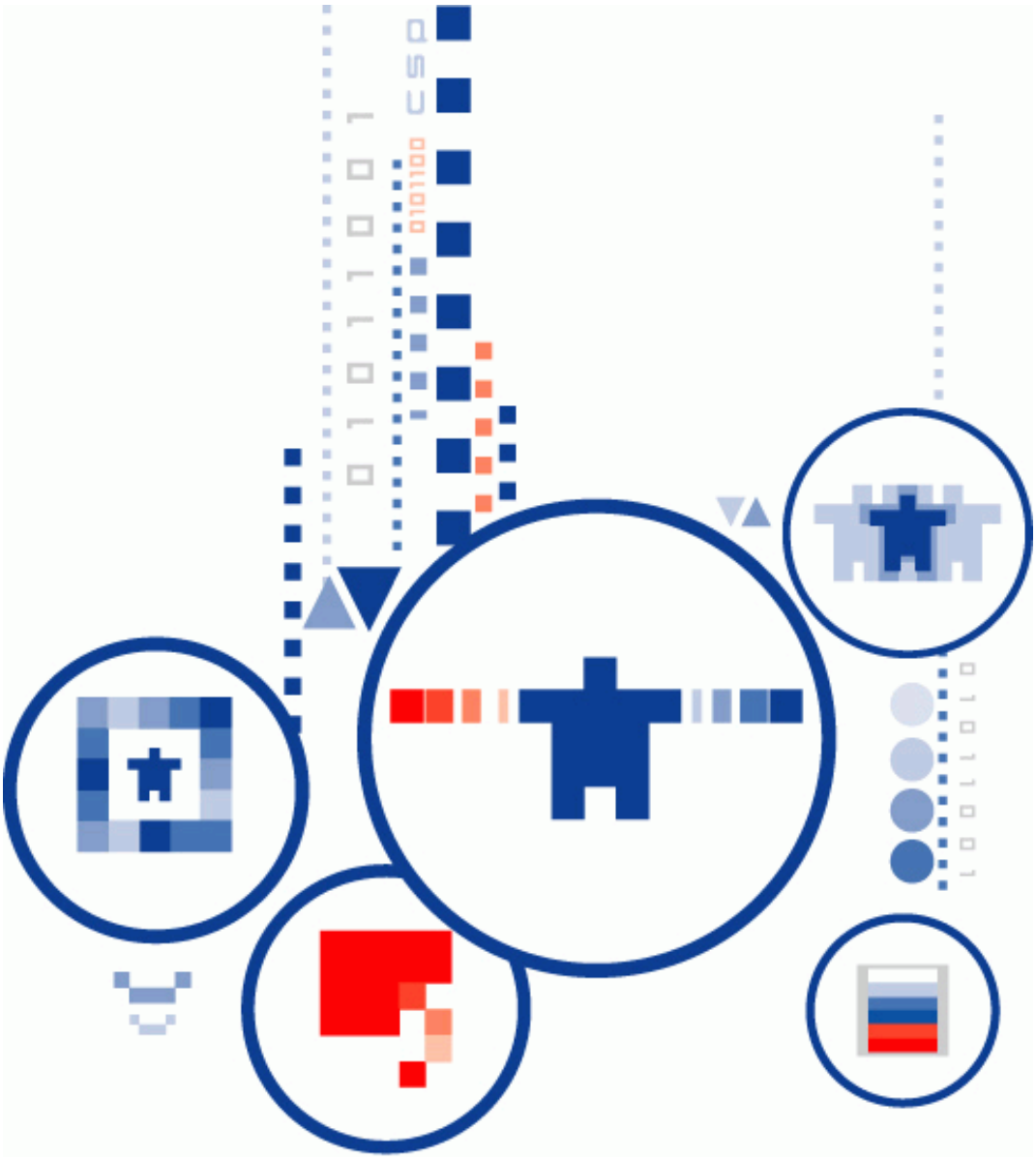

Сервер Электронной Подписи

«КриптоПро DSS»

Компонент ПАКМ «КриптоПро HSM»

