ТЕХНИЧЕСКИЙ КОМИТЕТ ПО СТАНДАРТИЗАЦИИ
«КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ»

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
ТК 26

МР
22

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Использование российских криптографических алгоритмов в протоколе получения актуальных статусов сертификатов (OCSP)

Москва
2022

Предисловие

1 РАЗРАБОТАНЫ Обществом с ограниченной ответственностью «КРИПТО-ПРО» (ООО «КРИПТО-ПРО»)

2 ВНЕСЕНЫ Техническим комитетом по стандартизации ТК 26 «Криптографическая защита информации»

Содержание

Введение

В настоящих рекомендациях описывается протокол получения актуального статуса сертификата ключа проверки электронной подписи (протокол OCSP).

Протокол OCSP позволяет приложениям определять статус указанного сертификата, а также предоставлять информацию о его аннулировании. Данный способ взаимодействия предполагает более быстрое предоставление актуальной информации, чем при использовании списка аннулированных сертификатов (CRL), а также позволяет получать дополнительную информацию об их состоянии.

П р и м е ч а н и е — Основная часть настоящих рекомендаций дополнена приложениями А–Б.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Использование российских криптографических алгоритмов в протоколе получения актуальных статусов сертификатов (OCSP)

Дата введения — 22 — —

1 Область применения

Протокол OCSP может использоваться различными службами, информационными системами и другими участниками какого-либо электронного взаимодействия для получения статуса сертификата вместо или в дополнение к проверке по списку аннулированных сертификатов (CRL).

Выдача актуального статуса сертификата при взаимодействии по протоколу OCSP осуществляется службой OCSP. Служба OCSP является доверенной третьей стороной, предоставляющей запрашивающей стороне сведения о статусе сертификата при условии соблюдения участниками взаимодействия правил и очередности действий протокола OCSP.

В случае если запрашивающей стороне не удается установить соединение со службой OCSP или данная служба недоступна по иным причинам, запрашивающая сторона может использовать CRL для получения статуса сертификата.

2 Нормативные ссылки

В настоящих рекомендациях использованы нормативные ссылки на следующие стандарты:

ГОСТ Р ИСО/МЭК 8824–93 Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии один (АСН.1)

ГОСТ Р 34.10–2012 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи

Издание официальное

ГОСТ Р 34.11–2012 Информационная технология. Криптографическая защита информации. Функция хэширования

Р 1323565.1.023–2018 «Информационная технология. Криптографическая защита информации. Использование алгоритмов ГОСТ Р 34.10–2012, ГОСТ Р 34.11–2012 в сертификате, списке аннулированных сертификатов (CRL) и запросе на сертификат PKCS #10 инфраструктуры открытых ключей X.509»

Р 1323565.1.025–2019 «Информационная технология. Криптографическая защита информации. Форматы сообщений, защищенных криптографическими методами»

Примечание — При пользовании настоящими рекомендациями целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования – на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящих рекомендаций в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящих рекомендациях применены следующие термины с соответствующими определениями.

3.1.1 **открытый ключ**: Общеизвестный ключ из ключевой пары субъекта.

3.1.2 **закрытый ключ**: Ключ из ключевой пары субъекта, известный только данному субъекту.

3.1.3 **сертификат открытого ключа** (сертификат): Документ, выданный и подписанный удостоверяющим центром и содержащий открытый ключ и информацию, идентифицирующую субъекта, владеющего соответствующим закрытым ключом.

3.1.4 **корень доверия**: сущность (субъект), являющаяся доверенной в случае проверки сертификата открытого ключа в процессе некоторого требующего защиты взаимодействия нескольких сторон.

3.1.5 **путь сертификации** (цепочка сертификатов): Упорядоченный список из одного или нескольких сертификатов открытого ключа, где первый сертификат содержит подпись корня доверия, а последний представляет собой сертификат, для проверки которого служит данный путь сертификации.

3.1.6 **доверенная третья сторона**: Организация или служба, которой доверяют все участники в процессе некоторого взаимодействия, требующего защиты.

3.1.7 **удостоверяющий центр** (УЦ): Доверенная третья сторона, которой один или несколько участников взаимодействия доверили создание сертификатов.

3.1.8 **корневой сертификат**: способ представления информации о корне доверия, представляющий собой самоподписанный сертификат удостоверяющего центра.

3.1.9 **список аннулированных сертификатов** (CRL): Подписанный электронный документ, содержащий сведения о сертификатах открытых ключей, которые выдавший их удостоверяющий центр более не признает действительными.

3.1.10 **точка распределения списка аннулированных сертификатов** (CDP): Источник данных, служащий для распространения списков аннулированных сертификатов.

3.1.11 **служба получения актуального статуса сертификата** (служба OCSP): доверенная третья сторона, предоставляющая сведения об актуальных статусах сертификатов.

3.1.12 **электронная цифровая подпись** (signature): Строка бит, полученная в результате процесса формирования подписи.

(ГОСТ Р 34.10–2012)

Примечание — В настоящих рекомендациях в целях сохранения терминологической преемственности по отношению к действующим отечественным нормативным документам и опубликованным научно-техническим изданиям установлено, что термины "электронная подпись", "цифровая подпись" и "электронная цифровая подпись" являются синонимами.

3.1.13 **хэш-функция**: Функция, отображающая строки бит в строки бит фиксированной длины и удовлетворяющая следующим свойствам:

1) по данному значению функции сложно вычислить исходные данные, отображаемые в это значение;

2) для заданных исходных данных сложно вычислить другие исходные данные, отображаемые в то же значение функции;

3) сложно вычислить какую-либо пару исходных данных, отображаемых в одно и то же значение.

(ГОСТ Р 34.11–2012)

Примечание — В настоящих рекомендациях в целях сохранения терминологической преемственности с действующими отечественными нормативными документами и опубликованными научно-техническими изданиями установлено, что термины «хэш–функция», «криптографическая хэш–функция», «функция хэширования» и «криптографическая функция хэширования» являются синонимами.

3.1.14 **хэш-код**: Строка бит, являющаяся выходным результатом хэш-функции.

(ГОСТ Р 34.11–2012)

3.1.15 **всемирное координированное время** (UTC): Временная шкала, основанная на секундах, определенная и рекомендованная Международным консультативным комитетом по радио, и поддерживаемая Международным бюро мер и весов. По-другому называется «время зулу» и обозначается буквой Z.

3.2 Обозначения

В настоящих рекомендациях используют следующие обозначения:

AIA — Authority Information Access — расширение сертификата, используемое для указания дополнительных способов проверки статуса сертификата;

CDP — Certificate (Revocation List) Distribution Point — точка распределения списков аннулированных сертификатов;

CRL — Certificate Revocation List — список аннулированных сертификатов;

DER — Distinguished Encoding Rules — правила кодирования структур;

HTTP — HyperText Transfer Protocol — протокол передачи гипертекста;

IETF — Internet Engineering Task Force – аббревиатура является сокращением названия группы инженерной поддержки Интернет, некоммерческой организации, являющейся открытым международным сообществом проектировщиков, учёных, сетевых операторов и провайдеров, созданным IAB (Internet Architecture Board) в 1986 году, которое занимается развитием протоколов и архитектуры Интернета;

OCSP — Online Certificate Status Protocol — протокол получения актуального статуса сертификата;

RFC — Request For Comments – данной аббревиатурой и числом IETF маркирует документы, содержащие технические спецификации и стандарты;

SMTP — Simple Mail Transfer Protocol — простой протокол передачи почты;

URI — Uniform Resource Identifier — универсальный идентификатор ресурсов;

URL — Uniform Resource Locator — универсальный указатель ресурсов;

UTC (Coordinated — всемирное координированное время;

Universal
Time)

УЦ — удостоверяющий центр.

4 Обзор протокола OCSP

Протокол OCSP предназначен для получения статуса сертификата вместо или в дополнение к проверке по списку аннулированных сертификатов (CRL). Для этого в рамках данного протокола осуществляется взаимодействие службы получения актуального статуса сертификата (службы OCSP) и запрашивающей стороны (С). Взаимодействие перечисленных участников заключается в формировании запроса к службе OCSP и проверке ее ответа (см. рисунок 1).

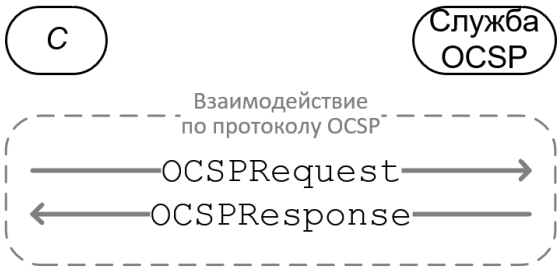

Рисунок 1 — Схема обмена сообщениями по протоколу OCSP

В первом сообщении обмена в рамках протокола OCSP запрашивающая сторона посылает в службу OCSP запрос актуального статуса сертификата (сертификатов), который содержит `OCSPRequest`, как это описано в 7.1. Во втором сообщении служба OCSP отправляет запрашивающей стороне ответ `OCSPResponse` (см. 7.2), который содержит информацию о статусе сертификата или сведения об ошибке.

Порядок формирования запроса описан в 5.1. Порядок формирования ответа службой OCSP описан в 5.2. Порядок обработки данного ответа запрашивающей стороной описан в 5.3.

Примечание — Ответы службы OCSP могут быть использованы повторно при проверке действительности сертификата. Например, если сертификат проверяется на момент подписи некоторого документа (при наличии достоверной информации о моменте подписания электронного документа).

Ответы службы OCSP подписаны закрытым ключом данной службы. Соответствующий открытый ключ содержится в сертификате службы OCSP. Требования к указанному сертификату и особенности его аннулирования описаны в разделе 6.

Форматы запроса и ответа описаны в разделе 7.

Требования к содержимому запроса к службе OCSP и ее ответа при использовании российских криптографических алгоритмов подписи и хэширования описаны в разделе 8.

Способы передачи запроса к службе OCSP и получения ответа описаны в разделе 9.

5 Порядок формирования запроса к службе OCSP и проверки ее ответа

В данном разделе описан порядок формирования запрашивающей стороной запроса актуального статуса сертификата (см. 5.1), формирования службой OCSP ответа (см. 5.2) и порядок обработки полученного ответа запрашивающей стороной (см. 5.3).

5.1 Порядок формирования запроса к службе OCSP

Идентификатор сертификата (или сертификатов, если их несколько), для которого требуется получить актуальный статус, помещается в запрос к службе OCSP. Для последующей проверки актуальности ответа службы в запрос может быть помещено большое случайное число. Дополнительно запрос к службе OCSP может быть подписан. Сформированный запрос отправляется в службу OCSP.

Удостоверяющий центр, предоставляющий возможность проверки статуса выпущенных им сертификатов по протоколу OCSP, должен включать в данные сертификаты расширение `AuthorityInfoAccess` (см. [1, подпункт 4.2.2.1]). При этом в поле `accessLocation` должно быть включено значение `uniformResourceIndicator` (URI, см. [2]), определяющее способ обращения к службе OCSP и дополнительную информацию (например, URL со схемой HTTP).

