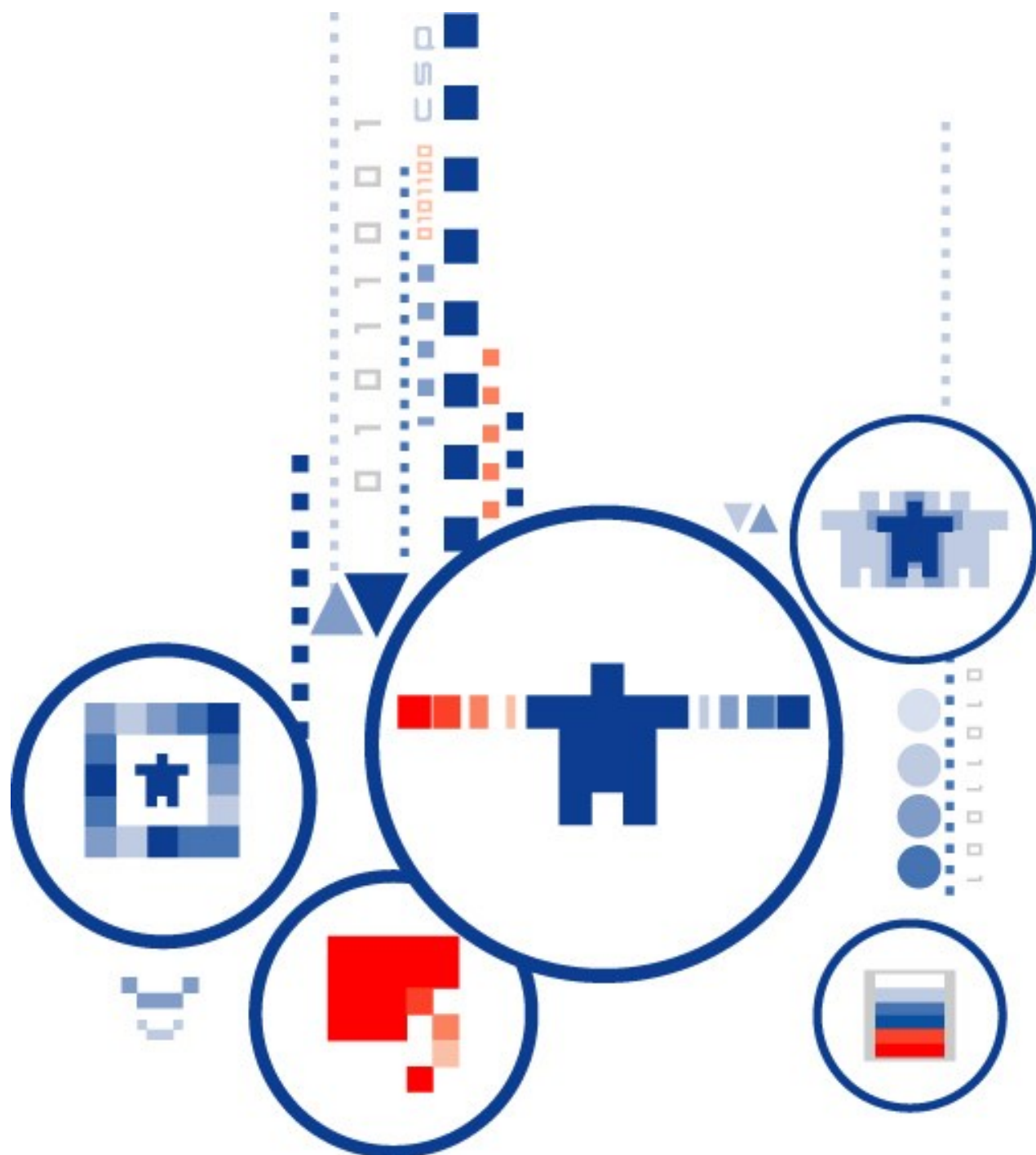




КриптоПро УЦ

программно-аппаратный комплекс
удостоверяющий центр

Версия 2.0

Руководство по эксплуатации

АННОТАЦИЯ

Настоящий документ содержит руководство по эксплуатации Программно-аппаратного комплекса (ПАК) «Удостоверяющий центр «КриптоПро УЦ» версии 2.0 (далее «КриптоПро УЦ»), который обеспечивает реализацию целевых функций удостоверяющего центра как организации.

Приведено описание программных компонентов, организационно-технических мероприятий, связанных с регистрацией пользователей, формированием служебных ключей и сертификатов пользователей и управления ЦР.

Документ предназначен для администраторов и пользователей как пособие по эксплуатации программного обеспечения программно-аппаратного комплекса «КриптоПро УЦ».

Информация о разработчике программно-аппаратного комплекса «КриптоПро УЦ» версии 2.0:

ООО "КРИПТО-ПРО"

127 018, г. Москва, ул. Сущевский вал, 18

Телефон: +7 (495) 995 4820

Факс: +7 (495) 995 4820

<https://www.cryptopro.ru>

E-mail: info@cryptopro.ru

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Используемые в настоящем документе термины описаны в таблице ниже.

Таблица 1. Список терминов

Термин	Определение
Владелец сертификата ключа проверки электронной подписи	Лицо, которому в установленном Федеральным законом (N 63-ФЗ, 2011) порядке выдан сертификат ключа проверки электронной подписи.
Демилитаризованная зона	Технология обеспечения защиты информационного периметра сети предприятия, при которой серверы, отвечающие на запросы из внешней сети, находятся в особом сегменте сети, называемом демилитаризованной зоной (ДМЗ). Компьютеры в ДМЗ ограничены в доступе к основным сегментам сети предприятия с помощью межсетевого экрана (МЭ) с целью минимизировать ущерб при взломе одного из общедоступных сервисов, находящихся в ДМЗ.
Межсетевой экран	Комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.
Сертификат ключа проверки электронной подписи	Электронный документ или документ на бумажном носителе, выданные удостоверяющим центром либо доверенным лицом удостоверяющего центра и подтверждающие принадлежность ключа проверки электронной подписи владельцу сертификата ключа проверки электронной подписи.
Квалифицированный сертификат ключа проверки электронной подписи (квалифицированный сертификат)	Сертификат ключа проверки электронной подписи, выданный аккредитованным удостоверяющим центром или доверенным лицом аккредитованного удостоверяющего центра либо федеральным органом исполнительной власти, уполномоченным в сфере использования электронной подписи.
Средства электронной подписи	Шифровальные (криптографические) средства, используемые для реализации хотя бы одной из следующих функций - создание электронной подписи, проверка электронной подписи, создание ключа электронной подписи и ключа проверки электронной подписи.
Список сертификатов, досрочно прекративших действие	Электронный документ с электронной подписью уполномоченного лица удостоверяющего центра, включающий в себя список серийных номеров сертификатов, действие которых было прекращено до окончания срока их действия. По тексту далее также может использоваться термин "список отозванных сертификатов".

Удостоверяющий центр	Юридическое лицо или индивидуальный предприниматель, осуществляющие функции по созданию и выдаче сертификатов ключей проверки электронных подписей, а также иные функции, предусмотренные Федеральным законом (N 63-ФЗ, 2011).
Электронная подпись	Информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связана с такой информацией и которая используется для определения лица, подписывающего информацию.
Электронный документ	Документ, информация в котором представлена в электронной форме, пригодной для обработки средствами вычислительной техники.
Представитель юридического лица	Указываемое наряду с наименованием юридического лица, физическое лицо, действующее от имени юридического лица без доверенности в соответствии с гражданским законодательством Российской Федерации.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Устойчивые русскоязычные сокращения, используемые в этом документе, описаны в таблице ниже.

Таблица 2. Список сокращений

Сокращение	Значение
АРМ	Автоматизированное рабочее место
ДМЗ	Демилитаризованная зона
МЭ	Межсетевой экран
НСД	Несанкционированный доступ
ОС	Операционная система
ПАК	Программно-аппаратный комплекс
ПО	Программное обеспечение
СКЗИ	Средство криптографической защиты информации
СКПЭП	Сертификат ключа проверки электронной подписи
СУБД	Системы управления базами данных
УЦ	Удостоверяющий центр
ЭП	Электронная подпись

Используемые иностранные сокращения, как правило, англоязычные, приведены в таблице ниже.

Таблица 3. Список иностранных сокращений

Сокращение	Значение	Перевод
AIA	Authority Info Access	Доступ к информации о Центре сертификации
CRL	Certificate Revocation List	Список отозванных сертификатов, СОС
CDP	CRL Distribution Point	Пункт распространения CRL
DN	Distinguished Name	Отличительное имя
IETF	Internet Engineering Task Force	Специальная комиссия интернет разработок
ITU	International Telecommunication Union	Международный союз электросвязи, МСЭ
ITU-T	Telecommunication Standardization Sector (ITU's)	Сектор стандартизации электросвязи (в МСЭ), МСЭ-Т
LDAP	Lightweight Directory Access Protocol	Облегченный протокол доступа к Справочнику
RDN	Relative Distinguished Name	Относительное отличительное имя

Сокращенные названия логических и программных компонентов КриптоПро УЦ используемые в этом документе, приведены в таблице ниже.

Таблица 4. Сокращённые названия компонентов КриптоПро УЦ

Сокращение	Значение
ЦС	Логический компонент УЦ «Центр сертификации»
ЦР	Логический компонент УЦ «Центр регистрации»
АРМ Оператора ЦР	Автоматизированное рабочее место Оператора Центра регистрации
АРМ РКС	Логический и программный компонент УЦ «АРМ разбора конфликтных ситуаций»
КриптоПро УЦ	Программно-аппаратный комплекс «Удостоверяющий центр «КриптоПро УЦ» версии 2.0
Сервер ЦС	Роль УЦ «Сервер центра сертификации»
Сервер ЦР	Роль УЦ «Сервер центра регистрации»
Сервер CDP	Служба роли сервера ЦР «Сервер CDP»
Сервер AIA	Служба роли сервера ЦР «Сервер AIA»
Справочник сертификатов LDAP	Служба роли сервера ЦР «Справочник сертификатов LDAP»

Содержание

Термины и определения	3
Перечень сокращений	5
1 Сервер Центра сертификации	11
1.1 Создание запросов и выпуск сертификатов	11
1.1.1 Выпуск корневого сертификата ЦС	11
1.1.2 Создание запроса на сертификат	12
1.1.3 Выпуск сертификата по запросу	14
1.1.4 Файл со значениями расширений	14
1.1.5 Установка сертификатов в хранилище	17
1.2 Экземпляры Центра сертификации	18
1.2.1 Просмотр списка экземпляров ЦС	18
1.2.2 Вывод информации об экземпляре ЦС	18
1.2.3 Добавление нового экземпляра ЦС	18
1.2.4 Установка экземпляра ЦС по умолчанию	18
1.2.5 Удаление экземпляра ЦС	19
1.2.6 Настройки ЦС	19
1.2.6.1 Редактирование настроек ЦС	19
1.2.6.2 Список адресов точек распространения CRL (CDP)	20
1.2.6.3 Список адресов точек доступа к информации о ЦС (AIA)	21
1.2.6.4 Режим контроля соответствия требований к квалифицированному сертификату.	21
1.2.6.5 Сроки действия	22
1.2.6.6 Точность часов ЦС	22
1.2.6.7 Интервалы перекрытия CRL	22
1.2.6.8 Автозагрузка ключа ЦС	23
1.2.6.9 Алгоритм формирования серийных номеров	23
1.2.6.10 Состав расширения «Идентификатор ключа центра сертификатов»	23
1.2.6.11 Политика криптопровайдеров	23
1.2.6.12 Запрет на выпуск сертификатов с алгоритмом ключа ГОСТ Р 34.10-2001 ..	24
1.2.6.13 Точность часов ЦС	24
1.2.6.14 Шаблон по умолчанию	24
1.2.6.15 Расширение «Доступ к сведениям Центра сертификации» в сертификатах службы OCSP	24
1.2.6.16 Расширение «Размещения опубликованных CRL»	24
1.2.6.17 Расширение «Следующая публикация CRL»	24
1.2.6.18 Кодировка имени субъекта	25
1.2.6.19 Контроль длин атрибутов имени субъекта	25
1.2.6.20 Состав расширения «Использование ключа» в сертификатах ЦС	25
1.2.6.21 Некритическое расширение «Использование ключа»	25
1.2.7 Добавление ключа подписи ЦС	25
1.2.8 Замена ключа подписи ЦС	26
1.2.9 Загрузка ключей подписи ЦС	27
1.2.10 Получение подробной информации о ключе подписи ЦС	28
1.2.11 Получение списка ключей подписи ЦС	28
1.2.12 Выгрузка ключей подписи ЦС	29
1.2.13 Изменение назначения ключа подписи ЦС	29

1.2.14	Вывод из эксплуатации ключа подписи ЦС.....	30
1.3	Шаблоны сертификатов	30
1.3.1	Структура шаблона сертификата	31
1.3.1.1	Свойства шаблона	31
1.3.1.2	Субъект сертификата	31
1.3.1.3	Срок действия сертификата	32
1.3.1.4	Параметры создания ключа	34
1.3.1.5	Расширения сертификата	34
1.3.2	Список шаблонов ЦС	47
1.3.3	Получение шаблона	47
1.3.4	Добавление нового шаблона	47
1.3.5	Редактирование шаблона	47
1.3.6	Удаление шаблона	48
1.3.7	Создание файла с закодированным расширением «Шаблон сертификата»	48
1.3.8	Создание файла с закодированным расширением «Тип идентификации при выдаче сертификата»	48
1.4	Списки отзыва (CRL)	48
1.4.1	Выпуск CRL	48
1.4.2	Получение CRL, действующих на указанный момент времени	49
1.4.3	Расписание выпуска CRL	49
1.4.3.1	Формат расписания выпуска CRL	49
1.4.3.2	Получение расписания выпуска CRL	50
1.4.3.3	Изменение расписания выпуска CRL	50
1.5	Отзыв сертификатов	51
1.5.1	Создание запроса на отзыв сертификата	51
1.5.2	Отправка запроса на отзыв	51
1.6	Получение сертификатов	52
1.7	Получение запросов на сертификат	52
1.8	Получение запросов на отзыв	53
2	Сервер Центра регистрации	55
2.1	Операторы	55
2.1.1	Добавление оператора	55
2.1.2	Обновление данных оператора	56
2.1.3	Получение данных об операторе	56
2.1.4	Получение списка операторов	56
2.1.5	Активация оператора	57
2.1.6	Деактивация оператора	57
2.2	Группы операторов	57
2.2.1	Добавление группы	57
2.2.2	Переименование группы	58
2.2.3	Получение списка групп	58
2.2.4	Получение информации о группе	58
2.2.5	Удаление группы	58
2.2.6	Добавление оператора в группу	58
2.2.7	Удаление оператора из группы	59
2.3	Политики имен	59
2.3.1	Добавление (регистрация) политики имён	61
2.3.2	Получение политики имён	61

2.3.3	Получение списка зарегистрированных политик имён	61
2.3.4	Удаление политики имён	61
2.3.5	Выбор политики имён по умолчанию	62
2.3.6	Обновление политики имён	62
2.4	Шаблоны сертификатов	62
2.4.1	Добавление шаблона сертификата	63
2.4.2	Управление режимом автообновления	63
2.4.3	Получение шаблона сертификата	63
2.4.4	Получение списка шаблонов сертификатов	63
2.4.5	Удаление шаблона сертификата	64
2.4.6	Обновление шаблона сертификата	64
2.5	Издатели сертификатов	64
2.5.1	Добавление (регистрация) издателя сертификатов	64
2.5.2	Получение списка зарегистрированных издателей сертификатов	65
2.5.3	Удаление зарегистрированного издателя сертификатов	65
2.5.4	Выбор издателя по умолчанию	65
2.6	Папки	65
2.6.1	Создание папки	66
2.6.2	Переименование папки	66
2.6.3	Вывод списка папок	66
2.6.4	Получение информации о папке	66
2.6.5	Удаление папки	67
2.6.6	Назначение политики имён для папки	67
2.6.7	Определение набора шаблонов сертификатов	67
2.6.8	Определение набора разрешенных издателей	68
2.7	Настройка безопасности	69
2.7.1	Добавление разрешений	69
2.7.2	Получение информации о разрешениях	70
2.7.3	Удаление разрешений	70
2.8	Настройка публикации сертификатов подписи в AIA	71
2.9	Настройка публикации выпущенных CRL в CDP	71
3	Сервер Реестра сертификатов	72
4	АРМ Оператора ЦР	73
4.1	Подключение Оператора к веб-интерфейсу Центра регистрации	73
4.2	Описание веб-интерфейса	73
4.3	Папки	74
4.4	Создание (регистрация) пользователя	75
4.5	Пользователи	77
4.6	Выпуск сертификата	77
4.6.1	Создание нового запроса на сертификат	77
4.6.2	Загрузка запроса на сертификат из файла	79
4.7	Сертификаты	80
4.8	Отзыв сертификата	81
4.9	Запросы на сертификат	81
4.10	Запросы на отзыв	82
5	Импорт данных из КриптоПро УЦ 2.0 ЖТЯИ.00078-01	84
5.1	Импорт данных Центра сертификации	84
5.1.1	Подготовка к импорту	84

5.1.2	Разворачивание БД ЦС.....	84
5.1.3	Импорт данных ЦС.....	84
5.1.4	Перенос ключей подписи.....	85
5.1.5	Настройки ЦС.....	85
5.1.6	Установка экземпляра ЦС по умолчанию.....	85
5.1.7	Выпуск CRL	85
5.2	Импорт данных Центра регистрации.....	85
5.2.1	Подготовка к импорту	85
5.2.2	Разворачивание БД ЦР	86
5.2.3	Создание операторов ЦР	86
5.2.4	Импорт данных ЦР	86
5.2.5	Настройка издателя по умолчанию	87
5.3	Импорт данных Реестра сертификатов.....	87
5.3.1	Подготовка к импорту	87
5.3.2	Разворачивание БД Реестра сертификатов	88
5.3.3	Импорт данных.....	88
6	Публикация сертификатов в LDAP каталог	89
6.1	Структура настроек модуля публикации	91
6.1.1	Включение публикации	91
6.1.2	Подключение к LDAP каталогу	91
6.1.3	Фильтрация сертификатов.....	91
6.1.4	Правила преобразования субъекта сертификата.....	93
6.1.5	Правила поиска и заполнения объектов LDAP каталога	93
6.2	Пример настроек модуля публикации.....	94
7	Журналирование.....	97

1 Сервер Центра сертификации

1.1 Создание запросов и выпуск сертификатов

1.1.1 Выпуск корневого сертификата ЦС

Выпуск корневого сертификата ЦС осуществляется командой **pkica ca new-root-cert**.

Параметры (звездочкой * помечены обязательные):

-o --out-file	Имя файла для сохранения сертификата. Если не задано, то сертификат будет выведен на консоль в кодировке base64.
-v --validity	Срок действия корневого сертификата. Задается в формате U:N, где U - единицы измерения срока действия, а N - количество таких единиц. N - целое число, U может принимать одно из значений Years, Months, Weeks, Days, Hours, Minutes. Значение по умолчанию: Years:16 (16 лет).
-V --key-validity	Срок действия ключа подписи корневого сертификата. Задается в формате U:N, где U - единицы измерения срока действия, а N - количество таких единиц. N - целое число, U может принимать одно из значений Years, Months, Weeks, Days, Hours, Minutes. Значение по умолчанию: Years:3 (3 года).
--path-length	Ограничение на длину пути.
-i --key-index	* Индекс ключа ЦС.
-s --subject	* Субъект.
--extensions-file	Файл с набором расширений сертификата. Формат файла описан в разделе 1.1.4.
--serial-number-algorithm	Алгоритм серийного номера. Допустимые значения: CryptoPro, Msca. Значение по умолчанию: CryptoPro.
-c --container-name	* Имя контейнера закрытого ключа.
--hash-algorithm	Имя или OID алгоритма хэширования. Если не задано, то будет использован алгоритм по умолчанию для провайдера.
-l --key-length	Длина ключа. Если не задана, то будет использовано значение по умолчанию для провайдера.
--use-existing-key	Использовать существующий ключ.
-e --exportable	Сделать ключ экспортируемым.
--prompt-for-media	Всегда показывать окно выбора носителя (только для КриптоПро CSP).

-p --provider-type	Тип провайдера закрытого ключа. Значение по умолчанию: 80 (ГОСТ 2012 256 бит).
-n --provider-name	Имя провайдера закрытого ключа сертификата ЦС. Если не указано, то будет использован провайдер заданного типа по умолчанию.
--pin	ПИН-код на контейнер. Если не задан, то будет запрошен во время выполнения.
--no-pin	Флаг указывает, что контейнер не имеет ПИН-кода. Подавляет запрос ПИН-кода во время выполнения.
--always-set-pin	Всегда явно задавать ПИН-код при обращении к контейнеру, даже если ПИН-код пустой.
-k --key-spec	Тип ключа. Допустимые значения: Signature, KeyExchange. Значение по умолчанию: Signature (ключ подписи).
--key-location	Расположение контейнера закрытого ключа. Допустимые значения: CurrentUser, LocalMachine. Значение по умолчанию: CurrentUser.
--silent	Работа с криптопровайдером в "тихом" режиме.

1.1.2 Создание запроса на сертификат

Эта операция нужна в сценариях, когда настраивается подчиненный ЦС или выпускаются служебные TLS сертификаты. После создания запрос должен быть передан на ЦС для обработки. Создание запроса осуществляется командой **pkica ca new-cert-req**.

Параметры (звездочкой * помечены обязательные):

-o --out-file	Имя файла для сохранения сертификата. Если не задано, то сертификат будет выведен на консоль в кодировке base64.
-t --template	OID шаблона сертификата.
--template-major-version	Старшая версия шаблона. Значение по умолчанию: 1.
--template-minor-version	Младшая версия шаблона. По умолчанию не задана.
--is-ca	Запрос на сертификат ЦС (добавить расширение Основные ограничения и указать признак ЦС).
--path-length	Ограничение на длину пути (для расширения Основные ограничения).
-i --key-index	Индекс ключа ЦС. Если задан, то будет добавлено расширение Версия ЦС.