DSS Client SDK. Руководство разработчика.

iOS

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ 2
1. Общие сведения о DSS Client SDK 5
2. Интеграция DSS Client SDK в программный проект 7
3. Сценарии использования 8
3.1. Анонимная регистрация пользователя 8
3.2. Регистрация устройства с использованием начального вектора аутентификации 9
3.3. Добавление (привязка) еще одного устройства к учетной записи пользователя 10
4. Использование DSS Client SDK. Классы и функции 11
4.1. Настройка TLS-соединения 11
4.2. Класс CryptoProDSS 11
4.2.1. Метод init 11
4.3. Класс Auth 12
4.3.1. Метод scanQR 12
4.3.2. Метод _init 12
4.3.3. Метод kinit 12
4.3.4. Метод addNewDevice 13
4.3.5. Метод confirm 13
4.3.6. Метод verify 14
4.3.7. Метод setPassAuth 14
4.3.8. Метод changePassAuth 14
4.3.9. Метод renameAuth 15
4.3.10. Метод removeAuth 15
4.3.11. Метод getAuthList 15
4.3.12. Метод confirmNewDevice 15
4.3.13. Метод checkStatus 16
4.4. Класс Cert 16
4.4.1. Метод getCert 16
4.4.2. Метод setCert 16
4.4.3. Метод getCertList 17
4.4.4. Метод setNameCert 17
4.4.5. Метод suspendCert 17
4.4.6. Метод resumeCert 18
4.4.7. Метод revokeCert 18
4.4.8. Метод setDefaultCert 19
4.4.9. Метод deleteCert 19
4.5. Класс Policy 19
4.5.1. Метод getOperations 19
4.5.2. Метод getHistoryOperations 20
4.5.3. Метод getParamDSS 20

4.5.4. Метод setPersonalisation 20
4.5.5. Метод getUserDevices 20
4.5.6. Метод getCaParams 21
4.5.7. Метод initBioRng 21
4.6. Класс Docs 21
4.6.1. Метод uploadDocument 21
4.6.2. Метод downloadDocument 22
4.7. Класс Sign 22
4.7.1. Метод signMT 22
4.7.2. Метод signMO 22
4.7.3. Метод deferredRequest 23
4.7.4. Метод deferredRequest 23
5. Типы данных 25
5.1. Тип DSSUser 25
5.2. Тип RegisterInfo 25
5.3. Тип ProtectionType 26
5.4. Тип Certificate 26
5.5. Тип state 27
5.6. Тип type (тип загруженного объекта) 27
5.7. Тип OperationsInfo 27
5.8. Тип OperationDescription 27
5.9. Тип Document 28
5.10. Тип Operation 28
5.11. Тип OperationHistory 29
5.12. Тип PolicyPayload 29
5.13. Тип keyProtectionFlags 29
5.14. Тип SignServerPolicy 30
5.15. Тип CAPolicy 30
5.16. Тип ProcessingTemplateInfo 31
5.17. Тип Devices 31
5.18. Тип Deviceinfo 31
5.19. Тип UploadFile 32
5.20. Тип UploadDocInfo 32
5.21. Тип parameters 32
5.22. Тип DocumentSelectedMode 33
5.23. Тип ConfirmationSendingMode 33
5.24. Тип ApproveRequestMT 33
5.25. Тип SignatureResult 33
5.26. Тип OperationResultInfo 33
5.27. Тип approveRequestMO 34
5.28. Тип ApprovedOperation 34
5.29. Тип ConfirmedDocument 34
5.30. Тип DeclinedDocument 34
5.31. Тип Error 35
Приложение 1. Файл стилизации графического интерфейса DSS Client SDK 36
Приложение 2. Значения по умолчанию файла стилизации графического интерфейса DSS Client SDK 39

Приложение 3. Сообщения об ошибках 42

1. Общие сведения о DSS Client SDK

DSS Client SDK для встраивания в мобильное приложение представляет собой набор программных компонентов (framework) для использования в мобильных приложениях с функциями клиента КриптоПро DSS.

