ГОСУДАРСТВЕННЫЙ СТАНДАРТ
РЕСПУБЛИКИ БЕЛАРУСЬ

СТБ 34.101.87

Информационные технологии и безопасность
ИНФРАСТРУКТУРЫ АУТЕНТИФИКАЦИИ

Інфармацыйныя тэхналогіі і бяспека
ІНФРАСТРУКТУРЫ АЎТЭНТЫФІКАЦЫІ

Госстандарт
Минск

УДК 004.4.056.55(083.74)(476) МКС 35.240.40 КП 05

Ключевые слова: информационные технологии, безопасность, аутентификация, идентификация, федерация доверия, требования безопасности, технология OIDC

Предисловие

Цели, основные принципы, положения по государственному регулированию и управлению в области технического нормирования и стандартизации установлены Законом Республики Беларусь «О техническом нормировании и стандартизации».

1 РАЗРАБОТАН учреждением Белорусского государственного университета «Научно-исследовательский институт прикладных проблем математики и информатики»

2 УТВЕРЖДЕН и ВВЕДЕН В ДЕЙСТВИЕ постановлением Госстандарта Республики Беларусь от 1 декабря 2022 г. № 116

3 ВВЕДЕН ВПЕРВЫЕ

Издан на русском языке

Содержание

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ

Информационные технологии и безопасность ИНФРАСТРУКТУРЫ АУТЕНТИФИКАЦИИ

Інфармацыйныя тэхналогіі і бяспека ІНФРАСТРУКТУРЫ АЎТЭНТЫФІКАЦЫІ

Information technology and security
Authentication frameworks

Дата введения 2023-07-01

1 Область применения

Настоящий стандарт устанавливает правила построения инфраструктур аутентификации и требования к блокам инфраструктур. Охватываются вопросы идентификации пользователей, в том числе их регистрации и подтверждения личности, организации аутентификации, распространения сведений об аутентификации среди сторон, связанных отношениями доверия. Представлена технология OpenID Connect (OIDC), ориентированная на построение в Интернете крупных открытых инфраструктур аутентификации.

Настоящий стандарт применяется при разработке информационных систем, в которых используются «цифровые образы» пользователей, а также сопровождающих эти системы средств аутентификации и средств криптографической защиты информации.

2 Нормативные ссылки

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

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

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

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

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

СТБ 34.101.47-2017 Информационные технологии и безопасность. Криптографические алгоритмы генерации псевдослучайных чисел

СТБ 34.101.65-2014 Информационные технологии и безопасность. Протокол защиты транспортного уровня (TLS)

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

СТБ 34.101.78-2019 Информационные технологии и безопасность. Профиль инфраструктуры открытых ключей

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных документов на официальном сайте Национального фонда технических нормативных правовых актов в глобальной компьютерной сети Интернет.

Если ссылочные документы заменены (изменены), то при пользовании настоящим стандартом следует руководствоваться действующими взамен документами. Если ссылочные документы отменены без замены, то положение, в котором дана ссылка на них, применяется в части, не затрагивающей эту ссылку.

3 Термины и определения и сокращения

3.1 В настоящем стандарте применяют термины, установленные в СТБ 34.101.19, СТБ 34.101.23, СТБ 34.101.27, СТБ 34.101.31, СТБ 34.101.45 и СТБ 34.101.65, а также следующие термины с соответствующими определениями:

3.1.1 авторизация: Назначение прав доступа.

Примечание — В настоящем стандарте права назначает служба идентификации, права назначаются прикладной системе, права касаются ресурсов пользователя (их владельца).

3.1.2 активация: Перевод объекта или устройства в оперативное состояние, которое открывает доступ к другим объектам и устройствам.

Примечание — Доступ, открываемый при активации одного объекта или устройства, может касаться активации другого. Например, активация пароля состоит в проверке того, что владелец криптографического токена ввел его правильно и вследствие этого активировал токен, т. е. получил доступ к его объектам.

3.1.3 аттестат: Информационный объект, который связывает идентификационные данные пользователя с его токенами аутентификации, создается при регистрации пользователя, используется в процессе аутентификации.

3.1.4 аутентификатор: Полученные с помощью токена аутентификации данные, которые подтверждают владение им или его знание.

3.1.5 аутентификация: Проверка подлинности стороны.

3.1.6 билет: Информационный объект, который выпускается по итогам успешной аутентификации пользователя или после предъявления другого билета и подтверждает определенные события, факты или состояние.

Примечание — Билет может содержать секретные данные определенного уровня энтропии.

3.1.7 билет аутентификации; БА: Билет, который содержит утверждения аутентификации и, возможно, другие утверждения.

3.1.8 билет доступа; БД: Билет, который подтверждает права доступа к определенным ресурсам определенного пользователя.

3.1.9 билет обновления; БО: Билет, при предъявлении которого можно получить новый билет аутентификации, доступа или обновления без повторной аутентификации.

3.1.10 билет сеанса; *куки*; БС: Билет, который подтверждает открытие сеанса с другой стороной.

Примечание — В интернет-системах билет сеанса обычно разделяется между браузером и сервером и называется куки (от англ. cookie).

3.1.11 биометрические данные: Описание биологических и (или) поведенческих отличительных характеристик пользователя, которое используется для его автоматического распознавания.

Примечание 1 — Примеры биометрических характеристик: папиллярная структура Гальтона, топография лица, текстура кожи лица, топография кисти руки, топография пальца, структура радужной оболочки глаза, структура сосудов кисти руки, папиллярная структура ладони, изображение сетчатки глаза, динамика рукописной подписи, динамика нажатия клавиш, голос.

Примечание 2 — Фотография пользователя, которая при его распознавании обрабатывается вручную, не относится к биометрическим данным. В настоящем стандарте под фотографиями понимаются именно такие, не биометрические. При этом стандарт допускает, что в состав

биометрических данных входят фотографии, которые обрабатываются автоматически, с помощью специальных алгоритмов.

3.1.12 защищенное соединение: Соединение, которое обеспечивает конфиденциальность, контроль целостности и, возможно, подлинности сообщений.

Примечание — Контроль подлинности сообщений от стороны A к стороне B обеспечивается после того, как B провела аутентификацию A.

3.1.13 идентификатор: Данные, ассоциированные с определенной стороной и позволяющие отличить ее от других сторон.

3.1.14 идентификационные данные: Данные, которые однозначно характеризуют определенную сторону в определенном контексте.

3.1.15 идентификационный атрибут: Компонент идентификационных данных.

3.1.16 идентификация: Назначение уникального идентификатора или сравнение предъявляемого идентификатора с назначенными идентификаторами.

Примечание — В настоящем стандарте под идентификацией в основном понимается назначение идентификаторов с предшествующими регистрацией, включая сбор идентификационных данных, и подтверждением личности.

3.1.17 инфраструктура аутентификации: Совокупность сторон, которые реализуют сервисы единой аутентификации и авторизации, централизованного управления ресурсами пользователей, а также сторон, которые используют эти сервисы.

3.1.18 клиентская программа; КП: Программа, которая организует взаимодействие между пользователем, его токенами аутентификации, прикладной системой и службой идентификации.

3.1.19 код авторизации (OIDC): Ссылочный билет.

3.1.20 коммуникационная схема (OIDC): Схема взаимодействия сторон, в том числе описание узлов, последовательность пересылаемых сообщений и общее содержание сообщений.

3.1.21 конвертованные данные: Данные, защищенные на секретном ключе и сопровождаемые этим ключом, защищенным в свою очередь на открытом ключе получателя.

3.1.22 криптографический токен; КТ: Аппаратный или программный токен аутентификации, секретом которого является личный или секретный криптографический ключ, а аутентификатором — электронная цифровая подпись, имитовставка или их производные.

Примечание — На криптографическом токене могут размещаться идентификационные данные владельца.

3.1.23 одноразовый пароль; OTP (one-time password): Пароль, действие которого ограничено сеансом аутентификации или промежутком времени.

3.1.24 подписанные данные: Данные, сопровождаемые электронной цифровой подписью отправителя.

3.1.25 подтверждение личности: Проверка того, что пользователь с заявленными идентификационными данными действительно существует, данные касаются именно этого пользователя и однозначно его характеризуют.

3.1.26 пользователь: Физическое лицо, регистрируемое или зарегистрированное в инфраструктуре аутентификации для доступа к сервисам аутентификации и авторизации.

3.1.27 прикладная система; ПС: Сторона, которая использует сервисы аутентификации и авторизации, предоставляемые инфраструктурой аутентификации.

Примечание — Прикладная система инициирует аутентификацию пользователя и свою авторизацию на доступ к его ресурсам. Прикладная система имеет собственные ресурсы-услуги, которые могут быть предоставлены пользователям после их аутентификации.

3.1.28 регистрационный центр; РЦ: Сторона, которая проводит сбор идентификационных данных пользователя или проверяет и заверяет их для службы идентификации.

3.1.29 ресурс: Данные или сервис определенной стороны.

3.1.30 сеанс: Логическая связь между двумя сторонами, которая описывается идентификатором, параметрами защиты и другими согласованными между сторонами данными, которые могут быть использованы в нескольких соединениях.

3.1.31 секрет аутентификации: Секретные данные, которые содержатся в токене аутентификации и используются для построения аутентификаторов.

3.1.32 сервер ресурсов; СР: Сторона, которой пользователи делегировали управление своими ресурсами и которая открывает доступ к этим ресурсам другим сторонам после авторизации.

3.1.33 служба идентификации; СИ: Сторона, которая проводит идентификацию и аутентификацию пользователей и авторизует доступ к их ресурсам.

3.1.34 соединение: Непостоянный канал связи между двумя сторонами.

3.1.35 ссылочный билет: Вспомогательный одноразовый билет, который ссылается на другой билет или билеты.

3.1.36 статический пароль: Секрет аутентификации, который способен запомнить человек.

3.1.37 сторона: Активный элемент: лицо, устройство, процесс, сервер, центр, служба.

3.1.38 терминал: Сторона, которая представляет службу идентификации и организует по ее поручению аутентификацию пользователей или самостоятельно проводит аутентификацию.

3.1.39 токен аутентификации; ТА: Устройство или данные, которыми сторона владеет и которые использует для аутентификации.

3.1.40 удостоверение: Документ на физическом носителе, выпущенный доверенной стороной и содержащий идентификационные данные пользователя.

3.1.41 узел (OIDC): Конечная коммуникационная точка сети прикладного уровня, которая характеризуется интерфейсом, построенным на базе протокола HTTP, в том числе: уникальным сетевым именем, методом, принимаемыми параметрами, возвращаемыми значениями.

3.1.42 уровень гарантий аутентификации: Степень уверенности в том, что для пользователя, прошедшего аутентификацию относительно некоторых утверждений, утверждения действительно выполняются.

3.1.43 уровень гарантий идентификации: Степень уверенности в том, что зарегистрированный и идентифицированный пользователь действительно тот, за кого себя выдает.

3.1.44 уровень гарантий федерации: Степень защиты утверждений, распространяемых в федерации.

3.1.45 утверждение: Характеристика стороны или события, в том числе данные о пользователе или сведения о прохождении им аутентификации.

3.1.46 утверждение аутентификации: Сведения об успешной аутентификации пользователя.

3.1.47 фактор аутентификации: Одна из трех категорий токенов аутентификации как таковых или данных, нужных для активации токенов: «что я знаю», «что я имею», «кто я».

3.1.48 федерация (доверия): Совокупность сторон, связанных отношениями доверия, полного или частичного.

Примечание — В настоящем стандарте речь идет о федерациях, которые используют сервисы инфраструктуры аутентификации, входят в состав инфраструктуры или пересекаются с нею. Доверие в федерации основано на аутентификации сторон. Доверие может транслироваться в авторизацию.

3.1.49 центр федерации; ЦФ: Сторона, которая является гарантом отношений доверия в федерации.

3.1.50 энтропия: Степень неопределенности.

Примечание — Если имеется N вариантов выбора объекта и все эти варианты примерно равновероятны, то говорят, что объект содержит $\log_2 N$ битов энтропии.

3.2 В настоящем стандарте применяют следующие сокращения:

ИОК — инфраструктура открытых ключей;

СКЗИ — средство криптографической защиты информации;

ЭЦП — электронная цифровая подпись;

HTTP (hypertext transfer protocol) — протокол передачи гипертекста [1];

URI (uniform resource identifier) — унифицированный идентификатор ресурса [1].

В приложении А для некоторых терминов настоящего стандарта представлены англоязычные прототипы, которые используются в спецификации OIDC [2] и базовых для нее спецификациях [3], [4].

4 Общие положения

4.1 Назначение

Аутентификация, проверка подлинности, является одной из услуг доверия. Пользователь, прошедший аутентификацию, приобретает «цифровой образ» (digital identity). Образ представляет пользователя в информационной системе, является его виртуальным посредником при доступе к сервисам системы. Для сервисов, к которым пользователь обращается многократно, обычно гарантируется неизменность образа при повторных обращениях. Образ при этом становится устойчивым, его принимают другие стороны и отождествляют с владельцем.

Аутентификации предшествует идентификация. Речь идет о регистрации пользователя в системе с назначением ему уникального идентификатора. В процессе идентификации, как правило, подтверждается личность регистрируемого пользователя, сохраняются его идентификационные данные.

Аутентификация продолжается распространением утверждений аутентификации в пределах федерации, т. е. среди сторон системы, связанных отношениями доверия. Эти отношения позволяют организовать надежную передачу информации о подлинности пользователей, прошедших аутентификацию, и, таким образом, создавать новые отношения доверия. Распространяемые утверждения могут сопровождаться разрешениями аутентифицированной стороны на доступ к собственным ресурсам, в том числе идентификационным данным. Другими словами, аутентификация в федерации может продолжаться авторизацией.

Настоящий стандарт устанавливает правила построения инфраструктур аутентификации, составленных из трех описанных выше блоков: идентификация (см. раздел 5), аутентификация (см. раздел 6), федерация (см. раздел 7). Устанавливаемые правила соответствуют стандартам [5], [6], [7], [8], [9].

Дополнительно в разделе 8 представлена реализация элементов аутентификации и федерации на основе технологии OIDC [2], в свою очередь основанной на технологии OAuth 2.0 [3]. В разделе унифицируются интерфейсы веб-сервисов, предназначенных для развертывания в Интернете крупных открытых инфраструктур аутентификации. Технология OIDC детализируется в приложениях Б–Д. Приложения Б–Д являются обязательными только при условии использования технологии.

Настоящий стандарт ориентирован на построение инфраструктур, которые обладают следующими свойствами и характеристиками.

1 Централизованная идентификация. Пользователи, которые регистрируются для доступа к услуге аутентификации, идентифицируются одной или несколькими СИ. Небольшое число СИ может обслужить большое число пользователей.

2 Распределенное подтверждение личности. Подтверждение проводят РЦ, обслуживающие локальные (например, территориальные или ведомственные) группы регистрируемых пользователей. Подтверждение личности как услуга может быть сделано максимально доступным для пользователей.

3 Централизованное управление идентификационными данными. РЦ передает СИ результаты подтверждения личности и, в случае успеха, идентификационные данные проверенного пользователя. Идентификационные данные централизованно хранятся на серверах СИ, что снижает издержки и угрозы раскрытия.

4 Централизованная аутентификация. ПС, взаимодействующие с пользователями и заинтересованные в проверке их подлинности, получают утверждения аутентификации от СИ, на услуги которых они подписаны. ПС могут сосредоточиться на своих функциональных обязанностях, а не на непрофильной аутентификации.

5 Масштабируемая аутентификация. СИ может делегировать организацию аутентификации терминалам, выступающим в роли агентов СИ. Использование терминалов позволяет масштабировать нагрузку при доступе к услуге аутентификации, повышает степень проникновения услуги.

6 Централизованная авторизация. Вместе с утверждениями аутентификации ПС получают от СИ авторизационные разрешения на доступ к ресурсам аутентифицированных пользователей. Ресурсы размещаются централизованно на выделенных СР, напрямую взаимодействующих с СИ.

4.2 Уровни гарантий

В настоящем стандарте каждый из блоков «идентификация», «аутентификация», «федерация» удовлетворяет определенному уровню гарантий: базовому (1), среднему (2) или высокому (3). Уровень гарантий определяет степень уверенности в соблюдении процедур и правил, за которые отвечает блок. Уровни кратко охарактеризованы в таблице 1.

Уровень гарантий блока следует выбирать с учетом критичности последствий возможных ошибок (идентификации, аутентификации или федерации). Рекомендации по выбору уровня гарантий даны в приложении Е.

Уровни гарантий блоков могут отличаться друг от друга. Например, в информационной системе может применяться средний уровень гарантий идентификации (проверяются удостоверения регистрируемого пользователя), высокий уровень гарантий аутентификации (аутентификация с помощью персонального аппаратного КТ) и базовый уровень гарантий федерации (распространяются подписанные билеты аутентификации OIDC).

Таблица 1 — Уровни гарантий

Уровень	Краткое описание
Идентификация	
1	Без подтверждения личности. При регистрации пользователь может указать идентификационные атрибуты, корректность которых подтверждается только им самим
2	Подтверждение личности через контроль удостоверений. Регистрация проходит виртуально (в рамках видеосеанса) или в личном присутствии пользователя
3	Подтверждение личности через контроль удостоверений квалифицированным персоналом. Регистрация проходит в личном присутствии пользователя. Сбор биометрических данных
Аутентификация	
1	Однофакторная аутентификация
2	Многофакторная аутентификация
3	Многофакторная аутентификация с использованием аппаратного устройства и КТ
Федерация	
1	Распространяемые утверждения подписываются
2	Распространяемые утверждения подписываются и зашифровываются (конвертуются)
3	Распространяемые утверждения подписываются и зашифровываются. Аутентифицированный пользователь, субъект утверждений, должен доказать владение ключом, ссылка на который сопровождает утверждения

4.3 Пакеты

Соответствие тому или иному уровню гарантий реализуется через выполнение требований безопасности, определенных в настоящем стандарте.

Требования безопасности схожего назначения группируются в пакеты. Пакет — это также процедуры, процессы и элементы, которых касаются требования. Перечень пакетов для каждого из блоков представлен в таблице 2.

Таблица 2 — Пакеты

Блок	Пакет	Код	Ссылка
Идентификация	Регистрация пользователей	РП	5.1
	Подтверждение личности	ПЛ	5.2
Аутентификация	Выпуск токенов	ВТ	6.1
	Управление аттестатами	УА	6.2
	Протоколы аутентификации	ПА	6.3
Федерация	Управление федерацией	УФ	7.1
	Управление билетами	УБ	7.2
	Управление сеансами	УС	7.3

В преамбуле пакета определяется его назначение, дается обзор понятий, необходимых для формулировки требований пакета и правильной их интерпретации.

Требования внутри пакета нумеруются последовательно, начиная с единицы. Номер требования через точку присоединяется к коду пакета, в результате получается полное имя требования: РП.1, УБ.2 и т. д.

Для каждого требования в круглых скобках перечисляются уровни, на которых требование выдвигается, например: (3), (1–3), (2, 3).

4.4 Стороны

Определены следующие роли сторон инфраструктуры аутентификации.

1 СИ. Проводит идентификацию пользователя, передает идентификационные данные пользователя и другие его ресурсы СР. По запросу ПС проводит аутентификацию пользователя. По согласованию с пользователем авторизует доступ ПС к его ресурсам. Управляет сетью терминалов, которые участвуют в аутентификации.

2 Терминал. По поручению СИ организует аутентификацию пользователей или самостоятельно проводит аутентификацию и сообщает результат СИ.

3 РЦ. Является посредником при взаимодействии между пользователями и СИ во время регистрации: проводит сбор идентификационных данных пользователей и подтверждение их личности, передает идентификационные данные СИ.

4 СР. Хранит идентификационные данные пользователей, управляет другими их ресурсами. Может входить в состав СИ.

5 ЦФ. Управляет отношениями доверия в федерации, связанной с инфраструктурой аутентификации.

6 Пользователь. Регистрируется с помощью РЦ, передает идентификационные данные и другие ресурсы для размещения на СР. По запросу ПС проходит аутентификацию перед ПС. При аутентификации авторизует ПС на доступ к своим ресурсам.

7 ПС. Регистрируется в федерации для доступа к услуге аутентификации. Организует аутентификацию пользователя перед СИ. После авторизационного разрешения пользователя получает доступ к его ресурсам, размещенным на СР.

В настоящем стандарте ЦФ представляет ИОК, в которой стороны инфраструктуры аутентификации (возможно, за исключением пользователей) получают сертификаты открытых ключей. Проверяя сертификат открытого ключа стороны A и убеждаясь в знании ею соответствующего личного ключа, сторона B убеждается в подлинности A и возникают отношения доверия. Кроме этого, сертификаты могут использоваться для создания защищенных соединений между сторонами, например, соединений протокола TLS, установленного в СТБ 34.101.65. Наконец сертификаты могут использоваться для проверки подписи билетов, для их конвертования, в других целях.

Инфраструктуры аутентификации настоящего стандарта совместимы с ИОК, установленными в СТБ 34.101.78.

Функциональные возможности ЦФ могут выходить за рамки стандартных сервисов ИОК. Например, ЦФ может выпускать в обращение персональные аппаратные КТ пользователей. При выпуске на КТ записываются идентификационные данные, личный ключ и сертификат владельца. КТ могут использоваться при аутентификации.

4.5 Клиентская программа

Пользователь взаимодействует с ПС и СИ с помощью КП. КП, как правило, выполняется на персональном компьютере или мобильном устройстве пользователя. Это может быть браузер, отдельное приложение или связка браузера с приложением.

КП может обрабатывать критические данные (например, секреты аутентификации), а также открытые данные, целостность и подлинность которых определяют надежность взаимодействия с ПС и СИ (например, запросы аутентификации).

КП не может обеспечить полную защиту обрабатываемых данных. Защита организуется средствами системной среды, в том числе через настройки операционной системы. Основные задачи защиты: невозможность чтения критических областей памяти вредоносными программами; невозможность перехвата данных, передаваемых по каналам управления; защита канала «браузер — приложение»; невозможность подмены открытых данных. Организация защиты выходит за рамки настоящего стандарта.