В поле `accessMethod` расширения `AuthorityInfoAccess` должен быть включен следующий объектный идентификатор:

`id-ad-ocsp`, «`1.3.6.1.5.5.7.48.1`».

Дополнительно способ обращения к службе OCSP может быть задан в настройках запрашивающей стороны.

5.2 Порядок формирования ответа службой OCSP

Служба OCSP формирует ответ, содержащий актуальный статус запрашиваемого сертификата (сертификатов) только при условии, что полученный запрос корректен (см. 7.1) и соответствует политике безопасности (регламенту) службы.

В случае успешной обработки запроса службой OCSP в ответе содержится следующее:

- время формирования ответа службой OCSP;

- сведения о службе OCSP, сформировавшей данный ответ;

- идентификатор используемого алгоритма подписи;

- подпись данных ответа;

- необязательная дополнительная информация, передаваемая в виде набора расширений (см. 7.3). При этом если в запросе для проверки актуальности ответа было указано большое случайное число, оно должно также присутствовать в ответе в соответствующем расширении (см. 7.3.1), если только служба OCSP не использует заранее сформированные ответы;

- ответ, содержащий статус для каждого из сертификатов в запросе.

Для каждого из сертификатов, который был указан в запросе к службе OCSP, в ответе содержится следующее:

- идентификатор данного сертификата;

- статус сертификата и причину аннулирования, если данный сертификат был аннулирован (см. 5.2.1);

- время, в течение которого статус сертификата можно считать актуальным (см. 5.2.2);

- необязательная дополнительная информация о данном сертификате, результатах его проверки, источниках информации и т.п., передаваемая в виде набора расширений (см. 7.3).

В случае возникновения во время обработки запроса каких-либо ошибок, ответ службы OCSP должен содержать только информацию о возникшей ошибке (см. 7.2.1).

5.2.1 Статусы сертификата в ответе службы OCSP

В ответе службы OCSP для каждого из указанных в запросе сертификатов могут содержаться следующие статусы.

- `good` (положительный ответ). Положительный ответ означает, что службе OCSP не известно о факте аннулирования или приостановки данного сертификата в пределах его срока действия. В то же время данный статус не означает, что сертификат выдавался, или что время, когда служба OCSP сформировала ответ, находится между датами начала и окончания срока действия сертификата. Расширения, присутствующие в ответе рядом с данным статусом, могут использоваться для передачи запрашивающей стороне дополнительной информации о сертификате — сведения о его выдаче, сроке действия и т.д. Некоторые возможные расширения и их идентификаторы приведены в 7.3.

- `revoked` (аннулирован). Данный статус означает, что сертификат был аннулирован или приостановлен. Данный статус также может означать, что удостоверяющий центр не выдавал сертификат, который указан в запросе. Подробная информация о случаях, в которых ответ службы OCSP содержит данный статус сертификата, приведена в 7.2.2.5.

- `unknown` (неизвестен). Данный статус означает, что служба OCSP не может определить статус сертификата. Статус указывается, если служба не имеет сведений о сертификате. Данная ситуация может возникать, в частности, когда сертификат выдан удостоверяющим центром, который данной службой OCSP не обслуживается.

Примечание — Выбор службой OCSP статуса «аннулирован» или «неизвестен» для случая, когда служба не имеет сведений о сертификате, определяется регламентом данной службы. Если в ответе указан статус «неизвестен», запрашивающей стороне предоставляется возможность самостоятельно определить статус сертификата при помощи другого источника информации (например, при помощи CRL).

Примечание — Если службе OCSP известно, что закрытый ключ определенного удостоверяющего центра был скомпрометирован, она может возвращать статус «аннулирован» для всех сертификатов, выпущенных данным УЦ.

5.2.2 Значения времени в ответе службы OCSP

Ответ службы OCSP может содержать следующие значения времени:

- `producedAt`: время, когда служба OCSP сформировала данный ответ (см. 7.2.2.2).

- `thisUpdate`: момент времени, на который было известно, что указанный статус сертификата является актуальным (см. 7.2.2.4). Данное значение указывается для каждого сертификата в ответе службы OCSP;

- `nextUpdate`: время, не позднее которого будет доступна более новая информация о статусе сертификата (см. 7.2.2.4);

- `revocationTime`: время, когда сертификат был аннулирован или приостановлен (см. 7.2.2.5).

Поля `thisUpdate` и `nextUpdate` определяют рекомендуемый интервал действительности ответа службы OCSP. Данный интервал должен интерпретироваться аналогично интервалу от `thisUpdate` до `nextUpdate` в списках аннулированных сертификатов, как это описано в [1]. Отсутствие поля `nextUpdate` означает, что новая информация о статусе сертификата может быть доступна в любой момент времени.

Служба OCSP может заранее формировать подписанные ответы, информирующие о статусе сертификата (или сертификатов, если их несколько). Время, когда было известно, что состояние является корректным, вносится в поле ответа `thisUpdate`. Время, не позже которого будет доступна новая информация, вносится в поле `nextUpdate`, а время, когда был сформирован ответ, вносится в поле `producedAt`.

5.3 Порядок проверки ответа службы OCSP

При получении ответа запрашивающая сторона должна проверить ответ на ошибки о состоянии и, если их нет, проверить различные поля в полученном ответе в соответствии с 7.2.2 настоящих рекомендаций. В частности, запрашивающая сторона должна убедиться в следующем:

- в ответе содержится информация о статусе именно того сертификата, который необходимо проверить;

- подпись в ответе действительна;

- ответ подписан именно той службой OCSP, к которой направлялся запрос. Данная проверка выполняется только при условии, что ее возможно осуществить;

- данная служба OCSP уполномочена предоставлять информацию о статусе проверяемого сертификата (см. 6.1);

- время, прошедшее с момента, когда был точно известен статус сертификата (поле `thisUpdate`, см. 5.2.2), не превышает допустимого времени, установленного регламентом запрашивающей стороны;

- время, когда был точно известен статус сертификата (поле `thisUpdate`, см. 5.2.2), не превышает локального системного времени запрашивающей стороны;

- если указано время, не позже которого информация о статусе сертификата обновится (поле `nextUpdate`, см. 5.2.2), следует убедиться, что оно еще не наступило.

Примечание — Данная проверка не должна выполняться в случаях, когда OCSP-ответ, полученный и использовавшийся ранее, используется повторно, как это описано в разделе 4.

Если любая из вышеперечисленных проверок завершилась неудачей, ответ службы OCSP следует отклонить.

Примечание — Настоящие рекомендации не устанавливают строгую последовательность проверок полученного от службы OCSP ответа. Данные проверки могут выполняться в произвольном порядке.

6 Сертификат службы OCSP

6.1 Определение полномочий службы OCSP

Ключ, которым подписывается ответ службы OCSP, может совпадать с ключом, которым подписан проверяемый сертификат. Если для подписи был использован другой ключ, запрашивающей стороне необходимо убедиться в том, что служба OCSP, подписавшая ответ, обладает соответствующими полномочиями. Указание полномочий на подпись службой OCSP ответов может быть установлено регламентом запрашивающей стороны, либо произведено путем выдачи сертификата специального вида службе OCSP.

В случае указания полномочий путем выдачи сертификата, в расширение `extendedKeyUsage` данного сертификата должно быть включен следующий идентификатор, как это описано в [1, подпункт 4.2.1.12]:

`id-kp-OCSPSigning`, «`1.3.6.1.5.5.7.3.9`».

Сертификат с указанным идентификатором должен быть выдан службе OCSP непосредственно удостоверяющим центром, выдавшим проверяемый сертификат.

Системы или приложения, проверяющие OCSP-ответы, должны иметь возможность проверить полномочия службы OCSP по значению `id-kp-OCSPSigning`,

как это описано выше. Они также могут предусматривать способы указания одной или нескольких служб OCSP и перечень УЦ, для которых каждая из служб OCSP является уполномоченной в соответствии с регламентом.

Службы или приложения принимают OCSP-ответ, если для сертификата, используемого при подписи данного ответа, выполняется по крайней мере одно из следующих условий:

- сертификат содержит значение `id-kp-OCSPSigning` в расширении `ExtendedKeyUsage` и подписан тем же ключом УЦ, который выдал проверяемый сертификат;

Примечание — в целях обеспечения возможности долговременного (архивного) хранения подписанных документов допускается применять сертификат службы OCSP, подписанный ключом УЦ, отличным от ключа, использованного данным УЦ при подписи проверяемого сертификата. Данный сертификат службы OCSP может применяться при невозможности выпуска сертификата службы OCSP на том же ключе УЦ в связи с окончанием срока действия этого ключа.

- полномочия службы OCSP (сертификат для подписи OCSP-ответа) на выдачу ответов о статусе сертификатов данного УЦ указаны в настройках системы или приложения;

- сертификат является сертификатом того УЦ, который выдал проверяемый сертификат.

Регламентом службы OCSP могут быть определены дополнительные проверки, применяемые непосредственно к ответу данной службы или к сертификату, используемому для проверки электронной подписи в ответе.

6.2 Проверка статуса сертификата службы OCSP

Запрашивающая сторона должна уметь проверять, был ли сертификат службы OCSP аннулирован или нет. УЦ могут решать этот вопрос одним из следующих способов:

- УЦ может указать, что запрашивающая сторона может не проверять статус сертификата службы OCSP. Для этого в сертификат службы OCSP должно быть включено расширение `id-pkix-ocsp-nocheck`. Расширение не должно быть критичным. Расширение должно содержать значение NULL, заданное в соответствии с подразделом 8.7 ГОСТ Р ИСО/МЭК 8824–93. Негативные последствия от компрометации ключа службы OCSP в данном случае аналогичны компрометации ключа, используемого УЦ для подписи списка аннулированных сертификатов. УЦ может выдавать сертификаты с данным расширением с коротким сроком действия и часто обновлять их. Для указания расширения должен быть использован следующий идентификатор:

`id-pkix-ocsp-nocheck`, «`1.3.6.1.5.5.7.48.1.5`».

Примечание — критичность расширения сертификата определяется наличием флага `critical` (см. [1, подраздел 4.1], указывающего на важность содержащейся в расширении

информации. Если запрашивающая сторона не распознает расширение, помеченное как критичное, ей следует отклонить объект, содержащий данное расширение.

- УЦ может указать, как необходимо проверять статус сертификата службы OCSP. Это можно сделать, указав точку распределения списков аннулированных сертификатов (CDP), описанных в [3]. В данном случае проверка должна выполняться по спискам аннулированных сертификатов (CRL). Для указания других способов проверки статуса сертификата службы OCSP допускается использовать расширение AIA, как это описано в [1, пункт 5.2.7].

- УЦ может не указывать способы проверки статуса сертификата службы OCSP, в этом случае решение о необходимости проверки статуса сертификата и о способе выполнения проверки принимается на основе регламента (политики безопасности) запрашивающей стороны.

7 Форматы запроса и ответа

В данном разделе приведено описание формата запроса к службе OCSP и формата ее ответа в соответствии с абстрактно-синтаксической нотацией версии 1 (АСН.1), определенной в ГОСТ Р ИСО/МЭК 8824–93. Формат запроса к службе OCSP описан в 7.1. Формат ответа службы OCSP описан в 7.2.