--subject-sign-tool	Наименование средства ЭП. Если задано, то будет добавлено соответствующее расширение.
--sign-tool-class	Класс средства ЭП. Если задан, то будет добавлено расширение Политики сертификата с соответствующим значением. Допустимые значения: KC1, KC2, KC3, KB1, KB2, KA1.
--extensions-file	Файл с набором расширений запроса. Формат файла описан в разделе 1.1.4.
-s --subject	* Субъект.
-c --container-name	* Имя контейнера закрытого ключа.
--hash-algorithm	Имя или OID алгоритма хэширования. Если не задано, то будет использован алгоритм по умолчанию для провайдера.
-l --key-length	Длина ключа. Если не задана, то будет использовано значение по умолчанию для провайдера.
--use-existing-key	Использовать существующий ключ.
-e --exportable	Сделать ключ экспортируемым.
--prompt-for-media	Всегда показывать окно выбора носителя (только для КриптоПро CSP).
-p --provider-type	Тип провайдера закрытого ключа. Значение по умолчанию: 80 (ГОСТ 2012 256 бит).
-n --provider-name	Имя провайдера закрытого ключа. Если не указано, то будет использован провайдер заданного типа по умолчанию.
--pin	ПИН-код на контейнер. Если не задан, то будет запрошен во время выполнения.
--no-pin	Флаг указывает, что контейнер не имеет ПИН-кода. Подавляет запрос ПИН-кода во время выполнения.
-k --key-spec	Тип ключа. Допустимые значения: Signature, KeyExchange. Значение по умолчанию: Signature (ключ подписи). Allowed values are: Signature, KeyExchange.
--key-location	Расположение контейнера закрытого ключа. Допустимые значения: CurrentUser, LocalMachine. Значение по умолчанию: CurrentUser.
--silent	Работа с криптопровайдером в "тихом" режиме.

1.1.3 Выпуск сертификата по запросу

Для выпуска на ЦС сертификата по запросу используется команда **pkica ca issue-cert**.

Параметры (звездочкой * помечены обязательные):

-f --file	* Имя файла запроса на сертификат.
-o --out-file	Имя файла для сохранения сертификата. Если не задано, то сертификат будет выведен на консоль в кодировке base64.
-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-t --timeout	Таймаут ожидания выпуска сертификата. Задается в формате ISO 8601: [d.]hh:mm[:ss]. Значение по умолчанию: 00:01 (1 минута). Если равен -1, то ожидание будет бесконечным; если 0, то будет выполнена отправка команды без ожидания выпуска сертификата.

Обращение этой команды к ЦС происходит через NATS, поэтому предварительно должен быть настроен и запущен сервис NATS Streaming.

1.1.4 Файл со значениями расширений

Конфигурационный файл имеет формат JSON и может содержать значения следующих расширений:

- **CertificateTemplateExtension** - шаблон сертификата. Значение расширения задается полями:
 - **TemplateOid** - идентификатор OID шаблона.
 - **MajorVersion** - старшая версия шаблона.
 - **MinorVersion** - младшая версия шаблона (необязательное).
- **CertPoliciesExtension** - политики сертификата. Значение задается полями:
 - **PolicyIdentifiers** - массив идентификаторов OID политик сертификата.
 - **Critical** - флаг критичности расширения. Если не указан, то расширение некритическое.
- **EnhancedKeyUsageExtension** - улучшенный ключ. Значение задается полями:
 - **Usages** - массив идентификаторов OID использования ключа.
 - **Critical** - флаг критичности расширения. Если не указан, то расширение некритическое.
- **IdentificationKindExtension** - тип идентификации. Значение задается полем **IdentificationKind**, которое может принимать одно из следующих значений: **Personal** (личное присутствие), **RemoteCert** (без личного присутствия с использованием квалифицированной ЭП), **RemotePassport** (без личного присутствия с использованием персональных данных, записанных на электронный

носитель из заграничного паспорта), **RemoteSystem** (без личного присутствия с использованием сведений из ЕСИА и ЕБС).

- **KeyUsageExtension** - использование ключа. Значение задается полем **KeyUsages**, которое может принимать одно или несколько значений из списка:
 - **EncipherOnly** - только шифрование.
 - **CrlSign** - подписание списка отзыва сертификатов (CRL).
 - **KeyCertSign** - подпись сертификатов.
 - **KeyAgreement** - согласование ключей.
 - **DataEncipherment** - шифрование данных.
 - **KeyEncipherment** - шифрование ключей.
 - **NonRepudiation** - проверка подлинности.
 - **DigitalSignature** - цифровая подпись.
 - **DecipherOnly** - только расшифровка.
- **SubjectKeyIdentifierExtension** - идентификатор ключа субъекта. Значение задается полем **HexValue**, которое содержит шестнадцатеричную строку.
- **PrivateKeyUsagePeriodExtension** - период использования закрытого ключа. Значение задается двумя необязательными полями **NotBefore** (дата начала действия) и **NotAfter** (дата окончания действия).
- **RawExtension** - произвольное расширение. Значение задается полями:
 - **Oid** - идентификатор OID расширения.
 - **RawData** - значение расширения, закодированное в base64 строку.
 - **Critical** - флаг критичности расширения. При установке этого флага в true следует учитывать, что в прикладных системах проверки, в которых используется сертификат с неизвестным критическим расширением, будут заканчиваться ошибки.
- **SubjectAlternativeNameExtension** - дополнительное имя субъекта. Значение задается полями:
 - **IpAddress** - строка, содержащая IP адрес, помещаемый в поле ipAddress расширения. Допустимо указывать несколько адресов, разделенных символом «;».
 - **DnsName** - строка, содержащая DNS имя, помещаемое в поле dNSName расширения. Допустимо указывать несколько имен, разделенных символом «;».
 - **Email** - строка, содержащая адрес электронной почты, помещаемый в поле rfc822Name расширения.
 - **DirectoryName** - массив пар Oid (идентификатор OID атрибута имени)-Value (значение атрибута имени в виде строки) для формирования поля directoryName расширения, содержащего имя каталога.

- **OtherName** - массив пар Oid (идентификатор OID)- Value (значение) для формирования набора полей otherName расширения, содержащих «Другие имена».
- **SubjectSignToolExtension** - средство ЭП владельца. Значение задается полем **SignTool**, содержащим название средства ЭП.

Пример содержимого файла:

```
[
{
  "SubjectAlternativeNameExtension": {
    "IpAddress": "127.0.0.1;192.168.0.1",
    "DnsName": "localhost;test.cp.ru",
    "Email": "email@cp.ru",
    "DirectoryName": [
      {
        "Oid": "2.5.4.3",
        "Value": "Test"
      },
      {
        "Oid": "2.5.4.10",
        "Value": "TestOrg"
      },
      {
        "Oid": "2.5.4.6",
        "Value": "RU"
      }
    ],
    "OtherName": [
      {
        "Oid": "1.3.6.1.4.1.311.20.2.3",
        "Value": "user@testorg.ru"
      }
    ]
  },
{
  "EnhancedKeyUsageExtension": {
    "Usages": [
      "1.3.6.1.5.5.7.3.4",
      "1.3.6.1.5.5.7.3.2"
    ]
  },
{
  "KeyUsageExtension": {
    "KeyUsages": "KeyEncipherment, DigitalSignature"
  },
{
  "RawExtension": {
    "Oid": "1.2.643.2.2.49.2",
    "RawData":
"0KLQtdGB0YLQvtCy0LDRjyDQu9C40YbQtdC90LfQuNGPINCa0YDQuNC/0YLQvtCf0YDQviBDU1Ag
NS4w",
    "Critical": false
  }
}
```

```

    },
    {
      "PrivateKeyUsagePeriodExtension": {
        "NotAfter": "2024-01-25T20:01:00+03:00"
      }
    },
    {
      "SubjectKeyIdentifierExtension": {
        "HexValue": "a2514e1bf4098f0ea749cbc18131f6bc1048a111"
      }
    },
    {
      "CertificateTemplateExtension": {
        "TemplateOid": "1.2.643.2.2.46.0.8",
        "MajorVersion": 1
      }
    },
    {
      "CertPoliciesExtension": {
        "PolicyIdentifiers": [
          "1.2.643.100.113.1",
          "1.2.643.100.113.2"
        ],
        "Critical": false
      }
    },
    {
      "SubjectSignToolExtension": {
        "SignTool": "КриптоПро CSP 5.0"
      }
    },
    {
      "IdentificationKindExtension": {
        "IdentificationKind": "RemoteCert"
      }
    }
  ]
}

```

1.1.5 Установка сертификатов в хранилище

Установка выпущенного сертификата в хранилище осуществляется с использованием утилит **certmgr** и **cryptsp** из состава КриптоПро CSP.

Для установки корневого сертификата используется команда **certmgr -install**. Пример команды установки в хранилище «Личные» пользователя:

```
/opt/cproscsp/bin/amd64/certmgr -install -store uMy -file <имя файла сертификата> -provtype 80 -container cont_ca -at_signature
```

Для установки прочих сертификатов также может быть использована команда **certmgr -install**. Но если запрос был создан командой **pkica ca new-cert-req** на этой же машине под этим же пользователем, то выпущенный по нему сертификат можно установить командой **cryptcp -instcert** без указания имени ключевого контейнера.

Пример команды для установки в хранилище «Личные» пользователя:

```
/opt/cproscsp/bin/amd64/cryptcp -instcert -from_request <имя файла с сертификатом>
```

Установка в хранилище «Личные» компьютера:

```
sudo /opt/cprosp/bin/amd64/cryptcp -instcert -from_request -dk -dm <имя файла с сертификатом>
```

Подробнее использование утилит описано в эксплуатационной документации СКЗИ КриптоПро CSP.

1.2 Экземпляры Центра сертификации

1.2.1 Просмотр списка экземпляров ЦС

Вывод списка экземпляров ЦС на консоль осуществляется командой **pkica ca authority list**.

1.2.2 Вывод информации об экземпляре ЦС

Вывод подробной информации (ключи подписи и настройки) о ЦС на консоль осуществляется командой **pkica ca authority show**.

Параметры:

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
--format	Формат вывода объектов на печать. Формат по умолчанию можно задать в конфигурации. Допустимые значения: Json, Yaml.

Пример:

```
./pkica ca authority show --name "CryptoPro CA"
```

1.2.3 Добавление нового экземпляра ЦС

Добавление нового экземпляра ЦС осуществляется командой **pkica ca authority new --name <имя ЦС>**. Если этот экземпляр единственный, то он будет установлен как экземпляр ЦС по умолчанию.

Пример:

```
./pkica ca authority new --name "CryptoPro CA"
```

1.2.4 Установка экземпляра ЦС по умолчанию

Экземпляр ЦС можно установить экземпляром по умолчанию. На экземпляре по умолчанию выполняются команды и обрабатываются запросы, в которых не указано имя ЦС. Установка экземпляра ЦС по умолчанию осуществляется командой **pkica ca authority set-default --name <имя ЦС>**.

Пример:

```
./pkica ca authority set-default --name "CryptoPro CA"
```

Для сброса значения установленного экземпляра ЦС по умолчанию используется эта же команда с параметром **--none**.

Пример:

```
./pkica ca authority set-default --none
```

1.2.5 Удаление экземпляра ЦС

Удаление экземпляра ЦС осуществляется командой **pkica ca authority remove --name <имя ЦС>**

Пример:

```
./pkica ca authority remove --name "CryptoPro CA"
```

1.2.6 Настройки ЦС

1.2.6.1 Редактирование настроек ЦС

Для редактирования настроек экземпляра ЦС их предварительно необходимо выгрузить в текстовый файл командой **pkica ca authority export-settings**.

Параметры (звездочкой * помечены обязательные):

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-f --file	* Имя файла, в который надо экспортировать настройки.
--format	Формат вывода объектов на печать. Формат по умолчанию можно задать в конфигурации. Допустимые значения: Json, Yaml.

Пример:

```
./pkica ca authority export-settings --name "CryptoPro CA" --file settings.txt --format Json
```

Затем в полученном файле необходимо изменить требуемые настройки, используя текстовый редактор, и импортировать их для того же экземпляра ЦС командой **pkica ca authority import-settings**.

Параметры (звездочкой * помечены обязательные):

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-f --file	* Имя файла, из которого надо импортировать настройки.
--format	Формат вывода объектов на печать. Формат по умолчанию можно задать в конфигурации. Допустимые значения: Json, Yaml.

Пример:

```
./pkica ca authority import-settings --name "CryptoPro CA" --file
settings.txt --format Json
```

1.2.6.2 Список адресов точек распространения CRL (CDP)

Имя настройки – **CrlPublicationUrls**.

Формат – набор структур, содержащих назначение и массив адресов для точки распространения. Каждая структура содержит поля **«Назначение» (Usage)** и **«Адреса» (Urls)**, и соответствует одной точке распространения.

Поле **«Назначение»** может принимать одно или несколько из следующих значений, разделенных запятой:

- **CdpIncludeInBase** (1) – включать URL в расширение «Размещения опубликованных CRL» базового списка отзыва.
- **CdpIncludeInCert** (2) – включать адрес в расширение «Точки распространения списков отзыва (CRL)» сертификата.
- **CdpIncludeInCrlFreshestCrlExtension** (4) – включать адрес в расширение «Новейший CRL» списка отзыва.
- **CdpIncludeInDelta** (64) – включать URL в расширение «Размещения опубликованных CRL» разностного списка отзыва.
- **CdpIncludeInCrlIdpExtension** (128) – включать адрес в расширение «Точка выдачи CRL» списка отзыва. Если настроить добавление этого расширения в список отзыва, то при проверке цепочки будет дополнительно проверяться, что эта точка выдачи действительно для проверяемого сертификата.

Поле **«Адреса»** является массивом адресов URL, которые содержатся в точке распространения. В каждом адресе можно использовать следующие подстановочные символы:

- **%4** – индекс ключа ЦС. Пустая строка, в случае если индекс равен нулю.
- **%8** – индекс ключа ЦС. Пустая строка, в случае если индекс равен нулю. Используется только в адресах точек CDP.
- **%9** – символ "-". Используется только в адресах CDP для разностных CRL, в противном случае пустая строка.
- **%15** – отпечаток сертификата издателя.
- **%17** – идентификатор ключа сертификата издателя.

Пример:

```
...
  "CrlPublicationUrls": [
    {
      "Usage": "CdpIncludeInCert",
      "Urls": [
        "http://cdp1.ca.local/%17.crl"
      ]
    },
    {
      "Usage": "CdpIncludeInBase, CdpIncludeInCert",
```

```

    "Urls": [
      "http://cdp2.ca.local/%17.crl",
      "http://cdp2.ca.local/%17.crl"
    ]
  },
  ...

```

1.2.6.3 Список адресов точек доступа к информации о ЦС (AIA)

Имя настройки – **CaCertPublicationUrls**.

Формат – набор структур, содержащих назначение и массив адресов для точки доступа. Каждая структура содержит поля «**Назначение**» (**Usage**) и «**Адрес**» (**Url**), и соответствует одной точке доступа.

Поле «**Назначение**» может принимать одно из следующих значений:

- **AiaIncludeInCert** (2) – включать адрес в расширение «Доступ к сведениям Центра сертификации».
- **OcspIncludeInCert** (32) – включать адрес в расширение «Точки распространения CRL» как адрес службы актуальных статусов сертификатов (OCSP).

Поле «**Адрес**» - адрес URL, содержащийся в точке распространения. В адресе можно использовать подстановочные символы как для адресов CDP из Список адресов точек распространения CRL (CDP) .

Пример:

```

...
"CaCertPublicationUrls": [
  {
    "Usage": "AiaIncludeInCert",
    "Url": "http://ca.local/ca.crt"
  },
  {
    "Usage": "OcspIncludeInCert",
    "Url": "http://ca.local/ocsp.srf"
  }
],
...

```

1.2.6.4 Режим контроля соответствия требований к квалифицированному сертификату.

Имя настройки - **QualifiedCertMode**. Определяет режим, в котором проверяется соответствие требований нормативным документам, определяющих состав квалифицированного сертификата.

Варианты настройки:

- **Full** – полный контроль соответствия выпущенного сертификата требованиям. В этом режиме проверяется состав субъекта на наличие требуемых и их формат, а также наличие обязательных для квалифицированного сертификата расширений. Этот режим должен быть обязательно включен, если УЦ выпускает квалифицированные сертификаты.

- **AttrFormat** – контроль формата атрибутов имени. В этом режиме проверяется, что атрибуты из нормативных документов (ИНН, ИНН ЮЛ, СНИЛС, ОГРН, ОГРНИП) являются числовыми строками определенной длины, а также проверяется формат некоторых других атрибутов. При этом формат атрибутов в субъекте проверяется только при их наличии.
- **None** – контроль отключен. Этот режим допустимо использовать только в целях тестирования.

1.2.6.5 Сроки действия

Определяет максимальные сроки действия сертификатов и их закрытых ключей. Раздел **CertValidityRestriction** является настройкой максимального срока действия сертификатов, а раздел **PrivateKeyValidityRestriction** – максимального срока действия закрытых ключей. Каждый раздел содержит следующие свойства:

- **PeriodType** – единицы измерения интервала времени. Допустимые значения:
 - **Years** – годы;
 - **Months** – месяцы;
 - **Weeks** – недели;
 - **Days** – дни;
 - **Hours** – часы;
 - **Minutes** – минуты.
- **UnitCount** – длина интервала времени в заданных единицах, является целым числом.

Пример:

```
...
  "CertValidityRestriction": {
    "PeriodType": "Years",
    "UnitCount": 16
  },
  "PrivateKeyValidityRestriction": {
    "PeriodType": "Months",
    "UnitCount": 15
  },
  ..
```

1.2.6.6 Точность часов ЦС.

Точность часов ЦС (в минутах) определяется настройкой **ClockSkewMinutes**, содержащей целое неотрицательное число. Для получения времени начала действия сертификатов и CRL значение этой настройки вычитается из текущего времени ЦС.

1.2.6.7 Интервалы перекрытия CRL

Интервал перекрытия – это интервал времени, в течение которого действуют («перекрываются») два CRL – текущий и следующий. В течение этого интервала времени клиенты могут скачивать новый CRL, пользуясь при этом текущим. Таким образом, у клиента всегда будет действующий CRL, без временных разрывов. Раздел **CrlOverlapPeriod** является

настройкой интервала перекрытия базовых CRL, а **CrlDeltaOverlapPeriod** – настройкой интервала перекрытия разностных CRL. Каждая настройка содержит свойства **PeriodType** и **UnitCount**, описанные в разделе Сроки действия

1.2.6.8 Автозагрузка ключа ЦС

Разрешает автозагрузку ключей для экземпляра ЦС. Выполняется при выпуске CRL и при подписи сертификата. Данная настройка работает только при условии отсутствия PIN-кода на контейнере ключа. Чтобы разрешить автозагрузку ключа нужно присвоить полю **AutoLoadKey** значение true.

1.2.6.9 Алгоритм формирования серийных номеров

По умолчанию, выпускаемые сертификаты содержат уникальные серийные номера размером 18 байт (36 символов в шестнадцатеричном представлении).

Алгоритм генерации серийных номеров задается настройкой **SerialNumberAlgorithm**, которая может принимать следующие значения:

- CryptoPro - алгоритм генерации серийных номеров длиной не более 18 байт. Это значение установлено по умолчанию.
- Msca - алгоритм генерации 10 байтовых серийных номеров службы сертификации Microsoft (https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-wcce/19b706b2-3db3-4a06-8aff-29ea7327e9ef)

При эксплуатации УЦ для выпуска квалифицированных СКПЭП разрешено использовать любой из этих алгоритмов.

1.2.6.10 Состав расширения «Идентификатор ключа центра сертификатов»

В соответствии с п.24 приказа ФСБ РФ от 27 декабря 2011 г. N 795 "Об утверждении Требований к форме квалифицированного сертификата ключа проверки электронной подписи" в расширение «Идентификатор ключа центра сертификатов» (Authority Key Identifier, OID 2.5.29.35) должно находиться значение серийного номера соответствующего сертификата, аккредитованного УЦ. Такая логика работы модуля политики КриптоПро УЦ 2.0 настроена по умолчанию при разворачивании нового экземпляра Центра Сертификации. При необходимости ее изменения для выпуска неквалифицированных сертификатов нужно установить параметр ЦС **DisableAkiIssuerNameAndSerial** в true.

После этого в расширение «Идентификатор ключа центра сертификатов» будет попадать только идентификатор ключа субъекта сертификата Центра Сертификации.

Аналогичным способом можно управлять составом расширения «Идентификатор ключа центра сертификатов» в списках отзыва. По умолчанию в списках отзыва это расширение содержит только идентификатор ключа сертификата ЦС. Для того, чтобы добавить в это расширение издателя и серийный номер, необходимо установить параметр **EnableAkiIssuerNameAndSerialInCrl** в true.

1.2.6.11 Политика криптопровайдеров

По умолчанию на ЦС действует политика криптопровайдеров, согласно которой набор доступных алгоритмов открытого ключа выпускаемых сертификатов ограничивается алгоритмами, реализуемыми СКЗИ «КриптоПро CSP» и ПАКМ «КриптоПро HSM».

Политика криптопровайдеров задается настройкой **CspPolicy**, которая может принимать два значения:

- CryptoPro – набор криптопровайдеров ограничен криптопровайдерами СКЗИ «КриптоПро CSP» и ПАКМ «КриптоПро HSM»;
- Unrestricted – доступны все криптопровайдеры.

При эксплуатации УЦ для выпуска квалифицированных СКПЭП должна использоваться политика криптопровайдеров CryptoPro.

1.2.6.12 Запрет на выпуск сертификатов с алгоритмом ключа ГОСТ Р 34.10-2001

После 31.12.2018 создание квалифицированных сертификатов ключей проверки электронной подписи в соответствии с ГОСТ Р 34.10-2001 не допускается, в связи с чем УЦ запрещает выпуск таких сертификатов. Для того, чтобы в целях тестирования разрешить выпуск СКПЭП с алгоритмом ключа ГОСТ Р 34.10-2001, нужно установить параметр **DisableGost2001CertIssuingCheck** в true.

При эксплуатации УЦ для выпуска квалифицированных СКПЭП запрет должен быть включен.

1.2.6.13 Точность часов ЦС

Параметр **ClockSkewMinutes** задаёт значение в минутах, определяющее максимально возможное расхождение времени на компонентах УЦ, а также в информационных системах, использующих сертификаты, изданные УЦ. Данный параметр используется при вычислении сроков действия выпускаемых сертификатов и CRL, а также для вычисления интервала перекрытия, если интервал перекрытия не задан явно.

1.2.6.14 Шаблон по умолчанию

Наличие шаблона в запросе на сертификат - обязательное условие для обработки этого запроса на ЦС. Если в целях тестирования требуется обрабатывать запросы, в которых шаблон не указан, то в параметре **DefaultTemplateOid** нужно указать идентификатор OID шаблона, по которому такие запросы будут обработаны.

1.2.6.15 Расширение «Доступ к сведениям Центра сертификации» в сертификатах службы OCSP

Из расширения «Доступ к сведениям Центра сертификации» (Authority Info Success) в сертификатах службы OCSP по умолчанию удаляются адреса самой службы OCSP, настроенные на ЦС (назначение **OcspIncludeInCert**, раздел 1.2.6.3). Если адреса службы OCSP из настроек ЦС должны присутствовать в сертификатах службы OCSP, то необходимо установить параметр **AllowOcspUrlFromSettingsInOcspCertificates** в true.