КП может быть выполнена в виде СКЗИ в соответствии с требованиями СТБ 34.101.27.

В ходе взаимодействия КП – СИ выполняется аутентификация пользователя. При аутентификации КП использует ТА пользователя. При использовании в качестве ТА аппаратного КТ единственными критическими данными, которые обрабатывает КП, является пароль доступа к КТ. Поэтому КТ желательно использовать в тех ситуациях, когда КП пользователя выполняется в потенциально агрессивной среде, например, на незащищенном компьютере общего пользования.

4.6 Обзор взаимодействия

Взаимодействие сторон инфраструктуры представлено на рисунке 1.

Пользователь регистрируется в инфраструктуре с помощью РЦ. При регистрации проводится сбор идентификационных данных пользователя. Эти данные РЦ пересылает СИ. СИ передает их в управление СР вместе с другими ресурсами пользователя.

Обычно при регистрации подтверждается личность пользователя, в том числе проверяется корректность собранных идентификационных данных. Подтверждение личности основано на проверке удостоверений пользователя.

Подтверждение личности может не проводиться. Например, допускается, что пользователь указывает при регистрации самозаявленные (никем не подтвержденные) имя и фамилию. Регистрация без подтверждения личности проводится напрямую между пользователем и СИ.

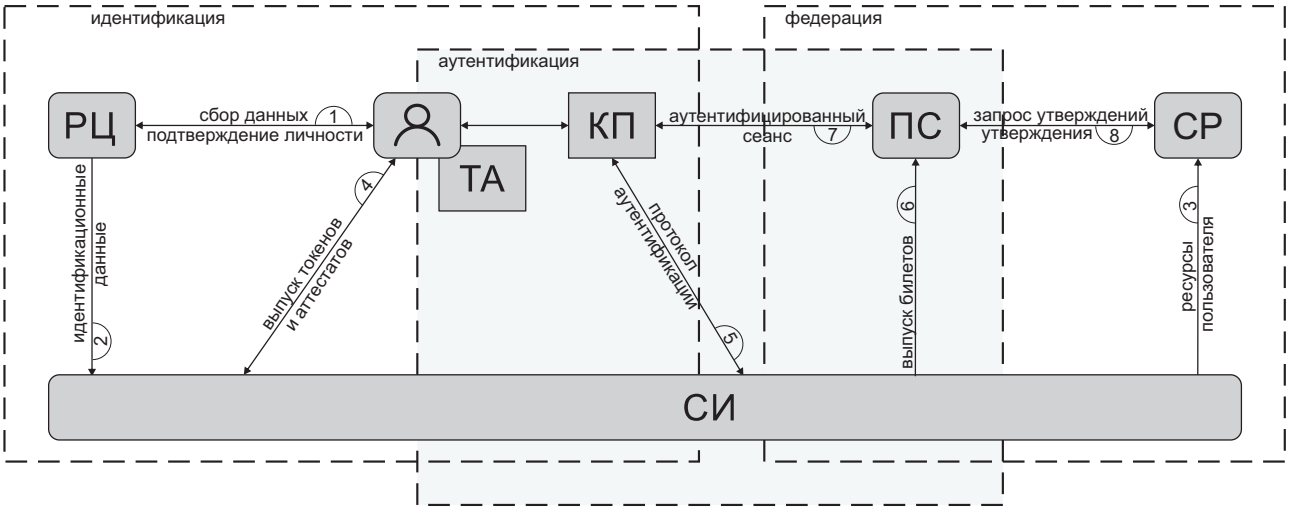

Рисунок 1 — Взаимодействие сторон инфраструктуры

В ходе регистрации пользователь и СИ согласуют перечень ТА, которые будут использоваться при аутентификации. СИ хранит информацию о связи между пользователями и его токенами в аттестатах.

СИ по запросу ПС, инициированному обращением пользователя, проводит аутентификацию пользователя. В ходе аутентификации пользователь доказывает владение одним или несколькими токенами, указанными в аттестате. В случае успеха СИ выдает ПС билеты: аутентификации (БА) и доступа (БД). БА подтверждает подлинность пользователя и может быть использован для открытия ему доступа к ресурсам ПС. БД авторизует ПС перед СР на доступ к ресурсам пользователя.

Кроме двух основных билетов, могут использоваться дополнительные. Например, билет обновления позволяет организовать перевыпуск других билетов без повторной аутентификации. Билет сеанса позволяет организовать аутентифицированный сеанс между пользователем и ПС.

Для доступа к услугам аутентификации и авторизации ПС предварительно регистрируется в СИ как член федерации.

5 Идентификация

5.1 Регистрация пользователей (РП)

5.1.1 Обзор

Пакет РП устанавливает требования по организации регистрации пользователей, сбору их идентификационных данных.

Регистрация может проходить тремя способами:

1 Удаленная регистрация. Выполняется по сетям связи без контроля пользователя со стороны операторов РЦ или СИ.

2 Виртуальная регистрация. Проходит в рамках видеосеанса между пользователем и оператором РЦ.

3 Личная регистрация. Проходит при непосредственном (физическом) взаимодействии пользователя и оператора РЦ.

РЦ формирует идентификационные данные пользователя или принимает данные, сформированные самим пользователем. Затем подтверждается личность пользователя (см. 5.2), в том числе проверяется корректность собранных данных. В случае успеха РЦ отправляет идентификационные данные СИ. СИ может передать их далее СР. При удаленной регистрации (она допускается только на уровне 1) сбор идентификационных данных может напрямую выполнять СИ.

Вместе с идентификационными данными РЦ может собирать биометрические. Биометрические данные позволяют удостовериться в совпадении пользователей при повторной регистрации или, наоборот, выявлять несовпадения. Кроме этого, биометрические данные выступают в роли свидетельства участия в регистрации и препятствуют отказу от регистрации. Собранные биометрические данные, как и идентификационные, отправляются СИ и могут далее пересылаться СР.

Еще одним инструментом неотказуемости является согласие пользователя на регистрацию в виде подписанного заявления.

5.1.2 Требования

Требование РП.1 (1). Должна проводиться удаленная, виртуальная или личная регистрация пользователей.

Требование РП.2 (2). Должна проводиться виртуальная или личная регистрация пользователей.

Требование РП.3 (3). Должна проводиться личная регистрация пользователей.

Требование РП.4 (1). Во время удаленной регистрации соединение между РЦ и пользователем должно быть защищено (с аутентификацией РЦ).

Требование РП.5 (1, 2). Во время виртуальной регистрации пользователь не должен покидать свое место регистрации (там, где установлена видеокамера). В регистрации со стороны РЦ должен лично участвовать оператор, и он также не должен покидать свое место регистрации. Оператор должен иметь возможность наблюдать все действия пользователя. Для проверки цифровых компонентов удостоверения место регистрации пользователя должно быть снабжено сканерами, карт-приемниками, другим необходимым оборудованием. Соединение между РЦ и пользователем должно быть защищено (с аутентификацией РЦ).

Требование РП.6 (1, 2). Операторы, которые проводят виртуальную регистрацию, должны проходить обучение, направленное на овладение навыками регистрации, на обнаружение мошенничества со стороны регистрируемых пользователей.

Требование РП.7 (3). Пользователь должен представить РЦ заявление о регистрации, подписанное собственноручно.

Требование РП.8 (2, 3). Во время регистрации пользователя должен проводиться сбор его идентификационных данных. Собираемые данные должны однозначно характеризовать пользователя в инфраструктуре. Должно быть получено явное согласие пользователя на сбор данных.

Примечание 1 — Обычный достаточный набор идентификационных данных — это полное имя пользователя, дата и место рождения.

Примечание 2 — При сборе идентификационных данных следует придерживаться следующих правил:
– не использовать идентификационные данные для проверки полномочий и статуса пользователя, для предоставления ему преференций;
– собирать только необходимые идентификационные данные;
– информировать пользователя о том, как собираемые данные будут использоваться;
– фиксировать претензии пользователей о недостатках при сборе данных.

Требование РП.9 (3). Вместе с идентификационными данными пользователя должны собираться биометрические.

Требование РП.10 (1–3). Идентификационные и биометрические данные должны передаваться между РЦ, СИ и СР по защищенным соединениям (с взаимной аутентификацией сторон).

Требование РП.11 (1–3). Должны быть разработаны и реализованы правила хранения идентификационных и биометрических данных у РЦ, СИ и СР. Доступ к хранимым данным должен быть ограничен. Доступ должны иметь только операторы и сервисы РЦ, СИ и СР, которым этот доступ необходим. Должны быть определены сроки хранения данных. При выводе РЦ, СИ и СР из обращения хранимые данные должны уничтожаться.

Требование РП.12 (2, 3). РЦ и СИ должны вести аудит событий регистрации. В записях аудита должны быть указаны представленные при регистрации удостоверения, виды собранных биометрических данных.

5.2 Подтверждение личности (ПЛ)

5.2.1 Обзор

Пакет РП устанавливает требования по подтверждению личности регистрируемых пользователей.

При подтверждении личности РЦ (через своего оператора) убеждается в том, что регистрируемый пользователь действительно существует, действительно характеризуется заявленными идентификационными данными и что эти данные корректны и подлинны.

Подтверждение личности может быть выполнено за рамками инфраструктуры аутентификации. Например, если аттестатом является сертификат открытого ключа, выпущенный в доверенной ИОК, то РЦ и СИ могут полагаться на подтверждение личности, выполненное при регистрации пользователя в ИОК. Требования по подтверждению в этом случае переносятся на внешние РЦ.

При подтверждении личности РЦ использует одно или несколько удостоверений пользователя. Удостоверение — это физический (бумага, пластик) документ, выпущенный доверенной стороной и содержащий идентификационные данные пользователя. При удаленной или виртуальной регистрации пользователь может предъявлять не само удостоверение, а его фотокопию.

Выделяются следующие классы удостоверений:

1 Слабое удостоверение. При выпуске удостоверения не проводилось подтверждение личности, но выпуск организован так, что с высокой достоверностью удостоверение действительно принадлежит предъявителю. Удостоверение содержит номер, который однозначно указывает на владельца или идентифицирует само удостоверение. Примеры слабых удостоверений — свидетельство о рождении, карточка сотрудника компании, диплом об образовании, кредитная карта.

2 Адекватное удостоверение. При выпуске удостоверения проводилось подтверждение личности. Выпуск организован так, что с высокой достоверностью удостоверение действительно принадлежит предъявителю. Удостоверение содержит либо номер, который однозначно указывает на владельца, либо фотографию, либо биометрические данные владельца. Если удостоверение содержит цифровые элементы, то их целостность и подлинность контролируются криптографическими методами. Если используются физические элементы защиты (водяные знаки), то их воспроизводство или обход требуют от противника применения дорогостоящих или недоступных технологий. Примеры адекватных удостоверений — студенческий билет, военный билет.

3 Сильное удостоверение. Адекватное удостоверение, которое выпущено государственным органом с соблюдением утвержденных процедур. Удостоверение обязательно содержит и номер, и фотографию, а возможно, и биометрические данные владельца. В удостоверении указано официальное имя владельца. Примеры сильных удостоверений — паспорт, водительские права, ID-карта.

Удостоверение (включая представленные в нем идентификационные данные) проверяется одним из четырех способов:

а) обращение к стороне, выдавшей удостоверение (прямой запрос, автоматизированные информационные системы);

б) контроль элементов удостоверения с помощью технологического оборудования и (или) программного обеспечения (фотоспектральные сканеры, компьютерное зрение);

в) ручная проверка обученными операторами (визуальный и тактильный осмотр элементов защиты, сличение фотографии);

г) проверка криптографических контрольных характеристик (проверка электронной цифровой подписи).

При выборе способа проверки следует учитывать, что при удаленной или виртуальной регистрации применение способов б)–г) затруднено или даже невозможно.

РЦ может проверять не только содержимое удостоверения, но и факт владения им. Например, для подтверждения владения кредитной картой пользователю может быть предложено оплатить с ее помощью услугу регистрации и получить квитанцию с секретным кодом проверки. Этот код пользователь предъявляет при регистрации. Похожим

образом РЦ проверяет владение физическим (почтовым) и электронными (номер сотового телефона, электронная почта) адресами.

Для повышения гарантий подтверждения личности РЦ дополнительно к удостоверениям может использовать записи актов гражданского состояния, данные геолокации, характеристики используемых пользователем устройств, физические особенности пользователя и др.

5.2.2 Требования

Требование ПЛ.1 (2, 3). Пользователь должен предъявить РЦ удостоверения, в которых представлены все его идентификационные данные, подлежащие сбору в ходе регистрации. РЦ должен проверить удостоверения и, в случае успеха, зафиксировать их реквизиты вместе с собираемыми идентификационными данными.

Требование ПЛ.2 (2). Пользователь должен предъявить, как минимум, либо одно сильное удостоверение, либо два адекватных.

Требование ПЛ.3 (3). Пользователь должен предъявить, как минимум, одно сильное удостоверение.

Требование ПЛ.4 (1–3). Проверка удостоверения должна покрывать:

– атрибуты удостоверения как физического документа (если вместо него не предъявлена фотокопия);

– представленные в удостоверении идентификационные данные владельца;

– реквизиты удостоверения;

– срок действия удостоверения.

Требование ПЛ.5 (1–3). Слабое удостоверение должно проверяться способом а).

Требование ПЛ.6 (1–3). Адекватное удостоверение должно проверяться одним или несколькими способами из списка а)–г).

Требование ПЛ.7 (1, 2). Сильное удостоверение должно проверяться одним или несколькими способами из списка а) – г).

Требование ПЛ.8 (3). Сильное удостоверение должно проверяться способом а), а также одним или несколькими способами из списка б)–г).

Требование ПЛ.9 (1–3). Если физический или электронный адрес пользователя не проверен как элемент удостоверения, то РЦ должен отправить по этому адресу секретный код и этот код пользователь должен предъявить для завершения регистрации. Код для проверки электронного адреса должен действовать не более суток. Код для проверки физического адреса должен действовать не более 21 дня. Код должен содержать не менее 32 бит энтропии.

6 Аутентификация

6.1 Выпуск токенов (ВТ)

6.1.1 Обзор

Пакет ВТ устанавливает требования по выбору типов токенов и их комбинаций, надежности токенов для соответствия тому или иному уровню гарантий аутентификации.

Все рассматриваемые в настоящем стандарте токены содержат секреты аутентификации: статический пароль, личный или секретный ключ, другое. Статический пароль способен запомнить человек, и поэтому он относится к фактору «что я знаю». Секретные и личные ключи пользователь запомнить и обработать не может, они не существуют автономно, а являются частью программного или аппаратного токена. Этот токен относится к фактору «чем я владею». Биометрические токены, основанные на биометрических данных и относящиеся к фактору «кто я», не содержат приемлемых секретов аутентификации. В

настоящем стандарте биометрические токены могут использоваться только для активации аппаратных.

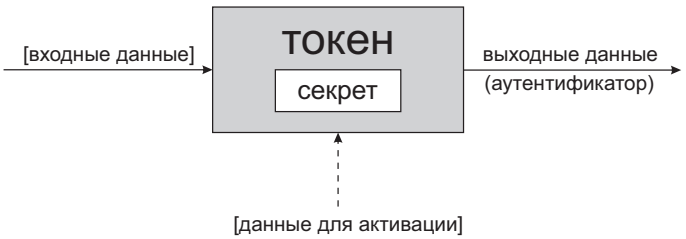

Рисунок 2 — Токен аутентификации

Токены используются для построения аутентификаторов (см. рисунок 2). Аутентификатор демонстрирует владение токеном аутентификации или его знание и, таким образом, подтверждает подлинность владельца.

Токены делятся на два класса: однофакторные и многофакторные. Однофакторные токены выдают аутентификатор сразу, многофакторные — только после активации. Для активации нужно задействовать дополнительный фактор, как правило, «что я знаю» (PIN-код) или «кто я» (отпечаток пальца).

Определены следующие типы ТА:

1 Статический пароль. Секретная последовательность символов, цифр, графических элементов и пр., которую способен запомнить человек. Разделяется между пользователем и СИ. Либо выбирается пользователем, либо генерируется СИ. Относится к фактору «что я знаю». Аутентификатором обычно является сам пароль.

2 Карта секретов. Физическое или электронное устройство, которое хранит набор секретов, разделяемых между пользователем и СИ. Относится к фактору «чем я владею». Аутентификатором является секрет по запрошенному СИ или выбранному пользователем номеру.

3 Сетевой токен. Устройство, связанное с СИ дополнительным соединением. Сетевой адрес устройства зафиксирован в момент регистрации. Относится к фактору «чем я владею». Аутентификатором является одноразовый секрет, например пароль, который СИ отправляет по дополнительному соединению и предлагает ввести по основному или, наоборот, отправляет по основному соединению и предлагает ввести по дополнительному. Как правило, дополнительное соединение — канал GSM, сетевой адрес — номер сотового телефона пользователя.

4 Однофакторный OTP-токен. Аппаратное устройство или программа, которые генерируют одноразовые пароли. Для генерации используется секретный ключ, размещенный в памяти устройства или в файле, сопровождающем программу. Относится к фактору «чем я владею». Активация устройства не требуется.

5 Однофакторный аппаратный КТ. Аппаратный криптографический токен, который не требуется активировать. Относится к фактору «чем я владею».

6 Программный КТ. Программный КТ, включающий специальный файл, в котором хранится защищенный секрет аутентификации. Ключ защиты строится по паролю. Относится к фактору «чем я владею». Пароль является дополнительным фактором «что я знаю».

7 Многофакторный OTP-токен. Аппаратное устройство или программа, которые генерируют одноразовые пароли. Для генерации используется секретный ключ, размещенный в памяти устройства, или в файле, сопровождающем программу. Относится к фактору «чем я владею». Для активации устройства нужен дополнительный фактор «что я знаю»

или «кто я». Ключ хранится в файле в защищенном виде. Ключ защиты файла строится по паролю. Пароль является дополнительным фактором «что я знаю».

8 Многофакторный аппаратный КТ. Аппаратный криптографический токен. Относится к фактору «чем я владею». Для активации токена нужен дополнительный фактор «что я знаю» и (или) «кто я».

Пользователь может использовать несколько независимых ТА, повышая при этом надежность аутентификации.

6.1.2 Требования

Требование ВТ.1 (1–3). Физический токен (карта кодов, сетевой токен, аппаратный OTP-токен или КТ) должен сопровождаться инструкциями владельцу. Инструкции должны содержать сведения об обращении с токеном и о действиях в случае его пропажи или кражи.

Требование ВТ.2 (1). Статический пароль должен содержать не менее 14 бит энтропии. Текстовый пароль, выбираемый пользователем, должен состоять из не менее чем 6 символов в алфавите из 90 и более символов. Секретный PIN-код должен состоять из не менее чем 4 цифр и выбираться СИ случайным образом.

Требование ВТ.3 (2, 3). Статический пароль должен содержать не менее 20 бит энтропии. Текстовый пароль, выбираемый пользователем, должен состоять из не менее чем 8 символов в алфавите из 90 и более символов. Секретный PIN-код должен состоять из не менее чем 6 цифр и выбираться СИ случайным образом.

Требование ВТ.4 (3). При регистрации статического пароля, выбранного пользователем, СИ должна проверить его отсутствие в списке паролей, которые часто используются, ожидаемы или скомпрометированы.

Примечание — Список запрещенных паролей может включать словари распространенных паролей, базы данных скомпрометированных паролей, тривиальные пароли из повторяющихся или соседних символов, а также пароли, построенные по имени пользователя или интернет-сервера.

Требование ВТ.5 (1–3). КП не должна хранить подсказки, которые помогают пользователю вспомнить забытый статический пароль.

Примечание — При этом КП может отображать подсказки, переданные от СИ или терминала по защищенному соединению.

Требование ВТ.6 (1–3). При вводе пароля его символы (по отдельности или все вместе) могут отображаться только на короткое время, нужное пользователю для проверки корректности ввода.

Требование ВТ.7 (1–3). Карта секретов должна генерироваться СИ по секретному ключу, который содержит не менее 128 бит энтропии. Аутентификатор карты должен содержать не менее 20 бит энтропии. Аутентификатор не должен использоваться дважды.

Требование ВТ.8 (1–3). Сетевой токен должен иметь уникальный сетевой адрес, сообщения на который может принимать только сам токен.

Примечание — В частности, сетевой токен не может использовать адрес электронной почты, поскольку сообщения на этот адрес могут принимать почтовые клиенты на разных устройствах. Мессенджер, который допускает установку на нескольких устройствах, также не может использоваться для приема сообщений.

Требование ВТ.9 (1–3). Сетевой токен должен взаимодействовать с СИ по дополнительному соединению, отличному от основного соединения КП – СИ. Аутентификатор, который СИ передает сетевому токену по дополнительному соединению, должен быть зашифрован. Перед отправкой аутентификатора сетевой токен должен быть аутентифи-

цирован перед СИ. Для шифрования и аутентификации должны использоваться либо СКЗИ, удовлетворяющие требованиям СТБ 34.101.27, либо стандартные криптографические механизмы сетей сотовой связи.

Примечание — В настоящем стандарте не рассматривается сценарий, когда токен получает один и тот же аутентификатор по основному и дополнительному соединениям, и пользователь подтверждает совпадение аутентификаторов по дополнительному (защищенному) соединению.

Требование BT.10 (1–3). Аутентификатор сетевого токена должен содержать не менее 20 бит энтропии. Аутентификатор не должен действовать более 5 мин. СИ должна принимать аутентификатор лишь однажды в течение периода действия.

Требование BT.11 (1–3). Владелец сетевого токена должен быть проинструктирован о необходимости настройки токена так, чтобы в неактивном (заблокированном) состоянии токен не отображал аутентификаторы.

Примечание — Если в качестве токена выступает смартфон, то его настройка может состоять в подавлении отображения сообщений с аутентификаторами на экране блокировки. При этом уведомления о приеме сообщений можно не блокировать.