7.1 Формат запроса

Запрос к службе OCSP представляет собой структуру `OCSPRequest` и выглядит следующим образом.

```
OCSPRequest ::= SEQUENCE {
    tbsRequest                  TBSRequest,
    optionalSignature   [0]  EXPLICIT Signature OPTIONAL }
```

Поле `tbsRequest` несет в себе содержательную часть запроса к службе OCSP и представляет собой структуру типа `TBSRequest`, описанную в 7.1.1. Запрос может быть подписан, в этом случае в нем присутствует поле `optionalSignature`.

Поле `optionalSignature` содержит идентификатор алгоритма подписи, используемого для подписи запроса к службе OCSP, значение подписи и представляет собой структуру типа `Signature`, описанную в 7.1.2. Дополнительно данное поле может содержать сертификаты, необходимые службе OCSP для проверки подписанного запроса.

7.1.1 Структура TBSRequest

Структура `TBSRequest` представляет собой содержательную часть запроса к службе OCSP и выглядит следующим образом.

```
TBSRequest ::= SEQUENCE {
    version             [0] EXPLICIT Version DEFAULT v1,
    requestorName       [1] EXPLICIT GeneralName OPTIONAL,
    requestList             SEQUENCE OF Request,
    requestExtensions   [2] EXPLICIT Extensions OPTIONAL }
```

Поля структуры `TBSRequest` должны быть заполнены в соответствии со следующим описанием.

`version` — Поле `version` описывает версию запроса к службе OCSP. В соответствии с настоящими рекомендациями поле `version` должно иметь значение `v1`. Поле `version` должно выглядеть следующим образом.

```
Version ::= INTEGER { v1(0) }
```

`requestorName` — Необязательное поле `requestorName` содержит имя запрашивающей стороны и имеет тип `GeneralName`, описанный в [1, подпункт 4.2.1.6].

Примечание — Если запрос к службе OCSP подписан, данное поле должно присутствовать (см. 7.1.2).

`requestList` — Поле `requestList` содержит запрос актуального статуса для каждого из сертификатов, статус которых необходимо получить. Каждый отдельный запрос представляет собой структуру `Request`, описанную в 7.1.1.1.

`requestExtensions` — Необязательное поле `requestExtensions` содержит расширения, относящиеся ко всем сертификатам в запросе к службе OCSP, и имеет тип `Extensions`, описанный в [1, подраздел 4.1]. Возможные расширения и их идентификаторы приведены в 7.3.

7.1.1.1 Структура Request

Структура `Request` содержит сведения о сертификате, актуальный статус которого необходимо получить, а также дополнительные расширения. Данная структура выглядит следующим образом.

```
Request ::= SEQUENCE {
    reqCert                     CertID,
    singleRequestExtensions [0] EXPLICIT Extensions OPTIONAL }
```

Поля структуры `Request` должны быть заполнены в соответствии со следующим описанием.

reqCert — Поле reqCert структуры Request содержит сведения о сертификате, актуальный статус которого необходимо получить, и представляет собой структуру CertID, описанную в 7.1.1.2.

singleRequestExtensions — Необязательное поле singleRequestExtensions структуры Request содержит расширения, относящиеся к определенному сертификату в запросе к службе OCSP, и имеет тип Extensions, описанный в [1, подраздел 4.1]. Возможные расширения и их идентификаторы приведены в 7.3.

7.1.1.2 Структура CertID

Структура CertID содержит сведения о сертификате, актуальный статус которого необходимо получить, и выглядит следующим образом.

```
CertID ::= SEQUENCE {
    hashAlgorithm           AlgorithmIdentifier,
    issuerNameHash          OCTET STRING, -- Хэш-код
                                          -- отличительного
                                          -- имени издателя
    issuerKeyHash           OCTET STRING, -- Хэш-код открытого
                                          -- ключа издателя
    serialNumber            CertificateSerialNumber }
```

Поля структуры CertID должны быть заполнены в соответствии со следующим описанием.

hashAlgorithm — Поле hashAlgorithm структуры CertID представляет собой структуру типа AlgorithmIdentifier, определенную в Р 1323565.1.023–2018, пункт 5.1.1.

Порядок включения в данную структуру идентификаторов российских криптографических алгоритмов хэширования описан в 8.1.

issuerNameHash — Поле issuerNameHash структуры CertID содержит хэш-код, вычисленный от отличительного имени издателя проверяемого сертификата. Хэш-код должен быть вычислен от закодированного в DER представления поля issuer проверяемого сертификата (см. Р 1323565.1.023–2018, пункт 4.2.1) с использованием алгоритма хэширования, идентификатор которого указан в поле hashAlgorithm структуры CertID в запросе к службе OCSP, и в соответствии с Р 1323565.1.025–2019, пункты 9.2.1–9.2.2.

issuerKeyHash — Поле issuerKeyHash структуры CertID содержит хэш-код, вычисленный от значения открытого ключа издателя проверяемого сертификата. Хэш-код должен быть вычислен от значения поля subjectKey сертификата издателя проверяемого сертификата (см. Р 1323565.1.023–2018, подраздел 5.2) с использованием алгоритма хэширования, идентификатор которого указан в поле hashAlgorithm структуры CertID в запросе к службе OCSP, и в соответствии с Р 1323565.1.025–2019, пункты 9.2.1–9.2.2. При вычислении хэш-кода следует исключить тег и длину указанного поля.

Хэш-код от значения открытого ключа издателя проверяемого сертификата используется для определения издателя в дополнение к хэш-коду от его отличительного имени, так как некоторые УЦ могут иметь одинаковые имена (уникальность имени рекомендуется, но не может быть гарантирована). При этом открытые ключи двух УЦ никогда не совпадают, если только оба центра не решили использовать один закрытый ключ, или в случае компрометации ключа.

serialNumber — Поле serialNumber имеет тип CertificateSerialNumber и содержит серийный номер сертификата, актуальный статус которого необходимо получить. Тип CertificateSerialNumber описан в [1, подраздел 4.1].

7.1.2 Структура Signature

Структура Signature содержит идентификатор алгоритма электронной подписи, используемого для подписи запроса к службе OCSP и значение подписи. Дополнительно данное поле может содержать сертификат (сертификаты), необходимые службе OCSP для проверки подписанного запроса.

Структура Signature выглядит следующим образом.

```
Signature ::= SEQUENCE {
    signatureAlgorithm      AlgorithmIdentifier,
    signature               BIT STRING,
    certs                [0] EXPLICIT SEQUENCE OF Certificate
                                                 OPTIONAL }
```

Поля структуры Signature должны быть заполнены в соответствии со следующим описанием.

`signatureAlgorithm` — Поле `signatureAlgorithm` структуры `Signature` представляет собой структуру типа `AlgorithmIdentifier`, определенную в Р 1323565.1.023–2018, пункт 5.1.1.

Порядок включения в данную структуру идентификаторов российских криптографических алгоритмов подписи описан в 8.2.

`signature` — Поле `signature` структуры `Signature` содержит значение подписи, вычисленное для закодированного в DER содержимого структуры `TBSRequest` (см. 7.1.1) в соответствии с Р 1323565.1.025–2019, пункты 7.7.1–7.7.2.

`certs` — Необязательное поле `certs` структуры `Signature` содержит один или несколько сертификатов, необходимых службе OCSP для проверки подписи запроса. Каждый сертификат должен быть помещен в структуру типа `Certificate`, описанную в Р 1323565.1.023–2018, подраздел 4.2. Обычно данное поле содержит цепочку сертификатов за исключением корневого.

7.2 Формат ответа

Ответ службы OCSP представляет собой структуру `OCSPResponse` и выглядит следующим образом.

```
OCSPResponse ::= SEQUENCE {
    responseStatus          OCSPResponseStatus,
    responseBytes       [0] EXPLICIT ResponseBytes OPTIONAL }
```

Поле `responseStatus` определяет статус ответа службы OCSP и имеет тип `OCSPResponseStatus`, описанный в 7.2.1.

Поле `responseBytes` содержит идентификатор типа ответа службы OCSP и сам ответ. Указанные данные помещаются в структуру `ResponseBytes`, описанную в 7.2.2. Поле `responseBytes` должно присутствовать в ответе службы OCSP только в случае, когда значение поля `responseStatus` равно 0, что означает успешный статус ответа (см. 7.2.1).

7.2.1 Тип OCSPResponseStatus

Тип `OCSPResponseStatus` представляет собой числовое значение, определяющее статус ответа службы OCSP. Если значение равно 0, в ответе службы OCSP присутствует поле `responseBytes` (см. 7.2.2). В остальных случаях поле

`responseBytes` отсутствует, а значение `OCSPResponseStatus` содержит тип возникшей ошибки. В случае ошибки ответ службы OCSP не содержит подписи данной службы.

Выделяют следующие статусы ответа службы OCSP.

```
OCSPResponseStatus ::= ENUMERATED {
    successful          (0),  -- Ответ содержит информацию о
                              -- статусе (статусах) сертификата
                              -- (сертификатов).
    malformedRequest    (1),  -- Некорректный запрос. Данный
                              -- ответ означает, что полученный
                              -- запрос не соответствует
                              -- синтаксису протокола OCSP.
    internalError       (2),  -- Внутренняя ошибка. Данный
                              -- ответ означает, что при
                              -- обработке запроса службой OCSP
                              -- произошла ошибка. Запрос
                              -- необходимо повторить и, при
                              -- возможности, отправить запрос
                              -- в другую службу OCSP.
    tryLater            (3),  -- Попробуйте позже. Данный ответ
                              -- означает, что служба OCSP
                              -- работает корректно, но не
                              -- может ответить в данный момент
                              -- времени.
                        (4)   -- Не используется.
    sigRequired         (5),  -- Требуется подпись. Данный
                              -- ответ означает, что
                              -- запрашивающей стороне
                              -- необходимо подписать свой
                              -- запрос к службе OCSP.
    unauthorized        (6)   -- Неуполномоченный. Данный ответ
                              -- может быть возвращен в двух
                              -- случаях. В первом случае
                              -- служба OCSP не уполномочена
                              -- сформировать ответ, который
                              -- будет принят запрашивающей
                              -- стороной (см. 6.1). Во втором
                              -- случае запрашивающая сторона
                              -- не уполномочена направлять
                              -- запрос к данной службе OCSP.
                              -- Определение полномочий
                              -- запрашивающей стороны не
                              -- относится к протоколу OCSP и
                              -- выходит за рамки действия
                              -- настоящих рекомендаций.
```

```
}
```

Примечание — Статус ответа службы OCSP предназначен для указания успеха или ошибки обработки запроса службой и не имеет отношения к статусу сертификата (см. 7.2.2.5), включаемому службой в содержательную часть ответа.

7.2.2 Структура ResponseBytes

Структура `ResponseBytes` представляет собой содержательную часть ответа службы OCSP и выглядит следующим образом.

```
ResponseBytes ::= SEQUENCE {
    responseType            OBJECT IDENTIFIER,
    response                OCTET STRING }
```

Поля структуры `ResponseBytes` должны быть заполнены в соответствии со следующим описанием.

