МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

МИНИСТЕРСТВО ЮСТИЦИИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ЗАРЕГИСТРИРОВАНО
Регистрационный № 61867
от "28" декабря 2020 г.

ПРИКАЗ

06.11.2020 № 580

Москва

Об утверждении порядка создания и проверки метки доверенного времени

В соответствии с пунктом 19 статьи 2 Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи» (Собрание законодательства Российской Федерации, 2011, № 15, ст. 2036; 2019, № 52, ст. 7794) и пунктом 1 Положения о Министерстве цифрового развития, связи и массовых коммуникаций Российской Федерации, утвержденного постановлением Правительства Российской Федерации от 2 июня 2008 г. № 418 (Собрание законодательства Российской Федерации, 2008, № 23, ст. 27.08; 2018, № 40, ст. 6142),

ПРИКАЗЫВАЮ:

1. Утвердить прилагаемый порядок создания и проверки метки доверенного времени.

2. Направить настоящий приказ на государственную регистрацию в Министерство юстиции Российской Федерации.

3. Настоящий приказ вступает в силу с 1 января 2021 г. и действует до 1 января 2027 г.

Министр М.И. Шадаев

УТВЕРЖДЕН
приказом Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации
от 06.11.2020 г. № 580

ПОРЯДОК
создания и проверки метки доверенного времени

1. Настоящий Порядок определяет правила создания и проверки метки доверенного времени.

2. Метка доверенного времени создается доверенной третьей стороной, удостоверяющим центром или оператором информационной системы (далее – служба меток доверенного времени) с использованием программных и (или) аппаратных средств, прошедших процедуру подтверждения соответствия требованиям, установленным в соответствии с пунктом 19 статьи 2 Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи» (далее – Федеральный закон «Об электронной подписи»).

3. Создание метки доверенного времени осуществляется службой меток доверенного времени по запросу участников электронного взаимодействия путем подписания электронной подписью службы меток доверенного времени текущего достоверного значения времени в соответствии с требованиями к структуре метки доверенного времени.

4. С целью обеспечения достоверной информации о моменте подписания электронного документа метка доверенного времени может быть присоединена к подписываемому документу или связана с подписываемым документом иным способом.

5. Формат запроса метки доверенного времени и формат ответа службы меток доверенного времени на такой запрос, предусматривающий обеспечение целостности и достоверности метки доверенного времени, используются службой меток доверенного времени в соответствии с законодательством Российской Федерации в сфере технического регулирования и стандартизации.

6. Службы меток доверенного времени, создаваемые доверенной третьей стороной или удостоверяющим центром, при создании метки доверенного времени должны получать информацию о точном значении времени с учетом часового пояса и календарной дате от технических средств передачи эталонных сигналов времени и частоты, а также информацию о точном значении времени и календарной дате (далее – технические средства передачи эталонных сигналов времени), функционирующих в соответствии с Положением о Государственной службе времени, частоты и определения параметров вращения Земли, утвержденным постановлением Правительства Российской Федерации от 23 марта 2001 г. № 225

(Собрание законодательства Российской Федерации, 2001, № 14, ст. 1361; 2018, № 49, ст. 7600). В метку времени заносятся значения о точном значении времени в соответствии со всемирным координированным временем (далее – UTC).

7. Службы меток доверенного времени, создаваемые операторами информационных систем, получают информацию о дате и времени от технических средств передачи эталонных сигналов времени или от службы меток доверенного времени, указанных в пункте 6 настоящего Порядка, посредством информационно-телекоммуникационных сетей при условии обеспечения целостности передаваемой информации с использованием средств криптографической защиты информации, имеющих подтверждение соответствия требованиям, установленным согласно части 5 статьи 8 Федерального закона «Об электронной подписи».