Требование BT.12 (1–3). Для генерации одноразовых паролей в OTP-токенах должны использоваться механизмы HOTP или TOTP, определенные в СТБ 34.101.47. В механизмах должен использоваться секретный ключ, который содержит не менее 128 бит энтропии. При использовании механизма TOTP пароль должен меняться не реже одного раза в 2 мин. Пароль должен состоять из не менее чем 6 десятичных символов.

Требование BT.13 (1–3). OTP-токен должен удовлетворять требованиям СТБ 34.101.27: программный токен — требованиям уровня 1 или 2, аппаратный токен — требованиям уровня 3 или 4.

Требование BT.14 (1–3). КТ должен удовлетворять требованиям СТБ 34.101.27: программный токен — требованиям уровня 1 или 2, аппаратный токен — требованиям уровня 3 или 4. Секрет аутентификации должен содержать не менее 128 бит энтропии, аутентификатор — не менее 64 бит.

Требование BT.15 (1–3). Однофакторные аппаратные OTP-токен и КТ должны выдавать аутентификаторы только после физического воздействия (например, нажатия кнопки).

Требование BT.16 (1–3). Программный КТ должен активироваться статическим текстовым паролем, удовлетворяющим требованию BT.3. По этому паролю должен строиться ключ защиты секрета аутентификации. Должен использоваться алгоритм PBKDF2, определенный в СТБ 34.101.45, или схожий криптографический механизм. В PBKDF2 число итераций должно быть не меньше 10 000, битовая длина синхропосылки — не меньше 64.

Требование BT.17 (1–3). Многофакторные аппаратные OTP-токен и КТ должны активироваться либо статическим паролем в соответствии с требованиями СТБ 34.101.27 (пакет ИА), либо биометрическим токеном. Биометрическая аутентификация должна удовлетворять следующим условиям:

– доля ошибочных решений о соответствии биометрического образца биометрическому эталону не превышает $1/1\,000$;

– допускается не более 5 ошибок аутентификации подряд. При достижении порога числа неверных попыток должна выполняться либо задержка на 30 с, либо переход на другой фактор аутентификации. Задержка должна увеличиваться в 2 раза с каждой новой ошибкой аутентификации.

Требование ВТ.18 (1). Должен использоваться токен одного из допустимых типов: статический пароль, карта секретов, сетевой токен, OTP-токен (однофакторный или многофакторный), КТ (однофакторный или многофакторный, программный или аппаратный).

Требование ВТ.19 (2). Должен использоваться многофакторный OTP-токен, программный КТ, многофакторный аппаратный КТ или комбинация статического пароля с одним из следующих токенов: карта секретов, сетевой токен, однофакторный OTP-токен, однофакторный аппаратный КТ. При повторной аутентификации в рамках аутентифицированного сеанса с СИ достаточно дополнительно к БС использовать статический пароль.

Примечание — БС выступает в роли токена фактора «чем я владею».

Требование ВТ.20 (3). Должен использоваться многофакторный аппаратный КТ или одна из следующих комбинаций токенов:

– статический пароль и однофакторный аппаратный КТ;

– многофакторный OTP-токен и однофакторный аппаратный КТ;

– однофакторный аппаратный КТ и программный КТ;

– аппаратный OTP-токен (однофакторный или многофакторный) и программный КТ.

6.2 Управление аттестатами (УА)

6.2.1 Обзор

Пакет УА устанавливает требования по управлению аттестатами. Требования касаются вопросов выпуска аттестатов в процессе регистрации ТА, хранения аттестатов, перевыпуска аттестатов и токенов, их отзыва и вывода из обращения.

Аттестат представляет собой учетную запись пользователя, которая связывает его идентификатор, другие идентификационные данные с зарегистрированными ТА. Пользователь регистрирует либо собственный токен (например, статический пароль), либо токен, выпущенный СИ или другой стороной. Посредником при регистрации может быть РЦ. Рекомендуется регистрировать несколько токенов, чтобы можно было сохранить доступ к учетной записи при утрате одного из них.

Аттестаты бывают двух типов: открытые, которые можно раскрывать, и секретные, которые раскрывать нельзя. Примером открытого аттестата является сертификат открытого ключа, примером секретного — структура данных, которая состоит из идентификатора (логин) пользователя и хэш-значения его пароля или секретного ключа его OTP-токена. СИ обязательно хранит секретные аттестаты и, возможно, открытые.

Аттестат с хэш-значением пароля классифицируется как секретный, потому что хэш-значение содержит информацию о пароле и дает возможность его определить через перебор. Для усложнения перебора используются два дополнительных механизма. Во-первых, вместе с паролем хэшируется волатильная синхропосылка, часто называемая «солью». Зависимость хэш-значений от синхропосылок препятствует упрощению перебора за счет предварительных вычислений. Во-вторых, алгоритм хэширования искусственно усложняется. Например, вместо одной выполняется несколько итераций хэширования, или в процессе хэширования используется память нерационально большого объема. Большая вычислительная сложность алгоритма хэширования навязывает высокую сложность перебора. В определенных случаях хэш-значения паролей могут использоваться как ключи, и тогда вместо алгоритмов хэширования речь идет об алгоритмах построения ключа. Алгоритм PBKDF2, определенный в СТБ 34.101.45, является примером алгоритма последнего типа. Сложность PBKDF2 определяется числом внутренних итераций, которое может регулироваться.

СИ может перевыпускать токены и соответствующие аттестаты. Обычной причиной перевыпуска является истечение срока действия. Перевыпуск может быть также инициирован запросом пользователя. При перевыпуске может потребоваться перерегистрация пользователя с участием РЦ. СИ может предусматривать переходный период, когда действуют и старые аттестаты, и новые.

СИ отвечает за отзыв аттестатов и токенов, за вывод их из эксплуатации по окончании жизненного цикла. СИ очищает записи в базе данных секретных аттестатов и поддерживает список отзыва открытых аттестатов.

СИ ведет аудит событий, связанных с управлением аттестатами и токенами. События касаются перечисленных выше процессов.

6.2.2 Требования

Требование УА.1 (1–3). Доступ к секретным аттестатам должен быть ограничен. Доступ должны иметь только администраторы и сервисы СИ, которым этот доступ необходим.

Требование УА.2 (1, 2). Секреты аутентификации в составе аттестатов не должны храниться в открытом виде. Для защиты секретов должны использоваться СКЗИ, удовлетворяющие требованиям СТБ 34.101.27 (уровни 2–4). Статический пароль либо должен защищаться с помощью СКЗИ, либо вместо него должны храниться хэш-значение или ключ, полученные с помощью алгоритма контролируемо большой вычислительной сложности. В алгоритме должны использоваться синхропосылки, битовая длина которых не меньше 64. Если используется алгоритм PBKDF2, то число итераций в нем должно быть не меньше 10 000.

Требование УА.3 (3). Секреты аутентификации в составе аттестатов не должны храниться в открытом виде. Для защиты секретов должны использоваться СКЗИ, удовлетворяющие требованиям СТБ 34.101.27 (уровни 2–4).

Требование УА.4 (2, 3). СИ должна разработать и ввести политику перевыпуска аттестатов и токенов. Для перевыпуска действующего токена пользователь должен подтвердить владение им. По истечении срока действия использование старых токенов и аттестатов должно быть запрещено. Новые статические пароли должны отличаться от старых. Соединение, которое используется при перевыпуске, должно быть защищено (с аутентификацией СИ).

Требование УА.5 (2, 3). СИ должна приостанавливать действие токенов немедленно после уведомления от владельца о компрометации (пропаже или краже). СИ должна регистрировать уведомления и расследовать их. Если в результате расследования уведомления выясняется, что оно является ложным, то действие токенов должно быть возобновлено.

Примечание — При приеме уведомления СИ следует провести аутентификацию пользователя с помощью сохранившихся (нескомпрометированных) ТА, согласованных при регистрации. Уровень гарантий аутентификации при приеме уведомления может быть ниже, чем в обычной ситуации.

Требование УА.6 (2). СИ должна аннулировать токены и аттестаты в течение 72 ч после уведомления о том, что аттестат перестал действовать или токен скомпрометирован (при условии справедливости уведомления).

Требование УА.7 (3). СИ должна аннулировать токены и аттестаты в течение 24 ч после уведомления о том, что аттестат перестал действовать или токен скомпрометирован (при условии справедливости уведомления).

Требование УА.8 (2, 3). СИ должна вести аудит событий, связанных с регистрацией, использованием, перевыпуском, выводом из эксплуатации токенов и аттестатов.

Требование УА.9 (2). СИ должна хранить записи аудита в течение не менее 7 лет и 6 месяцев после вывода аттестата из эксплуатации.

Требование УА.10 (3). СИ должна хранить записи аудита в течение не менее 10 лет и 6 месяцев после вывода аттестата из эксплуатации.

6.3 Протоколы аутентификации (ПА)

6.3.1 Обзор

Пакет ПА устанавливает требования к протоколам аутентификации, которые выполняются между пользователем и СИ и подтверждают владение пользователем зарегистрированными ТА.

Технически, в ходе выполнения протокола пользователь по запросу СИ активирует токен и предъявляет СИ аутентификатор, который доказывают владение им. При формировании аутентификатора может учитываться запрос СИ. СИ проверяет аутентификатор, используя выпущенный при регистрации токена аттестат. В сеансе протокола может использоваться не один, а несколько токенов и, соответственно, аутентификаторов.

При выборе или проектировании протокола аутентификации следует учитывать возможные атаки. Некоторые из них описаны в разделе Ж.3.

Одним из инструментов защиты от атак является контроль числа попыток аутентификации, которое может выполнить противник в течение определенного времени. Для организации контроля СИ может блокировать аутентификацию на определенное время после определенного числа неверных попыток. Интервал блокировки может быть постоянным или растущим вплоть до удачной попытки.

СИ может использовать следующие дополнительные методы контроля числа попыток аутентификации:

1 После очередной неверной попытки пользователь должен проходить антибот-тест вплоть до верной попытки. Антибот-тест позволяет отличить человека от программы (бота). В англоязычной литературе тест известен под аббревиатурой CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart).

2 Сетевые адреса, с которых предпринимается много неверных попыток аутентификации, блокируются. Пользователю разрешается проходить аутентификацию только с определенных адресов.

3 Запоминаются шаблоны поведения пользователя или его операционная среда. Пресекаются попытки, не соответствующие шаблонам или среде.

Протокол со стороны СИ может выполнять терминал. Эта ситуация во всех требованиях пакета подразумевается, хотя оговаривается явно только в требованиях ПА.6, ПА.7.

6.3.2 Требования

Требование ПА.1 (1–3). Если энтропия аутентификатора меньше 64 бит, то СИ должна организовать защиту от его подбора противником. Защита должна быть построена так, чтобы за 30 суток противник не мог выполнить более 100 попыток аутентификации.

Требование ПА.2 (1–3). Обмен данными протокола, от конфиденциальности, целостности и подлинности которых зависят гарантии аутентификации, должен выполняться по защищенному соединению (с аутентификацией СИ).

Примечание — Соединение может создаваться в ходе выполнения протокола, и тогда первые его сообщения, отвечающие за создание соединения или обмен открытыми данными, могут быть не защищены или для них может не обеспечиваться конфиденциальность.

Требование ПА.3 (3). Открытые данные, по которым можно получить информацию об идентификационных атрибутах пользователя, должны передаваться по защищенному соединению.

Требование ПА.4 (2, 3). При создании защищенного соединения должны использоваться волатильные данные, которые делают сообщения сеанса протокола контекстно зависимыми и препятствуют принятию сообщений одного сеанса в другом.

Требование ПА.5 (2, 3). Не должны использоваться протоколы аутентификации, стойкость которых снижается при определенных настройках, согласуемых между пользователем и СИ в начале выполнения протокола.

Требование ПА.6 (1–3). Если аутентификацию по поручению СИ выполняет внешний терминал, то соединение между СИ и терминалом должно быть защищено (с взаимной аутентификацией сторон).

Требование ПА.7 (1–3). По данным, которые СИ передает внешнему терминалу, должно быть вычислительно трудно определить секрет аутентификации пользователя.

Примечание — Терминалу может передаваться сертификат открытого ключа пользователя, и тогда в ходе аутентификации терминал проверяет, что пользователь знает соответствующий личный ключ. Терминалам не должны передаваться секретные аттестаты с хэш-значениями паролей, ключами OTP-токенов, картами секретов или ключами для их генерации.

7 Федерация

7.1 Управление федерацией (УФ)

7.1.1 Обзор

Пакет УФ устанавливает требования по взаимодействию сторон федерации, в которой СИ распространяет утверждения аутентификации.

Гарантом отношений доверия в федерации является ЦФ. Отношения доверия строятся на сертификатах открытых ключей, которые получают СИ, ПС и другие стороны федерации. Сертификаты выпускаются в ИОК, которую представляет ЦФ.

Для получения услуг аутентификации ПС регистрируется перед СИ. ПС пересылает СИ данные, необходимые для оказания услуги: имя, описание, сетевой адрес, другое. Стороны при необходимости обмениваются данными, которые позволят им проверять подлинность друг друга и устанавливать защищенные соединения. Регистрация может быть автоматизирована и выполняться через специальный сервис, реализуемый СИ. По окончании регистрации СИ назначает ПС уникальный идентификатор. Он будет указываться в БА, выдаваемых по запросам данной ПС.

7.1.2 Требования

Требование УФ.1 (1–3). ЦФ должен представлять СИ, СР и ПС сервисы управления сертификатами открытых ключей: выпуска, отзыва, проверки статуса.

Требование УФ.2 (1–3). СИ при присоединении к федерации должна согласовать с ЦФ уровни гарантий идентификации, аутентификации и федерации, которые СИ планирует поддерживать.

Требование УФ.3 (1–3). ЦФ должен контролировать:

– соблюдение сторонами инфраструктуры аутентификации правил (политик) федерации;

– формат билетов, которые выпускает СИ;

– правила использования идентификационных данных пользователей, которым следует ПС.

Требование УФ.4 (1–3). Данные, которыми обмениваются ПС и СИ в процессе регистрации, должны передаваться по защищенному соединению (с аутентификацией обеих сторон).

Требование УФ.5 (1–3). СИ должна согласовывать с пользователем выпуск билетов для ПС, которая запросила аутентификацию. Пользователю должна быть предоставлена возможность автоматического согласования выпуска по запросам определенной ПС. Пользователь должен иметь возможность отменить автоматическое согласование.

7.2 Управление билетами (УБ)

7.2.1 Обзор

Пакет УБ устанавливает требования к билетам, которые выпускаются СИ и распространяются в федерации, в том числе требования по содержанию билетов, их защите, логике их обработки.

СИ по результатам успешной аутентификации пользователя выпускает БА. Этот билет содержит утверждения о событии аутентификации — идентификаторы пользователя, СИ и целевой ПС, время выпуска, срок действия и др. Идентификатор пользователя является не только утверждением аутентификации, но и утверждением о самом пользователе. Билет может содержать и другие утверждения о пользователе.

БА используется для доступа к ресурсам ПС со стороны пользователя: ПС открывает доступ только после проверки корректности билета. Кроме этого, БА является квитанцией аутентификации, может предъявляться другим сторонам и таким образом служить основой сервисов одного окна (single sign-on). Наконец пользователь может обменять БА на БС и поддерживать с помощью последнего аутентифицированный сеанс с ПС или СИ.

Дополнительно к билету аутентификации СИ выпускает БД и БО. БД позволяет получить ПС утверждения о пользователе не напрямую, а косвенно, через СР. В тех случаях, когда БД имеет небольшой срок действия, БО позволяет перевыпустить БД без повторной аутентификации.

СИ и другие стороны могут выпускать другие билеты. Концептуально билет — это информация о пользователе, ссылка на ресурсы пользователя или на другие билеты. Так, БА — это информация об аутентификации пользователя, БД — ссылка на ресурсы, БО — ссылка на БД, БС — ссылка на сеанс пользователя. При таком толковании взаимодействие сторон инфраструктуры аутентификации можно представить себе как обмен одних билетов на другие. Обмен начинается с аутентификаторов и заканчивается утверждениями.

Билеты делятся на два класса: именные и на предъявителя. Именной билет содержит идентификационные данные владельца. При необходимости владелец может доказать, что билет принадлежит ему. Билет на предъявителя является безличным. Решение о корректности такого билета принимается в момент предъявления. Воспользоваться билетом может любой предъявитель, и поэтому билет следует хранить в секрете.

Билеты, которые СИ выпускает по результатам успешной аутентификации пользователя (БА, БД, БО), могут передаваться ПС двумя способами:

– через КП пользователя (длинная передача);
– напрямую (короткая передача).

В последнем случае ПС предварительно получает от СИ специальный ссылочный билет (код авторизации в терминологии OIDC). ПС получает ссылочный билет через КП пользователя и пересылает его СИ. СИ в ответ отправляет ПС требуемые билеты.

При длинной передаче билеты передаются по соединениям СИ — КП и КП — ПС через потенциально небезопасную среду эксплуатации КП. При короткой передаче используется

только прямое соединение СИ — ПС. Короткая передача надежнее. С другой стороны, длинная передача не нуждается в дополнительном ссылочном билете, она эффективнее.

БД, БО и ссылки на них, которые выпускает СИ после предъявления корректного аутентификатора пользователя, выступают в роли вторичных аутентификаторов. Вторичный аутентификатор лишь косвенно свидетельствует о подлинности пользователя, поскольку ПС не может связать аутентификатор с пользователем. Вторичный аутентификатор, как правило, является секретным билетом на предъявителя.

7.2.2 Требования

Требование УБ.1 (1–3). БА должен содержать:
- идентификатор пользователя, который прошел аутентификацию;
- идентификатор ПС, которая запрашивала аутентификацию;
- уникальный идентификатор билета;
- время выпуска билета;
- время окончания действия билета;
- время первичной аутентификации (если доступно);
- подпись содержимого билета, выработанная СИ.

Подпись билета должна сопровождаться сертификатом открытого ключа СИ или ссылкой на сертификат.

Примечание 1 — Билет может дополнительно содержать: уровень гарантий аутентификации; сведения о выполненном протоколе аутентификации; дополнительные сведения о пользователе.

Примечание 2 — В некоторых случаях по соображениям приватности требуется, чтобы различные ПС получали разные идентификаторы одного и то же пользователя. Такие контекстно зависимые идентификаторы (псевдонимы) СИ может генерировать с помощью алгоритмов имитозащиты, используя свой секретный ключ и обрабатывая на нем первоначальный идентификатор пользователя и идентификатор целевой ПС.

Требование УБ.2 (2, 3). БА должен быть конвертован на открытом ключе ПС.

Требование УБ.3 (3). БА должен содержать открытый ключ пользователя, прошедшего аутентификацию, или ссылку на открытый ключ. Пользователь должен доказать владение личным ключом, соответствующим открытому, при предъявлении билета ПС.

Примечание 3 — Открытый ключ пользователя может распространяться в форме сертификата, зарегистрированного в качестве открытого аттестата. При этом соответствующий личный ключ размещается на КТ, а при аутентификации пользователь доказывает владение токеном и, как следствие, личным ключом. Возможны другие сценарии. Например, БА может ссылаться на одноразовые открытый и личный ключи, которые формируются непосредственно в ходе протокола аутентификации, или на долговременные ключи, которые не используются для аутентификации.

Примечание 4 — Для доказательства владения личным ключом при предъявлении БА пользователь может подписать билет или подпись СИ, вложенную в него. Если личный ключ используется в алгоритмах ЭЦП за пределами инфраструктуры аутентификации, то предварительно следует проверить подпись СИ. Это защитит пользователя от подписи сообщений, навязываемых под видом БА или его частей.

Требование УБ.4 (1–3). ПС не должна принимать БА, подпись которого некорректна. ПС не должна принимать БА, если сертификат открытого ключа подписавшей билет СИ недействителен. ПС не должна принимать БА, предназначенные не ей. ПС не должна принимать БА, время окончания действия которого раньше текущего момента времени или момента начала действия сертификата СИ. ПС не должна принимать БА,

время выпуска которого позже текущего момента времени или момента окончания действия сертификата СИ.

Требование УБ.5 (1–3). БД, БО и ссылочные билеты должны передаваться между сторонами федерации по защищенным соединениям.

Требование УБ.6 (1–3). БД и БО должны содержать не менее 128 бит энтропии. Ссылочный билет должен содержать не менее 64 бит энтропии и использоваться однократно.

Требование УБ.7 (3). Должна использоваться прямая передача БД и БО от СИ к ПС.

Требование УБ.8 (1–3). СИ и СР, которым ПС предъявляет БД и БО, должны связывать билеты с ответами. Связывание должно быть реализовано с помощью ЭЦП (ответ, включающий ссылку на билет, подписывается) или через передачу по защищенному соединению (с аутентификацией отправителя).

Требование УБ.9 (3). СР должен подписывать утверждения, которые формирует в ответ на БД.

7.3 Управление сеансами (УС)

7.3.1 Обзор

Пакет УС устанавливает требования по управлению аутентифицированными сеансами пользователя с ПС и СИ.

ПС открывает аутентифицированный сеанс после получения БА. Именно в этом сеансе пользователь выполняет операции, ПС оказывает ему цифровые услуги. Неформально речь идет о работе пользователя в личном кабинете. Сеанс с пользователем может открыть и СИ. В этом сеансе поддерживается статус аутентификации пользователя, упрощается повторная аутентификация.

Сеанс строится по схеме «клиент — сервер». В качестве клиента выступает КП пользователя, в качестве сервера — СИ или ПС.

Связь в сеансе реализуется одним или несколькими соединениями. Соединение, как правило, устанавливается на транспортном сетевом уровне, а сеанс — на прикладном, поэтому в сеансе могут быть неизвестны детали соединений. Даже если к моменту создания сеанса уже имеются защищенные соединения, в сеансе нельзя положиться на эту защиту.