1.2.6.16 Расширение «Размещения опубликованных CRL»

Расширение списка отзыва «Размещения опубликованных CRL» содержит URL, по которым этот список был опубликован (флаги **CdpIncludeInBase** и **CdpIncludeInDelta** из раздела 1.2.6.2). Для добавления в выпускаемые списки отзыва расширения «Размещения опубликованных CRL» (1.3.6.1.4.1.311.21.14) нужно установить настройку **PublishCrlLocationExtensionEnabled** в true.

1.2.6.17 Расширение «Следующая публикация CRL»

Для отключения добавления расширения «Следующая публикация CRL» в список отзыва (например, при выпуске финального CRL, раздел 1.4.1) нужно установить параметр **CrlDisableNextPublish** в true.

1.2.6.18 Кодировка имени субъекта.

Для кодировки атрибутов имени субъекта сертификата, содержащих символы только из набора PrintableString, но допускающих любые символы из юникода, по умолчанию используется тип ASN.1 UTF8String. Если эти атрибуты необходимо кодировать именно в PrintableString, то нужно установить параметр **DistinguishedNameForceUtf8Encoding** в false. Этот параметр аналогичен флагу CERT_RDN_FORCE_UTF8_UNICODE_FLAG (https://learn.microsoft.com/en-us/windows/win32/api/wincrypt/ns-wincrypt-cert_rdn_attr) и работает только на ОС Windows.

1.2.6.19 Контроль длин атрибутов имени субъекта.

Для того, чтобы включить контроль длин атрибутов имени субъекта на соответствие требованиям стандарта RFC 5280 (<https://www.rfc-editor.org/rfc/rfc5280#appendix-A.1>), необходимо установить параметр **DisableRfc5280CertAttrsCheck** в false. По умолчанию этот контроль отключен, так как в квалифицированных сертификатах достаточно часто встречаются атрибуты имени, не удовлетворяющие требованиям к длине RFC 5280, например, атрибут «Общее имя» (CN) может быть длиннее 64 символов.

1.2.6.20 Состав расширения «Использование ключа» в сертификатах ЦС

Расширение «Использование ключа» (2.5.29.15) в сертификате ЦС должно обязательно содержать биты «Подпись сертификатов» и «Подпись списков отзыва». Если к этим битам требуется добавить бит «Цифровая подпись» (т.е. ключ подписи ЦС планируется использовать для подписи чего-то еще, кроме выпущенных сертификатов и списков отзыва), то нужно установить параметр **AddKeyUsageDigitalSignatureToCaCert** в true.

1.2.6.21 Некритическое расширение «Использование ключа»

Расширение «Использование ключа» (2.5.29.15) всегда по умолчанию является критическим. Если для тестовых целей необходимо разрешить выпуск сертификатов с некритическим расширением, то нужно установить параметр **AllowNonCriticalKeyUsage** в true.

1.2.7 Добавление ключа подписи ЦС

Для добавления ключа подписи экземпляру ЦС используется команда **pkica ca signing-key add**. Перед выполнением этой команды требуется установить добавляемый сертификат в локальное хранилище с ссылкой на ключ.

Параметры (звездочкой * помечены обязательные):

-n --name	Имя экземпляра ЦС, которому надо добавить ключ. Если не задано, то будет использован экземпляр по умолчанию.
-t --thumbprint	* Отпечаток сертификата добавляемого ключа.
--store-name	Имя хранилища сертификатов, в котором находится сертификат ключа. Значение по умолчанию: Му.
--store-location	Расположение хранилища сертификатов, в котором находится сертификат ключа. Допустимые значения: CurrentUser, LocalMachine. Значение по умолчанию: CurrentUser.

-u --usage	Разрешенное использование ключа подписи. Допустимые значения: SignCertAndCrl - ключ может использоваться для подписи сертификатов и CRL; SignCrl - ключ может использоваться только для подписи CRL; Retired - ключ выведен из эксплуатации и не может использоваться. Значение по умолчанию: SignCertAndCrl.
--hash-algorithm	Имя или OID алгоритма хэширования. Если не задано, то будет использован алгоритм по умолчанию для провайдера.
--accept-new-subject	По умолчанию субъект сертификата должен совпадать с субъектом текущего сертификата ЦС. Флаг отключает это поведение и позволяет добавить сертификат с отличающимся субъектом.
--planned-key-change-date	Плановая дата смены ключа. В течение 30 дней до наступления этой даты, а также после наступления, при загрузке ключа будет выводиться сообщение о необходимости плановой смены.
--planned-retirement-date	Плановая дата вывода ключа из обращения. В течение 30 дней до наступления этой даты, а также после наступления, при загрузке ключа будет выводиться сообщение о необходимости вывода ключа из обращения.
--pin	ПИН-код на контейнер. Если не задан, то будет запрошен во время выполнения.
--no-pin	Флаг указывает, что контейнер не имеет ПИН-кода. Подавляет запрос ПИН-кода во время выполнения.
--skip-container-check	Пропустить проверку контейнера ключа (в этом случае ПИН-код на контейнер не потребуется).

Пример:

```
./pkica ca signing-key add --name "CryptoPro CA" --thumbprint
1889d03da1a04f2ffc31f29c0c20950a0456e950
```

1.2.8 Замена ключа подписи ЦС

Если необходимо заменить сертификат ЦС или обновить ссылку на ключ подписи ЦС, то нужно выполнить замену ключа подписи ЦС командой **pkica ca signing-key replace**. Команда заменяет сертификат с указанным индексом ключа на новый. Так же будет заменена ссылка на ключ подписи, поэтому новый сертификат должен быть установлен в локальное хранилище с ссылкой на ключ.

Параметры (звездочкой * помечены обязательные):

-n --name	Имя экземпляра ЦС, которому надо добавить ключ. Если не задано, то будет использован экземпляр по умолчанию.
-i --index	* Индекс ключа, который надо заменить.

-t --thumbprint	* Отпечаток нового сертификата ключа.
--store-name	Имя хранилища сертификатов, в котором находится сертификат ключа. Значение по умолчанию: My.
--store-location	Расположение хранилища сертификатов, в котором находится сертификат ключа. Допустимые значения: CurrentUser, LocalMachine. Значение по умолчанию: CurrentUser.
--hash-algorithm	Имя или OID алгоритма хэширования. Если не задано, то будет использован алгоритм по умолчанию для провайдера.
--accept-new-subject	По умолчанию субъект сертификата должен совпадать с субъектом текущего сертификата ЦС. Флаг отключает это поведение и позволяет добавить сертификат с отличающимся субъектом.
--planned-key-change-date	Плановая дата смены ключа. В течение 30 дней до наступления этой даты, а также после наступления, при загрузке ключа будет выводиться сообщение о необходимости плановой смены.
--planned-retirement-date	Плановая дата вывода ключа из обращения. В течение 30 дней до наступления этой даты, а также после наступления, при загрузке ключа будет выводиться сообщение о необходимости вывода ключа из обращения.
--pin	ПИН-код на контейнер. Если не задан, то будет запрошен во время выполнения.
--no-pin	Флаг указывает, что контейнер не имеет ПИН-кода. Подавляет запрос ПИН-кода во время выполнения.
--skip-container-check	Пропустить проверку контейнера ключа (в этом случае ПИН-код на контейнер не потребуется).

Пример:

```
./pkica ca signing-key replace --name "CryptoPro CA" --index 1 --thumbprint 1889d03da1a04f2ffc31f29c0c20950a0456e950
```

1.2.9 Загрузка ключей подписи ЦС

Для выпуска сертификатов и CRL ключи подписи должны быть загружены в сервис ЦС. Для этого используется команда **pkica ca signing-key load**.

Параметры (обязательные параметры помечены звездочкой *):

-n --name	Имя экземпляра ЦС, ключ которого надо загрузить. Если не задано, то будет использован экземпляр по умолчанию.
-i --index	* Индекс загружаемого ключа.

--pin	ПИН-код на контейнер. Если не задан, то будет запрошен во время выполнения.
--no-pin	Флаг указывает, что контейнер не имеет ПИН-кода. Подавляет запрос ПИН-кода во время выполнения.
-u --usage	Использование, для которого загружается ключ. Допустимые значения: SignCertAndCrl, SignCrl. Если не задано, то ключ будет загружен для того использования, для которого был зарегистрирован.

Одновременно может быть загружен только один ключ для подписи сертификатов и сколько угодно ключей для подписи CRL.

Пример:

```
./pkica ca signing-key load --name "CryptoPro CA" --index 1 --pin 08642 --usage SignCertAndCrl
```

1.2.10 Получение подробной информации о ключе подписи ЦС

Для получения детальной информации о ключе подписи ЦС используется команда **pkica ca signing-key show**.

Параметры (обязательные параметры помечены звездочкой *):

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-i --index	* Индекс ключа подписи.
--status	Показать статус загрузки ключа подписи (будет выполнено обращение к сервису ЦС).
--timeout	Ожидание ответа от ЦС о статусе загрузки ключа. Задается в формате ISO 8601: [d.]hh:mm[:ss]. Значение по умолчанию: 00:00:30 (30 секунд).

Пример:

```
./pkica ca signing-key show --name "CryptoPro CA" --index 0 --status
```

1.2.11 Получение списка ключей подписи ЦС

Для получения списка ключей подписи ЦС используется команда **pkica ca signing-key list**. Она выводит список ключей, для каждого ключа выводится индекс, отпечаток сертификата и использование, для которого ключ был зарегистрирован.

Параметры:

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-l --loaded	Получать только загруженные ключи (будет выполнено

	обращение к сервису ЦС).
--status	Показать статус загрузки ключа подписи (будет выполнено обращение к сервису ЦС).
--timeout	Ожидание ответа от ЦС о статусе загрузки ключа. Задается в формате ISO 8601: [d.]hh:mm[:ss]. Значение по умолчанию: 00:00:30 (30 секунд).

Пример:

```
./pkica ca signing-key list --name "CryptoPro CA" --status --loaded
```

1.2.12 Выгрузка ключей подписи ЦС

Если ключом более не требуется подписывать сертификаты или CRL, то его необходимо выгрузить из сервиса ЦС. Для этого используется команда **pkica ca signing-key unload**.

Параметры (обязательные параметры помечены звездочкой *):

-n --name	Имя экземпляра ЦС, ключ которого надо выгрузить. Если не задано, то будет использован экземпляр по умолчанию.
-i --index	* Индекс выгружаемого ключа.

Пример:

```
./pkica ca signing-key unload --name "CryptoPro CA" --index 1
```

1.2.13 Изменение назначения ключа подписи ЦС

Если в соответствии с эксплуатационной документацией ключ ЦС больше не может использоваться для подписи сертификатов, а должен использоваться только для подписи CRL, то его использование необходимо изменить на **SignCrl** (только подпись CRL). Для этого используется команда **pkica ca signing-key change-usage**.

Параметры (обязательные параметры помечены звездочкой *):

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-i --index	* Индекс изменяемого ключа.
-u --usage	* Разрешенное использование ключа подписи. Допустимые значения: • SignCertAndCrl - ключ может использоваться для подписи сертификатов и CRL; • SignCrl - ключ может использоваться только для подписи CRL; • Retired - ключ выведен из эксплуатации и не может использоваться.

--force	Флаг подавляет подтверждение вывода ключа из эксплуатации.
---------	--

Пример:

```
./pkica ca signing-key change-usage --name "CryptoPro CA" --index 1 -usage SignCrl
```

1.2.14 Вывод из эксплуатации ключа подписи ЦС

Ключ подписи ЦС после окончания срока действия должен быть выведен из эксплуатации. После вывода этот ключ не может быть использован ни для подписи сертификатов, ни для подписи CRL.

Порядок вывода ключа из эксплуатации с выпуском финального CRL - такого CRL, который действует до окончания срока действия сертификата подписи ЦС:

- 1) загрузить ключ с назначением Подпись CRL командой **pkica ca signing-key load --usage SignCrl**, как описано в п. 1.2.9
- 2) вывести ключ из эксплуатации командой **pkica ca signing-key change-usage --usage Retired --name <имя ЦС> --index <индекс ключа>**
- 3) выпустить финальный CRL командой **pkica ca crl issue**, как описано в п. 1.4.1. **Важно:** финальный CRL может быть выпущен только на ключе, выведенном из обращения.
- 4) выгрузить выведенный из эксплуатации ключ командой **pkica ca signing-key unload**, как описано в п. 1.2.12. **Важно:** выведенный из эксплуатации ключ более не сможет быть загружен для подписи сертификатов или CRL.

Пример:

```
# загрузка ключа
./pkica ca signing-key load --name "CryptoPro CA" --index 1 -usage SignCrl

# вывод из эксплуатации
./pkica ca signing-key change-usage --name "CryptoPro CA" --index 1 -usage Retired

# выпуск финального CRL
./pkica ca crl issue --name "CryptoPro CA" --index 1 --final

# выгрузка ключа
./pkica ca signing-key unload --name "CryptoPro CA" --index 1
```

В случаях, когда не требуется выпускать финальный CRL (например, контейнер закрытого ключа подписи уже недоступен), достаточно выполнить п.2 и, если ключ был загружен - выгрузить (п.4).

1.3 Шаблоны сертификатов

При формировании сертификата по запросу Центр сертификации руководствуется совокупностью правил и данных, называемой шаблоном сертификата. Каждый шаблон имеет уникальный идентификатор и имя. Каждый запрос, поступивший на ЦС, должен содержать информацию о шаблоне, который будет применен при обработке.

1.3.1 Структура шаблона сертификата

1.3.1.1 Свойства шаблона

У каждого шаблона сертификата есть следующие свойства:

- **Name** - уникальное короткое имя шаблона. Может использоваться для указания шаблона в запросе в расширении «Имя шаблона сертификата» (1.3.6.1.4.1.311.20.2).
- **Oid** - объектный идентификатор шаблона.
- **MajorVersion** - старшая версия шаблона.
- **MinorVersion** - младшая версия шаблона.
- **DisplayName** - отображаемое имя шаблона, например «Квал. сертификат ФЛ (16 лет)».
- **Description** - подробное описание шаблона, например «Используется для выпуска квалифицированных сертификатов физическим лицам, срок действия 16 лет. »

1.3.1.2 Субъект сертификата

В шаблоне обязательно должно быть указано, как должен быть сформирован субъект сертификата. Эта настройка содержится в объекте **SubjectName** и заключается в указании источника компонентов имени и правил формирования субъекта:

- **PrioritySource** - определяющее приоритет источника данных для формирования субъекта. Может принимать одно из следующих значений:
 - **Profile** - данные для формирования субъекта берутся в первую очередь из профиля пользователя в запросе (атрибут CpRaRequesterProfile), а если профиль отсутствует - то из субъекта
 - **Subject** - данные для формирования субъекта берутся в первую очередь из субъекта запроса, а если субъект не заполнен - то из профиля.
- **Rules** – набор правил для формирования субъекта. Каждое правило соответствует одному компоненту имени в субъекте и определяет, из какого компонента имени источника данных будет получено значение компонента имени субъекта.

Каждое правило массива Rules представляет структуру из двух полей:

- **RdnOid** – идентификатор компонента имени, который будет добавлен в субъект сертификата. Является обязательным.
- **TakeFrom** – идентификатор компонента имени из запроса, значение которого будет добавлено. Является необязательным, если не задан - будет добавлено значение компонента с идентификатором **RdnOid**.

Правила обрабатываются в обратном порядке, т.е. компонент, указанный в наборе первым – в субъект попадет последним, а указанный последним – наоборот, первым.

Если ни одно правило не задано, то есть набор пуст, то профиль или субъект (в зависимости от настройки **PrioritySource**) перекладывается в субъект целиком, без изменений.

Пример:

```
{
...
{
  "SubjectName": {
    "PrioritySource": "Profile",
    "Rules": [
      {
        "RdnOid": "2.5.4.3"
      },
      {
        "RdnOid": "2.5.4.4"
      },
      {
        "RdnOid": "2.5.4.42"
      },
      {
        "RdnOid": "2.5.4.6"
      },
      {
        "RdnOid": "2.5.4.8"
      },
      {
        "RdnOid": "2.5.4.7"
      },
      {
        "RdnOid": "2.5.4.9"
      },
      {
        "RdnOid": "2.5.4.10",
        "TakeFrom": "2.5.4.3"
      },
      {
        "RdnOid": "1.2.840.113549.1.9.1"
      },
      {
        "RdnOid": "1.2.643.100.3"
      },
      {
        "RdnOid": "1.2.643.3.131.1.1"
      }
    ]
  },
...
}
```

1.3.1.3 Срок действия сертификата

Срок действия сертификата определяется датами начала и окончания действия сертификата. Дата окончания сертификата вычисляется путем прибавления к дате начала действия сертификата временного интервала, указанного в объекте **CertValidity** шаблона на

сертификат. Значение временного интервала определяется объектом **ValidityPeriod**, у которого есть 2 поля, определяющее длительность срока действия:

- **PeriodType** – единицы измерения интервала времени. Допустимые значения:
 - **Years** – годы;
 - **Months** – месяцы;
 - **Weeks** – недели;
 - **Days** – дни;
 - **Hours** – часы;
 - **Minutes** – минуты.
- **UnitCount** – длина интервала времени в заданных единицах, является целым числом.

При этом срок действия сертификата, указанный в шаблоне, ограничен сроком действия сертификата, указанным в настройках Центра сертификации, а время окончания срока действия сертификата ограничено временем окончания сертификата издателя.

Так же у **CertValidity** есть настройки, определяющие, какие параметры срока действия можно задавать в запросе на сертификат:

- **SpecifyValidityPeriodInRequest** – определяет, разрешено ли указывать срок действия сертификата в запросе;
- **SpecifyStartDateInRequest** – определяет, разрешено ли указывать начало действия сертификата в запросе;
- **SpecifyExpirationDateInRequest** – определяет, разрешено ли указывать окончание действия сертификата в запросе.

Каждая из этих настроек может принимать одно из следующих значений:

- **Prohibit** – значение запрещено указывать;
- **Allow** – значение разрешено указывать;
- **Required** – значение требуется указывать. Запросы без этого значения будут отклонены.

Пример:

```
{
...
  "CertValidity": {
    "ValidityPeriod": {
      "PeriodType": "Years",
      "UnitCount": 18
    },
    "SpecifyValidityPeriodInRequest": "Prohibit",
    "SpecifyStartDateInRequest": "Prohibit",
    "SpecifyExpirationDateInRequest": "Prohibit"
  },
...
}
```

1.3.1.4 Параметры создания ключа

Настройка параметров создания ключа задается объектом **KeyCreationSpec**. Эти параметры не влияют непосредственно на процесс выпуска сертификата на Центре сертификации, но могут служить рекомендацией для клиентов, создающих запросы на сертификат. В частности, на АРМ Оператора ЦР при создании ключей для клиентов используются строго те параметры, которые определены в шаблоне.

Объект **KeyCreationSpec** имеет следующие свойства:

- **MachineKeySet** - флаг, указывающий нужно ли создавать ключ в хранилище локального компьютера. Если ключ создается в таком хранилище, то процесс, создающий этот ключ, должен быть запущен с права администратора.
- **KeySpec** - тип создаваемого ключа. Может принимать следующие значения:
 - **Signature** - ключ подписи.
 - **KeyExchange** - ключ обмена.
 - **Unspecified** - тип ключа не задан, определяется выбором пользователя.

При создании запросов на выпуск квалифицированного сертификата должен создаваться только ключ подписи.

- **ExportablePrivateKey** - разрешение на создание экспортируемого ключа.
- **Csps** - коллекция провайдеров, которые могут быть использованы для создания ключа. Каждый элемент **Csp** этой коллекции имеет следующие свойства.
 - **ProviderType** - тип провайдера (число).
 - **ProviderName** - имя провайдера (строка). Если не задано, то допустим любой провайдер типа **ProviderType**.
 - **Priority** - приоритет провайдера (число). Может использоваться для сортировки списка доступных для выбора провайдеров, например чем больше приоритет - тем выше провайдер в списке.

Если коллекция пуста, то для создания ключа может быть использован любой провайдер.

1.3.1.5 Расширения сертификата

Каждое настроенное в шаблоне расширение сертификата содержит следующую информацию:

DefaultValue	Значение расширения по умолчанию. Формат этого значения зависит от расширения, для которого оно задано.
AddOnlyIfRequested	Добавлять расширение в сертификат, только если оно присутствует в запросе.
PrioritySource	Приоритетный источник значения расширения. Допустимые значения – Template (шаблон) и Request (запрос).
AllowArbitraryValue	Для расширения разрешено произвольное значение.

RequireInRequest	Расширение обязательно должно присутствовать в запросе. При его отсутствии запрос будет отклонен.
AllowedValues	Список допустимых в запросе значений. Этот список определяет набор допустимых значений, которые могут быть заданы в запросе. Он имеет смысл для тех расширений, значение которых определяется конкретными данными, а не правилом. Например, значение расширения «Улучшенный ключ» задается фиксированным набором OID, а значение расширения «Срок действия ключа ЭП» вычисляется на основе данных шаблона, запроса и текущего времени. Помимо собственно допустимых значений для каждого может быть задана метка (Tag). Метка является текстовым идентификатором, определяющим значение расширения из списка допустимых. Задается в запросе в атрибуте EnrollmentNameValuePair, значения которого имеют вид <имя>=<значение>. Имя в этом случае это CpRaExtensions, а значение – строка, состоящая из пар вида [Oid=имя метки]. Oid – это идентификатор расширения. Имя метки – то имя, которое задается в настройке допустимых значений. В запросе имя метки может отсутствовать, тогда принимается, что расширение задано без значения, т.е. выполнено условие «Добавлять расширение в сертификат, только если оно присутствует в запросе». Если имя метки задано, то берется соответствующее ему допустимое значение из списка. Если в запросе присутствует метка, которая отсутствует в списке допустимых, то такой запрос считается ошибочным и сертификат по нему издан не будет.

1.3.1.5.1 Расширение «Использование ключа» (KeyUsage)

Информация о настройке расширения «Использование ключа» содержится в объекте **KeyUsageExtension**. Значением по умолчанию является использование ключа, которое должно попасть в сертификат. Это использование ключа определяется полем **KeyUsages**, которое может принимать одно или несколько из допустимых значений:

- **EncipherOnly** (1) - только шифрование.
- **CrlSign** (2) - подписание списка отзыва сертификатов (CRL).
- **KeyCertSign** (4) - подпись сертификатов.
- **KeyAgreement** (8) - согласование ключей.
- **DataEncipherment** (16) - шифрование данных.
- **KeyEncipherment** (32) - шифрование ключей.
- **NonRepudiation** (64) - проверка подлинности.
- **DigitalSignature** (128) - цифровая подпись.
- **DecipherOnly** (32768) - только расшифровка.