8. Проверка метки доверенного времени осуществляется службой меток доверенного времени при проверке действительности электронной подписи с использованием средств, прошедших процедуру подтверждения соответствия требованиям, установленным согласно части 5 статьи 8 Федерального закона «Об электронной подписи» в следующем порядке:

1) служба меток доверенного времени проверяет математическую корректность электронной подписи;

2) служба меток доверенного времени проверяет применимость метки доверенного времени к электронной подписи, для которой данная метка доверенного времени создана;

3) служба меток доверенного времени осуществляет сравнение даты и времени, указанных в полученной для электронной подписи метке времени, со сроками действия сертификата ключа проверки электронной подписи, соответствующего проверяемой электронной подписи. Дата и время должны находиться в пределах срока действия данного сертификата;

4) служба меток доверенного времени получает сведения об аннулировании сертификата ключа проверки электронной подписи на момент выдачи метки доверенного времени для данной электронной подписи. Дата и время аннулирования данного сертификата ключа проверки электронной подписи должны быть позже даты и времени, указанных в полученной для электронной подписи метке доверенного времени.

9. Создание метки доверенного времени осуществляется в соответствии со следующими требованиями:

1) Электронный документ должен содержать доказательства, что объекты данных существовали до определенного момента времени (далее – протокол штампов времени). В целях обеспечения указанных доказательств осуществляется взаимодействие службы штампов времени (TSA) и запрашивающей стороны посредством формирования запроса к TSA и проверки ее ответа.

2) Служба штампов времени при создании метки времени должна:

использовать значение UTC;

включать достоверное значение времени в каждый штамп;

включать уникальное целое число в каждый новый штамп;

создавать штамп при получении запроса от запрашивающей стороны, когда это возможно;

включать в каждый штамп времени идентификатор политики безопасности (регламента), согласно которой он был создан;

ставить штамп времени только для хэш-значения, вычисленного с использованием устойчивой однонаправленной хэш-функции;

проверять соответствие длины хэш-значения длине, определенной в алгоритме хэширования, идентификатор которого указан в запросе к TSA;

не подвергать хэш-значение, которому присвоен штамп, какой-либо проверке (кроме проверки его длины, как это указано в предыдущем пункте);

не включать в штампы времени какую-либо информацию, идентифицирующую запрашивающую сторону;

подписывать каждый штамп времени ключом, сгенерированным специально для этой цели;

включать дополнительную информацию в штамп времени, если этого требует запрашивающая сторона, используя только те расширения, которые поддерживаются службой штампов времени. Если это невозможно, служба штампов времени должна ответить сообщением об ошибке.

3) В первом сообщении обмена в рамках данного протокола запрашивающая сторона посылает в службу штампов времени запрос на штамп времени (далее – TimeStampReq). Во втором сообщении служба штампов времени отвечает запрашивающей стороне сообщением (далее – TimeStampResp).

При получении TimeStampResp, содержащего штамп времени (далее – TimeStampToken), запрашивающая сторона должна проверить ответ на ошибки о состоянии и, если их нет, проверить различные поля в TimeStampToken, а также действительность электронной подписи, которой подписан TimeStampToken, и убедиться, что данные с проставленным штампом времени соответствуют отправленным данным. Запрашивающая сторона должна проверить, что TimeStampToken содержит правильный идентификатор сертификата службы штампов времени (ESSCertIDv2), правильное хэш-значение (hashedMessage) и правильный идентификатор алгоритма хэширования (hashAlgorithm) в поле messageImprint.

После этого запрашивающая сторона должна проверить актуальность ответа, соотнеся его время с собственным доверенным источником времени, если источник существует, либо сравнив включенное в ответ значение nonce со значением в запросе, а также убедиться в действительности сертификата службы штампов времени с помощью проверки соответствующего списка отозванных сертификатов.