Для организации защиты сеанса сервер выпускает БС и передает его КП. Предъявляя билет при последовательных обращениях к серверу, КП подтверждает владение сеансом. Это подтверждение является основанием для оказания пользователю цифровых услуг, для упрощения повторной аутентификации. БС позволяет пользователю поддержать длительный сеанс с сервером без постоянной аутентификации.

Сеанс может завершиться по разным причинам. Во-первых, может просто истечь срок его действия. Во-вторых, пользователь может прекратить сеанс по собственной инициативе, завершив работу с КП или выполнив логаут. В-третьих, сеанс может завершить сервер после длительного бездействия пользователя.

Для продолжения сеанса проводится повторная аутентификация. Она может быть проще первоначальной (см. требование ВТ.19), может выполняться по другой схеме, например с использованием выпущенного ранее БА.

7.3.2 Требования

Требование УС.1 (1–3). БС должен генерироваться сервером (ПС или СИ) сразу после аутентификации или предъявления БА. БС должен передаваться КП пользователя по защищенному соединению (с аутентификацией отправителя).

Требование УС.2 (1–3). КП не должна использовать БС вне защищенных соединений с его отправителем.

Требование УС.3 (1–3). БС должен содержать не менее 128 бит энтропии.

Требование УС.4 (1–3). Сервер (ПС или СИ) должен хранить БС вместе со временем его выпуска. Билет должен уничтожаться на стороне сервера по окончании срока действия.

Требование УС.5 (1–3). КП пользователя не должна хранить БС в общедоступной области памяти. Билет должен уничтожаться на стороне пользователя по окончании срока действия.

Примечание — Для КП, выполненной в виде браузера, общедоступной является память, которая может быть прочитана с помощью JavaScript, например, локальное хранилище (local storage) HTML5.

Требование УС.6 (1). Срок действия БС не должен превышать 30 суток. После 12 ч бездействия пользователя должна выполняться повторная аутентификация.

Примечание — Пользователь может предупреждаться о приближении повторной аутентификации.

Требование УС.7 (2). Срок действия БС не должен превышать 12 ч. После 30 мин бездействия пользователя должна выполняться повторная аутентификация.

Требование УС.8 (3). Срок действия БС не должен превышать 12 ч. После 15 мин бездействия пользователя должна выполняться повторная аутентификация.

8 Реализация на основе OIDC

8.1 Общие сведения

Технология OIDC устанавливает интерфейсы веб-сервисов, реализующих элементы аутентификации и федерации. Работа с веб-сервисом состоит в обращении с запросом на поддерживаемый им коммуникационный узел. Ответы используются либо непосредственно, либо участвуют в формировании запросов на другие узлы. Перечень задействованных узлов представлен в таблице 3.

Таблица 3 — Узлы OIDC

Узел	Назначение	Размещение
Authorization	Аутентификация и авторизация	СИ
Redirection	Перенаправление	ПС
Token	Обмен кода авторизации на билеты	СИ
UserInfo	Предоставление сведений о пользователе	СР

Узел Redirection является техническим, он используется для передачи управления ПС через механизм перенаправления протокола HTTP. Узел UserInfo отвечает за обслуживание ресурсов пользователей. Кроме UserInfo в инфраструктуре аутентификации могут использоваться другие узлы ресурсов.

Правила обращения к узлам и очередность обращений описываются коммуникационной схемой. Схема выбирается при обращении на узел Authorization (см. таблицу 6).

В настоящем стандарте определяются 3 схемы: Code, Implicit и Hybrid. Первую схему рекомендуется применять при наличии прямого защищенного соединения между СИ и ПС с взаимной аутентификацией сторон. Вторая схема может использоваться при отсутствии прямого соединения: передача данных от СИ к ПС осуществляется через КП. Третья схема комбинирует элементы первой и второй: данные передаются от СИ к ПС и напрямую, и

через КП. При этом обеспечивается бóльшая гибкость, создаются возможности усиления гарантий безопасности. Основные различия между схемами перечислены в таблице 4.

Таблица 4 — Отличия между коммуникационными схемами

Свойство	Схема		
	Code	Implicit	Hybrid
Билеты возвращаются с узла Authorization	Да	Нет	Нет
Билеты возвращаются с узла Token	Нет	Да	Нет
Билеты не открываются КП	Да	Нет	Нет
ПС должна быть аутентифицирована перед СИ	Да	Нет	Да
Возможно обновление билетов	Да	Нет	Да
Выполнение за один цикл ПС – КП – СИ – ПС	Нет	Да	Нет
Основные пересылки между ПС и СИ	Да	Нет	Варьируется

Узлы OIDC описываются следующими элементами:
- сетевой адрес (URI);
- HTTP-метод обращения к узлу (GET или POST);
- формат запроса;
- формат ответов (успешного и об ошибке);
- схема кодирования параметров запросов и ответов.

Форматы запросов и ответов подробно определяются в приложении Б. В настоящем разделе форматы определяются кратко, в виде обзорных таблиц. В таблицах описываются параметры запросов и ответов. Для каждого параметра указывается обязательность его включения в запрос или ответ, приводится краткое описание, дается ссылка на подробное описание. Параметр, обязательный для включения, помечается знаком «+», рекомендуемый для включения — символом «р», условно обязательный — символом «у», опциональный — символом «о». Сочетание «ру» означает рекомендацию при выполнении определенного условия.

Предусмотрены следующие схемы кодирования параметров запросов и ответов:

1 Query. Имена и значения параметров кодируются в формате `"application/x-www-form-urlencoded"`, определенном в [10]. Результат кодирования добавляется к адресу целевого узла в качестве компонента `query` (после знака «?», см. [1]). Схема используется для кодирования прямых запросов и перенаправляемых ответов. В последнем случае целевым является узел перенаправления. Адрес этого узла, дополненный компонентом `query`, помещается в заголовок `Location` HTTP-ответа перенаправления (код 302).

2 Fragment. Имена и значения параметров кодируются в формате `"application/x-www-form-urlencoded"`. Результат кодирования добавляется к адресу целевого узла в качестве компонента `fragment` (после знака «#», см. [1]). Схема используется для кодирования перенаправляемых ответов. Адрес узла перенаправления, дополненный компонентом `fragment`, помещается в заголовок `Location` HTTP-ответа перенаправления.

3 Form. Имена и значения параметров кодируются в формате `"application/x-www-form-urlencoded"`. Результат кодирования помещается в тело HTTP-пакета. Название формата указывается в заголовке `Content-Type` (тип содержимого) пакета.

4 JSON. Имена и значения параметров кодируются в формате `"application/json"`, определенном в [11]. В результате кодирования получается объект (контейнер) JSON,

который помещается в тело HTTP-пакета. Название формата указывается в заголовке `Content-Type` пакета. Параметры являются записями первого уровня объекта JSON. Названия параметров и строковые значения представляются строками JSON. Числовые значения представляются числами JSON. Очередность параметров в контейнере не имеет значения.

5 JWT. Имена и значения параметров записываются в объект JSON, который затем встраивается в объект JWT (см. приложение Г). Полученный объект кодируется в формате `"application/jwt"`, определенном в [12]. Результат кодирования помещается в тело HTTP-пакета. Название формата указывается в заголовке `Content-Type` пакета.

6 Bearer. Используется только для БД. Билет, предваряемый префиксом `"Bearer"`, размещается в заголовке `Authorization` HTTP-пакета. Формат заголовка определен в [4].

7 WWW-Authenticate. Имена и значения параметров, описывающих ошибки, помещаются в заголовок `WWW-Authenticate` HTTP-ответа об ошибке. Формат заголовка определен в [4]. В заголовке размещается строка с префиксом `"Bearer"`, а далее через запятую перечисляются параметры и их значения.

В соответствии с требованиями, установленными в разделах 6 и 7, взаимодействие сторон OIDC выполняется по защищенным соединениям. В частности, запросы на узлы и ответы с них должны защищаться при пересылке. Для организации защиты рекомендуется использовать протокол TLS, определенный в СТБ 34.101.65.

8.2 Схема Code

Взаимодействие сторон по схеме Code выполняется следующим образом:

1 В сеансе работы с пользователем ПС готовит запрос авторизации/аутентификации.

2 ПС пересылает запрос КП пользователя, перенаправляя КП на узел Authorization.

3 СИ получает запрос, аутентифицирует пользователя, получает согласие пользователя на доступ ПС к ресурсам, указанным в запросе.

4 В случае успешной аутентификации и согласия пользователя СИ создает и сохраняет билеты аутентификации, доступа и, возможно, обновления. СИ также генерирует и сохраняет код авторизации, ссылающийся на билеты.

5 СИ перенаправляет КП на узел Redirection вместе с кодом авторизации.

6 ПС отправляет код авторизации на узел Token, ожидая в ответ билеты.

7 СИ аутентифицирует ПС, проверяет код авторизации, определяет соответствующие ему билеты и отправляет их ПС.

8 ПС извлекает из БА утверждения об аутентификации и о пользователе. Эти утверждения ПС использует для организации цифровых услуг.

9 ПС использует БД для доступа к ресурсам пользователя, обращаясь к узлу UserInfo или другим узлам ресурсов.

10 Располагая БО, ПС получает новый БД без повторного прогона схемы Code. Для этого ПС обращается к узлу Token, включая в запрос БО.

Если в схеме Code используется БО, то этот билет, как правило, имеет большой (несколько месяцев) или даже неограниченный срок действия, а БД — малый (в пределах 1 ч). Если БО не используется, то БД имеет большой или неограниченный срок действия. Билеты с неограниченным сроком действия могут использоваться сколь угодно долго, пока пользователь не отзовет их.

Перед взаимодействием по схеме Code ПС должна быть аутентифицирована перед СИ с использованием протокола, установленного для данной ПС при ее регистрации перед СИ.

8.3 Схема Implicit

Взаимодействие сторон по схеме Implicit выполняется следующим образом:

1 В сеансе работы с пользователем ПС готовит запрос авторизации / аутентификации.

2 ПС пересылает запрос КП пользователя, перенаправляя КП на узел Authorization.

3 СИ получает запрос, аутентифицирует пользователя, получает согласие пользователя на доступ ПС к ресурсам, указанным в запросе.

4 В случае успешной аутентификации и согласия пользователя СИ создает билеты аутентификации и (или) доступа и передает их КП, перенаправляя ее на узел Redirection.

5 В результате перенаправления КП передает билеты ПС.

6 ПС извлекает из БА утверждения об аутентификации и о пользователе. Эти утверждения ПС использует для организации цифровых услуг.

7 ПС использует БД для получения ресурсов пользователя, обращаясь к узлу UserInfo или к другим узлам ресурсов.

8.4 Схема Hybrid

Схема Hybrid отличается от схемы Code тем, что в ответе с узла Authorization дополнительно к коду авторизации возвращаются билеты доступа и (или) аутентификации. Эти же билеты возвращаются с узла Token в обмен на код авторизации. Билеты одного типа, возвращаемые с разных узлов, могут отличаться.

Схема Hybrid заимствует элементы схемы Implicit, в которой билеты также возвращаются с узла Authorization.

8.5 Узел Authorization

Узел Authorization обрабатывает запросы авторизации/аутентификации. К узлу обращается КП пользователя по направлению ПС. При обработке запроса СИ проводит аутентификацию пользователя, получает согласие пользователя на доступ ПС к его ресурсам, возможно, выпускает код авторизации, перенаправляет пользователя к ПС.

Адрес узла должен быть зафиксирован в документах инфраструктуры, или адрес должен быть сообщен ПС во время регистрации в инфраструктуре. Адрес не должен содержать компонент `fragment`.

Запрос на узел и соответствующие ответы описаны в таблице 5. Запрос и ответы могут включать параметры, дополнительные к перечисленным в таблице.

Узел должен поддерживать HTTP-метод GET и может поддерживать метод POST. При использовании GET запрос должен кодироваться по схеме Query, при использовании POST — по схеме Form. Нераспознанные параметры запроса и параметры без значений должны игнорироваться. Параметры не должны повторяться.

Ответ узла должен кодироваться по схеме Query или Fragment с перенаправлением на узел Redirection. Схема кодирования Query должна применяться только в коммуникационной схеме Code. Для перенаправления в ответе должен указываться код 302 HTTP.

В параметре `response_type` запроса на узел задается коммуникационная схема, которую требуется использовать. Возможные значения параметра и соответствующие им коммуникационные схемы представлены в таблице 6. В лексемах параметра перечисляются ожидаемые в ответе с узла Authorization билеты: `"code"` указывает на код авторизации, `"id_token"` — на БА, `"token"` — на БД.

Таблица 5 — Узел Authorization: запрос и ответы

Параметр	Включение	Описание	Ссылка
Запрос (GET/Query или POST/Form)			
`scope`	+	Запрашиваемая область действия	Б.1.29
`response_type`	+	Тип ответа	Б.1.28
`client_id`	+	Идентификатор ПС	Б.1.8
`redirect_uri`	+	Адрес перенаправления ответа	Б.1.23
`state`	р	Данные для перенаправления	Б.1.30
`response_mode`	о	Механизм возвращения ответа	Б.1.27
`nonce`	у	Волатильные данные для переноса в БА	Б.1.21
`display`	о	Тип интерфейса пользователя	Б.1.11
`prompt`	о	Перечень приглашений пользователю	Б.1.22
`max_age`	о	Срок действия утверждений аутентификации	Б.1.20
`ui_locales`	о	Предпочтительные языки интерфейса пользователя	Б.1.32
`id_token_hint`	о	Ранее выпущенный БА	Б.1.18
`login_hint`	о	Подсказка об идентификаторе пользователя	Б.1.19
`acr_values`	о	Предпочтительные уровни гарантий аутентификации	Б.1.3
`claims`	о	Перечень утверждений для включения в БА	Б.1.4
`claims_locales`	о	Предпочтительные языки утверждений	Б.1.5
`request`	о	Защищенный запрос в виде объекта JWT	Б.1.25
`request_uri`	о	Ссылка на защищенный запрос	Б.1.26
Успешный ответ в схеме Code (Query)			
`code`	+	Код авторизации	Б.1.10
`state`	у	Копия параметра `state` запроса	Б.1.30
Успешный ответ в схеме Implicit (Fragment)			
`access_token`	у	БД	Б.1.2
`token_type`	у	Тип БД	Б.1.31
`expires_in`	р	Срок действия БД	Б.1.15
`id_token`	+	БА	Б.1.17
`scope`	у	Разрешенная область действия	Б.1.29
`state`	у	Копия параметра `state` запроса	Б.1.30
Успешный ответ в схеме Hybrid (Fragment)			
`code`	+	Код авторизации	Б.1.10
`access_token`	у	БД	Б.1.2
`token_type`	у	Тип БД	Б.1.31
`expires_in`	р	Срок действия БД	Б.1.15
`id_token`	у	БА	Б.1.17
`scope`	у	Разрешенная область действия	Б.1.29
`state`	у	Копия параметра `state` запроса	Б.1.30
Ответ об ошибке (Query или Fragment)			
`error`	+	Код ошибки	Б.1.12
`error_description`	о	Описание ошибки	Б.1.13
`error_uri`	о	Адрес веб-страницы с информацией об ошибке	Б.1.14
`state`	у	Копия параметра `state` запроса	Б.1.30

Таблица 6 — Выбор коммуникационной схемы

Значение параметра `response_type`	Схема
`"code"`	Code
`"id_token"`	Implicit
`"id_token token"`	Implicit
`"code id_token"`	Hybrid
`"code token"`	Hybrid
`"code id_token token"`	Hybrid

8.6 Узел Redirection

Узел Redirection принимает ответы СИ с узла Authorization. Ответы пересылаются через КП с помощью механизма перенаправления HTTP. Получив ответ, ПС использует его данные для обращения к другим узлам.

Адрес узла должен быть согласован с СИ при регистрации ПС. Адрес не должен содержать компонент `fragment`.

ПС может использовать несколько узлов Redirection и, соответственно, регистрировать несколько адресов. Конкретный адрес указывается в параметре `redirect_uri` запроса авторизации/аутентификации.

8.7 Узел Token

Узел Token обрабатывает запросы на выпуск и обновление билетов. Узел используется в схемах Code и Hybrid. К узлу обращается ПС, предъявляя ранее полученные секретные данные: код авторизации или БО. СИ проверяет представленные данные и, в случае успеха, выпускает БД. СИ может дополнительно выпускать БА и БО.

Запрос на выпуск билета включает код авторизации, полученный КП от узла Authorization и перенаправленный ПС на узел Redirection. Запрос на обновление включает БО, полученный ранее от узла Token в результате обработки запроса на выпуск или предыдущего запроса на обновление. Тип запроса указывается в его параметре `grant_type`.

Адрес узла должен быть зафиксирован в документах инфраструктуры, или адрес должен быть сообщен ПС во время регистрации в инфраструктуре. Адрес не должен содержать компонент `fragment`.

Перед обращением к узлу ПС должна пройти аутентификацию перед СИ. Аутентификация может проводиться или завершаться в момент обращения, и тогда запрос ПС должен содержать аутентификационные данные. Перечень методов аутентификации должен быть согласован с СИ при регистрации ПС и связан с идентификатором ПС.

Запросы на узел и соответствующие ответы описаны в таблице 7. Запросы и ответы могут включать параметры, дополнительные к перечисленным в таблице.

ПС должна обращаться к узлу, используя HTTP-метод POST. Запрос должен кодироваться по схеме Form. Нераспознанные параметры запроса и параметры без значений должны игнорироваться. Параметры не должны повторяться.

Ответ узла должен кодироваться по схеме JSON. Если ответ содержит БД, БО или другие критические данные, то соответствующий пакет HTTP должен включать заголовок `Cache-Control` со значением `"no-store"` и заголовок `Pragma` со значением `"no-cache"`.

Таблица 7 — Узел Token: запросы и ответы

Параметр	Включение	Описание	Ссылка
Запрос на выпуск билетов (POST/Form)			
`grant_type`	+	Значение `"authorization_code"`	Б.1.16
`code`	+	Код авторизации	Б.1.10
`redirect_uri`	+	Копия одноименного параметра запроса на узел Authorization	Б.1.23
`client_id`	у	Идентификатор ПС	Б.1.8
`client_secret`	у	Секрет аутентификации ПС *	Б.1.9
`client_assertion_type`	у	Тип аутентификатора ПС	Б.1.7
`client_assertion`	у	Аутентификатор ПС	Б.1.6
Запрос на обновление билетов (POST/Form)			
`grant_type`	+	Значение `"refresh_token"`	Б.1.16
`refresh_token`	+	БО	Б.1.24
`client_id`	у	Идентификатор ПС	Б.1.8
`client_secret`	у	Секрет аутентификации ПС *	Б.1.9
`client_assertion_type`	у	Тип аутентификатора ПС	Б.1.7
`client_assertion`	у	Аутентификатор ПС	Б.1.6
`scope`	о	Запрашиваемая область действия	Б.1.29
Успешный ответ на запрос на выпуск билетов (JSON)			
`access_token`	+	БД	Б.1.2
`token_type`	+	Тип БД	Б.1.31
`expires_in`	р	Срок действия БД	Б.1.15
`id_token`	+	БА	Б.1.17
`refresh_token`	у	БО	Б.1.24
`scope`	у	Разрешенная область действия	Б.1.29
Успешный ответ на запрос на обновление билетов (JSON)			
`access_token`	+	Билет доступа	Б.1.2
`token_type`	+	Тип билета доступа	Б.1.31
`expires_in`	р	Срок действия БД	Б.1.15
`id_token`	о	БА	Б.1.17
`refresh_token`	у	БО	Б.1.24
`scope`	у	Разрешенная область действия	Б.1.29
Ответ об ошибке (JSON)			
`error`	+	Код ошибки	Б.1.12
`error_description`	о	Описание ошибки	Б.1.13
`error_uri`	о	Адрес веб-страницы с информацией об ошибке	Б.1.14

* — Может передаваться отдельно по правилам схемы аутентификации HTTP Basic.

Для информирования СИ о необходимости выпуска БО в параметр `scope` запроса на узел Authorization (см. таблицу 5) включается строка `"offline_access"`. Для информирования о необходимости перевыпуска БО эта же строка включается в параметр `scope` запроса на обновление. СИ может выпускать и перевыпускать БО в других ситуациях, без явного запроса со стороны ПС.

В запросах на узел Token параметры `client_id`, `client_secret`, `client_assertion_type` и `client_assertion` используются для аутентификации ПС в момент обращения. Предусмотрены четыре метода аутентификации:

1 Метод `client_secret_basic`. При регистрации ПС согласует с СИ секрет аутентификации `client_secret`, СИ связывает секрет с идентификатором ПС `client_id`. Секрет передается в качестве аутентификатора ПС в запросе на узел. Секрет сопровождается идентификатором ПС. СИ определяет по идентификатору зарегистрированный секрет, а затем сравнивает его с присланным. Секрет `client_secret` передается по правилам схемы аутентификации HTTP Basic (в заголовке `Authorization`).

2 Метод `client_secret_post`. Отличается от метода `client_secret_basic` только тем, что `client_secret` кодируется вместе с остальными параметрами запроса по схеме Form.

3 Метод `client_secret_jwt`. При регистрации ПС согласует с СИ секретный ключ имитозащиты `client_secret`. В качестве аутентификатора передается объект JWT, который включает идентификаторы ПС и СИ, уникальный идентификатор самого объекта, отметки времени, а также имитовставку всех этих данных, вычисленную на ключе `client_secret`. СИ определяет по идентификатору зарегистрированный ключ и проверяет на нем имитовставку. Объект JWT передается в параметре `client_assertion` запроса. Метод аутентификации указывается в параметре `client_assertion_type`.

4 Метод `private_key_jwt`. Отличается от метода `client_secret_jwt` тем, что вместо имитовставки используется ЭЦП. Данные объекта JWT подписываются на личном ключе ПС. СИ проверяет подпись на соответствующем открытом ключе. Сертификат открытого ключа передается СИ в момент регистрации ПС.