Пример:

```
{
...
  {
    "KeyUsageExtension": {
      "DefaultValue": {
        "KeyUsages": "NonRepudiation, DigitalSignature"
      },
      "AllowedValues": [],
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  },
...
}
```

1.3.1.5.2 Расширение «Улучшенный ключ» (EnhancedKeyUsage)

Информация о настройке расширения «Улучшенный ключ» содержится в объекте **EnhancedKeyUsageExtension**. Значением по умолчанию является область применения квалифицированного сертификата. Эта область определяется полем **EnhancedKeyUsages**, содержащим массивом идентификаторов OID, и полем **Critical**, указывающим, является ли расширение критичным. Каждый элемент массива обозначает одно из возможных применений сертификата.

Пример:

```
{
...
  {
    "EnhancedKeyUsageExtension": {
      "DefaultValue": {
        "Critical": false,
        "EnhancedKeyUsages": [
          "1.3.6.1.5.5.7.3.2",
          "1.3.6.1.5.5.7.3.4"
        ]
      },
      "AllowedValues": [],
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  },
...
}
```

1.3.1.5.3 Расширение «Шаблон сертификата» (CertificateTemplate)

Информация о настройке расширения «Шаблон сертификата» содержится в объекте **CertificateTemplateExtension**. Значением по умолчанию является объект, содержащий информацию о шаблоне, по которому будет выпущен сертификат.

У этого объекта есть следующие поля:

- **TemplateOid** – идентификатор OID шаблона.
- **MajorVersion** – старшая версия шаблона.
- **MinorVersion** – младшая версия шаблона (необязательно поле).

Пример:

```
{
...
{
  "CertificateTemplateExtension": {
    "DefaultValue": {
      "TemplateOid": "1.2.643.2.2.46.0.8",
      "MajorVersion": 1,
      "MinorVersion": 0
    },
    "AddOnlyIfRequested": false,
    "PrioritySource": "Template",
    "AllowArbitraryValue": false,
    "RequireInRequest": false
  }
},
...
}
```

1.3.1.5.4 Расширение «Точки распространения списков отзыва (CRL)» (CRL Distribution Points)

Информация о настройке расширения «Точки распространения списков отзыва (CRL)» содержится в объекте **CrlDistributionPointsExtension**. Значением по умолчанию является набор **Entries** массивов строк, который содержит адреса точек распространения CRL. Каждый такой массив соответствует одной точке распространения CRL.

Если настройка не содержит ни одного адреса, то адреса будут взяты из настройки ЦС Список адресов точек распространения CRL (CDP).

Пример:

```
{
...
{
  "CrlDistributionPointsExtension": {
    "DefaultValue": {
      "Entries": [
        [ "http://cdp1.ca.local/%17.crl" ],
        [ "http://cdp2.ca.local/%17.crl",
          "http://cdp2.ca.local/%17.crl" ]
      ]
    },
    "AddOnlyIfRequested": false,
    "PrioritySource": "Template",
    "AllowArbitraryValue": false,
    "RequireInRequest": false
  }
},
...
}
```

1.3.1.5.5 Расширение «Доступ к сведениям Центра сертификации» (Authority Info Access)

Информация о настройке расширения «Доступ к сведениям Центра сертификации» содержится в объекте **AuthorityInfoAccessExtension**. Значением по умолчанию является набор **Entries** структур, каждая из которых содержит информацию об одной точке доступа – назначении (**Usage**) и адрес (**Url**).

Назначение может принимать одно из следующих значений:

- **AiaIncludeInCert** (2) – включать адрес в расширение «Доступ к сведениям Центра сертификации».
- **OcspIncludeInCert** (32) – включать адрес в расширение «Точки распространения CRL» как адрес службы актуальных статусов сертификатов (OCSP).

Если настройка не содержит ни одного адреса, то адреса будут взяты из настройки ЦС. Список адресов точек доступа к информации о ЦС (AIA).

Пример:

```
{
...
  {
    "AuthorityInfoAccessExtension": {
      "DefaultValue": {
        "Entries": [
          {
            "Usage": "CdpIncludeInCert",
            "Url": "http://aia.ca.local/ca.crt"
          },
          {
            "Usage": "OcspIncludeInCert",
            "Url": "http://ocsp.ca.local/ocsp.srf"
          }
        ]
      },
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  }
...
}
```

1.3.1.5.6 Расширение «Дополнительное имя субъекта» (Subject Alternative Name)

Информация о настройке расширения «Доступ к сведениям Центра сертификации» содержится в объекте **SubjectAlternativeName**. Значением по умолчанию является набор полей структуры, каждое из которых содержит информацию о способе заполнения расширения.

Могут использоваться следующие поля:

- **DirectoryName** – поле «Имя каталога» (directoryName) расширения. Содержит массив правил **Rules** для формирования имени из запроса и свойство **PrioritySource**, определяющее приоритет источника данных для формирования имени.

Свойство **PrioritySource** может принимать одно из следующих значений:

- **Profile** – данные для формирования имени берутся в первую очередь из профиля пользователя в запросе (атрибут CpRaRequesterProfile), а если профиль отсутствует - то из субъекта.
- **Subject** – данные для формирования имени берутся в первую очередь из субъекта запроса, а если субъект не заполнен - то из профиля.

Каждое правило массива Rules представляет структуру из двух полей:

- **RdnOid** – идентификатор компонента имени, который будет добавлен в имя каталога. Является обязательным.
- **TakeFrom** – идентификатор компонента имени из запроса, значение которого будет добавлено. Является необязательным, если не задан - будет добавлено значение компонента с идентификатором **RdnOid**.
- **RequireDirectoryGuid** – флаг, определяющий добавлять ли идентификатор контроллера домена в расширение. Если флаг установлен, то значение компоненты имени из запроса с идентификатором 1.3.6.1.4.1.311.25.1 (при ее отсутствии - с идентификатором 1.2.840.113556.1.4.2) будет добавлено в расширение. Если флаг в шаблоне установлен, а соответствующие компоненты имени в запросе не заданы, то при обработке возникнет ошибка.
- **RequireDns** – флаг, определяющий добавлять ли DNS имя компьютера в расширение (поле dNSName). Обычно используется при выпуске сертификатов для веб-серверов. Если флаг установлен, то значение компоненты имени из запроса с идентификатором 1.2.643.2.2.44.5 будет добавлено в расширение. Допустимо использовать несколько DNS имен, для этого в имени из запроса они должны быть разделены символом «точка с запятой» (;). Если флаг в шаблоне установлен, а соответствующий компонент имени в запросе не задан или имеет неверный формат, то при обработке возникнет ошибка.
- **RequireIp** – флаг, определяющий добавлять ли IP адрес компьютера в расширение (поле ipAddress). Обычно используется при выпуске сертификатов для веб-серверов. Если флаг установлен, то значение компоненты имени из запроса с идентификатором 1.2.643.2.2.44.7 будет добавлено в расширение. Допустимо использовать несколько IP адресов, для этого в имени из запроса они должны быть разделены символом «точка с запятой» (;). Если флаг в шаблоне установлен, а соответствующий компонент имени в запросе не задан или имеет неверный формат, то при обработке возникнет ошибка.
- **RequireEmail** – флаг, определяющий добавлять ли email адрес в расширение (поле rfc822Name). Обычно используется при выпуске сертификатов для обмена сообщениями электронной почты. Если флаг установлен, то значение компоненты имени из запроса с идентификатором 1.2.840.113549.1.9.1 будет добавлено в расширение. Если флаг в шаблоне установлен, а соответствующий компонент имени в запросе не задан или имеет неверный формат, то при обработке возникнет ошибка.
- **OtherName** – поля «Другое имя» (otherName) расширения. Каждое поле представляет из себя последовательность OID – значение. Содержит массив правил **Rules** для заполнения полей и свойство **PrioritySource**, определяющее приоритет источника данных. Каждое правило массива **Rules** соответствует одному полю **OtherName** из расширения.

Свойство **PrioritySource** может принимать одно из следующих значений:

- **Profile** - данные для заполнения полей берутся в первую очередь из профиля пользователя в запросе (атрибут CpRaRequesterProfile), а если профиль отсутствует - то из субъекта.
- **Subject** - данные для заполнения полей берутся в первую очередь из субъекта запроса, а если субъект не заполнен - то из профиля.

Каждое правило массива **Rules** представляет структуру из двух полей:

- **RdnOid** - идентификатор компонента имени, который будет добавлен в поле «Другое имя». Является обязательным.
- **TakeForm** - идентификатор компонента имени из запроса, значение которого будет добавлено. Является необязательным, если не задан - будет добавлено значение компонента с идентификатором **RdnOid**.

Одно из частых использований полей «Другое имя» - настройка аутентификации по сертификатам в домене. В этом случае для контроллера домена в «Другом имени» задается пара из OID 1.3.6.1.4.1.311.25.1 и значения - идентификатора GUID контроллера домена, для пользователей домена - пара из OID 1.3.6.1.4.1.311.20.2.3 и значения - имени участника (userPrincipalName, upn).

Пример:

```
{
...
  {
    "SubjectAlternativeNameExtension": {
      "DefaultValue": {
        "DirectoryName": {
          "PrioritySource": "Subject",
          "Rules": [
            {
              "RdnOid": "2.5.4.6"
            },
            {
              "RdnOid": "2.5.4.3",
              "TakeFrom": "2.5.4.10"
            }
          ]
        },
        "RequireDns": true,
        "RequireIp": true,
        "RequireEmail": true,
        "OtherName": {
          "PrioritySource": "Profile",
          "Rules": [
            {
              "RdnOid": "1.3.6.1.4.1.311.20.2.3"
            },
            {
              "RdnOid": "1.3.6.1.4.1.311.25.1"
            },
            {
              "RdnOid": "2.5.4.13",
              "TakeFrom": "2.5.4.3"
            }
          ]
        }
      }
    }
  }
}
```

```

    },
    "AddOnlyIfRequested": false,
    "PrioritySource": "Template",
    "AllowArbitraryValue": false,
    "RequireInRequest": false
  }
}
...
}

```

1.3.1.5.7 Расширение «Тип идентификации при выдаче сертификата» (Identification Kind)

Информация о настройке расширения «Тип идентификации при выдаче сертификата» содержится в объекте **IdentificationKindExtension**. Значением по умолчанию является тип идентификации, который должен попасть в расширение.

Допустимы следующие значения:

- **Personal** - личное присутствие.
- **RemoteCert** - без личного присутствия с использованием квалифицированной ЭП.
- **RemotePassport** - без личного присутствия с использованием персональных данных, записанных на электронный носитель из заграничного паспорта.
- **RemoteSystem** - без личного присутствия с использованием сведений из ЕСИА и ЕБС.

Пример:

```

{
...
  {
    "IdentificationKindExtension": {
      "DefaultValue": {
        "IdentificationKind": "Personal"
      },
      "AllowedValues": [
        {
          "IdentificationKind": "Personal"
        },
        {
          "IdentificationKind": "RemoteCert"
        },
        {
          "IdentificationKind": "RemotePassport"
        },
        {
          "IdentificationKind": "RemoteSystem"
        }
      ],
      "AddOnlyIfRequested": false,
      "PrioritySource": "Request",
      "AllowArbitraryValue": false,
      "RequireInRequest": true
    }
  },
...
}

```

}

Согласно настройкам, приведенным в примере, требуется наличие расширения в запросе (т.к. именно оператор, создающий запрос, проводит идентификацию пользователя), значение расширения будет взято из запроса, определен набор допустимых значений (иные значения будут отвергнуты).

1.3.1.5.8 Расширение «Средства электронной подписи УЦ издателя» (Issuer Sign Tool)

Информация о настройке расширения «Средства электронной подписи УЦ издателя» содержится в объекте **IssuerSignToolExtension**. Значением по умолчанию является информация о средствах УЦ и ЭП, которая должна попасть в расширение.

У этого объекта есть следующие поля:

- **SignTool** – полное наименование средства ЭП, которое было использовано для создания ключа ЭП, ключа проверки ЭП и квалифицированного сертификата.
- **SignToolCert** – реквизиты заключения ФСБ России о подтверждении соответствия средства ЭП, которое было использовано для создания ключа ЭП, ключа проверки ЭП, требованиям, установленным в соответствии с Федеральным законом.
- **CaTool** – полное наименование средства аккредитованного УЦ, которое было использовано для создания ключа ЭП, ключа проверки ЭП и квалифицированного сертификата.
- **CaToolCert** – реквизиты заключения ФСБ России о подтверждении соответствия средства УЦ, которое было использовано для создания квалифицированного сертификата, требованиям, установленным в соответствии с Федеральным законом.

Пример:

```
{
...

  {
    "IssuerSignToolExtension": {
      "DefaultValue": {
        "SignTool": "СКЗИ КриптоПро CSP 5.0",
        "CaTool": "КриптоПро УЦ",
        "SignToolCert": "Сертификат соответствия ФСБ России на СКЗИ",
        "CaToolCert": "Сертификат соответствия ФСБ России на средства УЦ"
      },
      "AllowedValues": [],
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  },
...
}
```

1.3.1.5.9 Расширение «Средство электронной подписи владельца» (Subject Sign Tool)

Информация о настройке расширения «Средство электронной подписи владельца» содержится в объекте **SubjectSignToolExtension**. Значением по умолчанию является

информация о средстве ЭП владельца квалифицированного сертификата, которая должна попасть в расширение. У этого объекта есть единственное поле **SignTool**, которое содержит наименование такого средства.

Пример:

```
{
...
  {
    "SubjectSignToolExtension": {
      "DefaultValue": {
        "SignTool": "СКЗИ КриптоПро CSP"
      },
      "AllowedValues": [],
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  },
...
}
```

1.3.1.5.10 Расширение «Политики сертификата» (Certificate Policies)

Информация о настройке расширения «Политики сертификата» содержится в объекте **CertPoliciesExtension**. Значением по умолчанию является информация о политиках, в соответствии с которым должен использоваться квалифицированный сертификат. У этого объекта есть поле **PolicyIdentifiers**, содержащее массив идентификаторов OID политик. Каждый идентификатор обозначает соответствующий класс средств ЭП. Также, объект содержит флаг **Critical**, указывающий, является ли расширение критичным.

Пример:

```
{
...
  {
    "CertPoliciesExtension": {
      "DefaultValue": {
        "Critical": false,
        "PolicyIdentifiers": [
          "1.2.643.100.113.1",
          "1.2.643.100.113.2"
        ]
      },
      "AllowedValues": [],
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  },
...
}
```

1.3.1.5.11 Расширение «Период использования закрытого ключа» (Private Key Usage Period)

Информация о настройке расширения «Период использования закрытого ключа» содержится в объекте **PrivateKeyUsagePeriodExtension**. Значением по умолчанию является информация о сроке действия закрытого ключа квалифицированного сертификата.

Значение **PrivateKeyUsagePeriodExtension** используется не только для формирования соответствующего расширения в сертификате, но для контроля значения расширения из запроса. Если дата окончания срока действия закрытого ключа в запросе превышает дату окончания, вычисленную с помощью шаблона, то такой запрос будет отклонен. При этом нужно учитывать, что ключ начинает действовать одновременно с сертификатом, т.е. при вычислении даты начала действия так же берется поправка на точность часов ЦС (см. 1.2.6.6).

Пример:

В шаблоне настроен срок действия закрытого ключа 1 год, срок действия сертификата – 5 лет одинаковый в шаблоне и настройках ЦС. В запросе указаны время начала действия: 01.01.2024 12:00:00, дата окончания действия: 01.01.2025 12:00:00. Точность часов ЦС – 5 минут.

При обработке такого запроса 01.01.2024 в 12:01:00 возникнет ошибка, т.к. время начала действия сертификата будет равно 01.01.2024 11:56:00, оно же будет являться и временем начала действия закрытого ключа. Время окончания действия закрытого ключа, вычисленное с использованием шаблона, будет 01.01.2025 11:56:00. Это ограничение меньше, чем время окончания, указанное в запросе, поэтому запрос будет отклонен.

Но при обработке такого запроса 01.01.2024 в 12:10:00 аналогичным образом срок действия закрытого ключа будет вычислен как начало: 01.01.2024 12:05:00, окончание: 01.01.2025 12:05:00. В этом случае дата окончания в запросе меньше, чем вычисленное ограничение и такой запрос пройдет проверку по сроку действия закрытого ключа.

Если нужно использовать значение расширения «Период использования закрытого ключа» из запроса без какого-либо контроля по шаблону, то нужно задать значение по умолчанию этого расширения в шаблоне пустым и установить **AddOnlyIfRequested** в true.

Настройка временного интервала срока действия закрытого ключа аналогична настройке Срок действия сертификата.

Пример:

```
{
...
  {
    "PrivateKeyUsagePeriodExtension": {
      "DefaultValue": {
        "Period": {
          "PeriodType": "Months",
          "UnitCount": 15
        }
      },
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  },
...
}
```

1.3.1.5.12 Расширение «Основные ограничения» (Basic Constraints)

Информация о настройке расширения «Основные ограничения» содержится в объекте **BasicConstraintsExtension**. Значением по умолчанию является набор полей, определяющий состав сформированного по шаблону расширения, который определяет, может ли сертификат выступать в роли сертификата ЦС и задает ограничение на длину цепочки.

Используются следующие поля:

- **Ca** – определяет, является ли сертификат сертификатом ЦС. Может принимать значение **true** (является) или **false** (не является).
- **PathLength** – ограничение на длину пути цепочки (количество промежуточных сертификатов, которые могут следовать за данным ЦС в цепочке доверия). Если ограничение равно 0, то ЦС не может выпускать сертификаты подчиненных ЦС. Если ограничение отсутствует, то число промежуточных ЦС в цепочке неограниченно).
В сертификатах, не являющихся сертификатами ЦС, это ограничение всегда будет отсутствовать, вне зависимости от настройки значения по умолчанию.
- **Critical** – определяет, является ли расширения критическим. В сертификатах ЦС это расширение всегда критическое, вне зависимости от настройки значения по умолчанию.

Это расширение должно обязательно присутствовать в сертификате ЦС и так же может присутствовать во всех остальных сертификатах.

Пример:

```
{
...
{
    "BasicConstraintsExtension": {
        "DefaultValue": {
            "Ca": true,
            "PathLength": 1,
            "Critical": true
        },
        "AllowedValues": [],
        "AddOnlyIfRequested": false,
        "PrioritySource": "Template",
        "AllowArbitraryValue": false,
        "RequireInRequest": false
    }
},
...
}
```

1.3.1.5.13 Произвольное расширение

В сертификат так же возможно добавить произвольное расширение, не указанное в предыдущих разделах документации. Произвольное расширение задается в разделе **RawExtension**, его объектный идентификатор определяется свойством **Oid**. В значении по умолчанию в свойстве **RawValue** задается закодированное в ASN.1 значение самого расширения в виде строки в кодировке base64. Строка base64 не должна содержать символы «перевод строки» (0x0A) или «возврат каретки» (0x0D). ЦС никак не контролирует правильность этого значения, поэтому ответственность за корректную настройку этого расширения лежит целиком на администраторе УЦ.

Пример:

```
{
...
  {
    "RawExtension": {
      "Oid": "1.3.6.1.4.1.311.20.2",
      "DefaultValue": {
        "RawData": "DAR0ZXN0"
      },
      "AllowedValues": [],
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  },
...
}
```

В приведенном примере показана настройка расширения «Имя шаблона сертификата» (1.3.6.1.4.1.311.20.2). Значение по умолчанию расширения содержит строку UTF8String со значением «test» (в бинарном виде 0C 04 74 65 73 74), закодированную в base64.

1.3.1.5.14 Расширение «NTDS_CA_SECURITY_EXT»

Некритическое расширение «NTDS_CA_SECURITY_EXT» содержится в сертификатах, используемых в winlogon. Оно содержит SID объекта LDAP, закодированный в OCTET STRING. Описание этого расширения приведено в статье https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-wcce/e563cff8-1af6-4e6f-a655-7571ca482e71.

Информация о настройке расширения «NTDS_CA_SECURITY_EXT» содержится в объекте **NtdsCaSecurityExtension**. Значением по умолчанию является OID компонента, из которого будет взято значение SID в виде строки для формирования расширения. Если OID не задан, то будет использован стандартный OID из схемы ActiveDirectory 1.2.840.113556.1.4.146 (objectSid).

Пример:

```
{
...
  {
    "NtdsCaSecurityExtension": {
      "DefaultValue": {
        "SidAttrOid": "1.2.840.113556.1.4.146"
      },
      "AllowedValues": [],
      "AddOnlyIfRequested": false,
      "PrioritySource": "Template",
      "AllowArbitraryValue": false,
      "RequireInRequest": false
    }
  },
...
}
```

1.3.2 Список шаблонов ЦС

Для вывода списка шаблонов ЦС на консоль используется команда **pkica ca template list**. По умолчанию команда выводит для каждого шаблона краткую информацию: короткое имя, идентификатор OID и версию. Если же указан параметр **--detailed (-d)**, то шаблоны будут выведены полностью, со всем содержимым.

Для переноса шаблонов в настройки Центра регистрации нужно сохранить список шаблонов с полной информацией в файл в формате JSON:

```
./pkica ca template list --detailed --file templates.json --format Json
```

1.3.3 Получение шаблона

Для получения полной информации о конкретном шаблоне ЦС используется команда **pkica ca template get --name <имя шаблона> --file <имя файла для сохранения>**. Параметр **file** необязательный, если он не указан, то информация о шаблоне будет выведена на консоль. Если же параметр **file** задан, то шаблон будет сохранен в указанном файле в формате JSON.

Пример:

```
./pkica ca template get --name User --file user.json
```

1.3.4 Добавление нового шаблона

Для добавления нового шаблона ЦС из файла используется команда **pkica ca template add --file <файл с шаблоном в формате JSON>**.

Пример:

```
./pkica ca template add --file new-template.json
```

Если требуется добавить один или несколько шаблонов по умолчанию, то используется команда **pkica ca template add --default-template <имя шаблона по умолчанию>**. Определены следующие шаблоны по умолчанию: **SubCa** (подчиненный ЦС), **User** (пользователь), **Tls** (шаблоны для клиента и сервера TLS), **Qual** (набор шаблонов для квалифицированных сертификатов пользователя), **QualOcsp** (набор шаблонов для квалифицированных сертификатов службы OCSP), **QualTsp** (набор шаблонов для квалифицированных сертификатов службы TSP).

Пример:

```
./pkica ca template add --default-template User
```

1.3.5 Редактирование шаблона

Для обновления информации, содержащейся в шаблоне ЦС, используется команда **pkica ca template update --file <файл с шаблоном в формате JSON>**.

Для редактирования шаблона его нужно выгрузить в файл формата JSON командой **pkica ca template get** (см. 0), отредактировать этот файл в текстовом редакторе и обновить шаблон командой **pkica ca template update**.

Пример:

```
./pkica ca template get --name User --file user.json
vi user.json
./pkica ca template update --file user.json
```

Если требуется вернуть один из шаблонов по умолчанию к значениям по умолчанию, то используется команда **pkica ca template update --default-template <имя шаблона по умолчанию>**.

1.3.6 Удаление шаблона

Для удаления шаблона используется команда **pkica ca template remove --name <имя шаблона>**. Используйте эту команду с осторожностью, после удаления шаблона вся содержащаяся в нем информация будет потеряна!