4) Запрос к службе штампов времени должен представлять собой структуру типа TimeStampReq:

```
TimeStampReq ::= SEQUENCE {
version                 INTEGER { v1(1) },
messageImprint          MessageImprint,
    -- OID алгоритма хэширования и хэш-значение от данных, для которых
    -- требуется штамп времени
reqPolicy       TSAPolicyId         OPTIONAL,
```

nonce INTEGER OPTIONAL,
certReq BOOLEAN DEFAULT FALSE,
extensions [0] IMPLICIT Extensions OPTIONAL }

Поля структуры TimeStampReq должны быть заполнены следующим образом:

version –поле version описывает версию запроса штампа времени. Поле version должно иметь значение 1;

messageImprint –поле messageImprint должно представлять собой структуру типа MessageImprint. Структура MessageImprint должна выглядеть следующим образом:

MessageImprint ::= SEQUENCE {
hashAlgorithm AlgorithmIdentifier,
hashedMessage OCTET STRING }

Служба штампов времени отказывает в выдаче штампа времени, возвращая сообщение со значением badAlg в поле PKIFailureInfo, если хэш-алгоритм не распознан.

Поле hashedMessage структуры MessageImprint должно содержать хэш-значение от данных, для которых требуется штамп времени. Служба штампов времени проверяет совпадение длины хэш-значения с длиной, установленной примененным хэш-алгоритмом, идентификатор которого указан в поле AlgorithmIdentifier.

По результатам такой проверки, в случае несовпадения указанных длин, служба штампов времени отказывает в выдаче штампа времени, возвращая сообщение со значением badAlg в поле badDataFormat;

reqPolicy –поле reqPolicy (при наличии) имеет тип TSAPolicyId и обозначает регламент службы штампов времени, согласно которому следует выдать TimeStampToken. TSAPolicyId определяется следующим образом:

TSAPolicyId ::= OBJECT IDENTIFIER;

nonce –поле nonce (при наличии) имеет тип INTEGER и позволяет клиенту проверить актуальность ответа, когда локальное время недоступно. Значение поля nonce должно представлять собой уникальное 64-битное целое число, сгенерированное клиентом случайным образом. При отсутствии значения nonce в ответе, ответ клиентом должен отклонен;

certReq –поле certReq (при наличии) имеет тип BOOLEAN. Если поле certReq присутствует и имеет значение «true», сертификат ключа проверки подписи службы штампов времени должен быть помещен в поле certificates в структуре SignedData.

Если поле certReq отсутствует и имеет значение «false», поле certificates в структуре SignedData в ответе службы штампов времени не указывается;

extensions –поле extensions (расширения) имеет тип Extensions и позволяет добавить дополнительную информацию в запрос.
Служба штампов времени отказывает в выдаче штампа времени, возвращая сообщение об ошибке (unacceptedExtension), если поле Extensions не распознано.

5) Ответ службы штампов времени должен представлять собой структуру TimeStampResp и выглядеть следующим образом:

```
TimeStampResp ::= SEQUENCE {
  status              PKIStatusInfo,
  timeStampToken      TimeStampToken    OPTIONAL }
```

6) Структура PKIStatusInfo должна содержать информацию о статусе запроса к службе штампов времени и выглядеть следующим образом:

```
PKIStatusInfo ::= SEQUENCE {
  status       PKIStatus,
  statusString PKIFreeText    OPTIONAL,
  failInfo     PKIFailureInfo OPTIONAL }
```

Поле status имеет тип PKIStatus.
Поле statusString имеет тип PKIFreeText.
Поле failInfo имеет тип PKIFailureInfo.

7) Тип PKIStatus структуры PKIStatusInfo должен представлять собой числовое значение, определяющее статус службы штампов времени. Если значение равно 0 или 1, штамп времени TimeStampToken должен присутствовать. Если присутствует иное значение (кроме 0 или 1), штамп времени TimeStampToken не должен присутствовать.
Служба штампов времени должна поддерживать только следующие возможные статусы:

```
PKIStatus ::= INTEGER {
  granted
  -- когда PKIStatus содержит значение 0, TimeStampToken
  -- присутствует, как и запрашивалось.
  grantedWithMods
  -- когда PKIStatus содержит значение 1, TimeStampToken
  -- присутствует с изменениями.
  rejection
  -- штамп времени не получен, причина указана в информационном сообщении
  waiting
  -- запрос штампа времени еще не обработан, дополнительная информация ожидается позже.
  revocationWarning
  -- это сообщение предупреждает о том, что аннулирование неизбежно
  revocationNotification
  -- уведомление о том, что аннулирование имело место.
```

8) Тип PKIFreeText структуры PKIStatusInfo должен представлять собой объяснение причины отклонения запроса штампа времени в виде текста.