`responseType` — Поле `responseType` структуры `ResponseBytes` содержит идентификатор типа ответа службы OCSP. Ответы службы OCSP могут быть различных типов. В настоящих рекомендациях описано использование базового типа ответа, представленного в виде структуры `BasicOCSPResponse` (см. 7.2.2.1). Для данного типа ответа должен быть использован следующий идентификатор:

`id-pkix-ocsp-basic`, «`1.3.6.1.5.5.7.48.1.1`».

`response` — Поле `response` структуры `ResponseBytes` представляет собой ответ службы OCSP того типа, идентификатор которого указан в поле `responseType`. В случае использования базового типа ответа поле `response` содержит закодированную в DER структуру `BasicOCSPResponse` (см. 7.2.2.1).

7.2.2.1 Структура BasicOCSPResponse

Структура `BasicOCSPResponse` содержит ответ службы OCSP базового типа и выглядит следующим образом.

```
BasicOCSPResponse ::= SEQUENCE {
    tbsResponseData         ResponseData,
    signatureAlgorithm      AlgorithmIdentifier,
    signature               BIT STRING,
    certs               [0] EXPLICIT SEQUENCE OF Certificate
                                                OPTIONAL }
```

Поля структуры BasicOCSPResponse должны быть заполнены в соответствии со следующим описанием.

tbsResponseData — Поле tbsResponseData структуры BasicOCSPResponse содержит данные ответа службы OCSP и представляет собой структуру типа ResponseData, описанную в 7.2.2.2.

signatureAlgorithm — Поле signatureAlgorithm структуры BasicOCSPResponse представляет собой структуру типа AlgorithmIdentifier, определенную в Р 1323565.1.023–2018, пункт 5.1.1.

Порядок включения в данную структуру идентификаторов российских криптографических алгоритмов подписи описан в 8.2.

signature — Поле signature структуры BasicOCSPResponse содержит значение подписи, вычисленное для закодированного в DER содержимого структуры ResponseData (см. 7.2.2.2) в соответствии с Р 1323565.1.025–2019, пункты 7.7.1–7.7.2.

certs — Необязательное поле certs структуры BasicOCSPResponse содержит один или несколько сертификатов, которые могут быть использованы запрашивающей стороной для проверки подписи ответа службы OCSP. Каждый сертификат должен быть помещен в структуру типа Certificate, описанную в Р 1323565.1.023–2018, подраздел 4.2. Данное поле не следует включать в ответ, если у службы OCSP нет указанных сертификатов.

Содержимое структуры BasicOCSPResponse должно быть представлено в виде OCTET STRING и закодировано в DER.

7.2.2.2 Структура ResponseData

Структура ResponseData содержит данные ответа службы OCSP и выглядит следующим образом.

```
ResponseData ::= SEQUENCE {
    version             [0] EXPLICIT Version DEFAULT v1,
    responderID             ResponderID,
    producedAt              GeneralizedTime,
    responses               SEQUENCE OF SingleResponse,
    responseExtensions  [1] EXPLICIT Extensions OPTIONAL }
```

Поля структуры ResponseData должны быть заполнены в соответствии со следующим описанием.

version — Поле version описывает версию ответа службы OCSP. В соответствии с настоящими рекомендациями поле version должно иметь значение v1. Поле version должно выглядеть следующим образом.

```
Version ::= INTEGER { v1(0) }
```

responderID — Поле responderID структуры ResponseData содержит сведения о службе OCSP, создавшей ответ, и представляет собой структуру типа ResponderID, описанную в 7.2.2.3.

producedAt — Поле producedAt обозначает время, когда служба OCSP создала данный ответ, и имеет тип GeneralizedTime, описанный в разделе 32 ГОСТ Р ИСО/МЭК 8824–93. Дополнительные требования к формату значения GeneralizedTime в рамках протокола OCSP приведены в [1, подпункт 4.1.2.5.2].

responses — Поле responses структуры ResponseData содержит информацию об актуальных статусах сертификатов, указанных в запросе к службе OCSP в поле requestList структуры TBSRequest (см. 7.1.1). Информация о каждом сертификате помещается в структуру SingleResponse, которая в свою очередь включается в поле responses. Формат содержимого структуры SingleResponse описан в 7.2.2.4.

responseExtensions — Необязательное поле responseExtensions содержит расширения, относящиеся ко всем сертификатам в ответе службы OCSP, и имеет тип Extensions, описанный в [1, подраздел 4.1]. Возможные расширения и их идентификаторы приведены в 7.3.

При этом если в запросе для проверки актуальности ответа было указано большое случайное число, оно должно также присутствовать в ответе в соответствующем расширении (см. 7.3.1), если только служба OCSP не использует заранее сформированные ответы.

7.2.2.3 Структура ResponderID

Структура ResponderID содержит сведения о службе OCSP, создавшей ответ и выглядит следующим образом.

```
ResponderID ::= CHOICE {
    byName              [1] Name,
    byKey               [2] KeyHash }
```

Поля структуры ResponderID должны быть заполнены в соответствии со следующим описанием.

byName — Поле byName структуры ResponderID имеет тип Name и содержит информацию, идентифицирующую службу OCSP, создавшую данный ответ. Информация, идентифицирующая службу OCSP, должна соответствовать содержимому поля subject структуры TBSCertificate (см. Р 1323565.1.023–2018, пункт 4.2.1) сертификата службы OCSP. Подробное описание типа Name приведено в [1, подпункт 4.1.2.4].

Примечание — Рекомендуется использовать именно данный способ указания сведений о службе OCSP, создавшей ответ.

byKey — Поле byKey структуры ResponderID имеет тип KeyHash и должно содержать хэш-код, вычисленный от поля subjectPublicKey сертификата службы OCSP (см. Р 1323565.1.023–2018, пункт 5.2) с использованием алгоритма хэширования SHA-1. При вычислении хэш-кода следует исключить тег и длину указанного поля. Результат должен быть представлен в виде OCTET STRING и закодирован в DER.

```
KeyHash ::= OCTET STRING -- хэш-код SHA-1 открытого ключа
                         -- отвечающей стороны (за
                         -- исключением полей тега и длины.
```

7.2.2.4 Структура SingleResponse

Структура SingleResponse содержит информацию об актуальном статусе сертификата, рекомендуемый интервал действительности ответа службы OCSP и другие сведения о сертификате. Структура SingleResponse должна выглядеть следующим образом.

```
SingleResponse ::= SEQUENCE {
    certID                  CertID,
    certStatus              CertStatus,
    thisUpdate              GeneralizedTime,
    nextUpdate          [0] EXPLICIT GeneralizedTime OPTIONAL,
    singleExtensions    [1] EXPLICIT Extensions OPTIONAL }
```

Поля структуры SingleResponse должны быть заполнены в соответствии со следующим описанием.

certID — Поле certID структуры SingleResponse содержит сведения о сертификате, актуальный статус которого содержится в ответе. Данное поле имеет тип CertID (см.7.1.1.2).

certStatus — Поле certStatus структуры SingleResponse содержит актуальный статус сертификата и имеет тип CertStatus, описанный в 7.2.2.5.

thisUpdate — Поле thisUpdate структуры SingleResponse содержит время, на которое было известно, что указанный статус является актуальным. Данное поле имеет тип GeneralizedTime, описанный в разделе 32 ГОСТ Р ИСО/МЭК 8824–93. Дополнительные требования к формату значения GeneralizedTime в рамках протокола OCSP приведены в [1, подпункт 4.1.2.5.2].

Ответы, в которых поле thisUpdate содержит более позднее значение, чем локальное системное время запрашивающей стороны, следует считать недостоверными.

nextUpdate — Поле nextUpdate структуры SingleResponse содержит время, не позднее которого будет доступна более новая информация о статусе сертификата. Данное поле имеет тип GeneralizedTime, описанный в разделе 32 ГОСТ Р ИСО/МЭК 8824–93. Дополнительные требования к формату значения GeneralizedTime в рамках протокола OCSP приведены в [1, подпункт 4.1.2.5.2].

Ответы, в которых поле nextUpdate содержит более раннее значение, чем локальное системное время запрашивающей стороны, следует считать недостоверными.

П р и м е ч а н и е — Если значение поля nextUpdate не указано, то возможность использования данного OCSP-ответа, либо необходимость осуществления повторного запроса к службе OCSP определяется регламентом запрашивающей стороны.

singleExtensions — Необязательное поле singleExtensions содержит расширения, относящиеся к определенному сертификату в ответе службы OCSP, и имеет тип Extensions, описанный в [1, подраздел 4.1]. Возможные расширения и их идентификаторы приведены в 7.3.

7.2.2.5 Тип CertStatus

Тип `CertStatus` содержит актуальный статус сертификата. Если статус сертификата отличен от `good` (действителен), в поле включается дополнительная информация о статусе данного сертификата.

Служба OCSP должна поддерживать только следующие возможные статусы:

```
CertStatus ::= CHOICE {
    good                [0] IMPLICIT NULL,
    revoked             [1] IMPLICIT RevokedInfo,
    unknown             [2] IMPLICIT UnknownInfo }
```

`good` (положительный ответ) — Состояние `good` означает положительный ответ на запрос о состоянии (см. подробнее 5.2.1).

`revoked` (аннулирован) — Данный статус означает, что сертификат был аннулирован или приостановлен. Данный статус также может означать, что удостоверяющий центр не выдавал сертификат, который указан в запросе. (см. подробнее 5.2.1).

Статус `revoked` представляет собой структуру `RevokedInfo` и выглядит следующим образом.

```
RevokedInfo ::= SEQUENCE {
    revocationTime          GeneralizedTime,
    revocationReason    [0] EXPLICIT CRLReason
OPTIONAL }
```

Поле `revocationTime` содержит информацию о времени, когда сертификат был аннулирован или приостановлен. Данное поле имеет тип `GeneralizedTime`, описанный в разделе 32 ГОСТ Р ИСО/МЭК 8824–93. Дополнительные требования к формату значения `GeneralizedTime` в рамках протокола OCSP приведены в [1, подпункт 4.1.2.5.2].

Поле `revocationReason` указывает на причину аннулирования сертификата и имеет тип `CRLReason`, описанный в [1, пункт 5.3.1].

В случае если службе OCSP известно, что удостоверяющий центр не выдавал сертификат, который указан в запросе, в поле `responseExtensions` структуры `ResponseData` (см. 7.2.2.2), должно быть включено расширение `id-pkix-ocsp-extended-revoke`, содержащее дополнительную информацию об аннулировании и описанное в 7.3.8. При этом должны выполняться следующие условия.

1) В поле revocationTime структуры RevokedInfo должны быть помещены дата и время, обозначающие 1 января 1970 года.

2) Поле revocationReason структуры RevokedInfo должно содержать причину аннулирования «certificateHold (6)» (см. [1, пункт 5.3.1]).

3) Поле singleExtensions структуры SingleResponse (см. 7.2.2.4), сформированной для данного сертификата, не должно содержать ссылок на CRL (см. 7.3.2) и расширений записей в CRL (см. 7.3.5).

unknown (неизвестен) — Данный статус означает, что служба OCSP не может определить статус сертификата (см. подробнее 5.2.1).

