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 34.101.17-2012

МИ.10117.10.01

Листов 26

Минск 2022

Предисловие

Настоящая методика испытаний предназначена для использования в испытательных лабораториях при проведении сертификационных испытаний средств криптографической защиты информации на соответствие требованиям СТБ 34.101.17-2012 «Информационные технологии и безопасность. Синтаксис запроса на получение сертификата».

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 34.101.17-2012 «Информационные технологии и безопасность. Синтаксис запроса на получение сертификата».

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

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

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

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

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

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

2 Термины, обозначения и сокращения

В настоящем документе применяются термины и обозначения СТБ 34.101.17, а также следующие сокращения:

АСН.1 абстрактно-синтаксическая нотация версии 1;

ЕСПД единая система программной документации;

СКЗИ средство криптографической защиты информации;

ТНПА технический нормативный правовой акт;

ЭЦП электронная цифровая подпись.

3 Объект и цель испытаний

На испытания представляется средство криптографической защиты информации (СКЗИ), реализующее управление запросами на получение сертификата согласно СТБ 34.101.17, и документация на СКЗИ.

Целью испытаний является проверка соответствия процедур формирования и разбора запросов на получение сертификата, реализованных в объекте испытаний, требованиям СТБ 34.101.17.

4 Требования к объекту испытаний

Программа объекта испытаний может реализовывать следующие процедуры, определенные в СТБ 34.101.17:

– формирование запроса на получение сертификата;

– разбор (обработка) запроса на получение сертификата.

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 34.101.17, поддерживаемые объектом испытаний;

– программа, реализующая процедуры СТБ 34.101.17, 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 34.101.17. Если программа не поддерживает некоторые процедуры, определенные в СТБ 34.101.17, 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 34.101.17, которые в составе других программ имеют действующие сертификаты соответствия требованиям СТБ 34.101.17, то проверки по тестированию и анализу исходных текстов для

данных реализаций могут не проводиться. При этом для подтверждения соответствия объекта испытаний требованиям СТБ 34.101.17 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 34.101.17 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 34.101.17, предварительно должна пройти успешные испытания по согласованной с Органом по сертификации методике испытаний.

Испытываемая программа может не поддерживать необязательный функционал, определенный в СТБ 34.101.17 (например, формирование атрибутов запроса). При этом сужение программой обязательного функционала, определенного в СТБ 34.101.17, 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 34.101.17, включая:

- формирование запроса на получение сертификата (см. п. 6.2.1);
- разбор запроса на получение сертификата (см. п. 6.2.2).

Тестирование процедуры формирования запроса выполняется путем формирования программой запросов с последующим визуальным сравнением текстового представления форматов сформированных запросов с форматами, определенными в СТБ 34.101.17.

Тестирование процедуры разбора запроса проводится путем выполнения программой разбора запросов с последующим сравнением полученных результатов с ожидаемыми.

В тестах используются запросы на получение сертификата, 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 34.101.17. 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 запроса на получения сертификата (см. п. 6.3.3);

– корректности процедуры разбора запроса на получения сертификата (см. п. 6.3.4);

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

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

При анализе исходных текстов реализации процедуры формирования запроса на получение сертификата выполняются проверки из п. 6.3.1 – 6.3.3, 6.3.5, 6.3.6, а для реализации процедуры разбора запроса на получение сертификата — проверки из п. 6.3.1, 6.3.4 – 6.3.6. 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 34.101.17 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 34.101.17 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 34.101.17 процедуры;

– описание функций, реализующих определенные в СТБ 34.101.17 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 34.101.17 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 запроса

Входными данными, задаваемыми при тестировании процедуры формирования запроса на получение сертификата, являются допустимые значения для параметров вызова программы, определяющих состав и содержимое компонент запроса.

В тестах для хранения запросов на получение сертификата используются бинарные файлы, содержащие закодированных значений типов АСН.1, составляющих запрос.

При тестировании процедуры формирования запроса выполняются следующие тесты.