8.8 Узел UserInfo

Узел UserInfo обрабатывает запросы на предоставление сведений об аутентифицированном пользователе. К узлу обращается ПС, предъявляя ранее полученный БД. СР проверяет билет и, в случае успеха, возвращает утверждения о пользователе.

Запрос на узел и соответствующие ответы описаны в таблице 8. Запрос и ответы могут включать параметры, дополнительные к перечисленным в таблице.

Таблица 8 — Узел UserInfo: запрос и ответы

Параметр	Включение	Описание	Ссылка
Запрос (GET/Bearer или POST/Bearer)			
`access_token`	+	БД	Б.1.2
Успешный ответ (JSON или JWT)			
`sub`	+	Идентификатор пользователя	В.2.2
`iss`	ру	Идентификатор СИ	В.2.1
`aud`	ру	Идентификатор ПС	В.2.3
Ответ об ошибке (WWW-Authenticate)			
`error`	ру	Код ошибки	Б.1.12
`error_description`	о	Описание ошибки	Б.1.13
`error_uri`	о	Адрес веб-страницы с информацией об ошибке	Б.1.14
`scope`	о	Требуемая область действия	Б.1.29

Узел должен поддерживать HTTP-методы GET и POST. Для обращения к узлу рекомендуется использовать метод GET. Отправляемый БД должен кодироваться по схеме Bearer.

Успешный ответ с утверждениями о пользователе должен кодироваться по схемам JSON и JWT. Схема JWT применяется тогда, когда утверждения подписываются и (или) конвертуются. Утверждения могут только подписываться, могут только конвертоваться и могут сначала подписываться, а затем конвертоваться.

Ответ об ошибке должен кодироваться по схеме WWW-Authenticate.

Приложение А
(справочное)
Англоязычные термины OAuth 2.0 и OIDC

Настоящий стандарт	OAuth 2.0 / OIDC
Билет аутентификации; БА	ID token
Билет обновления; БО	refresh token
Билет доступа; БД	access token
Запрос авторизации/аутентификации	authorization/authentication request
Запрос билетов	token request
Идентификатор	identifier
Клиентская программа; КП	user-agent
Код авторизации	authorization code
Коммуникационная схема	flow
Несущественное утверждение	voluntary claim
Пользователь	end-user
Прикладная система; ПС	client, relying party
Сервер ресурсов; СР	resource server
Служба идентификации; СИ	authorization server, OpenID provider
Существенное утверждение	essential claim
Узел	endpoint
Утверждение	claim

Приложение Б
(обязательное)
Запросы и ответы OIDC

Б.1 Параметры

Б.1.1 Общие сведения

В настоящем подразделе перечисляются параметры запросов и ответов OIDC, устанавливаются правила формирования, передачи и обработки параметров, определяются форматы параметров. Параметры перечисляются в алфавитном порядке.

При определении форматов параметров используется синтаксис ABNF (Augmented Backus-Naur Form), установленный в [13]. Применяются следующие определения ABNF, дополнительные к введенным в [13]:

```
VSCHAR = %x20-7E
NQCHAR = %x21 / %x23-5B / %x5D-7E
NQSCHAR = %x20-21 / %x23-5B / %x5D-7E
UNICODECHARNOCRLF = %x09 / %x20-7E / %x80-D7FF / %xE000-FFFD
    / %x10000-10FFFF
base64url-char = ALPHA / DIGIT / "_" / "-"
base64url = *base64url-char
jws = base64url 2( "." base64url )
    ; header.payload.signature
jwe = base64url 4( "." base64url )
    ; header.encrypted-key.iv.ciphertext.tag
jwt = jws / jwe
```

Дополнительно используется определение `URI-reference`, установленное в [1].

Параметрами ответа с узла UserInfo являются утверждения `sub`, `iss`, `aud`. Они описываются в В.2.

Б.1.2 Параметр `access_token`

Параметр `access_token` содержит БД, выпущенный СИ.

Параметр `access_token` возвращается в ответах с узлов Authorization и Token, передается в запросах на узел UserInfo и другие узлы ресурсов. Параметр должен включаться в успешный ответ с узла Authorization, если и только если параметр `response_type` запроса на узел содержит строку `"token"` (см. таблицу 6).

БД должен содержать не менее 128 бит энтропии.

Формат параметра:

```
access-token = 22*VSCHAR
```

Б.1.3 Параметр `acr_values`

Параметр `acr_values` содержит перечень предпочтительных уровней гарантий аутентификации пользователя. Уровни кодируются лексемами, разделяемыми пробелами. Уровни перечисляются в порядке убывания предпочтения.

Параметр `acr_values` указывается в запросе на узел Authorization.

СИ должна учитывать указанные в `acr_values` предпочтения при выборе методов аутентификации пользователя. СИ может указать выбранный уровень в утверждении `acr` БА (см. В.2.8).

Примечание — Предпочтительные уровни гарантий аутентификации могут запрашиваться также через параметр `claims` (см. Б.1.4, В.4.4).

Формат параметра:

```
acr-values = 1*VSCHAR
```

Базовый уровень гарантий рекомендуется кодировать строкой `"1"`, средний — строкой `"2"`, высокий — `"3"`. Могут вводиться подуровни: `"11"` (1-й подуровень базового уровня), `"12"` (2-й подуровень) и т. д.

Б.1.4 Параметр `claims`

Параметр `claims` содержит перечень утверждений, которые требуется включить в БА и (или) ответы с узлов ресурсов. Перечень представляет собой объект JSON, в котором указываются имена утверждений, их ожидаемые значения (например, идентификатор пользователя) и атрибуты (например, признак обязательности включения в билет). Правила формирования перечня определены в В.4.

Параметр `claims` указывается в запросе на узел Authorization.

Формат параметра:

```
claims = 1*VSCHAR
```

Б.1.5 Параметр `claims_locales`

Параметр `claims_locales` содержит перечень предпочтительных языков утверждений. Перечень состоит из тегов, формируемых по правилам [14]. Тег кодирует язык, а также, возможно, алфавит (кириллица или латиница), региональные особенности, диалект и пр. Теги разделяются пробелами и перечисляются в порядке убывания предпочтения.

Параметр `claims_locales` указывается в запросе на узел Authorization.

СИ и СР должны выбирать язык утверждений с учетом перечисленных в `claims_locales` тегов и их очередности. При обработке запроса СИ не следует возвращать ошибку, даже если ни один из запрашиваемых языков не поддерживается.

Формат параметра:

```
claims-locales = locales-name *( SP locales-name )
locales-name = 1*locales-char
locales-char = "-" / DIGIT / ALPHA
```

Б.1.6 Параметр `client_assertion`

Параметр `client_assertion` представляет собой объект JWT, который выступает в роли аутентификатора ПС при обращении к СИ. Объект содержит волатильные контекстно зависимые данные, которые сопровождаются имитовставкой или подписью ПС.

Параметр `client_assertion` указывается в запросе на узел Token. Параметр должен включаться в запрос при использовании методов аутентификации `client_secret_jwt` и `private_key_jwt`.

Формат параметра:

```
client-assertion = jwt
```

Исходными данными передаваемого в `client_assertion` объекта JWT являются утверждения. Обязательные утверждения: `iss`, `sub`, `aud`, `jti`, `exp`, необязательное: `iat` (см. раздел В.2). Могут использоваться другие утверждения. Утверждения, которые СИ не может распознать, должны игнорироваться.

В утверждениях `iss` и `sub` должен быть указан идентификатор ПС, в утверждении `aud` — идентификатор СИ. В качестве идентификатора СИ может выступать адрес узла Token. При обработке объекта СИ должна проверять, что утверждение `aud` действительно относится к ней.

В утверждении `jti` ПС должна указать уникальный идентификатор объекта JWT. СИ может проверять уникальность, сохраняя предъявляемые объекты вплоть до окончания их действия.

Момент окончания действия должен указываться в утверждении `jti`. СИ должна отклонять просроченные объекты (с поправкой на возможную рассинхронизацию). СИ может отклонять объекты, продолжительность действия которых неоправданно велика.

В утверждении `iat` ПС может указать момент выпуска объекта. СИ может отклонять объекты, которые начали действовать чересчур давно.

Б.1.7 Параметр `client_assertion_type`

Параметр `client_assertion_type` описывает метод аутентификации ПС, который требует передачи аутентификатора в параметре `client_assertion`.

Параметр `client_assertion_type` указывается в запросе на узел Token. Параметр должен включаться в запрос при использовании методов аутентификации `client_secret_jwt` и `private_key_jwt`.

Формат параметра:

```
client-assertion-type = 1*( "_" / "-" / ":" / DIGIT / ALPHA )
```

При использовании методов аутентификации `client_secret_jwt` и `private_key_jwt` параметр должен принимать значение

```
"urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
```

Б.1.8 Параметр `client_id`

Параметр `client_id` содержит идентификатор ПС. Идентификатор назначает СИ при регистрации ПС.

Параметр `client_id` указывается в запросах на узлы Authorization и Token. Параметр должен включаться в запрос на узел Token, если аутентификация выполняется в момент обращения к узлу и для этого используется метод `client_secret_basic` или `client_secret_post`.

Формат параметра:

```
client-id = *VSCHAR
```

Б.1.9 Параметр `client_secret`

Параметр `client_secret` содержит зарегистрированный секрет аутентификации ПС.

Параметр `client_secret` указывается в запросе на узел Token. Параметр должен включаться в запрос, если аутентификация выполняется в момент обращения к узлу и для этого используется метод `client_secret_basic` или `client_secret_post`.

Формат параметра:

```
client-secret = 20*VSCHAR
```

Б.1.10 Параметр `code`

Параметр `code` содержит код авторизации, выпущенный СИ по результатам обработки запроса авторизации/аутентификации в схеме Code или Hybrid.

Параметр `code` возвращается в ответе с узла Authorization, передается в запросе на узел Token.

При выпуске кода авторизации СИ должна связать его с идентификатором ПС и адресом перенаправления (параметры `client_id` и `redirect_uri` запроса на узел Authorization) и учитывать связывание при приеме кода. Код авторизации должен содержать не менее 64 бит энтропии. Для снижения издержек при компрометации кода срок его действия должен быть небольшим. Рекомендуемый срок — не более 10 мин.

ПС должна использовать код авторизации однократно. При повторном получении одного и того же кода СИ не должна его обрабатывать и, более того, СИ следует отменить все билеты, выпущенные ранее на основании этого кода.

Формат параметра:

```
code = 10*VSCHAR
```

Б.1.11 Параметр `display`

Параметр `display` определяет предпочтительный тип интерфейса пользователя во время аутентификации перед СИ и дачи разрешения на доступ к его ресурсам. Интерфейс реализует КП.

Параметр `display` указывается в запросе на узел Authorization.

При обработке запроса СИ может определить возможности клиентской программы и самостоятельно выбрать подходящий тип интерфейса.

Формат параметра:

```
display = 1*( "_" / DIGIT / ALPHA )
```

Стандартные значения параметра:

– `"page"` — полностраничный режим, значение по умолчанию;

– `"popup"` — всплывающее окно. Окно должно иметь подходящий размер и не должно закрывать главное окно КП;

– `"touch"` — интерфейс сенсорного устройства;

– `"wap"` — интерфейс сотового телефона (не смартфона).

Б.1.12 Параметр `error`

Параметр `error` описывает код ошибки.

Параметр `error` включается в ответ об ошибке. Параметр следует включать в ответ с узла UserInfo, если ошибка касается технологии OIDC и не может быть описана стандартным кодом ошибки HTTP.

Формат параметра:

```
error = 1*NQSCHAR
```

Стандартные коды ошибок:

– `"invalid_request"` — неверный формат запроса: отсутствует требуемый параметр, некорректное значение параметра, параметр встречается более одного раза, другое. Код `"invalid_request"` должен возвращаться вместе с кодом 400 (Bad Request) HTTP;

– `"invalid_client"` — ошибка аутентификации ПС: неизвестная ПС, аутентификация не выполнялась, неподдерживаемый метод аутентификации, другое. Код `"invalid_client"` может возвращаться вместе с кодом 401 (Unauthorized) HTTP. Если

секрет аутентификации передавался в заголовке `Authorization` HTTP-запроса (например, использовался метод `client_secret_basic`), то код 401 является обязательным и в HTTP-ответ должен быть включен заголовок `WWW-Authenticate` с информацией об использованном методе аутентификации;

– `"invalid_grant"` — код авторизации, аутентификатор пользователя или БО недействителен: некорректен, уже не действует, отозван, не соответствует адресу перенаправления, выпущен для другой ПС;

– `"invalid_token"` — БД недействителен: некорректен, уже не действует, отозван, другое. Код `"invalid_token"` должен возвращаться вместе с кодом 401 (Unauthorized) HTTP;

– `"insufficient_scope"` — область действия БД недостаточна для успешной обработки запроса. Код `"insufficient_scope"` следует возвращать с кодом 403 (Forbidden) HTTP. Код `"insufficient_scope"` может сопровождаться параметром `scope`, в котором должна быть указана требуемая область действия;

– `"unauthorized_client"` — ПС не авторизована на обращение к узлу, запрос кода авторизации или БД;

– `"access_denied"` — отказ в обработке запроса со стороны пользователя или СИ;

– `"unsupported_response_type"` — СИ не поддерживает заявленное значение параметра `response_type`;

– `"unsupported_grant_type"` — СИ не поддерживает заявленное значение параметра `grant_type`;

– `"invalid_scope"` — указанное в параметре `scope` значение некорректно, не распознано, или выходит за пределы заданной пользователем области действия;

– `"server_error"` — СИ столкнулась с непредвиденными обстоятельствами, которые препятствуют обработке запроса;

– `"temporarily_unavailable"` — СИ в настоящее время неспособна обработать запрос из-за временной перегрузки или технического обслуживания.

Примечание — Коды `"server_error"`, `"temporarily_unavailable"` имеют такое же значение, что и коды 500 (Internal Server Error), 503 (Service Unavailable) HTTP. Однако коды HTTP не могут быть возвращены ПС через механизм перенаправления;

– `"interaction_required"` — СИ требует взаимодействия с пользователем в той или иной форме. Код `"interaction_required"` может возвращаться тогда, когда параметр `prompt` запроса авторизации/аутентификации принимает значение `"none"`, и при этом запрос не может быть обработан без отображения интерфейса пользователя для взаимодействия с ним;

– `"login_required"` — СИ требует аутентификации пользователя. Код `"login_required"` может возвращаться тогда, когда параметр `prompt` запроса авторизации/аутентификации принимает значение `"none"`, и при этом запрос не может быть обработан без отображения интерфейса пользователя для его аутентификации;

– `"account_selection_required"` — для аутентификации пользователь должен выбрать учетную запись. Пользователь может быть аутентифицирован перед СИ с использованием различных учетных записей, но не выбрана ни одна из них. Код `"account_selection_required"` может возвращаться тогда, когда параметр `prompt` запроса авторизации/аутентификации принимает значение `"none"`, и при этом запрос не может быть обработан без отображения интерфейса пользователя для выбора учетной записи;

– `"consent_required"` — СИ требует подтверждения согласия на доступ к ресурсам пользователя. Код `"consent_required"` может возвращаться тогда, когда параметр

prompt запроса авторизации/аутентификации принимает значение "none", и при этом запрос не может быть обработан без отображения интерфейса пользователя для подтверждения его согласия на доступ к ресурсам;

– "invalid_request_uri" — параметр request_uri в запросе авторизации/аутентификации ссылается на некорректный объект JWT или произошла ошибка при обращении по указанному в request_uri адресу;

– "invalid_request_object" — параметр request в запросах авторизации/аутентификации содержит некорректный объект JWT;

– "request_not_supported" — СИ не поддерживает параметр request в запросах авторизации/аутентификации;

– "request_uri_not_supported" — СИ не поддерживает параметр request_uri в запросах авторизации/аутентификации.

Б.1.13 Параметр error_description

Параметр error_description содержит информацию об ошибке в удобной для восприятия человеком форме.

Параметр error_description включается в ответ об ошибке.

Формат параметра:

```
error-description = 1*NQSCHAR
```

Б.1.14 Параметр error_uri

Параметр error_uri содержит сетевой адрес (URI) веб-страницы с информацией об ошибке в удобной для восприятия человеком форме.

Параметр error_uri включается в ответ об ошибке.

Формат параметра:

```
error-uri = URI-reference
```

Б.1.15 Параметр expires_in

Параметр expires_in определяет срок действия БД в секундах. Например, значение 300 означает, что БД действует в течение 5 мин с момента формирования ответа.

Параметр expires_in включается в ответы с узлов Authorization и Token. Параметр рекомендуется включать в ответ, если он содержит БД. Если параметр опущен, то СИ должна определить в документации или другим способом срок действия БД, используемый по умолчанию.

Формат параметра:

```
expires-in = 1*DIGIT
```

Б.1.16 Параметр grant_type

Параметр grant_type определяет тип запроса на узел Token.

Параметр grant_type включается в запрос на узел Token. В запросе на выпуск билетов параметр должен принимать значение "authorization_code". В запросе на обновление билетов параметр должен принимать значение "refresh_token".

Формат параметра:

```
grant-type = grant-name / URI-reference
grant-name = 1*name-char
name-char = "-" / "." / "_" / DIGIT / ALPHA
```

Б.1.17 Параметр `id_token`

Параметр `id_token` содержит БА, выпущенный СИ.

Параметр `id_token` возвращается в ответах с узлов Authorization и Token.

Формат параметра:

```
id-token = 1*VSCHAR
```

Содержание и формат БА детализируются в приложении Д.

Б.1.18 Параметр `id_token_hint`

Параметр `id_token_hint` содержит БА, ранее выпущенный СИ. БА подтверждает, что пользователь был аутентифицирован в одном из предыдущих сеансов с СИ.

Параметр `id_token_hint` указывается в запросе на узел Authorization.

СИ возвращает успешный ответ на запрос, если сеанс с пользователем, указанным в БА, не завершен или если сеанс был открыт в результате обработки запроса. В противном случае СИ следует возвратить ответ об ошибке, например, с кодом `"login_required"`.

Если параметр `prompt` запроса на узел Authorization принимает значение `"none"`, то параметр `id_token_hint` следует включать в запрос. Если `id_token_hint` все-таки не включен, то СИ может возвратить ошибку. Тем не менее, СИ следует давать по возможности успешные ответы даже при отсутствии в запросе `id_token_hint`.

БА, переданный через `id_token_hint`, может использоваться СИ, даже если идентификатор СИ не указан в утверждении `aud` билета (см. В.2.3).

Если БА, который требуется передать через `id_token_hint`, не только подписан, но и конвертован, то ПС перед передачей должна снять защиту с БА, а затем конвертовать билет на открытом ключе СИ.

Формат параметра:

```
id-token-hint = 1*VSCHAR
```

Б.1.19 Параметр `login_hint`

Параметр `login_hint` содержит идентификатор, который пользователь предположительно может использовать во время аутентификации перед СИ. Предполагаемый идентификатор может быть известен ПС, если в сеансе с пользователем тот передавал ей свои идентификационные данные. Передача предполагаемого идентификатора может упростить СИ поиск учетной записи пользователя.

Параметр `login_hint` указывается в запросе на узел Authorization.

Обработка параметра `login_hint` остается на усмотрение СИ.

Формат параметра:

```
login-hint = *UNICODECHARNOCRLF
```

Б.1.20 Параметр `max_age`

Параметр `max_age` описывает срок действия утверждений аутентификации, т. е. максимальный промежуток времени с момента (активной) аутентификации пользователя, в течение которого ПС готова полагаться на факт аутентификации. Срок действия задается в секундах.

Параметр `max_age` указывается в запросе на узел Authorization.

Если заданный в `max_age` срок действия истек (например, задано нулевое значение), то СИ должна инициировать аутентификацию пользователя.

Если параметр `max_age` включен в запрос, то выпускаемый БА должен содержать утверждение `auth_time` (см. В.2.6).

Формат параметра:

```
max-age = 1*DIGIT
```

Б.1.21 Параметр `nonce`

Параметр `nonce` содержит синхропосылку — волатильные данные, предназначенные для противодействия атакам на основе повторов. Синхропосылка повторяется в запросе аутентификации и соответствующем БА, делая их уникальными и связывая друг с другом.

Параметр `nonce` указывается в запросе на узел Authorization. Параметр должен включаться в запрос при использовании коммуникационных схем Implicit и Hybrid.

Синхропосылку готовит ПС. Синхропосылка должна содержать не менее 64 бит энтропии.

При обработке запроса СИ должна сохранить синхропосылку, переданную в параметре `nonce`, а затем перенести ее в выпускаемый БА. Синхропосылка повторяется в утверждении `nonce` билета.

Если синхропосылка включена в БА, то при получении билета ПС должна сравнить ее с синхропосылкой, отправленной в запросе.

Формат параметра:

```
nonce = 10*VSCHAR
```

Б.1.22 Параметр `prompt`

Параметр `prompt` содержит перечень приглашений пользователю для организации его повторной аутентификации и получения его согласия на доступ к ресурсам. Приглашения кодируются лексемами, разделяемыми пробелами.

Параметр `prompt` указывается в запросе на узел Authorization.

ПС может включать параметр в запрос для проверки того, что сеанс пользователя все еще активен или для привлечения внимания пользователя к запросу.

Формат параметра:

```
prompt = prompt-name *( SP prompt-name )
prompt-name = 1*prompt-char
prompt-char = "_" / DIGIT / ALPHA
```

Предусмотрены следующие приглашения:

– `"none"` — СИ не должна отображать элементы интерфейса пользователя и, таким образом, приглашение фактически отсутствует, являясь техническим. Если пользователь еще не аутентифицирован, или ПС не имеет предварительного согласия на доступ к ресурсам пользователя, или не выполнены какие-либо другие условия, то должна быть возвращена ошибка, например, с кодом `"login_required"` или `"interaction_required"`. Приглашение `"none"` может использоваться для проверки того, аутентифицирован пользователь или нет;

– `"login"` — СИ следует предложить пользователю пройти повторную аутентификацию. Если СИ не может повторно аутентифицировать пользователя, то должна быть возвращена ошибка, например, с кодом `"login_required"`;

– `"consent"` — СИ следует предложить пользователю подтвердить согласие на доступ ПС к его ресурсам. Если СИ не может получить подтверждение, то должна быть возвращена ошибка, например, с кодом `"consent_required"`;

– "select_account" — СИ следует предложить пользователю выбрать учетную запись. Предложение оказывается полезным, если у пользователя имеется несколько учетных записей и нужно определить подходящую для текущего сеанса. Если СИ не может получить информацию о сделанном пользователем выборе учетной записи, то должна быть возвращена ошибка, например, с кодом "account_selection_required".

