name 2, o = Test Certificates 2011, c = BY

Субъект: cn = Invalid Name Chaining Order EE Certificate Test2, o = Test Certificates 2011, c = BY

Кем подписан: NameOrderingCACert

Д.5.24 ValidNameChainingWhitespaceTest3EE

Основан на: базовый СОК конечного участника

Серийный номер: 11

Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY

Субъект: cn = Valid Name Chaining Whitespace EE Certificate Test3, o = Test Certificates 2011, c = BY

Кем подписан: GoodCACert

Д.5.25 ValidNameChainingWhitespaceTest4EE

Основан на: базовый СОК конечного участника

Серийный номер: 12

Эмитент: cn = \20\20\20Good CA, o = Test Certificates 2011\20\20\20, c = BY

Субъект: cn = Valid Name Chaining Whitespace EE Certificate Test4, o = Test Certificates 2011, c = BY

Кем подписан: GoodCACert

Д.5.26 ValidNameChainingCapitalizationTest5EE

Основан на: базовый СОК конечного участника

Серийный номер: 13

Эмитент: cn = GOOD CA, o = Test Certificates 2011\20\20\20, c = BY

Субъект: cn = Valid Name Chaining Capitalization EE Certificate Test5, o = Test Certificates 2011, c = BY

Кем подписан: GoodCACert

Д.5.27 RevokedsubCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 14
Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY
Субъект: cn = Revoked subCA, o = Test Certificates 2011, c = BY
Кем подписан: GoodCACert

Д.5.28 RevokedsubCACRL

Основан на: базовый СОС
Эмитент: cn = Revoked subCA, o = Test Certificates 2011, c = BY
Кем подписан: RevokedsubCACert

Д.5.29 InvalidRevokedCATest2EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = Revoked subCA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid Revoked CA Certificate Test2, o = Test Certificates 2011, c = BY
Кем подписан: RevokedsubCACert

Д.5.30 InvalidRevokedEETest3EE

Основан на: базовый СОК конечного участника
Серийный номер: 15
Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid Revoked EE Certificate Test3, o = Test Certificates 2011, c = BY
Кем подписан: GoodCACert

Д.5.31 BadCRLSignatureCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 8
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = Bad CRL Signature CA, o = Test Certificates 2011, c = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.32 BadCRLSignatureCACRL

Основан на: базовый СОС
Эмитент: cn = Bad CRL Signature CA, o = Test Certificates 2011, c = BY
Кем подписан: BadCRLSignature CACert (подпись изменена)

Д.5.33 InvalidBadCRLSignatureTest4EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = Bad CRL Signature CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid Bad CRL Signature EE Certificate Test4, o = Test Certificates 2011, c = BY
Кем подписан: BadCRLSignatureCACert

Д.5.34 BadCRLIssuerNameCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 9
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = Bad CRL Issuer Name CA, o = Test Certificates 2011, c = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.35 BadCRLIssuerNameCACRL

Основан на: базовый СОС
Эмитент: cn = Incorrect CRL Issuer Name, o = Test Certificates 2011, c = BY
Кем подписан: BadCRLIssuerNameCACert
Расположен: СОС по адресу cn = Bad CRL Issuer Name CA, o = Test Certificates 2011, c = BY

Д.5.36 InvalidBadCRLIssuerNameTest5EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = Bad CRL Issuer Name CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid Bad CRL Issuer Name EE Certificate Test5, o = Test Certificates 2011, c = BY
Кем подписан: BadCRLIssuerNameCACert

Д.5.37 LongSerialNumberCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 18
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = Long Serial Number CA, o = Test Certificates 2011, c = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.38 LongSerialNumberCACRL

Основан на: базовый СОС
Эмитент: cn = Long Serial Number CA, o = Test Certificates 2011, c = BY
Отозванные сертификаты:
serialNumber: 0x7F0102030405060708090A0B0C0D0E0F10111213
  crlEntryExtensions:
    reasonCodeExtension: не критическое
      reasons: keyCompromise
Кем подписан: LongSerialNumberCACert

Д.5.39 ValidLongSerialNumberTest16EE

Основан на: базовый СОК конечного участника
Серийный номер: 0x7F0102030405060708090A0B0C0D0E0F10111212
Эмитент: cn = Long Serial Number CA, o = Test Certificates 2011, c = BY
Субъект: cn = Valid Long Serial Number EE Certificate Test16, o = Test Certificates 2011, c = BY
Кем подписан: LongSerialNumberCACert