Тест IssueReqTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования запроса на получение сертификата.
2 Средствами испытываемой программы сформировать запрос на получение сертификата и экспортировать его на жесткий диск персонального компьютера в виде файла, содержащего закодированные значения типов АСН.1, составляющих запрос.
3 Преобразовать файл, полученный на шаге 2, в текстовое представление запроса.
4 Провести визуальный анализ текстового представления сформированного запроса на соответствие п. 5 СТБ 34.101.17.
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 запроса на получение сертификата не выявлено несоответствий СТБ 34.101.17 (т.е. запрос является значением типа `CertificationRequest`, п. 5.2 СТБ 34.101.17), при этом:
1) формат и содержание значения компонента `certificationRequestInfo` соответствует типу `CertificationRequestInfo` (п. 5.1 СТБ 34.101.17):

- компонент `version` содержит значение 0 типа `INTEGER`;
- компонент `subject` содержит значение типа `Name` (п. 6.1.2.4 СТБ 34.101.19) и определяет имя стороны, которая запрашивает сертификат;
- компонент `subjectPKInfo` содержит значение типа `SEQUENCE`, включающее два компонента, которые определяют открытый ключ (в виде значения типа `BIT STRING`) и алгоритм, с которым данный ключ используется (в виде значения типа `AlgorithmIdentifier`);
- компонент `attributes` (п. 5.1 СТБ 34.101.17) содержит значение типа `SET OF Attribute`, которое может включать атрибуты запроса (при наличии атрибутов проверка соответствия форматов и содержания значений атрибутов выполняется в тесте `AttCertReqTest`);

2) формат и содержание значения компонента `signatureAlgorithm` соответствует типу `AlgorithmIdentifier` (п. 6.1.1.2 СТБ 34.101.19) и определяет идентификатор алгоритма ЭЦП (включая его параметры, при необходимости), в соответствии с которым подписывалась информационная часть запроса на получение сертификата;
3) формат и содержание значения компонента `signature` соответствует типу `BIT STRING` и содержит значение ЭЦП информационной части запроса (закодированного значения компонента `certificationRequestInfo`), выработанной на личном ключе стороны, запрашивающей сертификат.
7 Возвратить `ОШИБКА`.

Примечание — Задаваемые в тесте IssueReqTest параметры вызова испытываемой программы в совокупности должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по формированию запроса на выпуск сертификата в части формирования основных компонент запроса.

Тест AttReqTest

1 Задать параметры вызова испытываемой программы, которые в соответствии с документацией необходимы для формирования запроса на получение сертификата.
2 Средствами испытываемой программы сформировать запрос на получение сертификата 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 запроса.
4 Провести визуальный анализ текстового представления сформированного запроса на соответствие п. 5.1 СТБ 34.101.17.
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 запроса на получение сертификата не выявлено несоответствий СТБ 34.101.17 (т.е. запрос является значением типа `CertificationRequest`, см. п. 5.2 СТБ 34.101.17), при этом:
1) компонент `attributes` (см. п. 5.1 СТБ 34.101.17) содержит значение типа `SET OF Attribute`, которое может включать атрибуты запроса, каждый из которых содержит компонент `type`, имеющий тип `OBJECT IDENTIFIER` и определяющий идентификатор атрибута, и компонент `values`, содержащий значение атрибута;
2) для каждого атрибута, содержащегося в компоненте `attributes`, его значение соответствует значению, определенному в документе на атрибут с заданным значением идентификатора.
7 Возвратить `ОШИБКА`.

Примечание 1 —Задаваемые в тесте AttReqTest параметры вызова испытываемой программы в совокупности должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по выпуску запросов на получение сертификатов в части формирования атрибутов.

Примечание 2 —Корректность формирования атрибутов должна проверяться по ТНПА или другим документам, определяющим данные атрибуты. Например, атрибут «время подписания» определяется в п. 15.4 СТБ 34.101.23. В документации на испытываемую программу должен быть определен перечень атрибутов, поддерживаемых программой, и должны быть указаны ссылки на документы, в которых определяются поддерживаемые атрибуты.