Пример:

```
./pkica ca template remove --name OldTemplate
```

1.3.7 Создание файла с закодированным расширением «Шаблон сертификата»

Если требуется создать файл с закодированным расширением «Шаблон сертификата» используется команда **pkica ca template encode-extension certificate-template --name <имя шаблона> --file <имя файла для сохранения>**.

Пример:

```
./pkica ca template encode-extension certificate-template --name User --file /tmp/templateext.der
```

1.3.8 Создание файла с закодированным расширением «Тип идентификации при выдаче сертификата»

Если требуется создать файл с закодированным расширением «Тип идентификации при выдаче сертификата» используется команда **pkica ca template encode-extension identification-kind --value <тип идентификации> --file <имя файла для сохранения>**.

Пример:

```
./pkica ca template encode-extension identification-kind --value Personal --file /tmp/idkindext.der
```

1.4 Списки отзыва (CRL)

1.4.1 Выпуск CRL

Для выпуска списка отзыва (CRL) используется команда **pkica ca crl issue**.

Параметры:

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-d --delta-only	Выпустить только разностный CRL (без базового).
-u --next-update	Время до которого действует CRL. Формат ISO 8601: yyyy-MM-ddThh:mm:ss[±hh:mm].
--final	Выпустить финальный CRL - такой CRL, который действует

	до окончания срока действия сертификата подписи ЦС.
--key-index	Индекс ключа, для которого нужно выпустить финальный CRL. Этот ключ предварительно должен быть загружен и выведен из эксплуатации. Подробнее см. в 1.2.14.
-o --output-folder	Имя папки, в которую надо сохранить выпущенные CRL.
--force	Флаг подавляет запрос подтверждений.

1.4.2 Получение CRL, действующих на указанный момент времени

Для получения CRL используется команда **pkica ca crl get**.

Параметры (обязательные параметры помечены звездочкой *):

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-v --valid-at	Момент времени, на который должны действовать CRL. По умолчанию будет выбран текущий момент. Формат ISO 8601: yyyy-MM-ddThh:mm:ss[+hh:mm].
--all	Получить все действующие на заданный момент времени CRL. На заданный момент времени для ключа подписи ЦС может существовать более одного действующего CRL (последовательно выпущенные CRL могут перекрываться по сроку действия или может быть выпущен внеочередной CRL). По умолчанию будут возвращены только CRL с максимальным номером. Укажите этот флаг, если хотите получить все действующие на заданный момент времени CRL (в этом случае имена файлов будут содержать номер CRL для обеспечения уникальности имен).
-o --output-folder	* Имя папки, в которую надо сохранить полученные CRL.
--force	Флаг подавляет запрос подтверждений перезаписи файлов.

1.4.3 Расписание выпуска CRL

ЦС выпускает CRL согласно настроенному расписанию. Расписание хранится на ЦС и может быть просмотрено или изменено с помощью подкоманд команды **pkica ca crl schedule**.

1.4.3.1 Формат расписания выпуска CRL

Расписание выпуска хранится в формате CRON и позволяет выполнять гибкую настройку периодичности выпуска CRL.

Строка расписания состоит из шести значений, разделенных пробелом. Можно использовать как классический формат CRON (пять разрядов), так и расширенный формат (шесть разрядов, с секундами).

- первая позиция – секунды, допустимые значения 0-59, *, /, данный разряд необязательный;
- вторая позиция – минуты, допустимые значения 0-59, *, /;
- третья позиция – часы, допустимые значения 0-23, *, /;
- четвертая позиция – день месяца, допустимые значения 1-31, *, /;
- пятая позиция – месяц, допустимые значения 1-12, *, /;
- шестая позиция – день недели, допустимые значения 0-7 (воскресенье - 0 или 7), *, /.

Символ "*" обозначает любое значение. Также можно указывать диапазоны значений, используя знак "-" (пример 1-15, что означает с 1 по 15 число). Еще один спецсимвол "/" позволяет определить шаг выполнения (пример */5 в поле часов – каждые 5 часов; 1-15/3 в поле день месяца – каждые 3 дня с 1 по 15 число). Значения и диапазоны можно объединять с помощью запятой (пример: 3,5-11/3,12 эквивалентно 3,5,8,11,12).

Примеры: **0 0 18 * * *** раз в день в 18 часов; **0 0 */5 * * *** каждые 5 часов, ***/5 * * *** * каждые 5 минут, **0 0 1 * *** каждое первое число месяца в полночь.

1.4.3.2 Получение расписания выпуска CRL

Получение расписания выпуска CRL выполняется с помощью команды **pkica ca crl schedule get**.

Параметры:

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-----------	---

Пример команды получения текущего расписания выпуска CRL:

```
./pkica ca crl schedule get --name "CryptoPro CA"
```

Команда вернет текущее расписание выпуска базового и разностного CRL.

1.4.3.3 Изменение расписания выпуска CRL

Изменение расписания выпуска CRL выполняется с помощью команды **pkica ca crl schedule update**.

Параметры:

-n --name	Имя экземпляра ЦС. Если не задано, то будет использован экземпляр по умолчанию.
-b --base	Расписание выпуска базового CRL в формате cron.
-d --delta	Расписание выпуска разностных CRL в формате cron. Примеры: 0 0 18 * * * раз в день в 18 часов, 0 0 */5 * * * каждые 5 часов. Если необходимо отключить выпуск разностных CRL, то в качестве значения параметра указать минус (-).

Пример команды изменения расписания:

```
./pkica ca crl schedule update --name "CryptoPro CA" --base "0 0 18 * * *" --delta -
```

Здесь настраивается выпуск базового CRL каждый день в 18 часов, выпуск разностного CRL отключен.

1.5 Отзыв сертификатов

При начальном развороте УЦ, «первые» сертификаты выпускаются напрямую на ЦС. Такие сертификаты нельзя отозвать через ЦР, так как они есть только на ЦС. Чтобы отозвать такие сертификаты надо создать запрос на отзыв и отправить его на ЦС.

1.5.1 Создание запроса на отзыв сертификата

Создание запроса на отзыв выполняется с помощью команды **pkica ca new-rev-req**.

Параметры (обязателен для указания файл с отзываемым сертификатом или серийный номер и отпечаток отзываемого сертификата):

-c --cert-file	Имя файла с отзываемым сертификатом.
-s --serial-number	Серийный номер отзываемого сертификата.
-t --thumbprint	Отпечаток отзываемого сертификата. Задается вместе с серийным номером отзываемого сертификата.
--reason	Причина отзыва. Допустимые значения: Unspecified, AffiliationChanged, CACompromise, CessationOfOperation, KeyCompromise, Superseded.
-o --out-file	Имя файла для сохранения запроса. Если не задано, то запрос будет выведен на консоль в кодировке base64.

1.5.2 Отправка запроса на отзыв

Для отзыва сертификата необходимо передать на ЦС запрос на отзыв с помощью команды **pkica ca revoke-cert**.

Параметры (звездочкой * помечены обязательные):

-f --file	* Имя файла с запросом на отзыв сертификата.
-t --timeout	Таймаут ожидания отзыва сертификата. Задается в формате ISO 8601: [d.]hh:mm[:ss]. Значение по умолчанию: 00:01 (1 минута). Если равен -1, то ожидание будет бесконечным; если 0, то будет выполнена отправка команды без ожидания отзыва сертификата.

Обращение этой команды к ЦС происходит через NATS, поэтому предварительно должен быть настроен и запущен сервис NATS Streaming.

1.6 Получение сертификатов

Для получения сертификатов используется команда **pkica ca cert get**

Параметры (обязателен для заполнения один из параметров поиска, помеченных знаком плюс (+)):

-i --id	+ Идентификатор сертификата.
-s --serial-number	+ Серийный номер сертификата.
-f --file	Имя файла, в который надо сохранить сертификат.
--force	Флаг подавляет запрос подтверждения перезаписи файла.

Для получения списка сертификатов используется команда **pkica ca cert list**

Параметры:

--from	Момент времени, с которого будут показаны сертификаты. Формат ISO 8601: yyyy-MM-ddThh:mm:ss[±hh:mm].
--to	Момент времени, до которого будут показаны сертификаты. Формат ISO 8601: yyyy-MM-ddThh:mm:ss[±hh:mm].
--take	Количество выводимых записей. Значение по умолчанию можно задать в конфигурации appsettings.json: "CaOptions": {"PageSize": <количество записей>},
--oldest-first	Выводить старые записи вначале. По умолчанию записи выводятся в обратном хронологическом порядке (самые свежие – сверху). Флаг позволяет вывести записи в хронологическом порядке (самые старые – сверху).
--format	Формат вывода объектов на печать (Json/Yaml)

1.7 Получение запросов на сертификат

Для получения запросов на сертификат используется команда **pkica ca cert-request get**

Параметры (звездочкой * помечены обязательные):

-i --id	* Идентификатор запроса.
-f --file	Имя файла, в который надо сохранить запрос.
--force	Флаг подавляет запрос подтверждения перезаписи файла.

Для получения списков запросов на сертификат используется команда **pkica ca cert-request list**.

Параметры:

--from	Момент времени, с которого будут показаны запросы. Формат ISO 8601: yyyy-MM-ddThh:mm:ss[±hh:mm].
--to	Момент времени, до которого будут показаны запросы. Формат ISO 8601: yyyy-MM-ddThh:mm:ss[±hh:mm].
--take	Количество выводимых записей. Значение по умолчанию можно задать в конфигурации appsettings.json: "CaOptions": {"PageSize": <количество записей>},
--oldest-first	Выводить старые записи вначале. По умолчанию записи выводятся в обратном хронологическом порядке (самые свежие – сверху). Флаг позволяет вывести записи в хронологическом порядке (самые старые – сверху).
--format	Формат вывода объектов на печать (Json/Yaml)

1.8 Получение запросов на отзыв

Для получения запросов на сертификат используется команда **pkica ca cert-request get**

Для получения запросов на отзыв сертификата используется команда **pkica ca rev-request get**

Параметры (звездочкой * помечены обязательные):

-i --id	* Идентификатор запроса.
-f --file	Имя файла, в который надо сохранить запрос.
--force	Флаг подавляет запрос подтверждения перезаписи файла.

Для получения списков запросов на отзыв сертификата используется команда **pkica ca rev-request list**.

Параметры:

--from	Момент времени, с которого будут показаны запросы. Формат ISO 8601: yyyy-MM-ddThh:mm:ss[±hh:mm].
--to	Момент времени, до которого будут показаны запросы. Формат ISO 8601: yyyy-MM-ddThh:mm:ss[±hh:mm].
--take	Количество выводимых записей. Значение по умолчанию можно задать в конфигурации appsettings.json: "CaOptions": {"PageSize": <количество записей>},
--oldest-first	Выводить старые записи вначале. По умолчанию записи выводятся в обратном хронологическом порядке (самые свежие – сверху). Флаг позволяет вывести записи в хронологическом порядке (самые старые – сверху).

--format	Формат вывода объектов на печать (Json/Yaml)
----------	--

2 Сервер Центра регистрации

2.1 Операторы

2.1.1 Добавление оператора

Для добавления оператора используется команда **pkica ra operator add**. Для оператора так же добавляется сертификат, который будет использоваться для доступа к ЦР.

Перед добавлением оператора требуется, чтобы предварительно были выпущены сертификат TLS аутентификации на Центре Регистрации (неквалифицированный) и сертификат подписи (квалифицированный), удовлетворяющие следующим требованиям:

- сертификат подписи должен содержать биты «Цифровая подпись» (0) и «Неотрекаемость» (1) в расширении «Использование ключа»
- сертификат аутентификации должен содержать назначение «Проверка подлинности клиента» (идентификатор 1.3.6.1.5.5.7.3.2) в расширении «Улучшенный ключ»

Имя входа (логин) рекомендуется задавать так, чтобы оно было понятным и легко запоминающимся, но при этом уникальным. Подходящие варианты - сокращение ФИО или адрес email.

Параметры (звездочкой * помечены обязательные):

-l --last-name	* Фамилия оператора.
-f --first-name	* Имя оператора.
-m --middle-name	Отчество оператора.
--tls-cert-file	* Путь к файлу сертификата аутентификации оператора.
--signing-cert-file	* Путь к файлу сертификата подписи оператора.
-g --group	Название группы, в которую надо добавить оператора.
--default-folder	Имя папки по умолчанию. Эта папка будет использоваться, когда оператор явно не указал папку при создании пользователя, получении данных и т.п.
--login	Имя входа (логин) оператора. Является уникальным отличительным свойством каждого оператора, служит для идентификации оператора на ЦР. Если не задано, то будет сгенерировано автоматически из ФИО.

Пример:

```
./pkica ra operator add --last-name Иванов --first-name Иван --tls-cert-file operator-auth.cer --signing-cert-file operator-sign.cer --group Операторы --login ivanov --default-folder Operators
```

2.1.2 Обновление данных оператора

Для обновления данных оператора используется команда **pkica ra operator update**.

Обязательный параметр для заполнения - имя входа (логин) оператора помечен звездочкой (*). Остальные параметры, отвечающие за изменение данных, необязательны. Будут изменены только те данные оператора, которые указаны.

--login	* Имя входа (логин) оператора (для поиска).
--new-login	Новое имя входа (логин) оператора.
-l --last-name	Новая фамилия оператора.
-f --first-name	Новое имя оператора.
-m --middle-name	Новое отчество оператора.
--tls-cert-file	Путь к новому файлу сертификата аутентификации оператора.
--signing-cert-file	Путь к новому файлу сертификата подписи оператора.
--default-folder	Имя папки ЦР по умолчанию.
--clear-default-folder	Сбросить папку по умолчанию.

Пример:

```
./pkica ra operator update --login ivanov --last-name Иванов --new-login ivanov1
```

2.1.3 Получение данных об операторе

Для получения данных об операторе используется команда **pkica ra operator get**. Выводится ФИО оператора, данные сертификатов подписи и аутентификации, дата создания оператора, разрешено ли подключение к ЦР (активирован/деактивирован), список групп, в которых состоит оператор и название папки по умолчанию.

Обязательные параметры помечены звездочкой (*):

--login	* Имя входа (логин) оператора (для поиска).
---------	---

Пример:

```
./pkica ra operator get --login ivanov
```

2.1.4 Получение списка операторов

Для получения списка операторов используется команда **pkica ra operator list**.

-g --group-name	Название группы оператора.
-l --last-name	Фамилия оператора.

-f --first-name	Имя оператора. Параметр может использоваться только совместно с фамилией.
-m --middle-name	Отчество оператора. Параметр может использоваться только совместно с фамилией и именем.
-t --take	Количество выводимых записей. Значение по умолчанию можно задать в конфигурации.

Пример:

```
./pkica ra operator list
```

2.1.5 Активация оператора

Для активации оператора используется команда **pkica ra operator activate**. Активированному оператору разрешено подключение к ЦР.

Параметры (звездочкой * помечены обязательные):

--login	* Имя входа (логин) оператора.
---------	--------------------------------

Пример:

```
./pkica ra operator activate --login ivanov
```

2.1.6 Деактивация оператора

Для деактивации оператора используется команда **pkica ra operator deactivate**. Деактивированному оператору запрещено подключение к ЦР.

Параметры (звездочкой * помечены обязательные):

--login	* Имя входа (логин) оператора.
---------	--------------------------------

Пример:

```
./pkica ra operator deactivate --login ivanov
```

2.2 Группы операторов

2.2.1 Добавление группы

Для добавления группы операторов используется команда **pkica ra group add**.

Параметр -n|--name (название группы) является обязательным и помечен звездочкой (*). Название группы должно быть уникальным (на Центре регистрации не должно существовать групп с тем же именем).

-n --name	* Название группы.
-----------	--------------------

Пример:

```
./pkica ra group add --name Test
```

2.2.2 Переименование группы

Для переименования группы используется команда **pkica ra group rename**.

Новое название группы должно быть уникальным также, как и старое. Старое и новое названия группы являются обязательными параметрами этой команды.

-n --name	* Текущее название группы (для поиска).
--new-name	* Новое название группы.

Пример:

```
./pkica ra group rename --name Test --new-name Prod
```

2.2.3 Получение списка групп

Для получения списка групп используется команда **pkica ra group list**.

Пример:

```
./pkica ra group list
```

2.2.4 Получение информации о группе

Для получения информации о группе используется команда **pkica ra group get**. Название группы - обязательный параметр этой команды. В выводе содержится информация о всех операторах, содержащихся в группе и о всех папках, к которым группа имеет доступ. Для каждой папки будет выведен список разрешений, установленных для группы.

-n --name	* Название группы.
-----------	--------------------

Пример:

```
./pkica ra group get --name Test
```

2.2.5 Удаление группы

Для удаления группы операторов используется команда **pkica ra group delete**.

При удалении группы в ней не должны находиться операторы. Перед удалением группы нужно удалить операторов из неё.

Параметры (звездочкой * помечены обязательные):

-n --name	* Название группы.
-----------	--------------------

Пример:

```
./pkica ra group delete --name Test
```

2.2.6 Добавление оператора в группу

Для добавления оператора в группу используется команда **pkica ra group add-operator**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Название группы.
-l --login	* Имя входа (логин) оператора.

Пример:

```
./pkica ra group add-operator --name Test --login ivanov
```

2.2.7 Удаление оператора из группы

Для удаления оператора из группы используется команда **pkica ra group remove-operator**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Название группы.
-l --login	* Имя входа (логин) оператора.

Пример:

```
./pkica ra group remove-operator --name Test --login ivanov
```

2.3 Политики имен

Политика имен определяет набор атрибутов, разрешенных для использования в имени пользователя, а также правила проверки значений этих атрибутов. Политика имен используется для контроля состава имени пользователя при регистрации и переименовании.

Пример описания атрибутов в политике имен:

```
{
  "nameAttributes": [
    {
      "oid": "2.5.4.3",
      "stringIdentifier": "CN",
      "displayName": "Общее имя",
      "required": true,
      "requiredForCertRequest": false,
      "excludeFromCertRequest": false,
      "retired": false
    },
    {
      "oid": "2.5.4.12",
      "stringIdentifier": "T",
      "displayName": "Должность или звание",
      "required": false,
      "requiredForCertRequest": false,
      "excludeFromCertRequest": false,
      "allowedValues": [
        "Генеральный директор",
        "Начальник отдела"
      ],
      "retired": false
    }
  ]
}
```

```

    },
    {
      "oid": "1.2.643.100.1",
      "stringIdentifier": "OGRN",
      "displayName": "ОГРН",
      "required": false,
      "requiredForCertRequest": false,
      "excludeFromCertRequest": false,
      "regex": "^[0-9]{13}$",
      "retired": false
    },
    ...
  ]
}

```

Каждый атрибут имеет следующие свойства:

- **oid** – объектный идентификатор атрибута имени пользователя. Является обязательным свойством.
- **stringIdentifier** – строковый идентификатор атрибута имени пользователя (например CN, SN, O и т.п.). Является обязательным свойством.
- **displayName** – отображаемое имя атрибута имени пользователя (например «Общее имя», «Фамилия», «Организация» и т.п.). Является обязательным свойством.
- **required** – флаг показывает является ли атрибут имени пользователя обязательным. Если в имени нового пользователя такой атрибут отсутствует, то и самого пользователя создать не удастся. По умолчанию флаг не установлен.
- **requiredForCertRequest** – флаг показывает является ли атрибут имени пользователя обязательным на момент создания/одобрения запроса на сертификат.
- **excludeFromCertRequest** – флаг показывает, что данный атрибут не должен включаться в запрос на сертификат.
- **minLength** – минимальная длина атрибута имени пользователя. По умолчанию не задана.
- **maxLength** – максимальная длина атрибута имени пользователя. По умолчанию не задана.
- **allowedValues** – набор допустимых значений атрибута имени пользователя. Если набор пуст - атрибут имени пользователя может принимать любые значения.
- **regex** – регулярное выражение, которому должен удовлетворять атрибут имени пользователя.
- **retired** – флаг указывает, что атрибут выведен из эксплуатации. Если установлен, то атрибут не может использоваться в имени пользователя при создании нового пользователя и при переименовании существующего пользователя. Если имя пользователя содержит атрибут, выведенный из эксплуатации, то для такого пользователя нельзя отправить запрос на сертификат. Сначала надо изменить данные пользователя и удалить этот атрибут из его имени. Атрибут, выведенный из эксплуатации, не рекомендуется удалять из политики имен, так как могут существовать пользователи, для которых задано

значение этого атрибута. Для такого атрибута рекомендуется устанавливать флаг **retired**.

2.3.1 Добавление (регистрация) политики имён

Политика имен регистрируется с помощью команды **pkica ra name-policy add**. Одна из политик может быть назначена политикой по умолчанию. Политика имен по умолчанию применяется для всех папок, для которых политика не назначена явно. После регистрации политики имён на уровне сервера, можно назначить политику имён для той или иной папки (см. набор команд для управления папками).

Параметры (звездочкой * помечены обязательные):

-n --name	* Наименование политики.
-f --file	Путь к файлу с политикой имен. Файл должен содержать определение политики в формате json. Если не указан, будет создана политика со значениями по умолчанию.
-d --default	Назначить политикой по умолчанию на уровне сервера. Политика имен по умолчанию применяется для всех папок, для которых политика не назначена явно.

Пример:

```
./pkica ra name-policy add --name ЮЛ --file name-policy.json
```

2.3.2 Получение политики имён

Для получения зарегистрированной политики имен используется команда **pkica ra name-policy get**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Наименование политики.
-f --file	Путь к файлу, в который надо сохранить политику имён. Если не указано, результат будет выведен на экран.

Пример:

```
./pkica ra name-policy get --name ЮЛ --file name-policy.json
```

2.3.3 Получение списка зарегистрированных политик имён

Для получения списка зарегистрированных политик имён используется команда **pkica ra name-policy list**.

Пример:

```
./pkica ra name-policy list
```

2.3.4 Удаление политики имён

Для удаления политики имён используется команда **pkica ra name-policy remove**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Наименование политики.
-----------	--------------------------

Пример:

```
./pkica ra name-policy remove --name ЮЛ
```

2.3.5 Выбор политики имён по умолчанию

Для выбора политики имён по умолчанию используется команда **pkica ra name-policy set-default**. Политика имен, назначенная политикой по умолчанию, применяется для всех папок, для которых политика не назначена явно.

Параметры (обязателен для заполнения один из параметров, помеченных знаком плюс (+)):

-n --name	+ Наименование политики.
--none	+ Сбросить значение политики имён по умолчанию.

Пример:

```
./pkica ra name-policy set-default --name ЮЛ
```

2.3.6 Обновление политики имён

Для обновления политики имён используется команда **pkica ra name-policy update**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Наименование политики.
-f --file	* Путь к файлу с политикой имен. Файл должен содержать определение политики в формате json.
-d --default	Назначить политикой по умолчанию на уровне сервера. Политика имен по умолчанию применяется для всех папок, для которых политика не назначена явно.