Д.5.40 MissingbasicConstraintsCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 22
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = Missing basicConstraints CA, o = Test Certificates 2011, c = BY
Расширение BasicConstraints: отсутствует
Кем подписан: TrustAnchorRootCertificate

Д.5.41 MissingbasicConstraintsCACRL

Основан на: базовый СОС
Эмитент: cn = Missing basicConstraints CA, o = Test Certificates 2011, c = BY
Кем подписан: MissingbasicConstraintsCACert

Д.5.42 InvalidMissingbasicConstraintsTest1EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = Missing basicConstraints CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid Missing basicConstraints EE Certificate Test1, o = Test Certificates 2011, c = BY
Кем подписан: MissingbasicConstraintsCACert

Д.5.43 basicConstraintsNotCriticalcAFalseCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 24
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = basicConstraints Not Critical cA False CA, o = Test Certificates 2011, c = BY
Расширения:
BasicConstraints: не критическое
  cA: FALSE
Кем подписан: TrustAnchorRootCertificate

Д.5.44 basicConstraintsNotCriticalcAFalseCACRL

Основан на: базовый СОС
Эмитент: cn = basicConstraints Not Critical cA False CA, o = Test Certificates 2011, c = BY
Кем подписан: basicConstraintsNotCriticalcAFalseCACert

Д.5.45 InvalidcAFalseTest3EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = basicConstraints Not Critical cA False CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid cA False EE Certificate Test3, o = Test Certificates 2011, c = BY
Кем подписан: basicConstraintsNotCriticalcAFalseCACert

Д.5.46 Запрос pathLenConstraint0CACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 26
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = pathLenConstraint0 CA, o = Test Certificates 2011, c = BY
Расширения:
BasicConstraints: критическое
  cA: TRUE
  pathLenConstraint: 0
Кем подписан: TrustAnchorRootCertificate

Д.5.47 pathLenConstraint0CACRL

Основан на: базовый СОС
Эмитент: cn = pathLenConstraint0 CA, o = Test Certificates 2011, c = BY
Кем подписан: pathLenConstraint0CACert

Д.5.48 pathLenConstraint0subCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 1
Эмитент: cn = pathLenConstraint0 CA, o = Test Certificates 2011, c = BY
Субъект: cn = pathLenConstraint0 subCA, o = Test Certificates 2011, c = BY
Кем подписан: pathLenConstraint0CACert

Д.5.49 pathLenConstraint0subCACRL

Основан на: базовый СОС
Эмитент: cn = pathLenConstraint0 subCA, o = Test Certificates 2011, c = BY
Кем подписан: pathLenConstraint0subCACert

Д.5.50 InvalidpathLenConstraintTest5EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = pathLenConstraint0 subCA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid pathLenConstraint EE Certificate Test5, o = Test Certificates 2011, c = BY
Кем подписан: pathLenConstraint0subCACert

Д.5.51 ValidpathLenConstraintTest7EE

Основан на: базовый СОК конечного участника
Серийный номер: 2
Эмитент: cn = pathLenConstraint0 CA, o = Test Certificates 2011, c = BY
Субъект: cn = Valid pathLenConstraint EE Certificate Test7, o = Test Certificates 2011, c = BY
Кем подписан: pathLenConstraint0CACert

Д.5.52 keyUsageCriticalkeyCertSignFalseCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 29
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = keyUsage Critical keyCertSign False CA, o = Test Certificates 2011, c = BY
Расширения:
KeyUsage: критическое
  keyCertSign: FALSE
Кем подписан: TrustAnchorRootCertificate

Д.5.53 keyUsageCriticalkeyCertSignFalseCACRL

Основан на: базовый СОС
Эмитент: cn = keyUsage Critical keyCertSign False CA, o = Test Certificates 2011, c = BY
Кем подписан: keyUsageCriticalkeyCertSignFalseCACert

Д.5.54 InvalidkeyUsageCriticalkeyCertSignFalseTest1EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = keyUsage Critical keyCertSign False CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid keyUsage Critical keyCertSign False EE Certificate Test1, o = Test Certificates 2011, c = BY
Кем подписан: keyUsageCriticalkeyCertSignFalseCACert