6.2.2 Процедура разбора запроса

Входными данными, задаваемыми при тестировании процедуры разбора запроса на получение сертификата, являются запросы на получение сертификата в виде бинарных файлов, содержащих закодированные значения типов АСН.1, составляющих запрос.

При тестировании процедуры разбора запроса на получение сертификата выполняются базовые тесты и тесты известного ответа. Базовые тесты являются обязательными. Тесты известного ответа являются дополнительными и предназначены для разбора запросов на получение сертификата открытого ключа алгоритмов подписи и транспорта ключа, определенных в СТБ 34.101.45.

Для базовых тестов запросы на получение сертификата предоставляются совместно с испытываемой программой или формируются экспертом (при наличии необходимого программного обеспечения). Перед тестированием процедуры разбора запроса на получение сертификата эксперт проводит визуальное сравнение текстового представления запроса с

форматом, определенном в СТБ 34.101.17. Для изменения запросов, которое выполняется в некоторых базовых тестах, могут использоваться программы, описанные в приложении Г.2.

Для тестов известного ответа в качестве входных данных используются запросы, определенные в приложении Д (закодированные ASN.1-файлы с данными запросами являются неотъемлемой частью настоящей методики).

Базовые тесты. При тестировании процедуры разбора запроса на получение сертификата выполняются следующие базовые тесты.

Тест ValidReqTest

1 Задать корректный запрос на получение сертификата.
2 Средствами испытываемой программы выполнить разбор запроса на получение сертификата.
3 Повторить шаги 1 — 2 не менее 9 раз, задавая различные корректные запросы на получение сертификата.
4 Возвратить `УСПЕХ`, если на шаге 2 запрос успешно проверен и признан корректным.
5 Возвратить `ОШИБКА`.

Примечание — Задаваемые в тесте ValidReqTest запросы должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по разбору запросов на получение сертификата.

Тест InvalidReqTest

1 Изменить запрос на получение сертификата таким образом, чтобы запрос стал некорректным.
2 Средствами испытываемой программы выполнить разбор измененного запроса.
3 Повторить шаги 1 — 2 не менее 9 раз, задавая различные измененные запросы.
4 Возвратить `УСПЕХ`, если на шаге 2 запрос отклонен и возвращена ошибка, соответствующая внесенным изменениям.
5 Возвратить `ОШИБКА`.

Примечание 1 — Изменение запроса может состоять, например, в изменении значения компонента `signature`.

Примечание 2 — Задаваемые в тесте InvalidReqTest запросы должны покрывать наиболее полный функционал, предоставляемый испытываемой программой по разбору запросов на получение сертификата.

Тесты известного ответа. При тестировании процедуры разбора запроса на получение сертификата выполняются следующие тесты известного ответа.

Тест VReqTest1

1 Задать в качестве запроса на выпуск сертификата файл «VReqTest1.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос успешно проверен и признан корректным.
4 Возвратить `ОШИБКА`.

Примечание 1 — Цель теста VReqTest1 — проверка способности реализации обрабатывать корректные подписи и другие значения компонент в запросе, включая атрибут `extensionRequest` (см. RFC 2985) с расширением `keyUsage` (см. СТБ 34.101.19).
Примечание 2 — Испытуемая программа может не распознавать атрибуты. При этом она может выдать сообщение о невозможности обработки атрибутов.

Тест ISignTest2

1 Задать в качестве запроса на выпуск сертификата файл «ISignTest2.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос отклонен.
4 Возвратить `ОШИБКА`.

Примечание — Запрос содержит некорректное значение подписи. Цель теста ISignTest2 — проверка способности реализации выявлять некорректные подписи в запросах.

Тест IVerTest3

1 Задать в качестве запроса на выпуск сертификата файл «IVerTest3.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос отклонен.
4 Возвратить `ОШИБКА`.

Примечание — Запрос содержит некорректное значение версии. Цель теста IVerTest3 — проверка способности реализации выявлять некорректный номер версии в запросах.

Тест IPubKeyIdTest4

1 Задать в качестве запроса на выпуск сертификата файл «IPubKeyIdTest4.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос отклонен.
4 Возвратить `ОШИБКА`.