Статус unknown имеет тип UnknownInfo и должен выглядеть следующим образом.

```
UnknownInfo ::= NULL
```

7.3 Расширения запроса и ответа

В данном разделе настоящих рекомендаций определен ряд расширений на основе модели расширения, определенной в [1, раздел 4]. Расширения запросов к службе OCSP и ее ответов имеют тип Extensions, описанный в [1, подраздел 4.1].

В общем виде использование расширений в протоколе OCSP выглядит следующим образом. Запрашивающая сторона может поместить одно или несколько расширений в запрос к службе OCSP (см. 5.1) в соответствии с порядком включения в запрос расширений, описанным в данном разделе. Служба OCSP обрабатывает запрос и включенные в него расширения, после чего формирует ответ (см. 5.2), который также может содержать расширения, обрабатываемые при проверке ответа (см. 5.3).

В запросе к службе OCSP допустимы следующие виды расширений:

- расширение для всего запроса, включаемое в поле requestExtensions структуры TBSRequest (см. 7.1.1);

- расширение для отдельного запроса статуса сертификата, включаемое в поле singleRequestExtensions структуры Request (см. 7.1.1.1).

В ответе службы OCSP допустимы следующие виды расширений:

- расширение для всего ответа, включаемое в поле responseExtensions структуры ResponseData (см. 7.2.2.2);

- расширение для одного ответа о статусе сертификата, включаемое в поле singleExtensions структуры SingleResponse (см. 7.2.2.4).

Поддержка всех расширений, описанных в настоящих рекомендациях, не является обязательной как при получении запроса службой OCSP, так и при проверке ее ответа. Поэтому приведенные в данном разделе расширения не должны быть критичными (см. [1, подраздел 4.1]. Настоящие рекомендации также допускают использование расширений, определенных в иных нормативных документах.

В таблице 1 приведен список расширений запроса к службе OCSP и ее ответа с указанием поля, куда может включаться каждое из расширений. В случае когда расширение может быть включено в некоторое поле запроса и/или ответа, на пересечении строки с именем расширения и столбца с именем поля находится знак «+», в остальных случаях – знак «-».

Т а б л и ц а 1 – Возможность включения расширений в поля запроса и ответа

Расширение	Поле, в которое включается расширение			
	`requestExtensions`	`singleRequestExtensions`	`responseExtensions`	`singleExtensions`
Одноразовый код (`Nonce`), см. 7.3.1	+	-	+	-
Ссылки на CRL (`CrlID`), см. 7.3.2	-	-	-	+
Принимаемые типы ответов (`Acceptable Response Types`), см. 7.3.3	+	-	-	-
Архивный срез (`Archive Cutoff`), см. 7.3.4	-	-	-	+
Расширения записей в CRL (`CRL Entry Extensions`), см. 7.3.5	-	-	-	+
Адрес службы (`Service Locator`), см. 7.3.6	-	+	-	-
Предпочтительные алгоритмы подписи (`Preferred Signature Algorithms`), см. 7.3.7	+	-	-	-
Дополнительная информация об аннулировании (`Extended Revoked Definition`), см. 7.3.8	-	-	+	-

7.3.1 Одноразовый код

Расширение, содержащее одноразовый код, связывает запрос к службе OCSP и ее ответ для предотвращения атак повторного воспроизведения. Для этого данное расширение включается как в запрос, так и в ответ службы OCSP для всего запроса (поле `requestExtensions` структуры `TBSRequest`, см. 7.1.1) и для всего ответа (поле `responseExtensions` структуры `ResponseData`, см. 7.2.2.2) соответственно. Данное расширение имеет следующий объектный идентификатор:

id-pkix-ocsp-nonce, «1.3.6.1.5.5.7.48.1.2».

Значение расширения должно выглядеть следующим образом.

```
Nonce ::= OCTET STRING
```

7.3.2 Ссылки на CRL

Служба OCSP может указать список аннулированных сертификатов (CRL), в котором найден аннулированный или приостановленный сертификат. Для этого используется расширение, позволяющее включить ссылку на CRL в ответ службы OCSP о статусе каждого из сертификатов (поле singleExtensions структуры SingleResponse, см. 7.2.2.4). Ссылка на CRL может быть представлена одним из следующих способов:

- URL, указывающий расположение CRL;

- номер CRL;

- время создания CRL.

Ссылки на CRL могут применяться в случаях, когда служба OCSP использует различные источники информации о статусе сертификатов, а также когда необходимо сохранить сведения о данных источниках в целях аудита. Данное расширение имеет следующий объектный идентификатор:

id-pkix-ocsp-crl, «1.3.6.1.5.5.7.48.1.3».

Значение расширения должно выглядеть следующим образом.

```
CrlID ::= SEQUENCE {
    crlUrl               [0] EXPLICIT IA5String OPTIONAL,
    crlNum               [1] EXPLICIT INTEGER OPTIONAL,
    crlTime              [2] EXPLICIT GeneralizedTime OPTIONAL }
```

Поля структуры CrlID должны быть заполнены в соответствии со следующим описанием.

crlUrl — Поле crlUrl структуры CrlID содержит ссылку, по которой доступен список аннулированных сертификатов, в формате IA5String, описанном в разделе 31 ГОСТ Р ИСО/МЭК 8824-93.

crlNum — Поле crlNum структуры CrlID содержит значение расширения номера списка в формате INTEGER, описанном в разделе 14 ГОСТ Р ИСО/МЭК 8824–93.

crlTime — Поле crlTime структуры CrlID содержит время выпуска соответствующего списка в формате GeneralizedTime, описанном в разделе 32 ГОСТ Р ИСО/МЭК 8824–93. Дополнительные требования к

формату значения GeneralizedTime в рамках протокола OCSP приведены в [1, подпункт 4.1.2.5.2].

7.3.3 Принимаемые типы ответов

Настоящие рекомендации устанавливают формат базового типа ответа службы OCSP id-pkix-ocsp-basic, описанный в 7.2.2. Дополнительно запрашивающая сторона может указать в запросе к службе OCSP типы ответов, которые она поддерживает. Для этого следует в поле requestExtensions запроса к службе OCSP (см. 7.1.1) поместить расширение со следующим объектным идентификатором:

id-pkix-ocsp-response, «1.3.6.1.5.5.7.48.1.4».

Значение расширения представляет собой структуру AcceptableResponses, содержит идентификаторы типов ответов, которые запрашивающая сторона может обработать, и должно выглядеть следующим образом.

```
AcceptableResponses ::= SEQUENCE OF OBJECT IDENTIFIER
```

7.3.4 Архивный срез

Служба OCSP может хранить информацию о статусе сертификата после истечения срока его действия. Время, полученное путем вычитания интервала хранения информации о статусе сертификата из времени формирования ответа службой OCSP (поле producedAt, см. 7.2.2.2), называется временем «архивного среза» статуса сертификата. Например, интервал хранения информации о выданных ответах некоторой службы OCSP составляет 7 лет. В момент времени T данная служба OCSP сформировала ответ об актуальном статусе некоторого сертификата. В расширении архивного среза для данного сертификата в данном ответе будет указано значение времени, соответствующее T за вычетом 7 лет.

Архивный срез позволяет получить дополнительную информацию о статусе сертификата для проверки действительности электронной подписи в момент ее создания, даже если сертификат в настоящее время уже недействителен.

Расширение архивного среза должно включаться в расширения одного ответа о статусе сертификата (поле singleExtensions структуры SingleResponse, см. 7.2.2.4). Данное расширение имеет следующий объектный идентификатор:

id-pkix-ocsp-archive-cutoff, «1.3.6.1.5.5.7.48.1.6».

Значение расширения должно выглядеть следующим образом.

```
ArchiveCutoff ::= GeneralizedTime
```

7.3.5 Расширения записей в CRL

Настоящие рекомендации допускают использование в ответе службы OCSP расширений записей в CRL, описанных в [1, подраздел 5.3]. Данные расширения должны включаться в расширения одного ответа о статусе сертификата (поле `singleExtensions` структуры `SingleResponse`, см. 7.2.2.4).

7.3.6 Адрес службы

Служба OCSP может работать в режиме, в котором она может переадресовывать полученные запросы другой службе OCSP. Информация о другой службе OCSP может быть передана в расширении для отдельного запроса статуса сертификата `ServiceLocator`, включаемом в поле `singleRequestExtensions` структуры `Request` (см. 7.1.1.1). Расширение `ServiceLocator` используется в установленных регламентом случаях, когда требуется передать запрос актуального статуса некоторого сертификата службе OCSP, наделенной полномочиями для данной проверки.

Расширение `ServiceLocator` имеет следующий объектный идентификатор:

`id-pkix-ocsp-service-locator`, «`1.3.6.1.5.5.7.48.1.7`».

Значение расширения представляет собой структуру `ServiceLocator` и должно выглядеть следующим образом.

```
ServiceLocator ::= SEQUENCE {
    issuer                  Name,
    locator                 AuthorityInfoAccessSyntax }
```

Поля структуры `ServiceLocator` должны быть заполнены в соответствии со следующим описанием.

`issuer` — Поле `issuer` структуры `ServiceLocator` содержит имя издателя проверяемого сертификата и имеет тип `Name`, описанный в [1, подпункт 4.1.2.4].

`locator` — Необязательное поле `locator` структуры `ServiceLocator` содержит дополнительную информацию об издателе проверяемого сертификата и имеет тип `AuthorityInfoAccessSyntax`, описанный в [1, подпункт 4.2.2.1].

7.3.7 Предпочтительные алгоритмы подписи

Протокол OCSP не устанавливает определенного списка разрешенных алгоритмов подписи ответа службой OCSP. В то же время протокол OCSP не предусматривает возможности для запрашивающей стороны указывать в запросе предпочтительные

алгоритмы подписи. Поэтому существует риск, что служба OCSP сформирует OCSP-ответ с использованием алгоритма, который не поддерживается запрашивающей стороной.

Регламент службы OCSP может предусматривать различные правила выбора предпочтительного алгоритма подписи. Например, служба OCSP может использовать алгоритм подписи, совпадающий с алгоритмом подписи сертификатов и CRL в обслуживаемом данной службой УЦ. Данные правила не могут быть применены в следующих случаях.

- Алгоритм, используемый УЦ для подписи сертификатов и CRL, неприменим к ключевой паре, используемой службой OCSP для подписи OCSP-ответов.

- Запрос статуса сертификата, о котором служба OCSP не имеет сведений, не содержит достаточной информации для выбора предпочтительного алгоритма подписи ответа.

Служба OCSP может применять алгоритмы подписи ответа, отличные от используемых обслуживаемым ей УЦ, в следующих случаях:

- если алгоритм подписи OCSP-ответа требует меньшего объема вычислений, чем алгоритм подписи самого проверяемого сертификата;

- если требуется защититься от возможной компрометации алгоритма подписи при помощи использования двух различных алгоритмов подписи.