Д.5.55 keyUsageNotCriticalCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 31
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = keyUsage Not Critical CA, o = Test Certificates 2011, c = BY
Расширения:
KeyUsage: не критическое
  keyCertSign: TRUE
Кем подписан: TrustAnchorRootCertificate

Д.5.56 keyUsageNotCriticalCACRL

Основан на: базовый СОС
Эмитент: cn = keyUsage Not Critical CA, o = Test Certificates 2011, c = BY
Кем подписан: keyUsageNotCriticalCACert

Д.5.57 ValidkeyUsageNotCriticalTest3EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = keyUsage Not Critical CA, o = Test Certificates 2011, c = BY
Субъект: cn = Valid keyUsage Not Critical EE Certificate Test3, o = Test Certificates 2011, c = BY

Кем подписан: keyUsageNotCriticalCACert

Д.5.58 keyUsageCriticalcRLSignFalseCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 32
Эмитент: `cn` = Trust Anchor, `o` = Test Certificates 2011, `c` = BY
Субъект: `cn` = keyUsage Critical cRLSign False CA, `o` = Test Certificates 2011, `c` = BY
Расширения:
`KeyUsage`: критическое
  `cRLSign`: `FALSE`
Кем подписан: TrustAnchorRootCertificate

Д.5.59 keyUsageCriticalcRLSignFalseCACRL

Основан на: базовый СОС
Эмитент: `cn` = keyUsage Critical cRLSign False CA, `o` = Test Certificates 2011, `c` = BY
Кем подписан: keyUsageCriticalcRLSignFalseCACert

Д.5.60 InvalidkeyUsageCriticalcRLSignFalseTest4EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: `cn` = keyUsage Critical cRLSign False CA, `o` = Test Certificates 2011, `c` = BY
Субъект: `cn` = Invalid keyUsage Critical cRLSign False EE Certificate Test4, `o` = Test Certificates 2011, `c` = BY
Кем подписан: keyUsageCriticalcRLSignFalseCACert

Д.5.61 GoodsubCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 17
Эмитент: `cn` = Good CA, `o` = Test Certificates 2011, `c` = BY
Субъект: `cn` = Good subCA, `o` = Test Certificates 2011, `c` = BY
Расширения:
`PolicyConstraints`: не критическое
  `requireExplicitPolicy`: 0
Кем подписан: GoodCACert

Д.5.62 GoodsubCACRL

Основан на: базовый СОС
Эмитент: `cn` = Good subCA, `o` = Test Certificates 2011, `c` = BY
Кем подписан: GoodsubCACert

Д.5.63 DifferentPoliciesTest4EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: `cn` = Good subCA, `o` = Test Certificates 2011, `c` = BY
Субъект: `cn` = Different Policies EE Certificate Test4, `o` = Test Certificates 2011, `c` = BY
Расширения:
`CertificatePolicies`: не критическое

"2.16.840.1.101.3.2.1.48.2"
Кем подписан: GoodsubCACert

Д.5.64 PoliciesP2subCA2Cert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 18
Эмитент: cn = Good CA, o = Test Certificates 2011, c = BY
Субъект: cn = Policies P2 subCA2, o = Test Certificates 2011, c = BY
Расширения:
CertificatePolicies: не критическое
"2.16.840.1.101.3.2.1.48.2"
PolicyConstraints: не критическое
requireExplicitPolicy: 0
Кем подписан: GoodCACert

Д.5.65 PoliciesP2subCA2CRL

Основан на: базовый СОС
Эмитент: cn = Policies P2 subCA2, o = Test Certificates 2011, c = BY
Кем подписан: PoliciesP2subCA2Cert

Д.5.66 DifferentPoliciesTest5EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = Policies P2 subCA2, o = Test Certificates 2011, c = BY
Субъект: cn = Different Policies EE Certificate Test5, o = Test Certificates 2011, c = BY
Кем подписан: PoliciesP2subCA2Cert

Д.5.67 PoliciesP3CACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 39
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: cn = Policies P3 CA, o = Test Certificates 2011, c = BY
Расширения:
CertificatePolicies: не критическое
"2.16.840.1.101.3.2.1.48.3"
PolicyConstraints: не критическое
requireExplicitPolicy: 0
Кем подписан: TrustAnchorRootCertificate