DSS Client SDK предоставляет **программный (API) и графический интерфейсы**, позволяющие выполнять следующие действия.

- Управление сертификатами пользователя:
  - Создание запроса на сертификат;
  - Установка сертификата;
  - Просмотр списка сертификатов и запросов;
  - Удаление сертификатов и/или запросов;
  - Отзыв сертификата;
  - Назначения сертификата по умолчанию;
  - Назначения дружественного имени сертификата.
- Отправка документов на подпись.
- Подтверждение или отклонение подписи документов:
  - В том числе подтверждение или отклонение действий пользователя в ИС.
- Управление учетной записью:
  - Регистрация новой учётной записи с привязкой мобильного устройства;
  - Привязка устройства к существующей учетной записи;
  - Просмотр списка устройств пользователя;
  - Удаление устройств пользователя;
  - Смена ПИН-кода для доступа к ключу аутентификации на устройстве пользователя;
  - Назначение дружественного имени ключа аутентификации.
- Просмотр истории операций пользователя.

DSS Client SDK предоставляет **графический интерфейс (окна)**, позволяющий выполнять следующие действия.

- Ввод нового пароля для защиты устанавливаемых векторов аутентификации.

  Ввод пароля с повторным вводом для подтверждения. Окно отображается если на стороне сервера разрешен только пароль для защиты векторов.
- Ввод TouchID/FaceID для защиты устанавливаемых векторов аутентификации.

  Ввод пароля и TouchID/FaceID. Окно отображается, если на сервере разрешена биометрия для защиты векторов аутентификации.

  В окне пользователю предлагается придумать ПИН-код для защиты вектора, после чего защитить его с помощью биометрии.
- Ввод пароля для доступа к векторам аутентификации.

  Окно отображается, если на сервере разрешен только пароль для защиты векторов.
- Ввод TouchID/FaceID для доступа к векторам аутентификации.

Ввод пароля или TouchID/FaceID: окно отображается, если на сервере разрешена биометрия для защиты векторов аутентификации.

- Подпись: сопровождающий текст и список документов.
- Просмотр документа.
- Просмотр «сырого» представления документа.

2. Интеграция DSS Client SDK в программный проект

Интеграция DSS Client SDK в программный проект состоит из следующих этапов.

1. Открыть проект в XCode.
2. В панели слева выбрать проект, затем в основном окне в списке **targets** выбрать цель сборки, перейти на вкладку **General**, выбрать там **Embedded Binaries**, нажать на плюс, в появившемся окне выбрать **Add other**, добавить **SDKFramework.framework**.

   При добавлении следует установить флажок **Copy items if needed** и **Create groups**.
3. Открыть **SDKFramework.framework** через **Finder**.
   a. Перетащить файл **CpMyDssRootCerts.json** папки **SDKFramework.framework** в директорию **Resources** проекта. Если папка **Resources** отсутствует в проекте, то необходимо его создать. При переносе файлов следует установить флажок **Create Folder referencies for any added folders**.
   b. Файлы из **en.lproj** и **ru.lproj** перетащить в ресурсы приложения. В меню выбрать **Create Folder Groups**. При этом они будут автоматически сгруппированы в двуязычные файлы локализации.
4. Если используется XCode, в левой панели выбрать проект, затем в основном окне в списке **targets** выбрать цель сборки, перейти на вкладку **Build Phases**, выбрать там **Add Build Phase — Add Run Script**. Появившуюся фазу сборки **Run Script** отредактировать, поместив в поле для скрипта «**<Путь к директории с фреймворком - CPROCSP.Framework>/ConfigureApplication**». Если путь содержит пробелы, перед ними надо поставить обратную косую черту (backslash): "\".
5. В свойствах проекта и в свойствах **target** должны быть отключены опции "**Dead Code Striping**", "**Strip during copy**", "**Strip linked products**", "**Enable bitcode**".
6. В свойствах проекта в разделе **Preprocessor Macros** прописать свойство **UNIX**.
7. В файле **info.plist** проекта указать:
   - **Privacy — Face ID Usage Description**
   - **Privacy - Photo Library Usage Description**
   - **Privacy - Camera Usage Description**