В большинстве случаев при выборе предпочтительного алгоритма подписи достаточно, чтобы данный алгоритм был не менее стойким, чем алгоритм подписи проверяемого сертификата. В то же время данное условие может быть неприменимо при долговременном архивном хранении, если статус сертификата запрашивается на момент времени в прошлом, а используемый алгоритм подписи более не считается стойким.

Далее в данном подразделе содержится следующая информация:

- описание формата расширения, позволяющего запрашивающей стороне указать набор предпочтительных алгоритмов подписи (см. 7.3.7.1);

- порядок выбора предпочтительного алгоритма подписи, позволяющий повысить вероятность успешного взаимодействия по протоколу OCSP в случаях, когда в запросе не указано ни одного предпочтительного алгоритма подписи (см. 7.3.7.2).

7.3.7.1 Формат расширения

Запрашивающая сторона может указать набор предпочтительных алгоритмов подписи. Для этого в запрос к службе OCSP в поле `requestExtensions` структуры `TBSRequest` (см. 7.1.1) должно быть помещено расширение `PreferredSignatureAlgorithms`. Данное расширение имеет следующий объектный идентификатор:

`id-pkix-ocsp-pref-sig-algs`, «`1.3.6.1.5.5.7.48.1.8`».

Значение расширения представляет собой структуру PreferredSignatureAlgorithms, содержащую последовательность структур PreferredSignatureAlgorithm, и должно выглядеть следующим образом.

```
PreferredSignatureAlgorithms ::= SEQUENCE OF
                                    PreferredSignatureAlgorithm
```

Каждая из структур PreferredSignatureAlgorithm содержит информацию об одном предпочтительном алгоритме подписи и должна выглядеть следующим образом.

```
PreferredSignatureAlgorithm ::= SEQUENCE {
    sigIdentifier           AlgorithmIdentifier,
    pubKeyAlgIdentifier     AlgorithmIdentifier OPTIONAL }
```

Поля структуры должны быть заполнены в соответствии со следующим описанием.

sigIdentifier — Поле sigIdentifier структуры PreferredSignatureAlgorithm содержит идентификатор предпочтительного алгоритма подписи и представляет собой структуру типа AlgorithmIdentifier, определенную в Р 1323565.1.023–2018, пункт 5.1.1.

pubKeyAlgIdentifier — Необязательное поле pubKeyAlgIdentifier представляет собой структуру SMIMECapability, описанную в [4], и содержит идентификатор алгоритма открытого ключа, который является предпочтительным для запрашивающей стороны при проверке OCSP-ответа, и дополнительные параметры данного алгоритма. Например, поле pubKeyAlgIdentifier может быть использовано запрашивающей стороной для того, чтобы указать, какая эллиптическая кривая используется указанным в нем алгоритмом.

При заполнении структуры PreferredSignatureAlgorithms должны выполняться следующие условия.

- Запрашивающая сторона должна поддерживать каждый из предпочтительных алгоритмов подписи, указанных в расширении PreferredSignatureAlgorithms.

- Алгоритмы подписи, указанные в структуре PreferredSignatureAlgorithms, должны быть расположены в порядке их предпочтения запрашивающей стороной.

7.3.7.2 Порядок выбора службой OCSP алгоритма подписи ответа

Порядок выбора службой OCSP предпочтительного алгоритма подписи может различаться в зависимости от следующих факторов.

- Служба OCSP не использует заранее сформированные ответы. В этом случае служба OCSP может максимизировать вероятность успешного взаимодействия по протоколу OCSP, применяя следующие правила для выбора алгоритма подписи ответа.

Примечание — Выбираемый алгоритм подписи должен соответствовать требованиям безопасности, устанавливаемыми, например, регламентом службы OCSP.

Примечание — Правила должны применяться в порядке их перечисления. При этом переход к следующему правилу может быть осуществлен только в случае, если предыдущее правило невыполнимо.

1. Выбрать алгоритм подписи, который был указан запрашивающей стороной как предпочтительный.

2. Выбрать алгоритм подписи, который был использован для подписи CRL удостоверяющего центра, выпустившего проверяемый сертификат.

3. Выбрать алгоритм подписи, идентичный алгоритму подписи запроса к службе OCSP `OCSPRequest` (см. 7.1).

4. Выбрать алгоритм подписи, указанный в качестве алгоритма подписи по умолчанию в регламенте службы OCSP.

5. Выбрать алгоритм подписи, предписываемый применимыми нормами технического регулирования или другими нормативно-правовыми актами.

- Служба OCSP использует заранее сформированные ответы. В этом случае может возникнуть ситуация, когда служба OCSP не учитывает информацию, включенную в запрос запрашивающей стороной. Однако служба OCSP может использовать данные запроса для отправки подходящего/соответствующего заранее сформированного ответа. Например, служба OCSP может анализировать запросы, полученные ранее, и формировать ответы об актуальном статусе одного и того же сертификата, подписанные с использованием различных алгоритмов подписи.

7.3.8 Дополнительная информация об аннулировании

В случае если службе OCSP известно, что удостоверяющий центр не выдавал сертификат, который указан в запросе (статус «`revoked`», см. 5.2.1), в поле `responseExtensions` структуры `ResponseData` (см. 7.2.2.2), должно быть включено расширение `id-pkix-ocsp-extended-revoke`, содержащее дополнительную информацию об аннулировании.

Присутствие данного расширения означает, что служба OCSP поддерживает расширенное определение статуса сертификата «аннулирован» для случая, когда УЦ не выдавал данный сертификат (см. 7.2.2.5). Это позволяет определить режим работы службы OCSP, например, при аудите. Расширение `id-pkix-ocsp-extended-revoke` может также быть включено в ответы службы OCSP, содержащие иные статусы сертификатов, если требуется оповестить получателя ответа, что служба OCSP поддерживает включение в ответ дополнительной информации об аннулировании. При этом запрашивающей стороне необязательно учитывать данное расширение при определении статуса сертификата (сертификатов) в ответе службы OCSP.

Данное расширение имеет следующий объектный идентификатор:

id-pkix-ocsp-extended-revoke, «1.3.6.1.5.5.7.48.1.9».

Расширение ExtendedRevoke не должно быть критичным. Расширение должно содержать значение NULL, заданное в соответствии с подразделом 8.7 ГОСТ Р ИСО/МЭК 8824–93.

8 Использование российских криптографических алгоритмов

В данном разделе устанавливаются требования к содержимому запроса к службе OCSP и ее ответа, описываемых в разделе 7, при использовании российских криптографических алгоритмов подписи и хэширования.

8.1 Использование российских алгоритмов хэширования

При использовании российских криптографических алгоритмов хэширования идентификатор используемого алгоритма хэширования должен быть включен в поле hashAlgorithm структуры CertID запроса к службе OCSP (см. 7.1.1.2).

Для алгоритма хэширования, определенного в ГОСТ Р 34.11–2012, необходимо использовать следующие идентификаторы, определенные в Р 1323565.1.025–2019:

- для алгоритма ГОСТ Р 34.11–2012 с длиной хэш-кода 256 бит:

id-tc26-gost3411-2012-256, «1.2.643.7.1.1.2.2»;

- для алгоритма ГОСТ Р 34.11–2012 с длиной хэш-кода 512 бит:

id-tc26-gost3411-2012-512, «1.2.643.7.1.1.2.3».

8.2 Использование российских алгоритмов подписи

При использовании российских криптографических алгоритмов подписи идентификатор используемого алгоритма подписи должен быть включен в поле signatureAlgorithm структуры BasicOCSPResponse ответа службы OCSP (см. 7.2.2.1) и может быть включен в поле optionalSignature структуры OCSPRequest запроса к службе OCSP (см. 7.1.1).

Для алгоритма подписи, определенного в ГОСТ Р 34.10–2012, необходимо использовать следующие идентификаторы, определенные в Р 1323565.1.023–2018, подпункт 5.1.1.1:

- для алгоритма ГОСТ Р 34.10–2012 с ключом подписи 256 бит:

id-tc26-signwithdigest-gost3410-12-256, «1.2.643.7.1.1.3.2»;

- для алгоритма ГОСТ Р 34.11–2012 с ключом подписи 512 бит:

id-tc26-signwithdigest-gost3410-12-512, «1.2.643.7.1.1.3.3».

9 Способы передачи

В данном разделе описаны некоторые способы передачи запросов к службе OCSP и получения ее ответов. Настоящие рекомендации допускают и другие способы передачи указанной информации.

9.1 Протокол OCSP по HTTP

В данном разделе описан способ передачи запросов к службе OCSP и получения ее ответов по протоколу HTTP.

9.1.1 Запрос к службе OCSP по HTTP

Для передачи запросов к службе OCSP по протоколу HTTP могут быть использованы методы GET или POST. В случае если требуется разрешить кэширование HTTP, небольшие запросы (размер которых после кодирования меньше 255 байт) могут быть переданы методом GET. В случае если кэширование HTTP не требуется обеспечивать или размер запроса больше 255 байт, то данный запрос следует передавать методом POST.

В целях обеспечения конфиденциальности взаимодействие со службой OCSP с использованием протокола HTTP может быть защищено при помощи протокола TLS или других протоколов более низкого уровня.

Запрос OCSP, отправленный с помощью метода GET, должен выглядеть следующим образом.

```
GET {Адрес службы OCSP}/{Запрос к службе OCSP OCSPRequest (см. 7.1), последовательно закодированный в DER, затем в base64 [5], после чего представленный в кодировке URL (Percent-Encoding, см. [2])}
```

Компонент `{Адрес службы OCSP}` может быть извлечен из поля `AuthorityInfoAccess` проверяемого сертификата или из локальных настроек запрашивающей стороны.

Запрос OCSP, отправленный с помощью метода POST, должен быть построен следующим образом:

- заголовок `Content-Type` должен иметь значение `application/ocsp-request`;

- в теле запроса должно содержаться значение закодированной в DER структуры `OCSPRequest` (см. 7.1).

9.1.2 Ответ службы OCSP по HTTP

Ответ службы OCSP по протоколу HTTP должен содержать следующие компоненты:

- заголовок `Content-Type` со значением `application/ocsp-response`;

- заголовок `Content-Length`, содержащий длину ответа;

Примечание — В ответе службы OCSP могут присутствовать другие заголовки HTTP, которые могут быть проигнорированы, если они неизвестны запрашивающей стороне.

- тело ответа, содержащее значение закодированной в DER структуры `OCSPResponce` (см. 7.2).

9.2 Протокол OCSP через электронную почту

В данном разделе описан способ передачи запросов к службе OCSP и получения ее ответов посредством электронной почты. Для этого используются следующие объекты MIME.

```
Content-Type: application/ocsp-request
Content-Transfer-Encoding: base64

Content-Type: application/ocsp-response
Content-Transfer-Encoding: base64
```

Каждый из описанных объектов MIME должен содержать данные, закодированные в DER и в base64 [5].

Для MIME-типов `application/ocsp-request` и `application/ocsp-response` реализация передачи может включать в себя дополнительные параметры `name` (имя) и `filename` (название файла) с целью сохранения информации об имени, когда запросы и ответы о штампах времени сохранены в виде файлов.