Примечание — Запрос содержит некорректное значение идентификатора открытого ключа. Цель теста IPubKeyIdTest4 — проверка способности реализации выявлять некорректный идентификатор открытого ключа.

Тест IPubKeyPrmTest5

1 Задать в качестве запроса на выпуск сертификата файл «IPubKeyPrmTest5.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос отклонен.
4 Возвратить `ОШИБКА`.

Примечание — Запрос содержит некорректное значение идентификатора для параметров, с которыми используется открытый ключ. Цель теста IPubKeyPrmTest5 — проверка способности реализации выявлять некорректное значение параметров открытого ключа.

Тест IPubKeyLenTest6

1 Задать в качестве запроса на выпуск сертификата файл «IPubKeyLenTest6.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос отклонен.
4 Возвратить `ОШИБКА`.

Примечание — Запрос содержит открытый ключ некорректной длины. Цель теста IPubKeyLenTest6 — проверка способности реализации выявлять несоответствие длины открытого ключа и уровня стойкости, который определяется параметрами открытого ключа.

Тест ISignAlgIdTest7

1 Задать в качестве запроса на выпуск сертификата файл «ISignAlgIdTest7.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос отклонен.
4 Возвратить `ОШИБКА`.

Примечание — Запрос содержит некорректное значение идентификатора алгоритма подписи. Цель теста ISignAlgIdTest7 — проверка способности реализации выявлять некорректное значение идентификатора криптографического алгоритма, используемого для подписи запроса.

Тест ISignAlgPrmTest8

1 Задать в качестве запроса на выпуск сертификата файл «ISignAlgPrmTest8.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос отклонен.
4 Возвратить `ОШИБКА`.

Примечание — Запрос содержит значение идентификатора для параметров алгоритма подписи (согласно п. Д.2 СТБ 34.101.45 компонент, определяющий параметры алгоритма подписи запроса, должен принимать значение NULL). Цель теста ISignAlgPrmTest8 — проверка способности реализации выявлять несоответствие значения параметра стандарту на криптографический алгоритм.

Тест VExtReqAttrTest9

1 Задать в качестве запроса на выпуск сертификата файл «VExtReqAttrTest9.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос успешно проверен и признан корректным.
4 Возвратить `ОШИБКА`.

Примечание 1 — Запрос содержит атрибут `extensionRequest` (см. RFC 2985) с расширениями `keyUsage` и `extKeyUsage` (см. СТБ 34.101.19). Цель теста VExtReqAttrTest9 — проверка способности реализации обрабатывать атрибут `extensionRequest`, содержащий несколько расширений, которые предназначены для включения в сертификат.
Примечание 2 — Испытуемая программа может не распознавать атрибуты. При этом она может выдать сообщение о невозможности обработки атрибутов.

Тест VUnknowAttrTest10

1 Задать в качестве запроса на выпуск сертификата файл «VUnknowAttrTest10.req».
2 Средствами испытываемой программы выполнить процедуру разбора запроса.
3 Возвратить `УСПЕХ`, если запрос успешно проверен и признан корректным.
4 Возвратить `ОШИБКА`.

Примечание 1 — Запрос дополнительно содержит неизвестный атрибут. Цель теста VUnknowAttrTest10 — проверка способности реализации обрабатывать неизвестные атрибуты.

Примечание 2 — Так как испытуемая программа не может распознать атрибут она может выдать сообщение о невозможности обработки атрибутов.

6.3 Анализ исходных текстов

6.3.1 Корректность использования криптографических алгоритмов

В программе для доказательства владения личным ключом и проверки подлинности запроса используются алгоритмы хэширования и алгоритмы электронной цифровой подписи.

Использование функций, реализующих криптографический алгоритм, должно выполняться в соответствии с СТБ 34.101.17, 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 процедуре формирования запроса на выпуск сертификата 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 запроса

При анализе корректности реализации процедуры формирования запроса на выпуск сертификата исходные тексты программы оцениваются частично, по выбору эксперта. Корректность реализации процедуры означает, что реализация функционально соответствует СТБ 34.101.17 и что реализация не содержит ошибок и уязвимостей.