Приглашение "none" не должно включаться в prompt с какими-либо другими приглашениями.

Если параметр scope запроса на узел Authorization включает значение "offline_access", то после обработки запроса ПС может иметь доступ к ресурсам пользователя, даже если тот не активен (офлайн). Такой запрос должен содержать параметр prompt со значением "consent", которое обязует СИ запросить разрешение пользователя на доступ к его ресурсам. Разрешение должно запрашиваться всякий раз при передаче scope со значением "offline_access", предыдущие разрешения могут оказаться недостаточными. Исключением из этого правила являются ситуации, когда пользователь дает разрешение на доступ другим способом, например во время регистрации.

Б.1.23 Параметр redirect_uri

Параметр redirect_uri содержит адрес узла Redirection. Адрес используется при перенаправлении ПС ответов СИ через КП пользователя.

Параметр redirect_uri указывается в запросах на узлы Authorization и Token.

ПС должна указать в запросе на узел Authorization один из адресов перенаправления, согласованных с СИ во время регистрации. СИ должна проверить, что передан зарегистрированный адрес. Адреса должны сравниваться как строки, посимвольно.

ПС должна указать в запросе на узел Token тот же адрес, что был передан ранее в запросе на узел Authorization. СИ должна проверить совпадение адресов.

Формат параметра:

```
redirect-uri = URI-reference
```

Б.1.24 Параметр refresh_token

Параметр refresh_token содержит БО, выпущенный СИ. БО используется при обновлении билетов, в том числе самого БО.

Параметр refresh_token возвращается в ответе с узла Token и передается в запросе на этот узел. БО должен включаться в успешный ответ, если параметр scope запроса авторизации/аутентификации содержит значение "offline_access".

БО должен содержать не менее 128 бит энтропии.

При обновлении БО область действия, заданная параметром scope, должна сохраняться. Получив обновленный БО, ПС должна использовать его вместо текущего.

Формат параметра:

```
refresh-token = 22*VSCHAR
```

Б.1.25 Параметр request

Параметр request содержит объект JWT, в котором в качестве утверждений представлены параметры запроса. Объект можно подписывать и (или) конвертовать и таким образом организовать защиту параметров запроса.

Параметр request указывается в запросе на узел Authorization.

Параметры `request` и `request_uri` не должны включаться в запрос одновременно. Объект JWT, передаваемый в `request`, не должен включать параметры `request` и `request_uri`.

В объекте JWT следует передавать долгосрочные параметры (которые могут быть предварительно подписаны), в том время как волатильные параметры (например, `nonce` или `state`) следует передавать обычным образом. Параметры объекта JWT имеют приоритет перед параметрами, переданными обычным образом.

Параметры `response_type` и `client_id` должны передаваться обычным образом. При включении параметров в объект JWT их значения должны повторяться.

Параметр `scope` также должен передаваться обычным образом и при этом содержать строку `"openid"`, которая указывает на использование технологии OIDC.

Если объект JWT подписывается, то в него следует включать утверждения `iss` и `aud` (см. В.2.1, В.2.3). В утверждении `iss` следует указать идентификатор ПС или другой стороны, подписавшей объект. В утверждении `aud` следует указать идентификатор СИ.

Объект JWT может не только подписываться, но и конвертоваться на открытом ключе СИ после подписания или конвертоваться без подписания. Конвертование перед подписанием запрещено.

Формат параметра:

```
request = jwt
```

Б.1.26 Параметр `request_uri`

Параметр `request_uri` содержит сетевой адрес объекта JWT с параметрами запроса. Содержание объекта и правила работы с ним определены при описании параметра `request`. Отличие только в том, что в `request` объект передается по значению, а в `request_uri` — по ссылке. Передача по ссылке может быть удобнее, если объект имеет большой размер.

Параметр `request_uri` указывается в запросе на узел Authorization.

Параметры `request` и `request_uri` не должны включаться в запрос одновременно.

Формат параметра:

```
request-uri = URI-reference
```

Если объект JWT не подписан или если подпись не может быть проверена СИ, то в адресе объекта JWT должна использоваться схема `https`. СИ должна иметь доступ к объекту по указанному адресу. Следует также предусмотреть доступ ПС.

Б.1.27 Параметр `response_mode`

Параметр `response_mode` содержит предпочтительный режим ответа СИ на запрос авторизации/аутентификации.

Параметр `response_mode` указывается в запросе на узел Authorization.

Параметр не рекомендуется использовать в тех случаях, когда запрашиваемый режим ответа является режимом по умолчанию в соответствии с типом, переданным в `response_type`.

Формат параметра:

```
response-mode = 1*( "_" / DIGIT / ALPHA )
```

Предусмотрены следующие режимы ответа:

– `"query"` — параметры ответа кодируются по схеме Query (см. 8.1);
– `"fragment"` — параметры ответа кодируются по схеме Fragment.

Б.1.28 Параметр `response_type`

Параметр `response_type` определяет тип ответа на запрос авторизации/аутентификации и неявно задает коммуникационную схему, которая будет использоваться.

Параметр `response_mode` указывается в запросе на узел Authorization.

Формат параметра:

```
response-type = response-name *( SP response-name )
response-name = 1*response-char
response-char = "_" / DIGIT / ALPHA
```

Допустимые значения параметра определены в таблице 6.

Б.1.29 Параметр `scope`

Параметр `scope` определяет область действия запроса авторизации, в том числе используемую технологию (OIDC), перечень выпускаемых билетов, права доступа к ресурсам пользователей. Область кодируется лексемами, разделяемыми пробелами. С увеличением числа лексем область становится шире.

Параметр `scope` указывается в запросах на узлы Authorization и Token, возвращается в ответах с узлов Authorization, Token и UserInfo. В запросах параметр определяет запрашиваемую область действия, в успешных ответах — разрешенную (согласованную) область действия, в ответах об ошибке (с узла UserInfo) — область действия, которая требуется для получения доступа.

Параметр `scope` должен включаться в успешный ответ, если согласованная область действия отличается от запрошенной. Если области совпадают, то включение не обязательно.

Область действия в запросе на обновление билетов не должна быть шире, чем область действия в запросе авторизации/аутентификации. Наоборот, область может сужаться. Если область действия не указывается в запросе на обновление билетов, то СИ должна полагать, что повторяется область соответствующего запроса авторизации/аутентификации.

Формат параметра:

```
scope = scope-token *( SP scope-token )
scope-token = 1*NQCHAR
```

Область действия должна содержать лексему `"openid"`, указывающую на использование технологии OIDC.

Область действия может включать лексему `"offline_access"`, если от СИ требуется возврат БО.

Область действия может содержать дополнительные лексемы, которые определяют запрашиваемые ПС в виде утверждений ресурсы пользователя (см. В.4.1).

Лексемы `scope`, которые не могут быть распознаны, должны игнорироваться.

Б.1.30 Параметр `state`

Параметр `state` содержит состояние ПС — волатильные данные, связывающие запросы авторизации/аутентификации с соответствующими ответами. Связывание блокирует атаки типа CSRF (см. раздел Ж.5).

Параметр `state` указывается в запросе на узел Authorization и возвращается в соответствующих ответах. Состояние должно включаться в ответ (успешный и об ошибке), если оно указано в запросе.

Состояние из запроса должно повторяться в ответе. При получении ответа ПС должна проверять данный факт.

Формат параметра:

```
state = 1*VSCHAR
```

Б.1.31 Параметр `token_type`

Параметр `token_type` определяет тип БД, выпущенного СИ.

Параметр `token_type` указывается в ответе с узла Token. Параметр должен включаться в ответ, если ответ содержит параметр `access_token`.

Формат параметра:

```
token-type = type-name / URI-reference
type-name = 1*name-char
name-char = "-" / "." / "_" / DIGIT / ALPHA
```

Параметр должен принимать значение `"Bearer"`. Использование других типов БД выходит за пределы настоящего стандарта.

Б.1.32 Параметр `ui_locales`

Параметр `ui_locales` содержит перечень предпочтительных языков интерфейса пользователя. Перечень состоит из тегов, формируемых по правилам [14]. Теги разделяются пробелами и перечисляются в порядке убывания предпочтения.

Параметр `ui_locales` указывается в запросе на узел Authorization.

СИ должна выбирать язык утверждений с учетом перечисленных в `ui_locales` тегов и их очередности. При обработке запроса СИ не следует возвращать ошибку, даже если ни один из запрашиваемых языков не поддерживается.

Формат параметра:

```
ui-locales = locales-name *( SP locales-name )
locales-name = 1*locales-char
locales-char = "-" / DIGIT / ALPHA
```

Б.2 Запрос авторизации/аутентификации

Б.2.1 Подготовка запроса

Запрос аутентификации формируется ПС после того как пользователь инициирует (с помощью КП) процесс аутентификации. Если пользователь может аутентифицироваться с помощью нескольких СИ, то ПС должна явно отслеживать выбор пользователем конкретной СИ, что подразумевает сохранение на ПС информации о выбранной СИ с последующим использованием данной информации для проверки того, что ответы на запросы были получены от нужной СИ. Дополнительно ПС может неявно отслеживать выбор пользователем СИ. Для неявного отслеживания ПС может использовать различные сетевые адреса (URI) перенаправления для различных СИ.

Б.2.2 Обработка запроса

При получении запроса аутентификации СИ должна проверить, что все включенные в запрос параметры являются корректными и все требуемые параметры включены в запрос. СИ должна игнорировать нераспознанные параметры.

Если запрос является корректным, то СИ в зависимости от параметров запроса либо аутентифицирует пользователя, либо определяет, что пользователь уже аутентифицирован. Интерфейс пользователя во время аутентификации определяется в зависимости от параметров запроса и методов аутентификации.

СИ должна попытаться аутентифицировать пользователя в следующих случаях:

1 Пользователь еще не аутентифицирован.

2 Запрос аутентификации содержит параметр `prompt` со значением `"login"`. В этом случае СИ должна повторно аутентифицировать пользователя даже в том случае, если пользователь уже аутентифицирован.

СИ нельзя взаимодействовать с пользователем в случае, если запрос аутентификации содержит параметр `prompt` со значением `"none"`. В данном случае СИ должна возвратить ошибку, если пользователь еще не аутентифицирован или не может быть аутентифицирован способом, не требующим интерактивного взаимодействия с ним.

Если в запросе аутентификации в утверждении `sub` передан идентификатор пользователя, то СИ должна возвратить положительный ответ на запрос в случае, когда пользователь с идентификатором `sub` уже имеет активный аутентифицированный сеанс с СИ или успешно аутентифицирован по данному запросу. Запрос аутентификации с определенным значением утверждения `sub` может быть сделан либо путем включения параметра `id_token_hint`, либо путем включения параметра `claims` с определенным значением для утверждения `sub` (см. раздел В.4).

При включении в параметр `scope` лексемы `"offline_access"` СИ должна его игнорировать, если выполняется хотя бы одно из условий:

– коммуникационная схема, заданная в параметре `response_type`, не предусматривает выпуск БО;

– параметр `prompt` не содержит значение `"consent"`.

Пример * перенаправленного ответа ПС, который инициирует КП на передачу запроса аутентификации на СИ:

```
HTTP/1.1 302 Found
Location: https://server.example.com/authorize?
  response_type=code
  &scope=openid%20profile%20email
  &client_id=s6BhdRkqt3
  &state=af0ifjcsldkj
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
```

Пример запроса, который КП высылает СИ при получении от ПС приведенного выше перенаправленного ответа:

```
GET /authorize?
  response_type=code
  &scope=openid%20profile%20email
  &client_id=s6BhdRkqt3
  &state=af0ifjcsldkj
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb HTTP/1.1
Host: server.example.com
```

Б.2.3 Обработка успешного ответа

При получении успешного ответа с узла Authorization ПС должна проверить, что все требуемые параметры включены в ответ и что все параметры в ответе являются корректными. ПС должна игнорировать нераспознанные параметры ответа.

*В примере здесь и далее не используется полужирный курсив и уменьшенный размер шрифта, которым в стандартах выделяют примеры.

Если в ответе возвращается БА, то ПС должна обработать его по правилам, установленным в разделе Д.2. Если в ответе возвращаются БА и БД, то ПС должна проверить, что хэш-значение в утверждении `at_hash` БА соответствует БД.

Пример успешного ответа для коммуникационной схемы Code:

```
HTTP/1.1 302 Found
Location: https://client.example.org/cb?
  code=SplxlOBeZQQYbYS6WxSbIA
  &state=af0ifjcsldkj
```

Пример успешного ответа для коммуникационной схемы Implicit:

```
HTTP/1.1 302 Found
Location: https://client.example.org/cb#
  access_token=mF_9.B5f-4.1JqM3dpR+G~
  &token_type=bearer
  &id_token=eyJ0...NiJ9.eyJ1c...I6IjIifX0.DeWt4Qu...ZXso
  &expires_in=300
  &state=af0ifjcsldkj
```

Б.2.4 Ответ об ошибке

Если запрос аутентификации признан некорректным по причине отсутствия, неправильности записи или несовпадения URI перенаправления, а также если в запросе идентификатор ПС отсутствует или указан неверно, то СИ должна проинформировать пользователя об ошибке, при этом СИ нельзя автоматически перенаправлять КП по неверному URI перенаправления. При ошибках протокола HTTP указываются подходящие коды статуса HTTP.

Если запрос аутентификации признан некорректным по причинам, не связанным с отсутствием или некорректностью URI перенаправления, или если пользователь не смог пройти аутентификацию или ему было отказано в прохождении аутентификации, то СИ должна возвратить ответ об ошибке с одним из следующих кодов: `invalid_request`, `access_denied`, `unauthorized_client`, `invalid_scope`, `unsupported_response_type`, `server_error`, `temporarily_unavailable`, `interaction_required`, `login_required`, `consent_required`, `account_selection_required`, `invalid_request_uri`, `invalid_request_object`, `request_not_supported`, `request_uri_not_supported`.

Пример ответа об ошибке с перенаправлением КП на ПС:

```
HTTP/1.1 302 Found
Location: https://client.example.org/cb?
  error=invalid_request
  &error_description=Unsupported%20response_type%20value
  &state=af0ifjsldkj
```

Б.3 Запрос на выпуск билетов

Б.3.1 Обработка запроса

При получении запроса на выпуск билетов СИ должна проверить, что все включенные в запрос параметры являются корректными и все требуемые параметры включены в запрос. СИ должна игнорировать нераспознанные параметры.

Если запрос является корректным, то СИ должна выполнить следующие действия:

1 Аутентифицировать ПС, если она еще не была аутентифицирована.

2 Убедиться, что код авторизации был выпущен для ПС, которая обратилась с запросом.

3 Проверить, что у кода авторизации не истек срок действия.

4 Проверить, если это возможно, что код авторизации ранее не использовался. Если код авторизации уже использовался, то СИ должна отказать в обработке запроса, содержащего данный код, и должна отозвать, если это возможно, билет, который был выпущен на основе данного кода авторизации.

5 Если запрос содержит параметр `state`, то проверить, что значение этого параметра совпадает со значением параметра `state` из соответствующего запроса аутентификации.

6 Убедиться, что значение параметра `redirect_uri` совпадает со значением параметра `redirect_uri` из запроса аутентификации. Если параметр `redirect_uri` отсутствует и для ПС зарегистрировано только одно возможное значение для параметра `redirect_uri`, то СИ может возвратить ошибку.

7 Проверить, что код авторизации был выпущен в ответ на запрос авторизации/аутентификации.

Пример запроса на выпуск билетов:

```
POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

grant_type=authorization_code
  &code=SplxlOBeZQQYbYS6WxSbIA
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
```

Б.3.2 Обработка успешного ответа

При получении успешного ответа ПС должна проверить, что все требуемые параметры включены в ответ и что все параметры ответа являются корректными. ПС должна игнорировать нераспознанные параметры ответа.

ПС должна обработать БА по правилам, установленным в разделе Д.2. Если БА включает утверждение `at_hash`, то ПС с его помощью может проверить соответствие БА и БД.

Пример успешного ответа, который включает БД и БА:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "mF_9.B5f-4.1JqM3/3R+G~",
  "token_type": "Bearer",
  "expires_in": 300,
  "id_token": "eyJ0...NiJ9.eyJ1c...I6IjIifX0.DeWt4Qu...ZXso"
}
```

Б.3.3 Ответ об ошибке

Если запрос на выпуск билетов признан некорректным по причине истечения срока действия или некорректности кода авторизации или по другим

причинам, то СИ должна возвратить ответ об ошибке с одним из следующих кодов: `invalid_request`, `invalid_client`, `invalid_grant`, `unauthorized_client`, `unsupported_grant_type`, `invalid_scope`.

Б.4 Запрос на обновление билетов

Б.4.1 Обработка запроса

При получении запроса на обновление билетов СИ должна проверить, что все включенные в запрос параметры являются корректными и все требуемые параметры включены в запрос. СИ должна игнорировать нераспознанные параметры.

Если запрос является корректным, то СИ должна выполнить следующие действия:

1 Убедиться, что БО был выпущен для ПС, от которой он был получен.

2 Убедиться, что ПС аутентифицирована в соответствии с методом, указанным для данной ПС при регистрации.

3 Проверить, что у БО не истек срок действия.

Пример запроса на обновление билетов:

```
POST /token HTTP/1.1
Host: server.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token
  &refresh_token=a7R~64-qTmR.IyT+FwT_t5
```

Б.4.2 Обработка успешного ответа

При получении успешного ответа ПС должна проверить, что все требуемые параметры включены в ответ и что все параметры ответа являются корректными. ПС должна игнорировать нераспознанные параметры ответа.

Если в ответе возвращается БА, то ПС должна обработать его по правилам, установленным в разделе Д.2 с учетом раздела Д.3. Если БА включает утверждение `at_hash`, то ПС с его помощью может проверить соответствие БА и БД.

Пример успешного ответа, который включает только БД:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "t0pL3a.Y8d_vI+5qF~2-h4",
  "token_type": "Bearer",
  "expires_in": 300
}
```

Б.4.3 Ответ при ошибке

Если запрос на обновление билетов признан некорректным, то СИ должна возвратить ответ об ошибке с одним из следующих кодов: `invalid_request`, `invalid_grant`, `invalid_client`, `unsupported_grant_type`.

Б.5 Запрос на узел UserInfo

Б.5.1 Обработка запроса

При получении запроса на узел UserInfo СР должен:

1 Проверить, что БД передан по схеме Bearer.

2 Проверить, что БД имеет корректный формат, включая длину.

3 Проверить, что срок действия БД не истек и что билет не отозван.

Пример запроса:

```
GET /userinfo HTTP/1.1
Host: server.example.com
Authorization: Bearer mF_9.B5f-4.1JqM3dpR+G~
```

Б.5.2 Обработка успешного ответа

При получении успешного ответа ПС должна выполнить следующие действия:

1 Если ответ конвертован, то снять с него защиту, используя ключи и алгоритмы, которые ПС определила для этих целей при регистрации. Если во время регистрации ПС конвертование ответов было для нее предусмотрено, но ответ не конвертован, то его необходимо отклонить.

2 Проверить, что значение утверждения `sub` из ответа совпадает со значением утверждения `sub` из БА.

3 Если ответ подписан, то проверить, что идентификатор СР как издателя ответа, который ПС получает при регистрации, совпадает со значением утверждения `iss`.

4 Если ответ подписан, то проверить, что утверждение `aud` включает идентификатор `client_id`, назначенный ПС при регистрации. Ответ должен быть отвергнут, если идентификатор ПС не включен в `aud` или если `aud` содержит идентификатор стороны, которой ПС не доверяет.

5 Проверить подпись БА. При проверке ПС должна использовать открытые ключи, предоставленные ей при регистрации или полученные другим доверенным способом.

Если какая-либо из проверок не проходит, то утверждения из ответа не должны использоваться ПС.