Если указанные дополнительные параметры включены, необходимо использовать соответствующие расширения файлов:

MIME-тип	Расширение файла
application/ocsp-request	.ORQ
application/ocsp-response	.ORS

Настоящие рекомендации допускают использование произвольного названия файла длиной не более восьми символов без учета трехбуквенного расширения.

9.3 Протокол OCSP на основе файлового протокола

В данном разделе описан способ передачи запросов к службе OCSP и получения ее ответов посредством файлового протокола (например, FTP).

Файл с запросом к службе OCSP или с ее ответом должен содержать только результат DER-кодирования запроса или ответа соответственно и не должен содержать посторонних заголовков или концевых меток.

Файл с запросом к службе OCSP должен иметь расширение `.ORQ`.

Файл с ответом службы OCSP должен иметь расширение `.ORS`.

Приложение А

(нормативное)

OCSP в АСН.1

В данном приложении приведены форматы запроса к службе OCSP и ее ответа в нотации АСН.1, определенной в ГОСТ Р ИСО/МЭК 8824–93.

```
OCSP-2013-88
        {iso(1) identified-organization(3) dod(6) internet(1)
        security(5) mechanisms(5) pkix(7) id-mod(0)
        id-mod-ocsp-2013-88(81)}

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

IMPORTS

     -- PKIX Certificate Extensions
        AuthorityInfoAccessSyntax, CRLReason, GeneralName
        FROM PKIX1Implicit88 { iso(1) identified-organization(3)
        dod(6) internet(1) security(5) mechanisms(5) pkix(7)
        id-mod(0) id-pkix1-implicit(19) }

        Name, CertificateSerialNumber, Extensions,
        id-kp, id-ad-ocsp, Certificate, AlgorithmIdentifier
        FROM PKIX1Explicit88 { iso(1) identified-organization(3)
            dod(6) internet(1) security(5) mechanisms(5) pkix(7)
            id-mod(0) id-pkix1-explicit(18) };

OCSPRequest ::= SEQUENCE {
    tbsRequest                 TBSRequest,
    optionalSignature   [0]  EXPLICIT Signature OPTIONAL }

TBSRequest ::= SEQUENCE {
    version             [0] EXPLICIT Version DEFAULT v1,
    requestorName       [1] EXPLICIT GeneralName OPTIONAL,
    requestList             SEQUENCE OF Request,
    requestExtensions   [2] EXPLICIT Extensions OPTIONAL }

Signature ::= SEQUENCE {
    signatureAlgorithm      AlgorithmIdentifier,
    signature               BIT STRING,
    certs               [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }

Version ::= INTEGER { v1(0) }
```

```
Request ::= SEQUENCE {
    reqCert                     CertID,
    singleRequestExtensions [0] EXPLICIT Extensions OPTIONAL }

CertID ::= SEQUENCE {
    hashAlgorithm           AlgorithmIdentifier,
    issuerNameHash          OCTET STRING, -- Хэш-код отличительного имени
                                          -- издателя
    issuerKeyHash           OCTET STRING, -- Хэш-код открытого ключа
                                          -- издателя
    serialNumber            CertificateSerialNumber }

OCSPResponse ::= SEQUENCE {
    responseStatus          OCSPResponseStatus,
    responseBytes       [0] EXPLICIT ResponseBytes OPTIONAL }

OCSPResponseStatus ::= ENUMERATED {
    successful          (0),  -- Ответ содержит информацию о
                              -- статусе (статусах) сертификата
                              -- (сертификатов).
    malformedRequest    (1),  -- Некорректный запрос. Данный ответ
                              -- означает, что полученный запрос
                              -- не соответствует синтаксису
                              -- протокола OCSP.
    internalError       (2),  -- Внутренняя ошибка. Данный ответ
                              -- означает, что при обработке
                              -- запроса службой OCSP произошла
                              -- ошибка. Запрос необходимо
                              -- повторить и, при возможности,
                              -- отправить запрос в другую службу
                              -- OCSP.
    tryLater            (3),  -- Попробуйте позже. Данный ответ
                              -- означает, что служба OCSP
                              -- работает корректно, но не может
                              -- ответить в данный момент времени.
                        (4)   -- Не используется.
    sigRequired         (5),  -- Требуется подпись. Данный ответ
                              -- означает, что запрашивающей
                              -- стороне необходимо подписать свой
                              -- запрос к службе OCSP.
    unauthorized        (6)   -- Неуполномоченный. Данный ответ
                              -- может быть возвращен в двух
                              -- случаях. В первом случае служба
                              -- OCSP не уполномочена сформировать
                              -- ответ, который будет принят
                              -- запрашивающей стороной (см. 6.1).
                              -- Во втором случае запрашивающая
                            -  - сторона не уполномочена
                            --   направлять запрос к данной службе
                              -- OCSP. Определение полномочий
                              -- запрашивающей стороны не
```

```
                                -- относится к протоколу OCSP и
                                -- выходит за рамки действия
                                -- настоящих рекомендаций.
}

ResponseBytes ::= SEQUENCE {
    responseType            OBJECT IDENTIFIER,
    response                OCTET STRING }

BasicOCSPResponse ::= SEQUENCE {
    tbsResponseData         ResponseData,
    signatureAlgorithm      AlgorithmIdentifier,
    signature               BIT STRING,
    certs               [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL }

ResponseData ::= SEQUENCE {
    version             [0] EXPLICIT Version DEFAULT v1,
    responderID             ResponderID,
    producedAt              GeneralizedTime,
    responses               SEQUENCE OF SingleResponse,
    responseExtensions  [1] EXPLICIT Extensions OPTIONAL }

ResponderID ::= CHOICE {
    byName              [1] Name,
    byKey               [2] KeyHash }

KeyHash ::= OCTET STRING -- хэш-код SHA-1 открытого ключа отвечающей
                         -- стороны (за исключением полей тега и длины).

SingleResponse ::= SEQUENCE {
    certID                  CertID,
    certStatus              CertStatus,
    thisUpdate              GeneralizedTime,
    nextUpdate          [0] EXPLICIT GeneralizedTime OPTIONAL,
    singleExtensions    [1] EXPLICIT Extensions OPTIONAL }

CertStatus ::= CHOICE {
    good                [0] IMPLICIT NULL,
    revoked             [1] IMPLICIT RevokedInfo,
    unknown             [2] IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE {
    revocationTime          GeneralizedTime,
    revocationReason    [0] EXPLICIT CRLReason OPTIONAL }

UnknownInfo ::= NULL

Nonce ::= OCTET STRING

ArchiveCutoff ::= GeneralizedTime

AcceptableResponses ::= SEQUENCE OF OBJECT IDENTIFIER
```

```
ServiceLocator ::= SEQUENCE {
    issuer                  Name,
    locator                 AuthorityInfoAccessSyntax }

CrlID ::= SEQUENCE {
    crlUrl              [0] EXPLICIT IA5String OPTIONAL,
    crlNum              [1] EXPLICIT INTEGER OPTIONAL,
    crlTime             [2] EXPLICIT GeneralizedTime OPTIONAL }

PreferredSignatureAlgorithms ::= SEQUENCE OF PreferredSignatureAlgorithm

PreferredSignatureAlgorithm ::= SEQUENCE {
    sigIdentifier           AlgorithmIdentifier,
    pubKeyAlgIdentifier     AlgorithmIdentifier OPTIONAL }

-- Object Identifiers

id-kp-OCSPSigning            OBJECT IDENTIFIER ::= { id-kp 9 }
id-pkix-ocsp                 OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-basic           OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }
id-pkix-ocsp-nonce           OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }
id-pkix-ocsp-crl             OBJECT IDENTIFIER ::= { id-pkix-ocsp 3 }
id-pkix-ocsp-response        OBJECT IDENTIFIER ::= { id-pkix-ocsp 4 }
id-pkix-ocsp-nocheck         OBJECT IDENTIFIER ::= { id-pkix-ocsp 5 }
id-pkix-ocsp-archive-cutoff  OBJECT IDENTIFIER ::= { id-pkix-ocsp 6 }
id-pkix-ocsp-service-locator OBJECT IDENTIFIER ::= { id-pkix-ocsp 7 }
id-pkix-ocsp-pref-sig-algs   OBJECT IDENTIFIER ::= { id-pkix-ocsp 8 }
id-pkix-ocsp-extended-revoke OBJECT IDENTIFIER ::= { id-pkix-ocsp 9 }

END
```

Приложение Б

(справочное)

Примеры

В данном приложении содержатся примеры запроса к службе получения актуальных статусов сертификатов и ее ответа при использовании российских криптографических алгоритмов. Значения закрытых и открытых ключей представлены в порядке от младшего байта к старшему (little-endian).

При создании примеров были использованы следующие сертификаты.

- Сертификат, статус которого проверяется и который используется для подписи запроса к службе OCSP. Данный сертификат представлен в кодировке Base64 [5] и выглядит следующим образом.

```
MIIBsDCCAV2gAwIBAgIBAjAKBggqhQMHAQEDAjAUMRIwEAYDVQQDEwlFeGFtcGxlQ0EwHhcNMjE
xMTI4MTQ0NDM2WhcNNDIxMDI4MTQ0NDM2WjAUMRIwEAYDVQQDEwlTZXJ2ZXJUTFMwXjAXBggqhQ
MHAQEBATALBgkqhQMHAQIBAQIDQwAEQCbq3kYBdV4pNi3Xt2MX7uQ0g4wA7XU8M3JZoIWfpPDUm
BzULffy1gjM67tbFEPWq1ihb7wSwRktmPm/5tyINuWjgZowgZcwHQYDVR0OBBYEFICwaDpHKbAs
RWB812hG6blB/VPWMA4GA1UdDwEB/wQEAwIB/jATBgNVHSUEDDAKBggrBgEFBQcDATAMBgNVHRM
BAf8EAjAAMEMGA1UdIwQ8MDqAFMn2c6pxSOgMz9zqZeYbt/1ouf4moRikFjAUMRIwEAYDVQQDEw
lFeGFtcGxlQ0GCCFIBBf3MjS+7MAoGCCqFAwcBAQMCA0EAL9OWI7BTH4csA0gZg+dKmBsZWrMrT
gSgPv7GJ7l826YOm3PmBUtC2+DQT8ghZQGcdLk14j3LQ7vpK/q5eLimvw==
```

В качестве закрытого ключа для подписи запроса к службе OCSP используется следующий ключ.

$d = 62DD7FCD96B4028493A7F78CFC65E59ED959AFA77B83E0055E8B89E27CE9E603_{16}$

В качестве соответствующего открытого ключа используется следующий ключ.