Д.5.68 PoliciesP3CACRL

Основан на: базовый СОС
Эмитент: cn = Policies P3 CA, o = Test Certificates 2011, c = BY
Кем подписан: PoliciesP3CACert

Д.5.69 DifferentPoliciesTest12EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = Policies P3 CA, o = Test Certificates 2011, c = BY
Субъект: cn = Different Policies EE Certificate Test12, o = Test Certificates 2011, c = BY
Расширения:
CertificatePolicies: не критическое
"2.16.840.1.101.3.2.1.48.4"
Кем подписан: PoliciesP3CACert

Д.5.70 distributionPoint1CACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 74
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.71 distributionPoint1CACRL

Основан на: базовый СОС
Эмитент: ou = distributionPoin1 CA, o = Test Certificates 2011, c = BY
Расширения:
IssuingDistributionPoint: критическое
distributionPoint:
fullName:
directoryName: cn = CRL1 of distributionPoint1 CA, ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY
Отозванные сертификаты: serialNumber: 2
Кем подписан: distributionPoint1CACert
Раположен: СОС по адресу cn = CRL1 of distributionPoint1 CA, ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY; СОС по адресу cn = CRLx of distributionPoint1 CA, ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY

Д.5.72 ValiddistributionPointTest1EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY
Субъект: cn = Valid distributionPoint EE Certificate Test1, o = Test Certificates 2011, c = BY
Расширения:
CRLDistributionPoints: не критическое
distributionPoint:
fullName:
directoryName: cn = CRL1 of distributionPoint1 CA, ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY
Кем подписан: distributionPoint1CACert

Д.5.73 InvaliddistributionPointTest2EE

Основан на: базовый СОК конечного участника
Серийный номер: 2
Эмитент: ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid distributionPoint EE Certificate Test2, o = Test Certificates 2011, c = BY
Расширения:
CRLDistributionPoints: не критическое
distributionPoint:
fullName:
directoryName: cn = CRL1 of distributionPoint1 CA, ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY
Кем подписан: distributionPoint1CACert

Д.5.74 InvaliddistributionPointTest3EE

Основан на: базовый СОК конечного участника
Серийный номер: 3
Эмитент: ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid distributionPoint EE Certificate Test3, o = Test Certificates 2011, c = BY
Расширения:
CRLDistributionPoints: не критическое
distributionPoint:
fullName:
directoryName: cn = CRLx of distributionPoint1 CA, ou = distributionPoint1 CA, o = Test Certificates 2011, c = BY
Кем подписан: distributionPoint1CACert

Д.5.75 NoissuingDistributionPointCACert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 76
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: ou = No issuingDistributionPoint CA, o = Test Certificates 2011, c = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.76 NoissuingDistributionPointCACRL

Основан на: базовый СОС
Эмитент: ou = No issuingDistributionPoint CA, o = Test Certificates 2011, c = BY
Кем подписан: NoissuingDistributionPointCACert
Раположен: СОС по адресу cn = CRL, ou = No IssuingDistributionPoint CA, o = Test Certificates 2011, c = BY

Д.5.77 ValidNoissuingDistributionPointTest10EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: ou = No issuingDistributionPoint CA, o = Test Certificates 2011, c = BY

Субъект: cn = Valid No issuingDistributionPoint EE Certificate Test10, o = Test Certificates 2011, c = BY

Расширения:

CRLDistributionPoints: не критическое

distributionPoint:

fullName:

directoryName: cn = CRL, ou = No issuingDistributionPoint CA, o = Test Certificates 2011, c = BY

Кем подписан: NoissuingDistributionPointCACert

Д.5.78 onlyContainsUserCertsCACert

Основан на: базовый СОК подчиненного УЦ

Серийный номер: 77

Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY

Субъект: cn = onlyContainsUserCerts CA, o = Test Certificates 2011, c = BY

Кем подписан: TrustAnchorRootCertificate

Д.5.79 onlyContainsUserCertsCACRL

Основан на: базовый СОС

Эмитент: cn = onlyContainsUserCerts CA, o = Test Certificates 2011, c = BY

Расширения:

IssuingDistributionPoint: критическое

onlyContainsUserCerts: TRUE

Кем подписан: onlyContainsUserCertsCACert

Д.5.80 InvalidonlyContainsUserCertsTest11EE

Основан на: базовый СОК подчиненного УЦ

Серийный номер: 1

Эмитент: cn = onlyContainsUserCerts CA, o = Test Certificates 2011, c = BY

Субъект: cn = Invalid onlyContainsUserCerts EE Certificate Test11, o = Test Certificates 2011, c = BY

Расширения:

KeyUsage: критическое

digitalSignature: TRUE

nonRepudiation: TRUE

keyEncipherment: TRUE

Кем подписан: onlyContainsUserCertsCACert

Д.5.81 onlyContainsAttributeCertsCACert

Основан на: базовый СОК подчиненного УЦ

Серийный номер: 79

Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY

Субъект: cn = onlyContainsAttributeCerts CA, o = Test Certificates 2011, c = BY

Кем подписан: TrustAnchorRootCertificate

Д.5.82 onlyContainsAttributeCertsCACRL

Основан на: базовый СОС
Эмитент: cn = onlyContainsAttributeCerts CA, o = Test Certificates 2011, c = BY
Расширения:
IssuingDistributionPoint: критическое
  onlyContainsAttributeCerts: TRUE
Кем подписан: onlyContainsAttributeCertsCACert

Д.5.83 InvalidonlyContainsAttributeCertsTest14EE

Основан на: базовый СОК конечного участника
Серийный номер: 1
Эмитент: cn = onlyContainsAttributeCerts CA, o = Test Certificates 2011, c = BY
Субъект: cn = Invalid onlyContainsAttributeCerts EE Certificate Test14, o = Test Certificates 2011, c = BY
Кем подписан: onlyContainsAttributeCertsCACert

Д.5.84 onlySomeReasonsCA3Cert

Основан на: базовый СОК подчиненного УЦ
Серийный номер: 82
Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY
Субъект: ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY
Кем подписан: TrustAnchorRootCertificate

Д.5.85 onlySomeReasonsCA3compromiseCRL

Основан на: базовый СОС
Эмитент: ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY
Расширения:
IssuingDistributionPoint: критическое
  distributionPoint:
    fullName:
      directoryName: cn = CRL, ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY
  onlySomeReasons:
    keyCompromise
    cACompromise
Кем подписан: onlySomeReasonsCA3Cert
Раположен: СОС по адресу cn = CRL, ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY

Д.5.86 onlySomeReasonsCA3otherreasonsCRL

Основан на: базовый СОС
Эмитент: ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY
Расширения:
IssuingDistributionPoint: критическое
  distributionPoint:
    fullName:

directoryName: cn = CRL, ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY
onlySomeReasons:
unused
affiliationChanged
superseded
cessationOfOperation
certificateHold
privilegeWithdrawn
aACompromise

Кем подписан: onlySomeReasonsCA3Cert

Раположен: СОС по адресу cn = CRL, ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY

Д.5.87 ValidonlySomeReasonsTest15EE

Основан на: базовый СОК конечного участника

Серийный номер: 1

Эмитент: ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY

Субъект: cn = Valid onlySomeReasons EE Certificate Test15, o = Test Certificates 2011, c = BY

Расширения:

CRLDistributionPoints: не критическое
distributionPoint:
fullName:
directoryName: cn = CRL, ou = onlySomeReasons CA3, o = Test Certificates 2011, c = BY

Кем подписан: onlySomeReasonsCA3Cert

Д.5.88 ValidUnknownNotCriticalCertificateExtensionTest1EE

Основан на: базовый СОК конечного участника

Серийный номер: 94

Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY

Субъект: cn = Valid Unknown Not Critical Certificate Extension EE Cert Test1, o = Test Certificates 2011, c = BY

Расширения:

PrivateExtension: не критическое
privateNumber: 0

Кем подписан: TrustAnchorRootCertificate

Д.5.89 InvalidUnknownCriticalCertificateExtensionTest2EE

Основан на: базовый СОК конечного участника

Серийный номер: 95

Эмитент: cn = Trust Anchor, o = Test Certificates 2011, c = BY

Субъект: cn = Invalid Unknown Critical Certificate Extension EE Cert Test2, o = Test Certificates 2011, c = BY

Расширения:

`PrivateExtension`: критическое
  `privateNumber`: 0
Кем подписан: TrustAnchorRootCertificate