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

1 Реализация должна подписывать запрос на личном ключе стороны, запрашивающей сертификат.

2 Реализация должна включать в запрос идентификатор криптографического алгоритма, используемого для подписи запроса.

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

6.3.4 Корректность процедуры разбора запроса

При анализе корректности реализации процедуры разбора запроса на получение сертификата исходные тексты программы оцениваются частично, по выбору эксперта. Корректность реализации процедуры означает, что реализация функционально соответствует СТБ 34.101.17 и что реализация не содержит ошибок и уязвимостей.

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

1 Реализация должна проверять ЭЦП запроса на открытом ключе, содержащемся в запросе.

2 Реализация должна проверять ЭЦП запроса алгоритмом с идентификатором, включенным в запрос.

3 Реализация должна проверять ЭЦП запроса алгоритмом с долговременными параметрами, информация о которых включена в запрос.

6.3.5 Корректность обработки исключительных ситуаций

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

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

Для анализа корректности обработки исключительных ситуаций эксперт проверяет, что:

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

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

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

6.3.6 Отсутствие недокументированных возможностей

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

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

Приложение А
Форма протокола анализа документации

Экз. {Поле 1}

Протокол № {Поле 2} **от** {Поле 3}
результатов анализа документации
программы {Поле 4}, реализующей управление запросами на получение сертификата согласно СТБ 34.101.17-2012

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 запросами на получение сертификата согласно СТБ 34.101.17-2012.

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

В поле 1 указывается номер экземпляра протокола.

В поле 2 указывается номер, однозначно идентифицирующий протокол.

В поле 3 указывается дата составления протокола.

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

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

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

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

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

В поле 13 указывается результат выполнения проверки: «положительно» — результат проверки положительный, «отрицательно» — результат проверки отрицательный. После завершения анализа документации и заполнения таблицы делается вывод о соответствии (не соответствии) документации программе объекта испытаний в части управления запросами на получение сертификата согласно СТБ 34.101.17. 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 запросами на получение сертификата согласно СТБ 34.101.17-2012

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	IssueReqTest	{Поле 8}
2	AttReqTest	{Поле 8}
3	ValidReqTest	{Поле 8}
4	InvalidReqTest	{Поле 8}
...	...	...

3. Заключение по результатам тестирования: программа {Поле 4} соответствует (не соответствует) требованиям, установленным в СТБ 34.101.17-2012.

Эксперт, {Поле 10} {Поле 11}
{Поле 9}

В поле 1 указывается номер экземпляра протокола.

В поле 2 указывается номер, однозначно идентифицирующий протокол.

В поле 3 указывается дата составления протокола.

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

В поле 5 указываются имена исходных файлов программ объекта испытаний.

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

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

После завершения тестирования и заполнения таблицы делается вывод о соответствии (не соответствии) программной реализации объекта испытаний СТБ 34.101.17. 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 запросами на получение сертификата согласно СТБ 34.101.17-2012

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	Корректности использования криптографических алгоритмов	{Поле 8}
2	Корректности управления секретными данными	{Поле 8}
3	Корректности процедуры формирования запроса на получения сертификата	{Поле 8}
4	Корректности процедуры разбора запроса на получения сертификата	{Поле 8}
5	Корректности обработки исключительных ситуаций	{Поле 8}
6	Отсутствия недокументированных возможностей	{Поле 8}

3. Заключение по результатам анализа исходных текстов программ: программа {Поле 4} соответствует требованиям, установленным в СТБ 34.101.17–2012.