$X_q = 26EADE4601755E29362DD7B76317EEE434838C00ED753C337259A0859FA4F0D4_{16}$

$Y_q = 981CD42DF7F2D608CCEBBB5B1443D6AB58A16FBC12C1192D98F9BFE6DC8836E5_{16}$

- Сертификат, используемый для подписи ответа службы OCSP. Данный сертификат представлен в кодировке Base64 [5] и выглядит следующим образом.

```
MIIBwzCCAXCgAwIBAgIBATAKBggqhQMHAQEDAjAUMRIwEAYDVQQDEwlFeGFtcGxlQ0EwHhcNMjE
xMTI3MDk0MjIwWhcNNDIxMDI3MDk0MjIwWjAWMRQwEgYDVQQDEwtPQ1NQU2VydmljZTBeMBcGCC
qFAwcBAQEBMAsGCSqFAwcBAgEBAgNDAARA+xPUdwMP8NtarpmT89Puzav9rQ40Cq8I1ypZhnsku
jBHTjh7yAmElARwrkwikQ/jGMTLjAROKHFnn3VV0QCQHqOBqzCBqDAdBgNVHQ4EFgQUJoSrectb
wJGGOPk3ry/VrNz1C+gwDgYDVR0PAQH/BAQDAgH+MBMGA1UdJQQMMAoGCCsGAQUFBwMJMAwGA1U
dEwEB/wQCMAAwQwYDVR0jBDwwOoAUyfZzqnFI6AzP3Opl5hu3/Wi5/iahGKQWMBQxEjAQBgNVBA
MTCUV4YW1wbGVDQYIIUgEF/cyNL7swDwYJKwYBBQUHMAEFBAIFADAKBggqhQMHAQEDAgNBAA1LM
wfJNkg+ZBFpXeZ3kH5RKWx6EOW9UBKapz9DJoYEP7NhcbzEgH1AJl/hAc9ZikTeuV3XQ3IBgNdn
BLDmIFs===
```

В качестве закрытого ключа для подписи ответа службы OCSP используется следующий ключ.

$d = 84D9AE62ECC2A97960068A9EE61724593FCE2B6A5FE54506064528D22B6E844C_{16}$

В качестве соответствующего открытого ключа используется следующий ключ.

X_q = FB13D477030FF0DB5AAE9993F3D3EECDABFDAD0E340AAF08D72A59867B24BA30$_{16}$

Yq = 474E387BC80984940470AE4C22910FE318C4CB8C044E2871679F7555D100901E$_{16}$

- Сертификат УЦ, выдавшего сертификат, используемый для подписи ответа службы OCSP, и сертификат, статус которого проверяется. Данный сертификат представлен в кодировке Base64 [5] и выглядит следующим образом.

```
MIIBsjCCAV+gAwIBAgIIUgEF/cyNL7swCgYIKoUDBwEBAwIwFDESMBAGA1UEAxMJRXhhbXBsZUN
BMB4XDTIxMTEwMjIwNDk1OVoXDTQyMTAyODIwNDk1OVowFDESMBAGA1UEAxMJRXhhbXBsZUNBMG
gwIQYIKoUDBwEBAQEwFQYJKoUDBwECAQEBBggqhQMHAQECAgNDAARABZaQRjw5FoKjKJeq+PRoo
jqvbSLT60RKeaiHIhqGc7qre7O92ridPqFMMHGuGCAiIk4K550YBCWWg+tNhIFh5aOBizCBiDAd
BgNVHQ4EFgQUyfZzqnFI6AzP3Opl5hu3/Wi5/iYwDgYDVR0PAQH/BAQDAgH+MBIGA1UdEwEB/wQ
IMAYBAf8CAQEwQwYDVR0jBDwwOoAUyfZzqnFI6AzP3Opl5hu3/Wi5/iahGKQWMBQxEjAQBgNVBA
MTCUV4YW1wbGVDQYIIUgEF/cyNL7swCgYIKoUDBwEBAwIDQQAtaycJR+WUCul17CKh4ah1AMI9A
IGE6yEAXDZt/DqaPA1ILgAi6Zhu1dTcwrKdh2+b4I8Es4ORITIsizpTVUgL
```

В качестве закрытого ключа для сертификата УЦ используется следующий ключ.

d = 9C85418F6353B11D7A8F7DC831CA6A82A45C2D9EAC2273DBAF3351A448C36427$_{16}$

В качестве соответствующего открытого ключа используется следующий ключ.

X_q = 059690463C391682A32897AAF8F468A23AAF6D22D3EB444A79A887221A8673BA$_{16}$

Y_q = AB7BB3BDDAB89D3EA14C3071AE182022224E0AE79D1804259683EB4D848161E5$_{16}$

Б.1 Запрос

Запрос к службе OCSP в абстрактно-синтаксической нотации версии 1 (АСН.1), определенной в ГОСТ Р ИСО/МЭК 8824–93.

```
SEQUENCE {
  SEQUENCE {
    [0] {
      INTEGER 0
      }
    [1] {
      [4] {
        SEQUENCE {
          SET {
            SEQUENCE {
              OBJECT IDENTIFIER commonName (2 5 4 3)
              PrintableString 'ServerTLS'
              }
            }
          }
        }
      }
    SEQUENCE {
      SEQUENCE {
        SEQUENCE {
          SEQUENCE {
            OBJECT IDENTIFIER '1 2 643 7 1 1 2 2'
            }
```

```
          OCTET STRING
            E6 4D 6A F1 11 DA 72 6B 3B D6 7B 66 91 52 A5 5F
            D9 9F 8F 31 DB 2E F3 61 BF C0 5A 62 FB 11 C7 EE
          OCTET STRING
            0D 23 82 46 B9 65 45 23 35 2B D3 73 AD 97 DB B3
            F4 34 58 C5 87 1C 0F 4A 69 EE 13 75 F3 9D 53 BB
          INTEGER 2
          }
        }
      }
    }
  [0] {
    SEQUENCE {
      SEQUENCE {
        OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'
        }
      BIT STRING
        C7 FC 0D DC 8D 3D E1 2B 0D 4F 8E F6 64 56 46 7A
        ED D1 66 17 36 17 9E 21 31 C6 83 A2 AB A5 23 A2
        68 5F E8 4D 03 E7 C2 CD 08 07 B8 F3 46 66 6D 05
        76 C0 D5 E7 60 1D 59 49 09 45 52 C4 95 A7 5A D3
      }
    }
  }
```

Запрос к службе OCSP в кодировке Base64 [5].

```
MIHNMHigAwIBAKEYpBYwFDESMBAGA1UEAxMJU2VydmVyVExTMFcwVTBTMAoGCCqFAwcBAQICBCD
mTWrxEdpyazvWe2aRUqVf2Z+PMdsu82G/wFpi+xHH7gQgDSOCRrllRSM1K9NzrZfbs/Q0WMWHHA
9Kae4TdfOdU7sCAQKgUTBPMAoGCCqFAwcBAQMCA0EAx/wN3I094SsNT472ZFZGeu3RZhc2F54hM
caDoqulI6JoX+hNA+fCzQgHuPNGZm0FdsDV52AdWUkJRVLElada0w==
```

Б.2 Ответ

Ответ службы OCSP в абстрактно-синтаксической нотации версии 1 (АСН.1), определенной в ГОСТ Р ИСО/МЭК 8824–93.

```
SEQUENCE {
  ENUMERATED 0
  [0] {
    SEQUENCE {
      OBJECT IDENTIFIER ocspBasic (1 3 6 1 5 5 7 48 1 1)
      OCTET STRING, encapsulates {
        SEQUENCE {
          SEQUENCE {
            [0] {
              INTEGER 0
              }
            [1] {
              SEQUENCE {
                SET {
```

```
              SEQUENCE {
                OBJECT IDENTIFIER commonName (2 5 4 3)
                PrintableString 'OCSPService'
                }
              }
            }
          }
        GeneralizedTime '20220421120000Z'
        SEQUENCE {
          SEQUENCE {
            SEQUENCE {
              SEQUENCE {
                OBJECT IDENTIFIER '1 2 643 7 1 1 2 2'
                }
              OCTET STRING
                E6 4D 6A F1 11 DA 72 6B 3B D6 7B 66 91 52 A5 5F
                D9 9F 8F 31 DB 2E F3 61 BF C0 5A 62 FB 11 C7 EE
              OCTET STRING
                0D 23 82 46 B9 65 45 23 35 2B D3 73 AD 97 DB B3
                F4 34 58 C5 87 1C 0F 4A 69 EE 13 75 F3 9D 53 BB
              INTEGER 2
              }
            [0]
            GeneralizedTime '20220421120000Z'
            }
          }
        }
      SEQUENCE {
        OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'
        }
      BIT STRING
        F9 AD 07 FA B0 B9 85 F3 47 3D 73 01 80 6C CF 16
        F8 09 C7 D0 4D 2F 44 82 B2 F2 E4 E0 D3 22 4B 8C
        68 5F E8 4D 03 E7 C2 CD 08 07 B8 F3 46 66 6D 05
        76 C0 D5 E7 60 1D 59 49 09 45 52 C4 95 A7 5A D3
      }
    }
  }
}
```

Ответ службы OCSP в кодировке Base64 [5].

```
MIIBCQoBAKCCAQIwgf8GCSsGAQUFBzABAQSB8TCB7jCBnKADAgEAoRgwFjEUMBIGA1UEAxMLT0N
TUFNlcnZpY2UYDzIwMjIwNDIxMTIwMDAwWjBqMGgwUzAKBggqhQMHAQECAgQg5k1q8RHacms71n
tmkVKlX9mfjzHbLvNhv8BaYvsRx+4EIA0jgka5ZUUjNSvTc62X27P0NFjFhxwPSmnuE3XznVO7A
gECgAAYDzIwMjIwNDIxMTIwMDAwWjAKBggqhQMHAQEDAgNBAPmtB/qwuYXzRz1zAYBszxb4CcfQ
TS9EgrLy5ODTIkuMaF/oTQPnws0IB7jzRmZtBXbA1edgHVlJCUVSxJWnWtM===
```

Библиография

[1] IETF RFC 5280 — Cooper, D., Santesson, S., Farrell, S., Boeyen, R., Housley, R., Polk, W., Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, IETF RFC 5280, 2008

[2] IETF RFC 3986 — Berners-Lee, T., W3C/MIT, Fielding, R., Day Software, Masinter, L., Adobe Systems, Uniform Resource Identifier (URI): Generic Syntax, IETF RFC 3986, 2005

[3] ISO/IEC 9594-8:2020 — Information Technology-Open Systems Interconnection — The Directory: Public-Key and Attribute Certificate Frameworks, International standard ISO/IEC 9594-8, 2020

[4] IETF RFC 5751 — Ramsdell, B., Brute Squad Labs, Turner, S., IECA, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2. Message Specification, IETF RFC 5751, 2010

[5] IETF RFC 4648 — Josefsson, S., The Base16, Base32, and Base64 Data Encodings, IETF RFC 4648, 2008

УДК 681.3.06:006.354 ОКС 35. 030 ОКСТУ 5002 П85

Ключевые слова: криптография, синтаксис, электронная подпись, сертификат, статус сертификата, протокол OCSP

Руководители рабочей группы ТК 26 по сопутствующим криптографическим алгоритмам и протоколам:

С.В. Смышляев, ООО «КРИПТО-ПРО», svs@cryptopro.ru

В.А. Шишкин, ТК 26, shishkin_va@tc26.ru

Авторы документа:

П.В. Смирнов, ООО «КРИПТО-ПРО», spv@cryptopro.ru, М.В. Парамонова, ООО «КРИПТО-ПРО», mparamonova@cryptopro.ru