Пример:

```
./pkica ra name-policy update --name ЮЛ --file name-policy.json
```

2.4 Шаблоны сертификатов

Шаблон сертификата содержит набор правил создания сертификата по запросу на сертификат, а также правил создания ключа. Структура шаблона описана в п. 1.3.1.

Для регистрации шаблонов на ЦР, потребуется файл с описанием одного или нескольких шаблонов. Подготовить такой файл возможно на ЦС в соответствии с п. 1.3.2 или 1.3.3 настоящего руководства.

2.4.1 Добавление шаблона сертификата

Добавление шаблона сертификата выполняется с помощью команды **pkica ra template add**. После регистрации шаблона на уровне сервера, можно назначить шаблон той или иной папке (см. набор команд для управления папками).

Параметры (звездочкой * помечены обязательные):

-f --file	* Путь к файлу, содержащему один или несколько шаблонов сертификата в формате JSON.
-----------	---

Пример:

```
./pkica ra template add --file template.json
```

2.4.2 Управление режимом автообновления

Помимо ручного управления шаблонами, можно настроить автообновление шаблонов с ЦС – при изменении шаблонов на ЦС изменения будут автоматически передаваться на ЦР. Управление режимом автообновления выполняется с помощью команды **pkica ra template auto-update**.

Параметры:

--enable	Включить режим автоматического обновления шаблонов.
--disable	Выключить режим автоматического обновления шаблонов.

Пример:

```
./pkica ra template auto-update --enable
```

При запуске команды без параметров выводится текущее состояние (включен/выключен) режима автообновления шаблонов.

2.4.3 Получение шаблона сертификата

Получение шаблона сертификата выполняется с помощью команды **pkica ra template get**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Имя шаблона сертификата.
-f --file	Имя файла, в который надо сохранить шаблон. Если не указано, то результат будет выведен на экран.

Пример:

```
./pkica ra template get --name User --file template.json
```

2.4.4 Получение списка шаблонов сертификатов

Получение списка шаблонов сертификатов выполняется с помощью команды **pkica ra template list**.

Параметры:

-f --file	Путь к файлу для сохранения списка шаблонов. Если не указан, то результат будет выведен на экран.
-d --detailed	Получить полную информацию о шаблонах.

Пример:

```
./pkica ra template list
```

2.4.5 Удаление шаблона сертификата

Удаление шаблона сертификата выполняется с помощью команды **pkica ra template remove**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Имя шаблона сертификата
-----------	---------------------------

Пример:

```
./pkica ra template remove --name User
```

2.4.6 Обновление шаблона сертификата

Обновление шаблона сертификата выполняется с помощью команды **pkica ra template update**.

Параметры (звездочкой * помечены обязательные):

-f --file	* Путь к файлу, содержащему один или несколько шаблонов сертификата в формате JSON.
-----------	---

Пример:

```
./pkica ra template update --file template.json
```

2.5 Издатели сертификатов

Настройка издателей сертификатов позволяет определить перечень издателей (экземпляров ЦС), на которых можно выпускать сертификаты.

2.5.1 Добавление (регистрация) издателя сертификатов

Добавление (регистрация) издателя сертификатов выполняется с помощью команды **pkica ra authority add**. После регистрации издателя на уровне сервера, можно назначить издателя той или иной папке (см. набор команд для управления папками).

Параметры (звездочкой * помечены обязательные):

-n --name	* Уникальное имя (псевдоним) издателя.
-d --default	Назначить издателем по умолчанию (на уровне сервера, на

	уровне папки может быть назначен иной издатель по умолчанию). При регистрации первого издателя на сервере, он будет назначен издателем по умолчанию в любом случае (независимо от того указан данный параметр или нет).
--	---

Пример:

```
./pkica ra authority add --name ЦС --default
```

2.5.2 Получение списка зарегистрированных издателей сертификатов

Получение списка зарегистрированных издателей сертификатов выполняется с помощью команды **pkica ra authority list**.

Пример:

```
./pkica ra authority list
```

2.5.3 Удаление зарегистрированного издателя сертификатов

Удаление зарегистрированного издателя сертификатов выполняется с помощью команды **pkica ra authority remove**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Имя (псевдоним) издателя.
-----------	-----------------------------

Пример:

```
./pkica ra authority remove --name ЦС
```

2.5.4 Выбор издателя по умолчанию

Выбор издателя по умолчанию выполняется с помощью команды **pkica ra authority set-default**. Издатель по умолчанию используется для обработки запросов на сертификат, для которых не указан ЦС.

Параметры (обязателен для заполнения один из параметров, помеченных знаком плюс (+)):

-n --name	+ Имя (псевдоним) издателя.
--none	+ Сбросить значение издателя по умолчанию.

Пример:

```
./pkica ra authority set-default --name ЦС
```

2.6 Папки

Папки служат контейнерами для пользователей, запросов, сертификатов. На папку могут быть выданы разрешения для одной или нескольких групп операторов. Каждая папка может иметь свои настройки политики имен, шаблонов сертификатов и т.д. Структура папок является "плоской" одноуровневой, вложенные папки не поддерживаются.

2.6.1 Создание папки

Для создания папки используется команда **pkica ra folder add**.

Параметр `-n|--name` (имя папки) является обязательным и помечен звездочкой (*). Имя папки должно быть уникальным (на Центре регистрации не должно существовать папок с тем же именем).

Параметры (звездочкой * помечены обязательные):

<code>-n --name</code>	* Имя папки.
------------------------	--------------

Пример:

```
./pkica ra folder add --name Пользователи
```

2.6.2 Переименование папки

Для переименования папки используется команда **pkica ra folder rename**.

Новое имя папки должно быть уникальным также, как и старое.

Параметры (звездочкой * помечены обязательные):

<code>-n --name</code>	* Текущее имя папки (для поиска).
<code>--new-name</code>	* Новое имя папки.

Пример:

```
./pkica ra folder rename --name Пользователи --new-name 'Тестовые пользователи'
```

2.6.3 Вывод списка папок

Для вывода списка папок используется команда **pkica ra folder list**.

Пример:

```
./pkica ra folder list
```

2.6.4 Получение информации о папке

Для получения информации о папке используется команда **pkica ra folder get**. Имя папки - обязательный параметр этой команды. В выводе содержится информация о всех группах операторов, имеющих доступ к этой папке. Для каждой группы будет выведен список разрешений.

Параметры (звездочкой * помечены обязательные):

<code>-n --name</code>	* Имя папки.
------------------------	--------------

Пример:

```
./pkica ra folder get --name Пользователи
```

2.6.5 Удаление папки

Для удаления папки используется команда **pkica ra folder delete**.

Удаляемая папка должно быть пуста - в ней не должно содержаться пользователей, и она не должна являться папкой по умолчанию для какого-либо оператора. Имя папки - обязательный параметр этой команды.

Параметры (звездочкой * помечены обязательные):

-n --name	* Имя папки.
-----------	--------------

Пример:

```
./pkica ra folder delete --name Пользователи
```

2.6.6 Назначение политики имён для папки

Если для папки не назначена политика имён, то будет использоваться политика имён по умолчанию уровня сервера (см. 2.3).

Если для папки назначена политика имён, то для папки перестает действовать политика имён по умолчанию, назначенная на уровне сервера. При отсутствии политики имён по умолчанию на уровне сервера и на уровне папки, в папке будет невозможно создавать пользователей. В каждой папке может использоваться только одна политика имён.

Назначение политики имён для папки выполняется с помощью команды **pkica ra folder assign-name-policy**.

Параметры (звездочкой * помечены обязательные параметры; один из параметров, отмеченных знаком плюс +, также должен быть указан):

-n --name	* Имя папки.
-p --policy	+ Наименование политики имён.
--none	+ Сбросить политику имён для папки (для папки будет использоваться политика по умолчанию, заданная на уровне сервера).

Пример:

```
./pkica ra folder assign-name-policy --name Пользователи --policy ЮЛ
```

2.6.7 Определение набора шаблонов сертификатов

Если для папки не назначены шаблоны, то в папке разрешено использовать любой из шаблонов, зарегистрированных на сервере. Назначение шаблонов для папки позволяет ограничить набор используемых шаблонов в папке.

Шаблоны для папки назначаются из числа зарегистрированных шаблонов на сервере, поэтому перед назначением шаблона для папки, необходимо добавить (зарегистрировать) его на уровне сервера (см. 2.4).

Назначение шаблона для папки выполняется командой **pkica ra folder assign-template**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Имя папки.
-t --template	* Имя или OID шаблона.

Пример:

```
./pkica ra folder assign-template --name Пользователи --template User
```

Отмена назначения шаблона для папки выполняется командой **pkica ra folder deassign-template**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Имя папки.
-t --template	* Имя или OID шаблона.

Пример:

```
./pkica ra folder deassign-template --name Пользователи --template User
```

2.6.8 Определение набора разрешенных издателей

Если для папки не назначены издатели, то в папке разрешено использовать любой из издателей, зарегистрированных на сервере. Назначение издателей для папки позволяет ограничить набор используемых издателей в папке.

Издатели для папки назначаются из числа зарегистрированных издателей на сервере, поэтому перед назначением издателя для папки, необходимо добавить (зарегистрировать) его на уровне сервера (см. 2.5).

Если для папки назначены издатели, то для папки перестает действовать настройка издателя по умолчанию уровня сервера. При необходимости, на уровне папки может быть свой издатель по умолчанию.

Повторное назначение издателя для папки (если такой издатель уже был назначен для данной папки) не приводит к дублированию издателя в настройках и может быть использовано, например, чтобы назначить издателя издателем по умолчанию.

Назначение издателя сертификатов для папки выполняется командой **pkica ra folder assign-authority**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Имя папки.
-a --authority	* Имя (псевдоним) издателя.
-d --default	Назначить издателем по умолчанию для папки (будет использоваться по умолчанию если не указан явно при отправке запроса на сертификат).

Пример:

```
./pkica ra folder assign-authority --name Пользователи --authority ЦС
```

Отмена назначения издателя сертификатов для папки выполняется командой **pkica ra folder deassign-authority**.

Параметры (звездочкой * помечены обязательные):

-n --name	* Имя папки.
-a --authority	* Имя (псевдоним) издателя.

Пример:

```
./pkica ra folder deassign-authority --name Пользователи --authority ЦС
```

2.7 Настройка безопасности

Каждое действие, выполняемое оператором, выполняется в какой-то папке. Эта папка либо задается явно, либо определяется по тому, в какой папке находится объект (пользователь, запрос, сертификат), для которого этого действие выполняется. Чтобы выполнить действие оператор должен обладать соответствующим разрешением на соответствующей папке.

Разрешения выдаются не каждому конкретному оператору, а группе операторов. А оператор приобретает разрешения, когда он включен в одну или несколько групп. Если оператор включен в несколько групп, то его разрешения суммируются.

Перечень разрешений:

- Чтение содержимого папки (**Read**);
- Управление пользователями (**ManageUser**);
- Отправка запросов на сертификат (**SubmitCertRequest**);
- Рассмотрение (одобрение/отклонение) запросов на сертификат (**ResolveCertRequest**);
- Отправка запросов на отзыв (**SubmitRevRequest**);
- Рассмотрение (одобрение/отклонение) запросов на отзыв (**ResolveRevRequest**).

2.7.1 Добавление разрешений

Разрешения добавляются командой **pkica ra permission grant**. В качестве параметров указывается папка (либо отдельным флагом можно указать все папки), на которой назначаются разрешения, группа (либо отдельным флагом можно указать все группы), для которой назначаются разрешения, и перечень дополнительных разрешений. Разрешение на чтение добавляется всегда при выполнении этой команды, даже если перечень дополнительных разрешений отсутствует (т.е. нельзя разрешить, к примеру, управление пользователями в папке, не разрешив при этом чтение в этой папке).

Параметры:

--folder	Имя папки.
--all-folders	Назначить разрешения для всех папок.

--group	Название группы.
--all-groups	Назначить разрешения для всех групп.
--manage-user	Управление пользователями.
--submit-cert-request	Отправка запросов на сертификат.
--resolve-cert-request	Рассмотрение (одобрение/отклонение) запросов на сертификат.
--submit-rev-request	Отправка запросов на отзыв.
--resolve-rev-request	Рассмотрение (одобрение/отклонение) запросов на отзыв.
--all	Разрешить всё.

Пример:

```
./pkica ra permission grant --folder Пользователи --group Операторы --manage-user --submit-cert-request --submit-rev-request
```

2.7.2 Получение информации о разрешениях

Для получения информации о назначенных разрешениях для заданной группы на заданную папку используется команда **pkica ra permission get**.

```
./pkica ra permission get --folder Пользователи --group Операторы
```

Параметры:

--folder	Имя папки.
--group	Название группы.

2.7.3 Удаление разрешений

Разрешения удаляются командой **pkica ra permission revoke**. В качестве параметров указываются папка (либо отдельным флагом можно указать все папки), для которой удаляются разрешения, группа (либо отдельным флагом можно указать все группы), для которой удаляются разрешения, и перечень удаляемых разрешений. Если перечень удаляемых разрешений не задан, то будут удалены все разрешения, включая разрешение на чтение.

Параметры:

--folder	Имя папки.
--all-folders	Удалить разрешения для всех папок.
--group	Название группы.
--all-groups	Удалить разрешения для всех групп.
--manage-user	Управление пользователями.

<code>--submit-cert-request</code>	Отправка запросов на сертификат.
<code>--resolve-cert-request</code>	Рассмотрение (одобрение/отклонение) запросов на сертификат.
<code>--submit-rev-request</code>	Отправка запросов на отзыв.
<code>--resolve-rev-request</code>	Рассмотрение (одобрение/отклонение) запросов на отзыв.

Пример:

```
./pkica ra permission revoke --folder Пользователи --group Операторы --
manage-user --submit-cert-request --submit-rev-request
```

2.8 Настройка публикации сертификатов подписи в AIA

Настройка публикации осуществляется в конфигурационных файлах `appsettings.json`, которые находятся в каталогах рядом с исполняемыми модулями `CryptoPro.Ra.Service` и `CryptoPro.CertRegistry.Service`. Если задать параметру `PublishCaCerts` значение `true`, то сертификаты подписи будут публиковаться в папку, указанную в параметре `AiaFolder`. Для публикации сертификатов в несколько разных папок, необходимо запустить соответствующее количество экземпляров сервиса. При этом требуется задать им разные значения `NodeId`.

```
"Aia": {
  "PublishCaCerts": true,
  "AiaFolder": "/var/lib/cpca/aia",
  "NodeId": "ra"
},
```

2.9 Настройка публикации выпущенных CRL в CDP

Настройка публикации осуществляется в конфигурационных файлах `appsettings.json`, которые находятся в каталогах рядом с исполняемыми модулями `CryptoPro.Ra.Service` и `CryptoPro.CertRegistry.Service`. Если задать параметру `PublishCrls` значение `true`, то выпускаемые CRL будут публиковаться в папку, указанную в параметре `CdpFolder`. Для публикации CRL в несколько разных папок, необходимо запустить соответствующее количество экземпляров сервиса. При этом требуется задать им разные значения `NodeId`.

```
"Cdp": {
  "PublishCrls": true,
  "CdpFolder": "/var/lib/cpca/cdp",
  "NodeId": "ra"
},
```

3 Сервер Реестра сертификатов

Сервер Реестра сертификатов выполняет функцию хранилища выпущенных сертификатов и CRL, получая информацию о них от Центр сертификации через NATS Streaming. Так же на сервере Реестра сертификатов могут быть запущены сервисы публикации CRL в CDP (см пункт 2.9), пользовательских сертификатов в LDAP каталог и сертификатов ЦС в AIA (см пункт 2.8).

4 АРМ Оператора ЦР

АРМ Оператора ЦР выполнен в виде веб-приложения, подключение к которому осуществляется с помощью браузера.

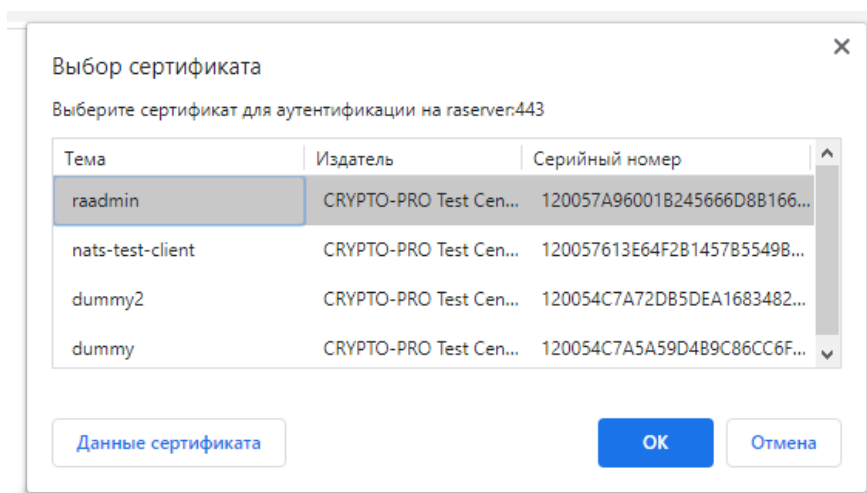
АРМ Оператора ЦР позволяет выполнять регистрацию пользователей, создание запросов на сертификат и выпуск сертификатов, создание запросов на отзыв и отзыв сертификатов пользователей.

Настройка рабочего места Оператора выполняется согласно «ЖТЯИ.00078-03 90 02 ПАК КриптоПро УЦ 2.0. Руководство по установке».

4.1 Подключение Оператора к веб-интерфейсу Центра регистрации

Подключение Оператора к веб-интерфейсу ЦР осуществляется с использованием браузера Chromium Gost или Яндекс Браузер по защищенному TLS-соединению с двусторонней аутентификацией в следующем порядке.

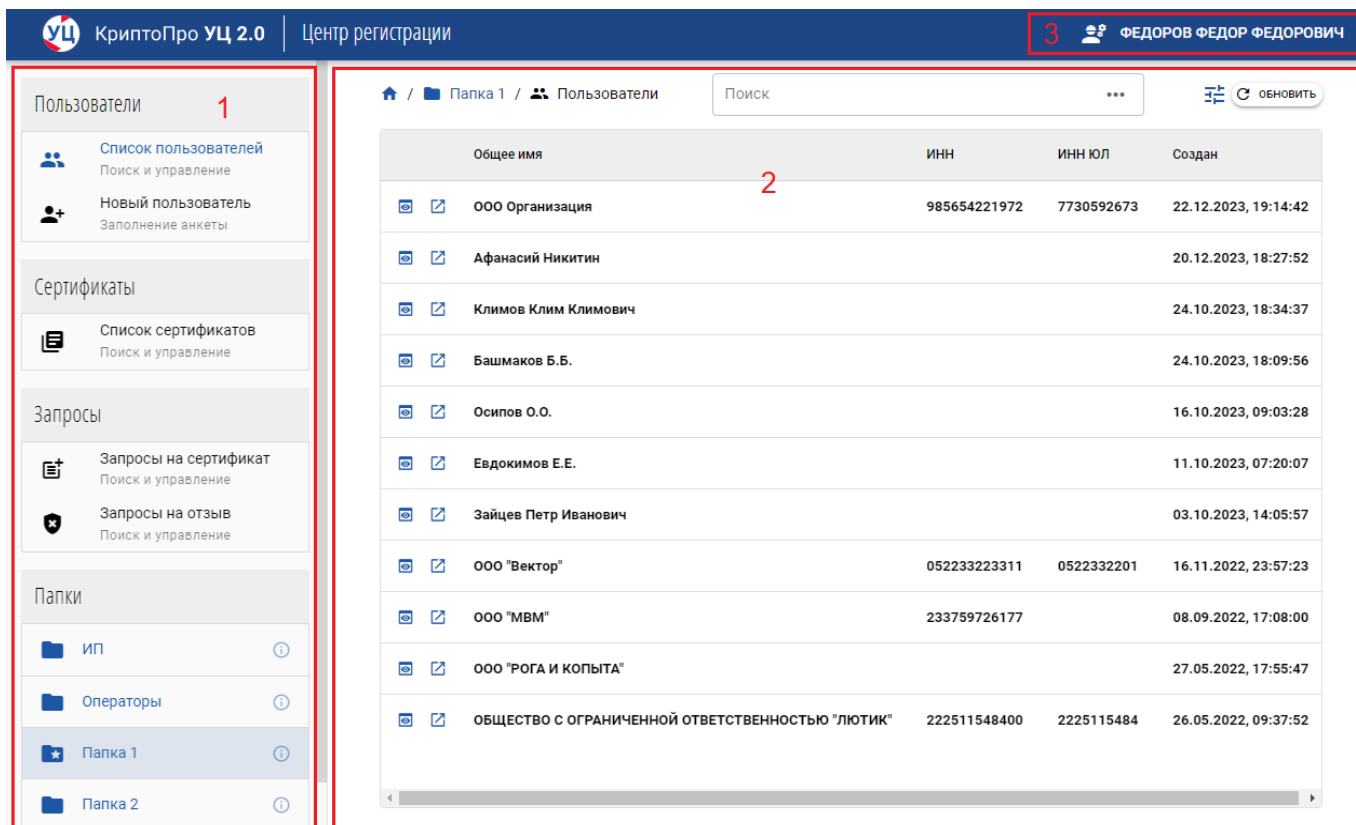
1. В адресной строке браузера необходимо указать адрес сайта, на котором развернут веб-интерфейс ЦР.
2. В окне выбора сертификата необходимо выбрать личный сертификат Оператора и нажать **ОК**.
3. В случае успешной аутентификации будет отображена страница веб-интерфейса ЦР.



4.2 Описание веб-интерфейса

В левой части окна находится навигационное меню (1). Меню содержит следующие основные разделы:

1. **Пользователи** - создание новых пользователей, поиск пользователей.
2. **Сертификаты** - поиск сертификатов.
3. **Запросы** - поиск запросов на сертификаты и запросов на отзыв сертификатов.

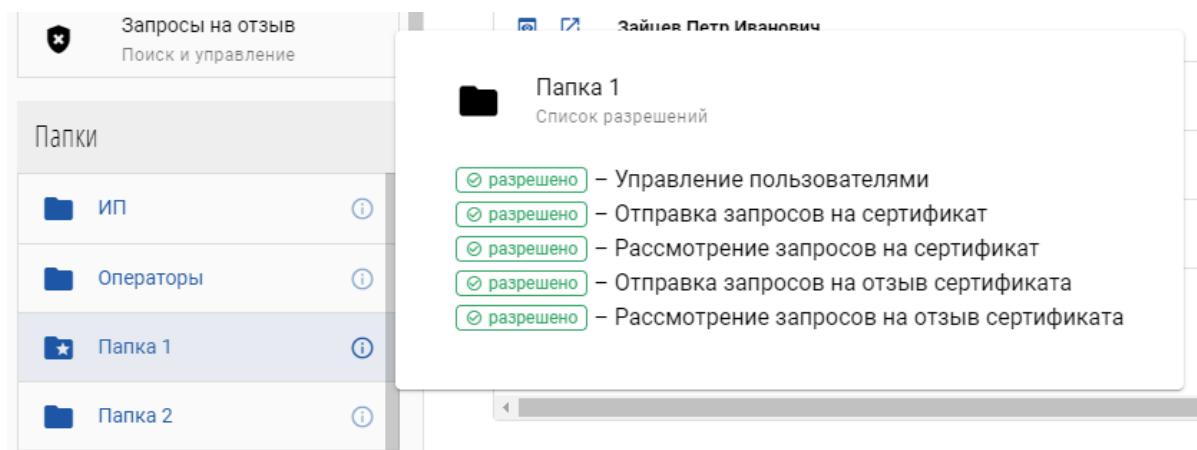


В центральной части окна находится основная рабочая область (2).