9) Тип PKIFailureInfo структуры PKIStatusInfo должен представлять собой числовое значение, определяющее тип произошедшей ошибки. Если TimeStampToken отсутствует, PKIFailureInfo показывает причину отклонения запроса штампа времени.

Служба штампов времени должна поддерживать только следующие возможные значения:

PKIFailureInfo ::= BIT STRING {
badAlg
-- нераспознанный или неподдерживаемый идентификатор алгоритма
badRequest
-- транзакция не разрешена или не поддерживается
badDataFormat
-- отправленные данные имеют неверный формат
timeNotAvailable
-- источник времени в службе штампов времени недоступен
unacceptedPolicy
-- служба штампов времени не поддерживает запрашиваемую политику безопасности
unacceptedExtension
-- служба штампов времени не поддерживает запрашиваемое расширение
addInfoNotAvailable
-- запрашиваемая дополнительная информация не распознается или недоступна
systemFailure
-- запрос не может быть обработан вследствие ошибки системы.

10). Структура TimeStampToken содержит штамп времени и должна представлять собой структуру типа ContentInfo, где поле contentType должно содержать идентификатор типа содержимого «Подписанные данные» signed-data, а поле content должно содержать соответствующую структуру SignedData. Штамп времени TimeStampToken должен выглядеть следующим образом:

TimeStampToken ::= ContentInfo
-- contentType is id-signedData (P 1323565.1.025–2019)
-- content is SignedData (P 1323565.1.025–2019)

Настоящие требования определяют дополнительные условия к содержимому следующих полей структуры SignedData:

- поля encapContentInfo;

- поля signedAttrs, входящего в структуру SignerInfo, включающего в себя подписанный атрибут, содержащий идентификатор сертификата TSA.

11) Поле encapContentInfo структуры SignedData должно представлять собой структуру типа EncapsulatedContentInfo. В него необходимо включать следующие значения:

- в поле eContentType – следующий объектный идентификатор штампа времени:

id-ct-TSTInfo OBJECT IDENTIFIER ::= { iso(1) member-body(2)

us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) ct(1) 4};

- в поле eContent – штамп времени в виде строки октетов (OCTET STRING). Значение поля eContent должно быть результатом DER-кодирования структуры TSTInfo.

Структура TSTInfo содержит штамп времени и должна быть представлена следующим образом:

```
TSTInfo ::= SEQUENCE {
  version                  INTEGER { v1(1) },
  policy                   TSAPolicyId,
  messageImprint           MessageImprint,
  -- должно содержать то же значение, что и поле TimeStampReq
  serialNumber             INTEGER,
  -- сторона, запрашивающая штамп времени, должна принимать целые числа длиной
до 160 бит.
  genTime                  GeneralizedTime,
  accuracy                 Accuracy                OPTIONAL,
  ordering                 BOOLEAN                 DEFAULT FALSE,
  nonce                    INTEGER                 OPTIONAL,
  -- должно присутствовать, если аналогичное поле имеется в
  -- TimeStampReq. В этом случае поле должно иметь то же значение.
  tsa                      [0] GeneralName         OPTIONAL,
      extensions           [1] IMPLICIT Extensions OPTIONAL }
```

12) Поля структуры TSTInfo должны быть заполнены в соответствии со следующими требованиями:

version – поле version имеет тип INTEGER и описывает версию запроса штампа времени. Поле version должно иметь значение 1. Запрашивающая сторона должна распознавать штампы времени версии 1;