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 34.101.17. 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 запрос

Компонент запроса или тип ASN.1	Значение компонента	Пояснения
`CertificationRequest`		
`certificationRequestInfo`		Значение данного компонента подписывается ЭЦП
`version`	0	Версия 0 запроса
`subject`		
`Name`		Используется тип СТБ 34.101.19
`c`	BY	Поле с идентификатором `countryName`
`o`	Test Requests 2016	Поле с идентификатором `organizationName`
`cn`	{Значение, соответствующее названию теста}	Поле с идентификатором `commonName`
`subjectPKInfo`		
`algorithm`		
`AlgorithmIdentifier`		
`algorithm`	`bign-pubkey`	Открытый ключ СТБ 34.101.45
`parameters`	`bign-curve256v1`	Стандартные параметры для уровня стойкости $l = 128$
`subjectPublicKey`		
`BIT STRING`	Значение открытого ключа	Длина 512 бит
`attributes`		
`extensionRequest`	Атрибут с идентификатором 1.2.840.113549.1.9.14	Содержит в закодированном виде расширения для включения в сертификат. Определяется в RFC 2985.
`keyUsage`	Расширение с идентификатором 2.5.29.15, определяющее использование ключа для выработки и проверки подписи, а также для шифрования ключа	Расширение `KeyUsage` согласно СТБ 34.101.19
`signatureAlgorithm`		
`AlgorithmIdentifier`		
`algorithm`	`bign-with-hbelt`	Алгоритмы подписи СТБ 34.101.45 с функцией хэширования СТБ 34.101.31
`parameters`	`NULL`	Функция хэширования определяется компонентом `AlgorithmIdentifier.algorithm`
`signature`	`BIT STRING`	Подпись от компонента `certificationRequestInfo`

Д.2 Тестовые запросы

Д.2.1 Запрос VReqTest1

Основан на: базовый запрос.
Общее имя (`subject.cn`): Valid Request Test1.

Д.2.2 Запрос ISignTest2

Основан на: базовый запрос.
Общее имя (`subject.cn`): Invalid Signature Test2.
Подпись (`signature`): некорректная.

Д.2.3 Запрос IVerTest3

Основан на: базовый запрос.
Версия (`version`): 1.
Общее имя (`subject.cn`): Invalid Version Test3.

Д.2.4 Запрос IPubKeyIdTest4

Основан на: базовый запрос.
Общее имя (`subject.cn`): Invalid Public Key Identifier Test4.
Идентификатор открытого ключа (`subjectPKInfo.algorithm.algorithm`): 1.2.112.0.2.0.34.101.45.3.1.

Д.2.5 Запрос IPubKeyPrmTest5

Основан на: базовый запрос.
Общее имя (`subject.cn`): Invalid Public Key Parameters Test5.
Параметры открытого ключа (`subjectPKInfo.algorithm.parameters`): 1.2.112.0.2.0.34.101.45.2.1.

Д.2.6 Запрос IPubKeyLenTest6

Основан на: базовый запрос.
Общее имя (`subject.cn`): Invalid Public Key Length Test6.
Открытый ключ (`subjectPKInfo.subjectPublicKey`): ключ длины 768 битов.

Д.2.7 Запрос ISignAlgIdTest7

Основан на: базовый запрос.
Общее имя (`subject.cn`): Invalid Signature Algorithm Identifier Test7.
Идентификатор алгоритма подписи (`signatureAlgorithm.algorithm`): 1.2.112.0.2.0.34.101.31.81.

Д.2.8 Запрос ISignAlgPrmTest8

Основан на: базовый запрос.
Общее имя (`subject.cn`): Invalid Signature Algorithm Parameters Test8.
Параметры алгоритма подписи (`signatureAlgorithm.parameters`): 1.2.112.0.2.0.34.101.45.3.1.

Д.2.9 Запрос VExtReqAttrTest9

Основан на: базовый запрос.
Общее имя (`subject.cn`): Valid Extension Request Attribute Test9.
Атрибуты (`attributes`): атрибут `extensionRequest` (см. RFC 2985) содержит дополнительно расширение сертификата `extKeyUsage` (см. СТБ 34.101.19), определяющее использование ключа для TLS-аутентификации интернет-клиента.

Д.2.10 Запрос VUnknowAttrTest10

Основан на: базовый запрос.
Общее имя (`subject.cn`): Valid Unknow Attribute Test10.
Атрибуты (`attributes`): дополнительный атрибут с идентификатором 2.16.840.1.101.2.1.12.2, содержащий значение 0 типа `INTEGER`.