В верхнем правом углу окна (3) отображается информация о текущем операторе, от имени которого установлено подключение к ЦР.

4.3 Папки

ЦР поддерживает разделение пользователей по разным папкам. Все пользователи, их сертификаты и запросы находятся в какой-то папке. Выбор текущей папки выполняется в нижней части навигационной области (1):



При наведении мышки на иконку с буквой «i» в строке с папкой, отображается подсказка с описанием разрешенных и запрещенных действий в этой папке. Выбор текущей папки выполняется щелчком мыши по нужной папке.

4.4 Создание (регистрация) пользователя

Для создания (регистрации) нового пользователя выберете пункт **Новый пользователь** в разделе **Пользователи** меню навигации. В рабочей области будет отображена форма для ввода данных о пользователе.

Заполнение профиля нового пользователя

✓ ЗАПОЛНИТЬ...

🗑 ОЧИСТИТЬ

🖨 ПЕЧАТЬ ЗАЯВЛЕНИЯ С ВВЕДЁННЫМИ ДАННЫМИ

Общее имя	Фамилия
Имя и отчество	Страна/регион
Область	Город
Адрес	Организация
Подразделение	Должность или звание
ОГРН	СНИЛС
ИНН	Адрес E-Mail
ОГРНИП	ИНН юл
Список имён DNS	Пол
Дата рождения (ДД.ММ.ГГГГ)	

СОХРАНИТЬ

Пользователь будет создан в папке  Папка 1

Заполните данные о пользователе. Данные о пользователе будут заноситься в сертификаты пользователя в качестве идентификационных данных владельца сертификата в поле Subject (Субъект) сертификата в соответствии с выбранным шаблоном сертификата. После заполнения формы нажмите кнопку **Сохранить**.

По мере заполнения формы пользователя выполняется автоматический поиск дубликатов и проверка вводимых значений на корректность.

В случае обнаружения дубликатов под соответствующим полем будет отображена ссылка **Найдены дубликаты**:

Общее имя
Осипов О.О.

[Найдены дубликаты](#)

При переходе по ссылке **Найдены дубликаты** будет отображен список дублирующих пользователей:

Найденные дубликаты (Общее имя: Осипов О.О.)			
Общее имя	ИНН	ИНН юл	Создан
  Осипов О.О.			16.10.2023, 09:03:28

В случае нарушения правил проверки полей, под соответствующим полем будет показано сообщение об ошибке:

Общее имя

!

Обязательное поле

При наличии запроса на сертификат, можно автоматически заполнить данные пользователя на основе этого запроса. Для этого в форме создания пользователя нажмите кнопку **Заполнить... > Из запроса на сертификат...** и выберете нужный файл запроса:

Заполнение профиля нового пользователя

[Заполнить...](#)
[Очистить](#)
[Печать заявления с введенными данными](#)

[Из запроса на сертификат...](#)
[Скопированными данными](#)

Заполнение профиля пользователя на основе скопированных данных

Поля формы пользователя будут автоматически заполнены из запроса. При необходимости, их можно отредактировать.

Также, можно заполнить данные пользователя данными другого существующего пользователя, для этого в форме существующего пользователя нажмите **Скопировать данные пользователя**:

Пользователь

[РЕДАКТИРОВАТЬ](#)
[ЗАПРОСИТЬ СЕРТИФИКАТ](#)
[ПЕРЕМЕСТИТЬ](#)
[СКОПИРОВАТЬ ДАННЫЕ ПОЛЬЗОВАТЕЛЯ](#)
[ЗАЯВЛЕНИЕ НА РЕГИСТРАЦИЮ](#)

ПРОФИЛЬ ПОЛЬЗОВАТЕЛЯ

Скопировать данные пользователя для дальнейшего использования при создании новых пользователей

ЗАПРОС

А в форме создания пользователя выберете **Заполнить... > Скопированными данными**.

4.5 Пользователи

Для поиска существующего пользователя перейдите в меню **Список пользователей**. Будет отображен список пользователей, в котором можно найти существующего пользователя, воспользовавшись строкой поиска в верхней части основной рабочей области (2).

Для открытия пользователя щелкните на нужной строке в списке пользователей.

Форма пользователя позволяет выполнять изменение учетных данных пользователя, выполнять выпуск сертификата пользователю, просматривать сертификаты и запросы пользователя, перемещать пользователя в другую папку.

The screenshot shows the 'Пользователь' (User) form. At the top, there is a title bar 'Пользователь'. Below it is a navigation bar with five icons and labels: 'РЕДАКТИРОВАТЬ' (Edit), 'ЗАПРОСИТЬ СЕРТИФИКАТ' (Request Certificate), 'ПЕРЕМЕСТИТЬ' (Move), 'СКОПИРОВАТЬ ДАННЫЕ ПОЛЬЗОВАТЕЛЯ' (Copy User Data), and 'ЗАЯВЛЕНИЕ НА РЕГИСТРАЦИЮ' (Registration Statement). Below the navigation bar are four tabs: 'ПРОФИЛЬ ПОЛЬЗОВАТЕЛЯ' (User Profile), 'СЕРТИФИКАТЫ' (Certificates), 'ЗАПРОСЫ НА СЕРТИФИКАТ' (Certificate Requests), and 'ЗАПРОСЫ НА ОТЗЫВ' (Certificate Withdrawals). The 'ПРОФИЛЬ ПОЛЬЗОВАТЕЛЯ' tab is active. It displays two columns of user information. The left column contains 'Общее имя' (Full Name) with the value 'ООО "Вектор"' and 'Имя и отчество' (Name and Patronymic) with the value 'Петров Иван Семенович'. The right column contains 'Фамилия' (Surname) with the value 'Петров' and 'Страна/регион' (Country/Region) with the value 'RU'.

Для редактирования учетных данных пользователя нажмите кнопку **Редактировать**. В открывшемся окне отредактируйте данные и нажмите кнопку **Сохранить**. При редактировании данных пользователя, также, как и при создании пользователя, выполняется автоматический поиск дубликатов, а также доступно заполнение данных из запроса на сертификат и скопированными данными другого пользователя.

Для просмотра сертификатов и запросов пользователя перейдите в соответствующую вкладку в форме пользователя.

4.6 Выпуск сертификата

4.6.1 Создание нового запроса на сертификат

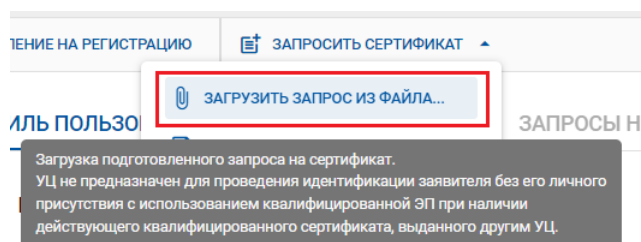
Перейдите в форму пользователя, для которого необходимо выпустить сертификат и нажмите кнопку **Запросить сертификат**. Откроется форма создания запроса на сертификат.

Укажите шаблон, по которому будет производиться формирование сертификата центром сертификации. Шаблон сертификата определяет правила формирования сертификата пользователя. Укажите параметры создания ключа или выберите создание запроса на основе существующего ключа. После заполнения формы нажмите кнопку **Отправить запрос**.

Созданный запрос на сертификат отобразится на вкладке **Запросы на сертификат** формы пользователя. Выпущенный сертификат будет доступен на вкладке **Сертификаты** формы пользователя.

4.6.2 Загрузка запроса на сертификат из файла

Перейдите в форму пользователя, для которого необходимо выпустить сертификат по запросу на сертификат из файла. В выпадающем списке кнопки **Запросить сертификат** выберите пункт **Загрузить запрос из файла**.



Будет предложено выбрать файл запроса и далее данные выбранного файла будут показаны в форме.

Загрузка запроса на сертификат из файла

Файл запроса на сертификат (.req или .p7s в кодировке DER)

ОБЩЕСТВО_С_ОГРАНИЧЕННОЙ_ОТВЕТСТВЕННОСТЬЮ_ЛЮТИК.req

Издатель для выпуска сертификата

root-ca

Субъект запроса

Общее имя

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
"ЛЮТИК"

Город

Москва

ОГРН

1112225000975

ИНН ЮЛ

2225115484

Страна/регион

RU

Адрес

656056, Г БАРНАУЛ, УЛ ПОЛЗУНОВА, Д 26, КВ 316

ИНН

222511548400

Свойства запроса

Шаблон сертификата

Пользователь

Тип идентификации

При личном присутствии

Подпись запроса

ПАК «КриптоПро УЦ 2.0» не предназначен для проведения удостоверяющим центром идентификации заявителя без его личного присутствия с использованием квалифицированной электронной подписи при наличии действующего квалифицированного сертификата, выданного другим удостоверяющим центром.

Сразу одобрить запрос

ОТПРАВИТЬ ЗАПРОС

Проверьте данные запроса и убедитесь в их корректности, убедитесь, что в запросе указан правильный шаблон сертификата и тип идентификации при выпуске сертификата. После чего отправьте запрос на сервер ЦР с помощью кнопки **Отправить запрос**.

ПАК «КриптоПро УЦ 2.0» не предназначен для проведения удостоверяющим центром идентификации заявителя без его личного присутствия с использованием квалифицированной электронной подписи при наличии действующего квалифицированного сертификата, выданного другим удостоверяющим центром.

Созданный запрос на сертификат отобразится на вкладке **Запросы на сертификат** формы пользователя.

4.7 Сертификаты

Для просмотра списка выпущенных сертификатов воспользуйтесь пунктом **Список сертификатов** меню навигации. Список сертификатов поддерживает поиск по значениям атрибутов субъекта, серийному номеру и отпечатку сертификата. Для этого воспользуйтесь строкой поиска в верхней части основной рабочей области (2).

Для просмотра сертификатов, выпущенных для пользователя, перейдите на вкладку **Сертификаты** формы пользователя.

Форма сертификата содержит информацию о сертификате (субъект сертификата и проч.) и позволяет выполнять действия с сертификатом: скачивание и установка сертификата, отзыв сертификата.

Сертификат

25.11.2024, 22:44:59 [ОБНОВИТЬ](#)

[СВЯЗИ](#)
[ПОЛЬЗОВАТЕЛЬ](#)
[СКАЧАТЬ](#)
[УСТАНОВИТЬ](#)
[О СЕРТИФИКАТЕ](#)
[ОТЗЫВАТЬ](#)

Субъект сертификата

Общее имя	Организация
ООО "Вектор"	ООО "Вектор"
Город	Область
Дербент	Республика Дагестан 05
Страна/регион	
RU	

Свойства сертификата

Издатель	Отпечаток
root-ca	88AE75212261A3795D4564810FF59A6FB43DB777
Статус сертификата	Действителен по
Сертификат действителен	18.01.2025, 14:28:15
Серийный номер	Срок действия закрытого ключа
0115CB0E0B080538B548EB1AC0018B428B	18.01.2025, 14:18:14
Действителен с	
18.10.2023, 14:18:15	
Время получения Центром регистрации	
18.10.2023, 14:28:15	

Кнопка **Связи** формы сертификата позволяет найти цепочку **Пользователь – Запрос на сертификат – Сертификат – Запрос на отзыв** для данного сертификата, которая представляет собой удобное средство навигации между объектами.

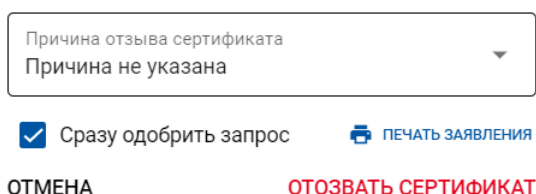
Кнопка **Пользователь** позволяет перейти к пользователю владельцу данного сертификата.

Для скачивания файла сертификата нажмите кнопку **Скачать** в форме сертификата. Файл с сертификатом будет сохранен на диск.

Для установки сертификата в контейнер с ключом ЭП необходимо нажать кнопку **Установить**.

4.8 Отзыв сертификата

Для создания запроса на отзыв сертификата откройте сертификат в форме просмотра сертификата и нажмите кнопку **Отозвать**. В открывшемся окне укажите причину отзыва сертификата. Затем нажмите кнопку **Отозвать сертификат**.



Запрос может быть создан без одновременного одобрения. В этом случае запрос будет создан и будет доступен в списке запросов на отзыв. Позднее его можно одобрить либо отклонить в форме запроса на отзыв.

После одобрения запрос на отзыв сертификата будет отправлен через Центр регистрации на Центр сертификации, после чего сертификат будет помещен в список отозванных сертификатов (CRL) при его формировании.

В списке сертификатов статус сертификата изменится на **Отозван**.

4.9 Запросы на сертификат

Для просмотра списка запросов на сертификат воспользуйтесь пунктом **Запросы на сертификат** меню навигации. Список запросов на сертификат поддерживает поиск по значениям атрибутов субъекта. Для этого воспользуйтесь строкой поиска в верхней части основной рабочей области (2).

Для просмотра запросов на сертификат пользователя, перейдите на вкладку **Запросы на сертификат** формы пользователя.

Форма запроса на сертификат содержит информацию о запросе (субъект запроса и проч.).

Запрос на сертификат

25.11.2024, 22:47:06

ОБНОВИТЬ

СВЯЗИ

ПОЛЬЗОВАТЕЛЬ

СКАЧАТЬ

ЗАЯВЛЕНИЕ

Субъект запроса

Общее имя

000 "Вектор"

Имя и отчество

Петров Иван Семенович

Область

Республика Дагестан 05

Организация

000 "Вектор"

ОГРН

1177746126040

ИНН

052233223311

ИНН юл

0522332201

Фамилия

Петров

Страна/регион

RU

Город

Дербент

Должность или звание

Босс

СНИЛС

15247513970

Адрес E-Mail

vector@xn--d1acufc.xn--p1ai

Свойства запроса

Издатель для выпуска сертификата

root-ca

Статус

Завершён

Шаблон сертификата

Пользователь

Время создания

18.10.2023, 14:28:14

Время рассмотрения

18.10.2023, 14:28:14

Тип идентификации

При личном присутствии

Создал

Федоров Федор Федорович

Рассмотрел

Федоров Федор Федорович

Кнопка **Связи** формы запроса на сертификат позволяет найти цепочку **Пользователь – Запрос на сертификат – Сертификат – Запрос на отзыв** для данного запроса, которая представляет собой удобное средство навигации между объектами.

Кнопка **Пользователь** позволяет перейти к пользователю владельцу данного запроса.

4.10 Запросы на отзыв

Для просмотра списка запросов на отзыв воспользуйтесь пунктом **Запросы на отзыв** меню навигации.

Для просмотра запросов на сертификат пользователя, перейдите на вкладку **Запросы на отзыв** формы пользователя.

Форма запроса на отзыв содержит информацию об отзываемом сертификате и запросе на отзыв.

Запрос на отзыв

25.11.2024, 22:48:04

ОБНОВИТЬ

СВЯЗИ

ПОЛЬЗОВАТЕЛЬ

СЕРТИФИКАТ

СКАЧАТЬ

ЗАЯВЛЕНИЕ

Отзываемый сертификат

Серийный номер

0154BABA5DC1CAF4A94F20B968018B4287

Отпечаток

B5865D7B62DFA11810C4FB4496568DEC71B0F224

Общее имя

ООО "Вектор"

Организация

ООО "Вектор"

Город

Дербент

Область

Республика Дагестан 05

Страна/регион

RU

Свойства запроса

Издатель отзываемого сертификата

root-ca

Причина отзыва

Причина не указана

Статус

Завершён

Создал

Федоров Федор Федорович

Время создания

18.10.2023, 14:24:41

Рассмотрел

Федоров Федор Федорович

Время рассмотрения

18.10.2023, 14:24:41

Кнопка **Связи** формы запроса на отзыв позволяет найти цепочку **Пользователь – Запрос на сертификат – Сертификат – Запрос на отзыв** для данного запроса, которая представляет собой удобное средство навигации между объектами.

Кнопка **Пользователь** позволяет перейти к пользователю владельцу данного запроса.

5 Импорт данных из КриптоПро УЦ 2.0 ЖТЯИ.00078-01

5.1 Импорт данных Центра сертификации

5.1.1 Подготовка к импорту

Для импорта данных ЦС необходимо иметь в наличии:

- работающую на MS SQL Server БД ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01. К MS SQL Server должны быть разрешены подключения с ЦС, на который выполняется импорт.
- ключи ЦС

Перед выполнением импорта ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01 должен быть остановлен (остановлены Служба сертификатов КриптоПро УЦ 2.0, Служба управления ключами КриптоПро УЦ 2.0, служба IIS).

5.1.2 Разворачивание БД ЦС

Импорт данных ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01 возможен только в существующую БД ЦС, поэтому предварительно ее надо развернуть, используя команду **pkica ca db new**. Перед выполнением команды следует остановить все сервисы УЦ.

5.1.3 Импорт данных ЦС

Импорт выполняется в автоматическом режиме, импортируются экземпляры ЦС, ключи подписи, шаблоны сертификатов, запросы на сертификат, сертификаты и CRL. Если на ЦС есть сертификаты в статусе «Приостановлен», то импорт выполнен не будет. Необходимо либо возобновить такие сертификаты, либо отозвать и после этого выполнить импорт повторно. Сертификаты, для которых задана дата отзыва в будущем, будут импортированы как действующие, а информация об отзыве в будущем будет потеряна, так как эта процедура теперь не поддерживается. Запросы на сертификат, по которым еще не было принято окончательное решение (выпущен сертификат либо отклонен/ошибка) импортированы не будут, поэтому рекомендуется перед импортом их обработать. Импорт может быть выполнен только один раз, после успешного заполнения БД ЦС последующие выполнения будут вызывать ошибку.

Импорт выполняется командой **pkica ca import-legacy**.

Параметры (обязательные параметры помечены звездочкой *):

-c --connection-string	* Строка подключения к базе данных ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01.
-t --timeout	* Таймаут SQL команды. Задается в формате ISO 8601: [d.]hh:mm[:ss]. Значение по умолчанию: 00:01 (1 минута).

Пример:

```
./pkica ca import-legacy --connection-string "Data Source=.;Initial
Catalog=CertificateService.Database;Username=sa;Password=*****;Pooling=True"
```

При выполнении импорта на Windows в качестве строки подключения допустимо использовать строку из настроек ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01:

```
./pkica ca import-legacy --connection-string "Data Source=.;Initial
Catalog=CertificateService.Database;Integrated Security=true;Pooling=True"
```

5.1.4 Перенос ключей подписи

После импорта данных ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01 вся информация о ключах подписи, включая ссылки на контейнеры закрытых ключей, так же будет импортирована без изменений. Поэтому, если после импорта планируется использовать ключи с новыми именами контейнеров или на других носителях, эти ссылки необходимо изменить в соответствии с п. 1.2.8. После выполнения этих операций все доступные ключи подписи нужно загрузить, чтобы убедиться, что ссылки настроены корректно.

5.1.5 Настройки ЦС

Настройки ЦС, включая расписание выпуска CRL, автоматически не импортируются. Поэтому после выполнения предыдущих шагов их надо перенести вручную, согласно пп. 1.2.6 и 1.4.2.

5.1.6 Установка экземпляра ЦС по умолчанию

Если экземпляр по умолчанию не был установлен перед импортом и его не удалось установить автоматически в процессе импорта из-за того, что существует несколько экземпляров ЦС, то его нужно установить после выполнения импорта в соответствии с разделом 1.2.4.

5.1.7 Выпуск CRL

ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01 не помещал в CRL сертификаты, срок действия которых истек, поэтому после импорта ключей подписи, сертификатов и CRL нужно выпустить новый CRL содержащий все сертификаты, которые когда-либо были отозваны. Для этого нужно задать значение **Unoptimized** для настройки **CrIssueMode** каждого перенесенного ЦС.

```
...  
"CrIssueMode": "Unoptimized"  
...
```

После этого нужно дождаться выпуска очередного CRL по расписанию либо выпустить внеочередной CRL вручную командой **pkica ca crl issue**, как описано в п. 1.4.1.

Включенная настройка **Unoptimized** в **CrIssueMode** указывает службе ЦС на необходимость полного перебора статусов всех выпущенных сертификатов, который занимает достаточно продолжительное время. Поэтому после выпуска CRL эту настройку нужно вернуть к значению по умолчанию, т.е. установить в **None** или вообще удалить из списка настроек ЦС.

5.2 Импорт данных Центра регистрации

5.2.1 Подготовка к импорту

Для импорта данных сервера ЦР необходимо иметь в наличии работающую на MS SQL Server БД ЦР КриптоПро УЦ 2.0 ЖТЯИ.00078-01. К MS SQL Server должны быть разрешены подключения с ЦР, на который выполняется импорт.

Перед выполнением импорта ЦР КриптоПро УЦ 2.0 ЖТЯИ.00078-01 должен быть остановлен (остановлены Служба регистрации КриптоПро УЦ 2.0, служба IIS).

5.2.2 Разворачивание БД ЦР

Импорт данных ЦР КриптоПро УЦ ЖТЯИ.00078-01 возможен только в существующую БД ЦР, поэтому предварительно ее надо развернуть, используя команду **pkica ra db new**. Перед выполнением команды следует остановить все сервисы УЦ.

5.2.3 Создание операторов ЦР

Для импорта данных необходимо, чтобы на ЦР, на который осуществляется импорт, предварительно были зарегистрированы один или несколько операторов в соответствии с разделом 2.1.1. В дальнейшем эти операторы станут владельцами импортированных пользователей, запросов и сертификатов.

5.2.4 Импорт данных ЦР

Импорт выполняется в автоматическом режиме, импортируются структура папок (древовидная разворачивается в одноуровневую плоскую), пользователи, сертификаты, запросы на сертификат и на отзыв сертификата.

Условия для успешного выполнения импорта:

- отсутствие сертификатов в статусе «Приостановлен». Перед выполнением импорта такие сертификаты следует возобновить или отозвать.
- отсутствие запросов на сертификат или отзыв, которые ожидают обработки на ЦС, (статус «Принят» или «Обрабатывается»). Перед выполнением импорта следует дождаться, когда обработка завершится.