Пример ответа в виде объекта JSON:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "sub": "248289761001",
  "name": "Victor Mitskevich",
  "given_name": "Victor",
  "family_name": "Mitskevich",
  "preferred_username": "v.mitskevich",
  "email": "vicm@example.com",
  "picture": "http://example.com/vicm/me.jpg"
}
```

Б.5.3 Ответ об ошибке

Если запрос признан некорректным, то СР должен возвратить ПС ответ об ошибке с одним из следующих кодов ошибки: `invalid_request`, `invalid_token`, `insufficient_scope`.

Приложение В
(обязательное)
Утверждения OIDC

В.1 Общие сведения

БА и ответ с узла UserInfo содержат утверждения: об аутентификации и о пользователе. Стандартные утверждения аутентификации определяются в разделе В.2, стандартные утверждения о пользователе — в разделе В.3. Утверждение `sub` содержит идентификатор пользователя, прошедшего аутентификацию, и относится одновременно к обеим группам утверждений.

Утверждения передаются в полях объектов JSON. Значение стандартного утверждения, если не оговорено противное, — строка JSON.

Перечень утверждений может расширяться. При введении дополнительных утверждений требуется определить имена и синтаксис их значений. Следует минимизировать возможность конфликта имен утверждений, например следуя правилам [12] и используя так называемые устойчивые к коллизиям (collision-resistant) имена.

Перечень нестандартных утверждений должен быть предварительно согласован между ПС и СИ. Нераспознанные утверждения должны игнорироваться.

Перечни утверждений, которые включаются в БА и ответ с узла UserInfo, регулируются ПС в запросе авторизации/аутентификации (см. раздел В.4). Запрашиваемые утверждения делятся на существенные и несущественные. Запрашивая утверждение как существенное, ПС информирует о том, что утверждение необходимо для выполнения определенных сервисов, связанных с пользователем. Несущественное утверждение желательно, но не необходимо. Независимо от того, является ли запрошенное утверждение существенным или несущественным, СИ при обработке запроса не должна генерировать ошибку, если утверждение не может быть возвращено по причине отсутствия необходимой информации или из-за отказа пользователя в передаче данной информации (если в описании утверждения не оговорено противное).

В.2 Утверждения аутентификации

В.2.1 Утверждение `iss`

Утверждение `iss` содержит идентификатор СИ, выпустившей БА. Утверждение должно включаться в БА.

Утверждение `iss` может включаться в ответ с узла UserInfo. В этом случае утверждение содержит идентификатор СР, выпустившего ответ. Утверждение следует включать в ответ, если ответ подписывается.

В качестве идентификатора `iss` должен использоваться сетевой адрес (URI). Адрес не должен содержать компоненты `query` и `fragment`. В адресе должна использоваться схема `https`.

В.2.2 Утверждение `sub`

Утверждение `sub` содержит идентификатор пользователя, прошедшего аутентификацию. Утверждение должно включаться в БА.

Утверждение `sub` должно включаться в ответ с узла UserInfo. В этом случае утверждение содержит идентификатор пользователя, которого касается ответ.

В качестве идентификатора `sub` должна использоваться строка, чувствительная к регистру. Строка должна умещаться в буфер из 255 октет. СИ должна назначать пользо-

вателям уникальные идентификаторы (по их запросу или по согласованию с ними). СИ не должна переназначать идентификаторы.

Примечание — Пара идентификаторов `sub` и `iss` может использоваться в роли глобального идентификатора пользователя. Этот идентификатор остается уникальным в инфраструктуре с несколькими СИ. Для сравнения, идентификация через утверждения `email` или `phone_number` не обеспечивает уникальность, поскольку разные СИ могут использовать один и тот же адрес электронной почты или номер телефона пользователя.

В.2.3 Утверждение `aud`

Утверждение `aud` описывает аудиторию БА — перечень сторон, которым билет предназначен. Утверждение должно включаться в БА.

Утверждение `aud` может включаться в ответ с узла UserInfo. В этом случае утверждение описывает аудиторию ответа. Утверждение следует включать в ответ, если ответ подписывается.

Аудитория должна задаваться строкой, чувствительной к регистру, или массивом таких строк. Каждая строка содержит идентификатор отдельной стороны аудитории.

Утверждение `aud` в БА должно включать идентификатор ПС, инициировавшей выпуск билета. В ответе с узла UserInfo в `aud` следует указать идентификатор ПС, которая предоставила БД. В качестве идентификатора ПС должен использоваться параметр `client_id` запроса авторизации/аутентификации.

В.2.4 Утверждение `exp`

Утверждение `exp` описывает время окончания действия БА. Утверждение должно включаться в БА.

При достижении времени, указанного в `exp`, БА должен быть признан действительным. Для учета возможных задержек передачи данных или рассинхронизации таймеров признание БА недействительным может выполняться с небольшой задержкой, как правило, не более чем несколько минут.

Время обработки данного параметра должно предшествовать времени, приведенному в значении параметра. Для учета возможных задержек передачи данных или рассинхронизации часов, средство, реализующее обработку данного параметра, может допускать небольшое отклонение при сравнении значений времени, как правило, не более чем на несколько минут.

Время окончания действия должно задаваться в секундах, которые прошли, начиная с полуночи 1 января 1970 г. по Гринвичу. Время должно быть представлено числом JSON.

В.2.5 Утверждение `iat`

Утверждение `iat` описывает момент выпуска БА. Утверждение должно включаться в БА.

Момент выпуска должно задаваться в секундах, которые прошли, начиная с полуночи 1 января 1970 г. по Гринвичу. Момент должен быть представлен числом JSON.

В.2.6 Утверждение `auth_time`

Утверждение `auth_time` описывает момент аутентификации пользователя, которого касается БА. Утверждение должно включаться в БА, если параметр `max_age` (см. Б.1.20) включен в запрос авторизации/аутентификации или если утверждение `auth_time` объявлено в запросе существенным (см. В.4.2). Утверждение может включаться в БА, даже если данные условия не выполняются.

Момент аутентификации должен задаваться в секундах, которые прошли, начиная с полуночи 1 января 1970 года по Гринвичу. Момент должен быть представлен числом JSON.

В.2.7 Утверждение `nonce`

Утверждение `nonce` содержит синхропосылку, указанную в одноименном параметре запроса авторизации/аутентификации (см. Б.1.21). Утверждение должно включаться в БА, если оно включено в запрос. Синхропосылка из запроса должна быть повторена в билете.

В.2.8 Утверждение `acr`

Утверждение `acr` описывает достигнутый уровень гарантий аутентификации. Утверждение может включаться в БА.

Рекомендуется описывать уровень гарантий по правилам, описанным в Б.1.3.

В.2.9 Утверждение `amr`

Утверждение `amr` содержит перечень использованных методов аутентификации. Под методами понимаются токены и протоколы аутентификации. Утверждение может включаться в БА.

Перечень `amr` должен представлять собой массив строк, чувствительных к регистру. Строка кодирует тип ТА или название протокола. Правила кодирования определяются за пределами настоящего стандарта.

В.2.10 Утверждение `azp`

Утверждение `azp` содержит идентификатор ПС, инициировавшей выпуск БА. Утверждение может включаться в БА.

В качестве идентификатора ПС должен использоваться параметр `client_id` запроса авторизации/аутентификации.

Утверждение `azp` полезно использовать тогда, когда в утверждении `aud` БА указан только один идентификатор, и это не идентификатор ПС.

Утверждение `azp` может включаться в БА, даже если оно дублирует утверждение `aud`.

В.2.11 Утверждение `jti`

Утверждение `jti` содержит уникальный идентификатор БА. Утверждение может включаться в БА для предотвращения его повторного использования.

В качестве идентификатора `jti` должна использоваться строка, чувствительная к регистру. Идентификаторы не должны повторяться.

В.2.12 Утверждение `at_hash`

Утверждение `at_hash` содержит половину хэш-значения БД, который выпускается вместе с БА. Утверждение связывает БД с БА. Утверждение может включаться в БА. Утверждение `at_hash` должно включаться в БА, если данный билет возвращается в ответе с узла Authorization вместе с БД (коммуникационные схемы Implicit и Hybrid).

БД должен хэшироваться с помощью алгоритма, указанного в параметре `alg` заголовка БА. Вторая (правая) половина хэш-значения должна отбрасываться. Оставшаяся часть должна кодироваться по правилам base64url [15], полученный код должен помещаться в `at_hash`.

B.3 Стандартные утверждения о пользователе

B.3.1 Утверждение `name`

Утверждение `name` содержит полное имя пользователя, все его части.

Части имени упорядочиваются в соответствии с региональными настройками и предпочтениями.

B.3.2 Утверждение `given_name`

Утверждение `given_name` содержит собственное имя пользователя.

У пользователя может быть несколько собственных имен, все они могут быть перечислены через пробел.

B.3.3 Утверждение `family_name`

Утверждение `family_name` содержит фамилию пользователя.

У пользователя может быть несколько фамилий, все они могут быть перечислены через пробел. Фамилия может отсутствовать.

B.3.4 Утверждение `middle_name`

Утверждение `middle_name` содержит отчество пользователя.

У пользователя может быть несколько отчеств, все они могут быть перечислены через пробел. Отчество может отсутствовать.

B.3.5 Утверждение `nickname`

Утверждение `nickname` содержит неофициальное имя пользователя. Это имя может совпадать с `given_name` (`"Виктор"`) или отличаться от него (`"Витя"`).

B.3.6 Утверждение `preferred_username`

Утверждение `preferred_username` содержит сокращенное имя пользователя, которое пользователь предпочитает использовать при взаимодействии с ПС. Например, `"v.mitskevich"` или `"vicm"`.

В качестве `preferred_username` может использоваться любая строка JSON, в том числе с пробелами, специальными символами `"@"`, `"/"` и др.

ПС не должна полагаться на уникальность `preferred_username`.

B.3.7 Утверждение `personal_number`

Утверждение `personal_number` содержит идентификационный (личный) номер пользователя.

Примечание — Утверждение `personal_number` расширяет перечень стандартных утверждений о пользователе, установленный в [2].

B.3.8 Утверждение `profile`

Утверждение `profile` содержит сетевой адрес (URL) веб-страницы с профайлом пользователя.

Следует ссылаться на страницу, которая действительно касается пользователя.

B.3.9 Утверждение `picture`

Утверждение `picture` содержит сетевой адрес (URL) файла с фотографией пользователя.

Следует ссылаться на файл графического формата (PNG, JPEG, GIF и др.), а не на веб-страницу, в которую встроена фотография. Следует использовать фотографии, которые действительно представляют пользователя, и избегать неформальных фотографий, выбранных пользователем.

В.3.10 Утверждение `website`

Утверждение `website` содержит сетевой адрес (URL) личной веб-страницы пользователя или его блога.

Следует давать ссылки на интернет-ресурсы, информация на которых публикуется пользователем или его организацией.

В.3.11 Утверждение `email`

Утверждение `email` определяет предпочтительный адрес электронной почты пользователя.

Адрес должен быть задан в формате `addr-spec`, определенном в [16].

ПС не должна полагаться на уникальность `email`.

В.3.12 Утверждение `email_verified`

Утверждение `email_verified` содержит признак проверки адреса электронной почты пользователя, указанного в утверждении `email`. Положительный признак означает, что СИ выполнила определенные действия и убедилась, что пользователь действительно владел адресом в момент проверки (например, ответил на электронное письмо). Способ проверки определяется за пределами настоящего стандарта с учетом специфики инфраструктуры аутентификации.

Признак проверки должен быть задан булевым значением JSON (`true` или `false`).

В.3.13 Утверждение `gender`

Утверждение `gender` определяет пол пользователя.

Пол `gender` должен принимать значения `"male"` (мужской), `"female"` (женский) и другие, определяемые за пределами настоящего стандарта.

В.3.14 Утверждение `birthdate`

Утверждение `birthdate` содержит дату рождения пользователя.

Дата рождения должна задаваться в формате `"YYYYMMDD"`, где `YYYY` — четыре цифры года, `MM` — две цифры номера месяца, `DD` — две цифры дня месяца. Год может принимать значение `"0000"`, что означает, что он опущен. Если требуется указать только год рождения, то должен использоваться формат `"YYYY"`.

В.3.15 Утверждение `zoneinfo`

Утверждение `zoneinfo` определяет часовой пояс пользователя.

Часовой пояс должен задаваться строкой из базы данных часовых поясов [17]. Например, `"Europe/Minsk"`.

В.3.16 Утверждение `locale`

Утверждение `locale` определяет локализацию пользователя в виде тега языка согласно [14].

Обычно локализация задается строкой из двух кодов, разделенных тире: кода языка согласно [18] в нижнем регистре и кода страны согласно [19] в верхнем регистре. Например, `"ru-BY"`.

В некоторых случаях в качестве разделителя вместо тире используется знак подчеркивания: "ru_BY". ПС может использовать именно такой синтаксис.

B.3.17 Утверждение phone_number

Утверждение phone_number определяет предпочтительный номер телефона пользователя.

Номер рекомендуется представлять в формате, определенном в [20]. Например, "+375 17 200 0001". В номере в качестве разделителей могут использоваться пробелы и символы "-", ".", "(", ")".

Если номер телефона содержит расширение, то его рекомендуется представлять в формате, определенном в [21]. Например, "+375 17 200 0001; ext=3456".

B.3.18 Утверждение phone_number_verified

Утверждение phone_number_verified содержит признак проверки номера телефона пользователя, указанного в утверждении phone_number. Положительный признак означает, что СИ выполнила определенные действия и убедилась, что пользователь действительно владел номером в момент проверки (например, предъявил одноразовый пароль, отправленный в виде SMS). Способ проверки определяется за пределами настоящего стандарта с учетом специфики инфраструктуры аутентификации.

Для того чтобы проверка была завершена успешно, номер phone_number, включая возможное расширение, должен соответствовать формату, определенному в [20], [21].

Признак проверки должен быть задан булевым значением JSON (true или false).

B.3.19 Утверждение address

Утверждение address определяет предпочтительный почтовый адрес пользователя.

Адрес должен быть представлен объектом JSON, поля которого соответствуют тем или иным атрибутам адреса. Значениями полей являются строки JSON.

Предусмотрены следующие поля:

– formatted — полный почтовый адрес для отображения на экране или печати на конвертах. В строке formatted могут быть переносы. Они кодируются либо символом «перенос строки» ("\n"), либо парой символов «возврат каретки, перенос строки» ("\r\n");

– street_address — полный адрес улицы, включая ее название, номер дома, номер квартиры и другие данные. В строке street_address могут быть переносы. Они кодируются так же, как переносы в formatted;

– locality — населенный пункт;

– region — регион, область, район;

– postal_code — почтовый индекс;

– country — страна.

По соображениям приватности или из-за отсутствия полной информации в объект address могут включаться только некоторые поля.

Полный адрес, который указывается в formatted, позволяет не включать в address другие поля. Если все-таки некоторое из этих полей включается, то следует обеспечить соответствие его значения сведениям в formatted.

B.3.20 Утверждение updated_at

Утверждение updated_at определяет время последнего обновления информации о пользователе.

Время последнего обновления должно задаваться в секундах, которые прошли, начиная с полуночи 1 января 1970 г. по Гринвичу. Время должно кодироваться числом JSON.

B.4 Запрос утверждений

B.4.1 Запрос утверждений с помощью параметра `scope`

Утверждения о пользователе могут быть запрошены с помощью параметра `scope`.

Параметр включается в запрос авторизации/аутентификации и учитывается СИ при выпуске БД и БА. Утверждения возвращаются в ответе с узла UserInfo при предъявлении БД. Если БД не выпускается, то утверждения возвращаются в БА.

Утверждения, которые запрашиваются с помощью параметра `scope`, считаются несущественными.

Перечень запрашиваемых утверждений регулируется следующими лексемами `scope`:

– `"profile"` — запрашиваются утверждения профиля по умолчанию: `name`, `family_name`, `given_name`, `middle_name`, `nickname`, `preferred_username`, `profile`, `picture`, `website`, `gender`, `birthdate`, `zoneinfo`, `locale`, `updated_at`;

– `"email"` — запрашиваются утверждения `email` и `email_verified`;

– `"address"` — запрашиваются утверждения `address`;

– `"phone"` — запрашиваются утверждения `phone_number` и `phone_number_verified`.

Пример запроса утверждений с помощью параметра `scope`:

```
scope=openid profile email phone
```

B.4.2 Запрос утверждений с помощью параметра `claims`

Утверждения могут быть запрошены с помощью параметра `claims`. Параметр включается в запрос авторизации/аутентификации и учитывается СИ при выпуске БД и БА. Утверждения возвращаются в ответе с узла UserInfo при предъявлении БД и (или) в БА. Параметр `claims` определяет перечень запрашиваемых утверждений, способ их возврата. В `claims` предусмотрена детализация запроса по каждому утверждению.

Запрос `claims` представляется объектом JSON, который включает два вложенных объекта: `userinfo` и `id_token`. В объекте `userinfo` перечисляются утверждения, которые следует возвратить с узла UserInfo, в объекте `id_token` — утверждения, которые следует возвратить в БА. Оба объекта являются необязательными.

Запрос `claims` может включать другие объекты и, соответственно, перечни утверждений. Например, дополнительный перечень может касаться дополнительного узла ресурсов. Объекты `claims`, которые не могут быть распознаны, должны игнорироваться.

Если в `claims` включен объект `userinfo`, то запрашиваемые в нем утверждения добавляются к перечню утверждений, запрашиваемых через параметр `scope`. Включение `userinfo` в `claims` должно сопровождаться добавлением лексемы `"token"` в параметр `response_type`, т. е. запросом БД.

Если в `claims` включен объект `id_token`, то запрашиваемые в нем утверждения добавляются к перечню утверждений об аутентификации, которые по умолчанию (без явного запроса) включаются в БА.

В объектах `userinfo` и `id_token` имена полей совпадают с именами запрашиваемых утверждений. Поле либо принимает значение `null`, либо значением поля является объект JSON с деталями запроса утверждения.

Значение `null` означает, что утверждение запрашивается в режиме по умолчанию, в частности, как несущественное.

B.4.3 Детали запроса утверждения в `claims`

Объект JSON с деталями запроса утверждения включает поля `essential`, `value` и `values`, каждое из которых является необязательным. Могут вводиться дополнительные поля. Поле, которое не может быть распознано, должно игнорироваться.

Поле `essential` содержит признак существенности утверждения. Например, утверждение `auth_time` запрашивается как существенное следующим образом:

```
"auth_time": {"essential": true}
```

По умолчанию `essential` принимает значение `false`.

Поле `value` задает определенное значение запрашиваемого утверждения. Например, сослаться на пользователя с идентификатором `"248289761001"` при запросе утверждений можно следующим образом:

```
"sub": {"value": "248289761001"}
```

В `value` должно указываться корректное значение утверждения.

Поле `values` задает набор определенных значений запрашиваемого утверждения. Значения задаются в порядке убывания приоритета. Например, запросить второй и первый уровни гарантии аутентификации, отдавая приоритет второму, можно следующим образом:

```
"acr": {"essential": true, "values": ["2", "1"]}
```

В `values` должны указываться корректные значения утверждения.

Пример запроса утверждений через `claims`:

```
{
  "userinfo":
  {
    "given_name": {"essential": true},
    "nickname": null,
    "email": {"essential": true},
    "email_verified": {"essential": true},
    "picture": null
  },
  "id_token":
  {
    "auth_time": {"essential": true},
    "acr": {"values": ["2", "1"]}
   }
}
```

В.4.4 Запрос утверждения `acr`

Запрашивая утверждение `acr` в параметре `claims`, ПС предлагает СИ поддержать один из уровней гарантий пользователя, указанных в качестве допустимых значений `acr`.

Если утверждение `acr` запрашивается как существенное, то СИ должна поддержать предложение ПС. Для этого СИ может провести повторную аутентификацию пользователя. Если СИ не может поддержать один из запрошенных уровней гарантий, то это должно трактоваться как неуспешная попытка аутентификации.

Если утверждение `acr` запрашивается как несущественное, то СИ не обязательно поддерживать один из запрошенных уровней. СИ может возвратить в `acr` достигнутый уровень или вообще не возвращать `acr`.

Передача предпочтительных уровней гарантий аутентификации в параметре `acr_values` (см. Б.1.3) также считается запросом утверждения `acr`. При этом утверждение запрашивается как несущественное. Поведение при запросе `acr` одновременно через `claims` и `acr_values` не определено.

Приложение Г
(обязательное)
Объект JWT

Г.1 Общие положения

БА и защищенные ответы с узла UserInfo представляют собой объекты JWT (JSON Web Token, [12]). Используются объекты трех типов:

1) объект JWS (JSON Web Signature, [22]) — подписанные утверждения;
2) объект JWE (JSON Web Encryption, [23]) — конвертованные утверждения;
3) составной объект JWE — конвертованный объект JWS, т. е. конвертованные подписанные утверждения.

БА представляет собой либо объект JWS, либо составной объект JWE. Ответы с узла UserInfo могут быть объектами любого типа.

Объект JWT состоит из нескольких кодовых слов, разделенных символом `"."` (точка). Кодовые слова описывают компоненты объекта. Компоненты представляют собой либо двоичные данные (по умолчанию, если не оговорено противное), либо объекты JSON. Компоненты кодируются по правилам base64url [15]. Компоненты, которые являются объектами JSON, предварительно кодируются по правилам UTF8 [24].

Число компонентов зависит от типа объекта. Первым компонентом всегда является заголовок. Он является объектом JSON.

Г.2 Объект JWS

Объект JWS состоит из трех компонентов:

1) заголовок;
2) подписываемые данные;
3) подпись.

Заголовок описывает алгоритмы выработки и проверки ЭЦП. Предусмотрены следующие варианты (см. определения алгоритмов и параметров в СТБ 34.101.45 и СТБ 34.101.77):

– `"BIGNS128"` — алгоритмы `bign-with-hbelt` с параметрами `bign-curve256v1`;
– `"BIGNS192"` — алгоритмы `bign-with-bash384` с параметрами `bign-curve384v1`;
– `"BIGNS256"` — алгоритмы `bign-with-bash512` с параметрами `bign-curve512v1`.

Выбранный вариант должен быть указан в параметре `alg` заголовка.

Пример заголовка объекта JWS: `{"alg":"BIGNS128"}`.

Подписываемые данные представляют собой объект JSON с утверждениями.

Подпись вычисляется от слова, составленного из кодового представления заголовка, затем точки, затем кодового представления подписываемых данных.

Г.3 Объект JWE

Объект JWE состоит из следующих компонентов:

1) заголовок;
2) защищенный ключ (токен ключа);
3) синхропосылка;
4) зашифрованные данные;
5) имитовставка.

Заголовок описывает алгоритмы защиты ключей и данных.

Предусмотрены следующие варианты алгоритмов защиты данных (см. определения алгоритмов в СТБ 34.101.31):

– `"BELT-DWP"` — алгоритмы аутентифицированного шифрования `belt-dwp`;

– `"BELT-CHE"` — алгоритмы аутентифицированного шифрования `belt-che`.

Выбранный вариант должен быть указан в параметре `enc` заголовка.

Предусмотрены следующие варианты алгоритмов защиты ключей (см. определения алгоритмов и параметров в СТБ 34.101.45):

– `"BIGNT128"` — алгоритмы транспорта ключа `bign-keytransport` с параметрами `bign-curve256v1`;

– `"BIGNT192"` — алгоритмы транспорта ключа `bign-keytransport` с параметрами `bign-curve384v1`;

– `"BIGNT256"` — алгоритмы транспорта ключа `bign-keytransport` с параметрами `bign-curve512v1`.

Выбранный вариант должен быть указан в параметре `alg` заголовка.

В заголовок составного объекта JWE (конвертованные подписанные данные) дополнительно должен быть включен параметр `cty` со значением `"JWT"`. Включение этого параметра в обычный объект JWE не рекомендуется.

Пример заголовков объекта JWE: `{"alg":"BIGNT128","enc":"BELT-DWP"}`, `{"alg":"BIGNT256","enc":"BELT-CHE","cty":"JWT"}`.

Ключ защиты данных должен вырабатываться случайным или псевдослучайным образом в соответствии с требованиями СТБ 34.101.31. Защищенный ключ указывается во второй компоненте объекта JWE.

Алгоритм аутентифицированного шифрования, который используется для установки защиты данных, принимает на вход критические данные, ассоциированные открытые данные, ключ защиты данных и синхропосылку. Алгоритм возвращает зашифрованные данные и имитовставку. Синхропосылка указывается в третьей компоненте объекта JWE, зашифрованные данные — в четвертой, имитоставка — в пятой.

Синхропосылка выбирается произвольным образом. Нулевая синхропосылка считается синхропосылкой по умолчанию, она может кодироваться пустой строкой.

Критическими данными является либо кодовое представление объекта JSON с утверждениями, либо вложенный объект JWS. В качестве ассоциированных открытых данных должно выступать кодовое представление заголовка объекта JWE.

Приложение Д
(обязательное)
Билет аутентификации OIDC

Д.1 Структура билета

БА включает утверждения аутентификации, определенные в разделе В.2. БА может дополнительно включать утверждения о пользователе, определенные в разделе В.3, а также другие утверждения. Перечень утверждений, которые следует включить в БА, уточняется в запросе авторизации/аутентификации (см. раздел В.4).

Пример набора утверждений БА:

```
{
  "iss": "https://server.example.com",
  "sub": "24400320",
  "aud": "s6BhdRkqt3",
  "nonce": "n-0S6_WzA2Mj",
  "exp": 1311281970,
  "iat": 1311280970,
  "auth_time": 1311280969,
  "acr": "30"
}
```

БА представляет собой объект JWT, в котором утверждения подписываются и, возможно, дополнительно конвертуются.

Д.2 Обработка билета

ПС должна обработать полученный БА следующим образом:

1 Если БА конвертован, то снять с него защиту, используя ключи и алгоритмы, которые ПС определила для этих целей при регистрации. Если во время регистрации ПС конвертование билетов было для нее предусмотрено, но БА не конвертован, то его необходимо отклонить.

2 Проверить подпись БА. При проверке ПС должна использовать открытые ключи, предоставленные ей при регистрации или полученные другим доверенным способом.

3 Если БА получен напрямую от аутентифицированной СИ (с узла Token), то проверка подписи БА может опускаться.

4 Проверить, что идентификатор СИ как издателя БА совпадает с идентификатором, указанным в утверждении `iss`.

5 Проверить, что утверждение `aud` включает идентификатор `client_id`, назначенный ПС при регистрации. БА должен быть отвергнут, если идентификатор ПС не включен в `aud` или если `aud` содержит идентификатор стороны, которой ПС не доверяет.

6 Если утверждение `aud` содержит несколько идентификаторов, то следует проверить, что утверждение `azp` включено в БА.

7 Если утверждение `azp` включено в БА, то следует проверить, что идентификатор `client_id`, который был назначен ПС при регистрации, совпадает с идентификатором в `azp`.

8 Проверить, что текущее время предшествует времени в утверждении `exp`.

9 Утверждение `iat` может быть использовано для отклонения билетов, которые были выпущены слишком давно в сравнении с текущим временем. Допустимое превышение

времени может являться специфическим для каждой ПС и определяется с учетом времени хранения на ПС синхропосылок запроса авторизации/аутентификации.

10 Если утверждение `jti` включено в БА, то проверить, что представленный в нем идентификатор не использовался в других доступных билетах.

11 Проверить наличие в БА утверждения `nonce`: оно должно присутствовать, если билет возвращен с узла Authorization или если билет возвращен с узла Token и запрос авторизации/аутентификации включал параметр `nonce`. При наличии утверждения `nonce` сравнить указанную в нем синхропосылку с синхропосылкой в параметре `nonce` запроса. Если обнаружено несовпадение синхропосылок или если сравнение не может быть выполнено (например, по причине истечения срока хранения синхропосылок) или если ПС обнаруживает, что БА с данной синхропосылкой уже обрабатывался, то отклонить билет. ПС может использовать дополнительные методы противодействия атакам, основанным на повторах БА.

12 Если запрашивалось утверждение `acr`, то проверить, что запрос корректно обработан (см. В.2.8).

13 Если запрашивалось утверждение `auth_time` (через параметр `claims` или `max_age`), то проверить, что запрос корректно обработан. Проанализировать момент аутентификации пользователя, указанный в утверждении, и запросить повторную аутентификацию, если время, которое прошло с этого момента, больше допустимого.

Если какая-либо из проверок завершается с ошибкой, то БА должен быть отклонен.

Любые утверждения, которые содержатся в БА и не могут быть распознаны, должны игнорироваться ПС.

Д.3 Обновление билета

При выпуске БА в результате обработки запроса на обновление должны соблюдаться следующие правила:

1 Утверждения `iss`, `sub`, `aud` в первоначальном и обновленном БА должны совпадать.

2 В утверждении `iat` должен быть указан момент выпуска обновленного БА.

3 Если первоначальный БА содержал утверждение `auth_time`, то оно должно быть повторено в обновленном БА.

4 Если первоначальный БА содержал утверждение `azp`, то оно должно быть повторено в обновленном БА. Если утверждение `azp` отсутствовало в первоначальном БА, то оно должно отсутствовать в обновленном.

В остальном правила выпуска обновленного БА не отличаются от правил выпуска первоначального.

Приложение Е
(рекомендуемое)
Выбор уровня гарантий

Е.1 Ошибки

Уровень гарантий идентификации, аутентификации и федерации выбирается по результатам анализа последствий ошибок. Ошибки могут быть вызваны атаками. Перечень основных атак представлен в приложении Ж.

Основная ошибка идентификации — это некорректное связывание аттестата (учетной записи) пользователя с ним самим. Другие ошибки: сбор некорректных идентификационных данных, сбор лишних данных.

Основная ошибка аутентификации — это некорректный вывод о подлинности стороны. Ошибка, как правило, индуцируется атаками противника.

Основная ошибка федерации — это распространение некорректных утверждений об аутентификации пользователя или об его идентификационных данных. Речь идет о подделке билетов или их попадании в руки противника. К ошибкам федерации также относится нарушение отношений доверия, например, между ПС и СИ.

Е.2 Последствия

Типовые категории последствий от ошибок идентификации, аутентификации и федерации и степень их критичности представлены в таблице Е.1. С помощью таблицы следует определить степень критичности ошибок для каждой из категорий.

Таблица Е.1 — Последствия ошибок

Категория последствий	Степень критичности		
	Низкая	Средняя	Высокая
Репутационные потери	Незначительные краткосрочные	Значительные краткосрочные или незначительные долгосрочные	Серьезные долгосрочные, потенциально затрагивающие многие стороны
Финансовые потери	Незначительные	Значительные	Серьезные или катастрофические
Вред организациям/институтам	Заметное снижение эффективности	Существенное снижение эффективности	Потеря работоспособности
Раскрытие конфиденциальной информации	Частичное раскрытие, несущественная информация	Существенная информация	Критическая информация
Личная безопасность	Легкие травмы, без медицинского вмешательства	Средний риск легких травм или незначительный риск травм, требующих медицинского вмешательства	Серьезные травмы или смерть
Правонарушение	Гражданское или уголовное, не требуется вмешательство сил правопорядка	Гражданское или уголовное, требуется вмешательство сил правопорядка	Гражданское или уголовное, требуется вмешательство крупных сил правопорядка

Е.3 Выбор уровня

Уровень гарантий выбирается с помощью таблицы Е.2. По таблице определяется минимальный уровень, для которого степень критичности последствий ошибок допустима относительно каждой из категорий последствий. Этот уровень считается рекомендуемым.

Таблица Е.2 — Допустимая степень критичности последствий ошибок

Категория последствий	Уровень гарантий		
	1	2	3
Репутационные потери	Низкая	Низкая Средняя	Низкая Средняя Высокая
Финансовые потери	Низкая	Низкая Средняя	Низкая Средняя Высокая
Вред организациям/институтам	Без последствий	Низкая Средняя	Низкая Средняя Высокая
Раскрытие конфиденциальной информации	Без последствий	Низкая Средняя	Низкая Средняя Высокая
Личная безопасность	Без последствий	Низкая	Низкая Средняя Высокая
Правонарушение	Без последствий	Низкая Средняя	Низкая Средняя Высокая

При выборе уровня гарантий идентификации анализ в соответствии с таблицей Е.2 не проводится и в качестве рекомендуемого сразу назначается уровень 1, если сбор идентификационных данных не предполагается или если корректность идентификационных данных, распространяемых после аутентификации их владельца, не требуется.

Рекомендуемый уровень 1 гарантий аутентификации меняется на уровень 2, если выполнено одно из следующих условий:

– предполагается сбор идентификационных данных;

– планируется открытая публикация идентификационных данных (даже самозаявленных);

– уровень гарантий идентификации выше первого.

При выборе уровня гарантий федерации анализ в соответствии с таблицей Е.2 не проводится и в качестве рекомендуемого сразу назначается уровень 1, если распространение утверждений в федерации не планируется.

Рекомендуемый уровень гарантий может усиливаться по желанию разработчика инфраструктуры аутентификации.

Приложение Ж
(справочное)
Атаки

Ж.1 Атаки при регистрации и подтверждении личности

Выдача себя за другого. Противник регистрируется, выдавая себя за другого пользователя. Например, противник проходит регистрацию, предъявляя чужое удостоверение.

Отказ от регистрации. Противник отказывается от факта регистрации в инфраструктуре. Например, противник утверждает, что кто-то выдал себя за него, предъявив поддельное удостоверение.

Раскрытие секрета. Секрет аутентификации, который СИ передает пользователю в момент регистрации, перехватывает противник. Например, противник перехватывает QR-код секретного ключа генерации одноразовых паролей.

Модификация секрета. Секрет аутентификации, который пользователь передает СИ в момент регистрации, модифицирует противник. Например, противник меняет пароль, который выбирает пользователь при регистрации на сайте СИ.

Неавторизованный выпуск. Противник навязывает СИ выпуск аттестата, предназначенного другому пользователю, но связанному с токеном противника. Например, противник получает сертификат открытого ключа для своего личного ключа, но на чужое доменное имя.

Ж.2 Атаки на токены

Кража. Противник совершает кражу физического токена. Объектом кражи может быть аппаратный КТ, OTP-токен, карта кодов, сетевой токен (сотовый телефон).

Дублирование. Противник дублирует токен без ведома владельца. Например, переписывает пароль, записанный на бумаге, копирует файлы программного КТ, фотографирует карту кодов.

Перехват. Противник перехватывает аутентификаторы в момент их ввода пользователем и определяет по ним будущие аутентификаторы. Например, противник наблюдает за клавиатурой визуально или использует вредоносные программы, которые фиксируют клавиатурные события операционной системы.

Взлом. Противник восстанавливает секрет аутентификации по токену или аутентификаторам, используя аналитические методы. Например, противник определяет секрет программного КТ, проводя словарную атаку на пароль защиты файла с секретом, или находит личный ключ аппаратного КТ, анализируя флуктуации времени выполнения криптографических операций.

Фишинг/фарминг. Противник узнает аутентификаторы, выдавая себя за СИ, и определяет будущие аутентификаторы. Например, противник переадресует пользователя на поддельную страницу аутентификации.

Социальная инженерия. Противник входит в доверие к пользователю и узнает его секрет аутентификации. Например, противник звонит по телефону, представляется системным администратором и просит сообщить пароль.

Угадывание. Противник угадывает аутентификатор в ходе выполнения протокола аутентификации. Например, противник угадывает статический пароль (предъявляя наиболее вероятные варианты) или одноразовый пароль, пробуя случайные варианты в комбинации с известными логинами.

Ж.3 Атаки на протоколы аутентификации

Подбор аутентификатора. Противник в последовательных сеансах аутентификации пытается угадать аутентификатор, проверяя различные его варианты. Например, аутентификатором является статический пароль, и противник проверяет часто используемые пароли. Противник может использовать словарь таких паролей (словарные атаки).

Фишинг. Противник вынуждает пользователя раскрыть секрет аутентификации, идентификационные данные и др. Используя раскрытые данные, противник выдает себя за пользователя перед СИ. Например, пользователь получает электронное письмо с приглашением пройти аутентификацию перед СИ, контролируемой противником, и в ходе аутентификации раскрывает свой статический пароль.

Фарминг. Противник перенаправляет пользователя на поддельную СИ, манипулируя таблицами сетевых адресов. Пользователь проходит аутентификацию, не подозревая, с кем он взаимодействует.

Перехват. Противник перехватывает сообщения протокола и обрабатывает их, надеясь получить информацию, которая позволит выдать себя за пользователя. Например, противник перехватывает хэш-значение статического пароля. Обработка состоит в переборе подходящих паролей, их хэшировании и сравнении результатов с данными перехвата.

Раскрытие. Противник перехватывает сообщения протокола и определяет приватные идентификационные данные или даже просто идентификатор пользователя, который выполняет протокол аутентификации. Например, используется клиентская аутентификация TLS, и пользователь передает СИ в открытом виде свой сертификат с персональными данными.

Повтор. Противник повторяет определенные сообщения предыдущего сеанса протокола, пытаясь выдать себя за пользователя в текущем сеансе. Например, противник предъявляет хэш-значение статического пароля, перехваченное в предыдущем сеансе.

Противник посередине. Противник встраивается во взаимодействие между пользователем и СИ во время выполнения протокола аутентификации. Противник выдает себя за пользователя перед СИ и (или) за СИ перед пользователем. Например, противник открывает поддельную СИ, которой пользователь пересылает свой одноразовый пароль. Противник сразу же отсылает этот пароль настоящей СИ, выдавая себя за пользователя.

Вредоносные программы. Противник встраивает в среду эксплуатации КП вредоносное программное обеспечение, с помощью которого контролирует ход протокола или компрометирует ТА.

Снижение уровня. Противник навязывает выполнение менее стойкого протокола или протокола с ослабленными параметрами.

Ж.4 Атаки на аттестаты

Раскрытие секретов при хранении. Противник получает доступ к базе данных СИ, в которой хранятся секретные аттестаты, и раскрывает секреты аутентификации. Например, противник получает файл пар (логин, пароль) и узнает пароли пользователей.

Модификация аттестатов при хранении. Противник получает доступ к базе данных СИ, в которой хранятся аттестаты, и изменяет их. Например, противник меняет соответствие в парах (логин, пароль) или устанавливает свои пароли.

Раскрытие секрета при перевыпуске. Секрет аутентификации, который СИ передает пользователю при перевыпуске аттестата, перехватывает противник.

Модификация секрета при перевыпуске. Секрет аутентификации, который пользователь передает СИ при перевыпуске, меняет противник.

Неавторизованный перевыпуск. Противник навязывает СИ перевыпуск аттестата другого пользователя, связывая аттестат со своим токеном. Противник перевыпускает аттестат, предъявляя просроченный или отозванный токен.

Использование отозванных токенов. Противник проходит аутентификацию с помощью уже отозванного токена, используя задержку в оповещении об отзыве или отсутствие оповещения. Например, оповещение реализуется еженедельной публикацией списка отозванных сертификатов. Противник пользуется отозванным токеном вплоть до публикации.

Использование просроченных токенов. Противник использует аппаратный токен по истечении срока его действия. Например, противник определяет конфиденциальное содержимое предыдущих сеансов связи, организованных с помощью токена.

Ж.5 Атаки на билеты и сеансы

Подделка БА. Противник выпускает поддельный БА или модифицирует уже выпущенный. Например, противник продлевает срок действия БА. Поддельный билет открывает несанкционированный доступ к ресурсам ПС.

Раскрытие утверждений. Противник раскрывает приватные утверждения о пользователе, которые содержатся в БА или в ответах на БД.

Оспаривание выпуска БА. СИ отказывается от факта выпуска БА, утверждая, что билет выпущен другой стороной.

Оспаривание предъявления БА. Пользователь отказывается от факта предъявления БА и, как следствие, от факта выполнения операций с ПС. Пользователь утверждает, что билет предъявила другая сторона.

Перенаправление БА. Противник использует БА, выпущенный для одной ПС, для получения доступа к ресурсам другой ПС.

Повторное использование БА. Противник использует БА, который уже ранее использовался законным пользователем билета.

Подделка аутентификатора. Противник выпускает вторичный аутентификатор другого пользователя, и этот аутентификатор принимается другими сторонами.

Перехват аутентификатора. Противник перехватывает вторичный аутентификатор другого пользователя. Перехват выполняется во время передачи аутентификатора от СИ к КП или от КП к ПС. Противник проводит атаки «кража сеанса», «противник посередине» и др.

Подмена ответов. Противник вмешивается в передачу данных между ПС и другими сторонами и в ответ на вторичный аутентификатор одного пользователя возвращает утверждения или билеты другого. Например, противник меняет порядок запросов и ответов.

Кража сеанса. Противник встраивается во взаимодействие между пользователем и сервером (СИ или ПС) сразу после успешного завершения протокола аутентификации. Противник выдает себя за пользователя перед сервером или наоборот. Например, противник перехватывает БС (куки) или угадывает его.

Примечание — Кража сеанса возможна даже при использовании защищенных TLS-соединений. Кража может быть организована с помощью атаки CSRF (Cross Site Request Forgery, межсайтовая подделка запроса). В атаке CSRF сервер противника содержит ссылку на узел ПС. Пользователь автоматически перенаправляется на этот узел при каждом заходе на сервер противника. Если в момент перенаправления пользователь прошел аутентификацию перед СИ по запросу ПС и информация об этом сохранилась в куки его браузера, то эти куки могут быть автоматически предъявлены ПС. ПС может оказать пользователю цифровые услуги даже без его ведома.

Библиография

[1] Berners-Lee T., Fielding R., Masinter L. Uniform Resource Identifier (URI): Generic Syntax. Request for Comments: 3986, 2005

[2] Sakimura N., Bradley J., Jones M., de Medeiros B., Mortimore C. OpenID Connect Core 1.0 incorporating errata set 1. Avail. at `http://openid.net/specs/openid-connect-core-1_0.html`, 2014

[3] Hardt D., ed. The OAuth 2.0 Authorization Framework. Request for Comments: 6749, 2012

[4] Jones M., Hardt D. The OAuth 2.0 Authorization Framework: Bearer Token Usage. Request for Comments: 6750, 2012

[5] ISO/IEC 29115:2013, Information technology — Security techniques — Entity authentication assurance framework
(Информационные технологии. Технологии безопасности. Инфраструктура аутентификации)

[6] Grassi P., Garcia M., Fenton J. NIST Special Publication 800-63-3. Digital Identity Guidelines. U.S. Department of Commerce, National Institute of Standards and Technologies, 2017

[7] Grassi P., Garcia M., Lefkovitz N., Danker J., Choong Y., Greene K., Theofanos M. NIST Special Publication 800-63-3A. Digital Identity Guidelines. Enrollment and Identity Proofing. U.S. Department of Commerce, National Institute of Standards and Technologies, 2017

[8] Grassi P., Fenton J., Newton E., Perlner R., Regenscheid A., Burr W., Richer J., Lefkovitz N., Danker J., Choong Y., Greene K., Theofanos M. NIST Special Publication 800-63-3B. Digital Identity Guidelines. Authentication and Lifecycle Management. U.S. Department of Commerce, National Institute of Standards and Technologies, 2017

[9] Grassi P., Richer J., Squire S., Fenton J., Nadeau E., Lefkovitz N., Danker J., Choong Y., Greene K., Theofanos M. NIST Special Publication 800-63-3C. Digital Identity Guidelines. Federation and Assertion. U.S. Department of Commerce, National Institute of Standards and Technologies, 2017

[10] HTML 4.01 Specification. W3C Recommendation 24 December 1999. Avail. at `https://www.w3.org/TR/1999/REC-html401-19991224/`, 1991

[11] Crockford D. The application/json Media Type for JavaScript Object Notation (JSON). Request for Comments: 4627, 2006

[12] Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). Request for Comments: 7519, 2015

[13] Crocker D., ed. Augmented BNF for Syntax Specifications: ABNF. Request for Comments: 5234, 2008

[14] Phillips A., Davis M. Tags for Identifying Languages. Request for Comments: 5646, 2009

[15] Josefsson S. The Base16, Base32, and Base64 Data Encodings. Request for Comments: 4648, 2006

[16] Resnick P., ed. Internet Message Format. Request for Comments: 5322, 2008

[17] Lear E., Eggert P. Procedures for Maintaining the Time Zone Database. Request for Comments: 6557, 2012

[18] ISO 639-1:2002, Codes for the representation of names of languages — Part 1: Alpha-2 code
(Коды для представления названий языков. Часть 1. Двухбуквенный код)

[19] ISO 3166-1:1997, Codes for the representation of names of countries and their subdivisions — Part 1: Country codes
(Коды для представления названий стран и их подразделений. Часть 1. Коды стран)

[20] E.164: The international public telecommunication numbering plan. International Telecommunication Union, 2010

[21] Schulzrinne H. The tel URI for Telephone Numbers. Request for Comments: 3966, 2004

[22] Jones M., Bradley J., Sakimura N. JSON Web Signature (JWS). Request for Comments: 7515, 2015

[23] Jones M., Hildebrand J. JSON Web Encryption (JWE). Request for Comments: 7516, 2015

[24] ISO/IEC 10646:2012, Information technology — Universal Coded Character Set (UCS)
(Информационные технологии. Универсальный набор кодированных символов (UCS))