Policy – поле policy имеет тип TSAPolicyId и должно обозначать политику безопасности (регламент службы штампов времени, в рамках которой был дан ответ). Если в TimeStampReq присутствует поле reqPolicy, оно должно иметь то же значение, иначе должна быть возвращена ошибка unacceptedPolicy.
Поле policyсодержит:
- условия использования штампа времени;
- информацию о доступности журнала штампов времени для проверки подлинности штампа времени;

messageImprint – поле messageImprint имеет тип MessageImprint и должно иметь то же значение, что и поле messageImprint в TimeStampReq;

serialNumber – поле serialNumber имеет тип INTEGER и является уникальным целым числом, присвоенным данной службой штампов времени каждому штампу времени TimeStampToken. Служба штампов времени обеспечивает уникальность этого числа в том числе в случаях сбоя в работе штампов времени;

genTime – поле genTime имеет тип GeneralizedTime и обозначает время создания штампа времени соответствующей службой, должно быть выражено в UTC.
Значения GeneralizedTime должны включать секунды. Следует использовать GeneralizedTime с точностью до секунд. В отличие от ограничений, поле GeneralizedTime может содержать и доли секунд.
Используется следующий синтаксис: ГГГГММДДччммсс[.s...]Z.
Предусмотрены следующие ограничения к DER-кодированию:
1) результат кодирования должен оканчиваться на Z. При этом десятичный знак, если он присутствует, представляется в виде точки.2) полночь представляется в следующей форме: ГГГГММДД000000Z, где ГГГГММДД обозначает наступивший день;

accuracy – поле accuracy должно представлять собой структуру типа Accuracy и обозначает отклонение времени от времени, значение которого содержится в поле GeneralizedTime. Структура Accuracy выглядит следующим образом:

```
Accuracy ::= SEQUENCE {
    seconds      INTEGER           OPTIONAL,
    millis   [0] INTEGER (1..999)  OPTIONAL,
          micros [1] INTEGER (1..999)  OPTIONAL
      }
```

Если поля seconds, millis или micros отсутствуют, то их значение должно быть равным 0.
Добавлением значения точности поле accuracy к GeneralizedTime вычисляется верхний предел значения времени в штампе времени, созданном службой штампов времени. Вычитанием точности из GeneralizedTime вычисляется нижний предел значения времени в штампе времени, созданном службой штампов времени. Точность представляется в секундах, миллисекундах и микросекундах в виде целого числа;

ordering – поле ordering (упорядочивание) имеет тип BOOLEAN и используется для обозначения работы службы штампов времени

в режиме, когда время создания двух штампов времени не может совпадать.

Значение «true» поля ordering означает необходимость упорядочивания штампов времени, последовательно выдаваемых службой штампов времени, по полю genTime вне зависимости от значения поля accuracy.

Отсутствие поля ordering или его значение «false» означают отсутствие необходимости упорядочивания штампов времени, последовательно выданных службой штампов времени по полю genTime. В этом случае упорядочивание штампов времени осуществляется только тогда, когда разница между genTime первого штампа и genTime второго штампа больше суммы значений точности genTime для каждого штампа;

nonce – поле nonce (при наличии) имеет тип INTEGER и позволяет клиенту проверить актуальность значения штампа времени, когда локальное время недоступно. Значение поля nonce должно представлять собой уникальное 64-битное целое число, сгенерированное клиентом случайным образом. При отсутствии значения nonce в ответе, ответ клиентом должен отклонен;

tsa – поле tsa имеет тип GeneralName и содержит название службы штампов времени. Значение поля tsa должно соответствовать одному из имен субъектов, включенных в сертификат, используемый для проверки подписи штампа времени;

extensions – поле extensions (расширения) имеет тип Extensions и предназначено для дополнительной информации, помещаемой в запрос.

13) Штамп времени не должен содержать иных подписей, кроме подписи службы штампов времени.

Идентификатор сертификата службы штампов времени ESSCertIDv2 должен быть включен в атрибут SigningCertificateV2.

Атрибут SigningCertificateV2 является подписанным атрибутом и должен быть включен в поле signedAttrs структуры SignerInfo.