Сертификаты и запросы, содержащие некорректные данные (например, некорректные атрибуты имени) будут пропущены. Так же при импорте можно пропустить отозванные или истекшие сертификаты и пользователей, для которых не импортировано ни одного сертификата и ни одного необработанного запроса на сертификат (в состоянии "Новый").

В ЦР КриптоПро УЦ ЖТЯИ.00078-01 концепция безопасности отличалась, поэтому нет возможности автоматически импортировать группы безопасности и назначить соответствующие разрешения на перенесенные папки. После выполнения импорта нужно дать необходимые вручную в соответствии с разделом 2.7.1.

Импорт может быть выполнен только один раз, после успешного заполнения БД ЦР последующие выполнения будут вызывать ошибку.

Импорт выполняется командой **pkica ra import-legacy**.

Параметры (обязательные параметры помечены звездочкой *):

-c --connection-string	* Строка подключения к базе данных ЦР КриптоПро УЦ 2.0 ЖТЯИ.00078-01 (в строке подключения обязательно должен быть указан параметр MultipleActiveResultSets = True).
-l --login	* Имя входа (логин) оператора. Пользователи, запросы и сертификаты будут импортированы от имени этого оператора.
--folder-id	Папка, из которой будет выполняться импорт данных. Если не указана, то будут импортированы данные из всех папок.
--include-subfolders	Импортировать все подпапки.

--skip-empty-users	Пропустить пользователей, для которых не импортировано ни одного сертификата и ни одного необработанного запроса на сертификат.
--skip-revoked-certs	Пропустить отозванные сертификаты.
--skip-expired-certs	Пропустить истекшие сертификаты.
-t --timeout	* Таймаут SQL команды. Задается в формате ISO 8601: [d.]hh:mm[:ss]. Значение по умолчанию: 00:01 (1 минута).

Пример:

```
./pkica ra import-legacy --connection-string "Data Source=.;Initial
Catalog=RegistrationService.Database;Username=sa;Password=*****;MultipleActi
veResultSets=True" --login ivanov --folder-id 1aae020d-c84f-406b-becc-
a8c000c648eb
```

При выполнении импорта на Windows в качестве строки подключения допустимо использовать строку из настроек ЦР КриптоПро УЦ 2.0 ЖТЯИ.00078-01:

```
./pkica ra import-legacy --connection-string "Data Source=.;Initial
Catalog=RegistrationService.Database;Integrated
Security=True;Pooling=True;MultipleActiveResultSets=True" --operator-id
01852fb9-5568-444b-b323-9b70e58c217a --folder-id 1aae020d-c84f-406b-becc-
a8c000c648eb
```

5.2.5 Настройка издателя по умолчанию

Если на ЦР не был настроен издатель по умолчанию и в процессе импорта так же не удалось его настроить, то эту настройку нужно задать после окончания импорта в соответствии с п. 2.5.4.

5.3 Импорт данных Реестра сертификатов

В КриптоПро УЦ 2.0 ЖТЯИ.00078-01 не было компонента Реестр сертификатов, поэтому в качестве источника данных для импорта так же используется ЦС КриптоПро УЦ.

5.3.1 Подготовка к импорту

Для импорта данных Реестр сертификатов необходимо иметь в наличии работающую на MS SQL Server БД ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01. К MS SQL Server должны быть разрешены подключения с сервера Реестра сертификатов, на который выполняется импорт.

Перед выполнением импорта ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01 должен быть остановлен (остановлены Служба сертификатов КриптоПро УЦ 2.0, Служба управления ключами КриптоПро УЦ 2.0, служба IIS).

5.3.2 Разворачивание БД Реестра сертификатов

Импорт данных ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01 возможен только в существующую БД Реестра сертификатов, поэтому предварительно ее надо развернуть, используя команду **pkica cert-registry db new**.

5.3.3 Импорт данных

Импорт выполняется в автоматическом режиме, импортируются сертификаты и списки отзыва. Если на ЦС есть сертификаты в статусе «Приостановлен», то импорт выполнен не будет. Необходимо либо возобновить такие сертификаты, либо отозвать и после этого выполнить импорт повторно. Сертификаты, для которых задана дата отзыва в будущем, будут импортированы как действующие. Импорт может быть выполнен только один раз, после успешного заполнения БД Реестра сертификатов последующие выполнения будут вызывать ошибку.

Импорт выполняется командой **pkica cert-registry import-legacy**.

Параметры (обязательные параметры помечены звездочкой *):

-c --connection-string	* Строка подключения к базе данных ЦС КриптоПро УЦ 2.0 ЖТЯИ.00078-01.
-t --timeout	* Таймаут SQL команды. Задается в формате ISO 8601: [d.]hh:mm[:ss]. Значение по умолчанию: 00:01 (1 минута).

Пример:

```
./pkica cert-registry import-legacy --connection-string "Data
Source=.;Initial Catalog=CertificateService.Database;Integrated
Security=True;Pooling=True"
```

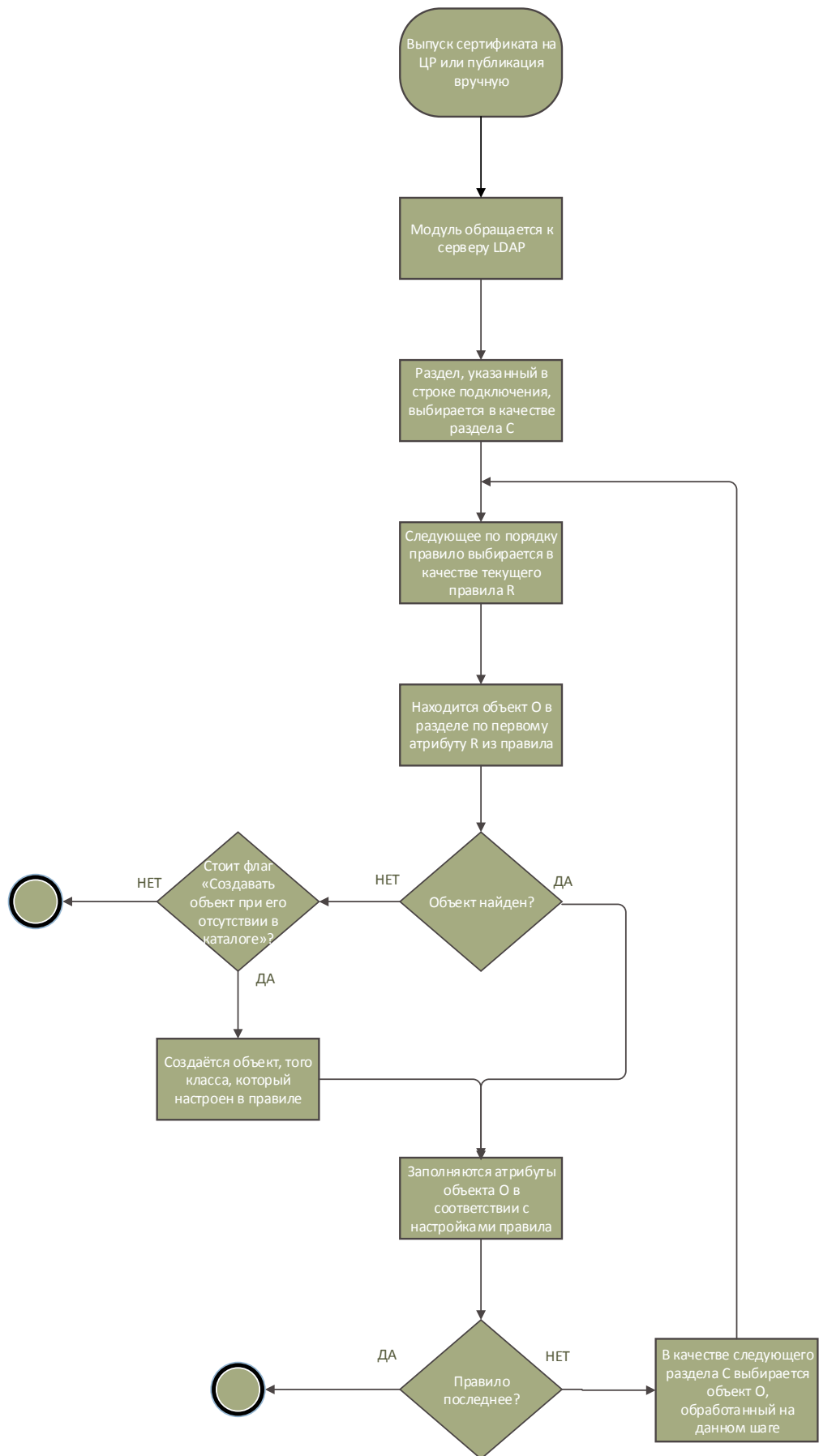
6 Публикация сертификатов в LDAP каталог

Публикация выпущенных сертификатов в LDAP каталог может быть настроена на компонентах УЦ Сервер Центра регистрации и Сервер Реестра сертификатов.

Модуль публикации сертификатов в LDAP Каталог имеет расширенные настройки для экспорта сертификатов как в Каталоги Microsoft (AD, AD LDS), так и других производителей, например, Oracle Directory. При настройке публикации сертификатов вначале нужно убедиться, что LDAP Каталог корректно работает с квалифицированными сертификатами, т.е. поддерживает поля субъекта ИНН, СНИЛС, ОГРН и т.д.

Модуль публикует сертификат в Каталог в соответствии со своими настройками по следующему алгоритму:

1. Модуль обращается к серверу, на котором расположен Каталог от имени пользователя, имеющего право записи в требуемый раздел Каталога (параметры сервера LDAP и администратора указываются в настройках подключения).
2. В соответствии с параметрами, указанными в строке подключения, выбирается раздел Каталога в качестве раздела **С**. Если в строке подключения не задан раздел для публикации, для публикации используется контекст именования по умолчанию (default naming context).
3. В качестве текущего правила **R** выбирается одно из правил по порядку, начиная с первого (см. 6.1.5).
4. По первому атрибуту из правила **R** находится объект **О** (атрибуты правил задаются в настройках подключения и их порядок может быть изменён путём перемещения в списке). Остальные атрибуты не используются для поиска.
5. Если объект **О** не найден, при наличии в настройках подключения флага «Создавать объект при его отсутствии в каталоге», создаётся объект того класса, который настроен в правиле. При отсутствии такого флага модуль прекращает обработку данного сертификата.
6. Если объект **О** найден или создан, его атрибуты заполняются значениями в соответствии с настройками остальных атрибутов в правиле.
7. Если правило последнее, то модуль прекращает обработку сертификата, иначе найденный объект **О** используется в качестве следующего раздела **С** и выполнение переходит к п.3.



Для того, чтобы подключить модуль, необходимо в файл конфигурации appsettings.json сервиса Центра регистрации или Реестра сертификатов добавить раздел **PublishLdap**.

6.1 Структура настроек модуля публикации

6.1.1 Включение публикации

Включение и выключение публикации определяется значением параметра **PublishCerts**. Если параметр **true**, то публикация включена, если **false** - то выключена.

6.1.2 Подключение к LDAP каталогу

Подключение к LDAP каталогу определяется следующими параметрами:

- **LdapServer** - строка подключения к серверу, на котором размещен LDAP каталог. Должна иметь вид: LDAP://имя_сервера.домен:порт/путь к разделу.
- **LdapUser** - имя пользователя каталога, под которым выполняется подключение. Этот пользователь должен обладать необходимыми правами в зависимости от настроек публикации (создание объектов, изменение атрибутов)
- **LdapPassword** - пароль пользователя, под которым выполняется подключение к LDAP каталогу.
- **UseAuthentication** - параметр, определяющий использовать ли аутентификацию при подключении. Если **true**, то будет использована аутентификация по логину/паролю, если **false**, то будет выполнено анонимное подключение без аутентификации.

6.1.3 Фильтрация сертификатов

Набор правил фильтрации определяет, какие сертификаты будут опубликованы. Если набор пуст, то будут опубликованы все сертификаты. Если же заданы какие-то правила, то будут опубликованы сертификаты, удовлетворяющие этим правилам.

Набор правил фильтрации задается разделом **FilterRules**. Если раздел отсутствует, то считается что правила не заданы.

Каждое правило задается набором, состоящим из следующих параметров:

- **Name** - понятное название правила
- **Enabled** - параметр, определяющий, включено ли правило. Если значение равно **true**, то правило включено, если **false** - то выключено.
- **Conditions** - набор условий правила. Правило считается выполненным, только если выполнены все его условия. Каждое условие задается следующим набором параметров:
 - **Comparison** - тип сравнения, используемый при проверке условия. Может принимать следующие значения:
 - **Equal** - равно
 - **NotEqual** - не равно

- **Contains** - содержит
- **NotContains** - не содержит
- **ContainedIn** - содержится
- **NotContainedIn** - не содержится
- **Source** - источник, откуда будет получено значение для сравнения с заданным. Может принимать следующие значения:
 - **Dn** - субъект сертификата
 - **San** - расширение сертификата «Дополнительное имя субъекта» (2.5.29.17)
 - **Eku** - расширение сертификата «Улучшенный ключ» (2.5.29.37)
 - **Ext** - набор, состоящий из объектных идентификаторов (OID) расширений сертификата
 - **Template** - объектный идентификатор (OID) шаблона сертификата из расширения «Шаблон сертификата» (1.3.6.1.4.1.311.21.7)
- **ComponentOid** - объектный идентификатор (OID), указывающий, какой компонент из источника (**Source**) нужно брать для проверки правила. Применяется, если в качестве источника указан **Dn** или **San**. Для **San** значение **ComponentOid** интерпретируется следующим образом:
 - **1.2.840.113549.1.9.1** - адрес email, будет взято поле rfc822Name расширения «Дополнительное имя субъекта»
 - **1.2.643.2.2.44.5** - список DNS имен, будет взято поле dNSName расширения «Дополнительное имя субъекта»
 - **1.3.6.1.4.1.311.20.2.3** - имя участника, будет взято значение userPrincipalName (UPN) из поле otherName расширения «Дополнительное имя субъекта»
 - **все остальные значения** - будет взят соответствующий компонент имени из поля directoryName расширения «Дополнительное имя субъекта»
- **Value** - целевое значение, для которого будет производится проверка правила

Так же можно задать логическое условие, применяемое к вычислению результата всех правил набора. Это условие задается параметром **RuleOperator** раздела **PublishLdap**.

Параметр может принимать следующие значения:

- **And** - логическое И, для публикации сертификата требуется выполнение всех правил в наборе. Это значение по умолчанию.
- **Or** - логическое ИЛИ, для публикации сертификата требуется выполнение хотя бы одного правила в наборе.

6.1.4 Правила преобразования субъекта сертификата

При публикации сертификатов в каталог встречаются ситуации, когда требуется заменять значения атрибутов имени субъекта сертификата. Сделать это можно с помощью набора правил замены, которые будут применяться после проверки правил фильтрации.

Набор задается разделом настроек **RegexRules**. Каждое значение из набора содержит следующие параметры:

- **FindPattern** - строка для поиска
- **ReplacePattern** - строка для замены

Замены осуществляются в строке субъекта, содержащей короткие имена компонент (CN, O, C и т.д.), а не объектные идентификаторы. Правила из набора применяются последовательно.

Пример: "CN=Иванов Иван,O="ООО ""Вектор""",C=RU". Для замены значения компонента имени Организация (O) нужно указать следующую настройку: **FindPattern**: O="ООО ""Вектор""", **ReplacePattern**: O="ООО ""Роткев""".

6.1.5 Правила поиска и заполнения объектов LDAP каталога

Настройки модуля в разделе **LdapMappings** содержат список правил, которые последовательно используются для поиска/создания и заполнения объектов в соответствии с описанным выше алгоритмом.

Каждая настройка содержит следующие параметры:

- **ObjectClass** - имя класса объекта LDAP каталога.
- **ObjectNotExistAction** - действие при отсутствии объекта в LDAP каталоге. Допустимы следующие значения:
 - **Create** - создать объект. Это значение по умолчанию
 - **Skip** - не выполнять публикацию сертификата.
 - **ThrowError** - прервать публикацию с ошибкой. Сертификат не будет опубликован, но сообщение останется необработанным в очереди NATS Streaming и через некоторое время будет предпринята повторная попытка обработки.
- **Attributes** - коллекция настроек атрибутов объекта LDAP. Каждая настройка содержит следующие параметры:
 - **Attribute** - имя атрибута
 - **Type** - тип источника значения. Может быть одним из следующих:
 - **Constant** - строковая константа
 - **Rdn** - компонент имени субъекта
 - **San** - значение из расширения «Дополнительное имя субъекта» (2.5.29.17)
 - **Certificate** - сертификат в бинарном виде

- **Value** - значение атрибута. Если в качестве **Type** задано Rdn, то в качестве значения нужно задать объектный идентификатор (OID) компонента имени, чье значение нужно использовать. Если в качестве **Type** задано San, то в качестве компонента имени так же нужно задать объектный идентификатор (OID), при этом он будет интерпретирован следующим образом:
 - **1.2.840.113549.1.9.1** - адрес email, будет взято поле rfc822Name расширения «Дополнительное имя субъекта»
 - **1.2.643.2.2.44.5** - список DNS имен, будет взято поле dNSName расширения «Дополнительное имя субъекта»
 - **1.3.6.1.4.1.311.20.2.3** - имя участника, будет взято значение userPrincipalName (UPN) из поле otherName расширения «Дополнительное имя субъекта»
 - **все остальные значения** - будет взят соответствующий компонент имени из поля directoryName расширения «Дополнительное имя субъекта»

Если в качестве **Type** задано Certificate, то значение задавать не нужно.

- **ReplaceValue** - заменять значение атрибута. Если этот параметр равен **true**, то при публикации нового значения все прежние значения, опубликованные ранее, будут удалены и заменены новым. Если параметр равен **false**, то новое значение будет добавлено к тем, что уже были опубликованы. При этом, если атрибут не многозначный, возникнет ошибка публикации.

Поиск объекта производится по значению, указанному в первом по порядку атрибуте. Если объект найден или создан, то его атрибуты заполняются в соответствии с остальными элементами списка атрибутов. Если в экспортируемом сертификате имеется несколько повторяющихся полей (например, несколько полей OrganizationalUnit), то поиск по этому полю будет выполнен подряд столько раз, сколько значений поля содержится в сертификате. При записи значения в атрибут, если в сертификате есть несколько значений для указанного атрибута, они все будут записаны. Если значений несколько, но атрибут не многозначный, вернется ошибка.

6.2 Пример настроек модуля публикации

В примере:

- задается подключение к серверу LDAP dc.test.local с публикацией в раздел DC=test,DC=local
- настраивается фильтрация = публикуемые сертификаты должны содержать в субъекте C=RU **И** расширение с шаблоном 1.2.643.2.2.46.0.8
- в субъекте, используемом при публикации, значение компонента O "test" будет заменено на "production".
- правила публикации:
 - в корневом разделе ищется объект с атрибутом cn=Users. Если такого объекта нет, то публикация завершается с ошибкой.

- у найденного объекта с `cn=Users` ищется дочерний объект с атрибутом `cn=<CN субъекта сертификата>`. Если такой объект не найден, то он создается с классом `inetOrgPerson`. Далее у объекта `cn=<CN субъекта сертификата>` в атрибут `userCertificate` записывается публикуемый сертификат.

```
{
...
"PublishLdap": {
  "PublishCerts": false,
  "LdapServer": "LDAP://dc.test.local/DC=test,DC=local",
  "LdapUser": "CN=LdapPublisher",
  "LdapPassword": "12345",
  "UseAuthentication": true,
  "LdapMappings": [
    {
      "ObjectClass": "container",
      "ObjectNotExistAction": "ThrowError",
      "Attributes": [
        {
          "Attribute": "cn",
          "Type": "Constant",
          "Value": "Users",
          "ReplaceValue": false
        }
      ]
    },
    {
      "ObjectClass": "inetOrgPerson",
      "ObjectNotExistAction": "Create",
      "Attributes": [
        {
          "Attribute": "cn",
          "Type": "Rdn",
          "Value": "2.5.4.3",
          "ReplaceValue": false
        },
        {
          "Attribute": "objectClass",
          "Type": "Constant",
          "Value": "inetOrgPerson",
          "ReplaceValue": false
        },
        {
          "Attribute": "userCertificate",
          "Type": "Certificate",
          "ReplaceValue": true
        }
      ]
    }
  ]
},
"RegexRules": [
  {
    "FindPattern": "O=test",
    "ReplacePattern": "O=production"
  }
],
"FilterRules": [
```

```
{
  "Name": "CN Rule",
  "Enabled": true,
  "Conditions": [
    {
      "Comparison": "Equal",
      "Source": "Dn",
      "ComponentOid": "2.5.4.6",
      "Value": "RU"
    }
  ]
},
{
  "Name": "Template Rule",
  "Enabled": true,
  "Conditions": [
    {
      "Comparison": "Equal",
      "Source": "Template",
      "Value": "1.2.643.2.2.46.0.8"
    }
  ]
}
],
"RuleOperator": "And"
},
...
}
```

7 Журналирование

Сервисы ЦС и ЦР ведут два вида журналов:

- журнал аудита – позволяет вести учёт всех событий, происходящих в системе;
- лог-файлы – содержат техническую информацию о событиях в системе, которая будет полезна для диагностики тех или иных проблем.

Параметры ведения журналов настраиваются в конфигурации сервисов ЦС и ЦР. Приведем основные настройки:

- `MinimumLevel` – задаёт минимальный уровень логирования, означающий, насколько подробная информация должна попасть в журнал. В журнал будет выводиться информация, соответствующая выбранному уровню и всем более низким уровням. Допустимые значения: `Verbose`, `Debug`, `Information`, `Warning`, `Error`, `Fatal`.
- `Filter` – позволяет настроить дополнительный фильтр для регулирования того, какие записи должны попадать в журнал.
- `WriteTo` – представляет собой массив, каждый из элементов которого задаёт метод вывода информации. Поддерживается вывод в файл и консоль. Разные способы вывода могут содержать дополнительные индивидуальные настройки.
- `outputTemplate` – формат записи журнала, позволяет настроить какие именно данные должны попадать в записи журнала.
- `path` – путь к файлу журнала. Если файл не существует, он будет автоматически создан.
- `rollingInterval` – период ротации файлов. По истечении указанного периода, создаётся новый файл журнала.
- `retainedFileCountLimit` – количество хранимых файлов журнала. При достижении этого количества файлов, старые файлы будут автоматически удалены.
- `flushToDiskInterval` – периодичность записи данных на диск.

Отметим, что для записи файлов журналов на диск у сервисов должны быть права на запись в соответствующую папку.

Файлы журналов создаются при наличии свободного места на диске. В связи с этим, во избежание переполнения диска, необходимо предусмотреть своевременный перенос файлов журналов на другие носители. Или настроить автоматическую очистку старых журналов, если отсутствует необходимость их хранения.