3. Сценарии использования

3.1. Анонимная регистрация пользователя

Процедура анонимной регистрации пользователя на сервисе включает в себя следующие шаги:

1. Метод **_init** (см. раздел 4.3.2). Установка вектора аутентификации на устройство пользователя.
2. Метод **confirm** (см. раздел 4.3.5). Подтверждение установки вектора аутентификации на устройстве пользователя.
3. Метод **scanQR** (см. раздел 4.3.1). Считывание **qr verification** и его загрузка в оперативную память
4. Метод **verify** (см. раздел 4.3.6). Отображение пользователю его профиля и используя загруженный qr, метод генерирует ответ с подтверждением привязки устройства к учетной записи пользователя на сервисе.

Важно: до выполнения этого шага должен быть создан профиль пользователя.

Пример:

```
do {
let auth = try Auth()

    // шаг 1.
    auth._init(view: <view>, view: <view>, dssUser: <DSSUser>,
    registerInfo: <RegisterInfo>, keyProtectionType:
    <keyProtectionType>, password: <password>) { error in
    // проверка наличия ошибки (если error равен nil, то функция
    завершилась успешно, иначе - продолжение сценария невозможно)

        // шаг 2.
        auth.confirm(view: <view>, kid: <kid>) { error in
        // проверка наличия ошибки (если error равен nil, то функция
        завершилась успешно, иначе - продолжение сценария невозможно)

            // шаг 3.
            auth.scanQR(view: <view>, base64QR: <base64 json from qr>) {
            type, error in
            // проверка наличия ошибки (если error равен nil, то функция
            завершилась успешно, иначе - продолжение сценария
            невозможно)
            // Ожидается, что 'type' будет равен строке 'Verification'

                // шаг 4.
                auth.verify(view: <view>, kid: <kid>, silent: <silent
                 mode>) { error in
                // проверка наличия ошибки (если error равен nil, то
                функция завершилась успешно, иначе - продолжение
                сценария невозможен)
                 }
            }
        }
    }
} catch {
    // обработка ошибок
}
```

3.2. Регистрация устройства с использованием начального вектора аутентификации

Процедура регистрации устройства на сервисе с использованием начального вектора аутентификации включает в себя следующие шаги:

1. Метод **scanQR** (см. раздел 4.3.1). Считывание **qr kinit** и его загрузка в оперативную память
2. Метод **kinit** (см. раздел 4.3.3). Установка ключей на устройстве с использованием считанного qr для отправки аутентифицированного запроса
3. Метод **confirm** (см. раздел 4.3.5). Подтверждение установки ключей на устройстве
4. Метод **verify** (см. раздел 4.3.6). Отображает пользователю его профиль и отправляет ответ с подтверждением привязки устройства к учетной записи пользователя на сервисе

Пример:

```
do {
let auth = try Auth()

    // шаг 1.
    auth.scanQR(view: <view>, base64QR: <base64 json from qr>) { type,
    error in
    // проверка наличия ошибки (если error равен nil, то функция
    завершилась успешно, иначе - продолжение сценария невозможен)
    // Ожидается, что 'type' будет равен строке 'Kinit'

        // шаг 2.
        auth.kinit(view: <view>, dssUser: <DSSUser>, registerInfo:
        <RegisterInfo>, keyProtectionType: <keyProtectionType>,
        activationCode: <activationCode>, password: <password>) { error
        in
        // проверка наличия ошибки (если error равен nil, то функция
        завершилась успешно, иначе - продолжение сценария невозможен)

            // шаг 3.
            auth.confirm(view: <view>, kid: <kid>) { error in
            // проверка наличия ошибки (если error равен nil, то функция
            завершилась успешно, иначе - продолжение сценария
            невозможен)

                // шаг 4.
                auth.verify(view: <view>, kid: <kid>, silent: <silent>)
                { error in
                // проверка наличия ошибки (если error равен nil, то
                функция завершилась успешно, иначе - продолжение
                сценария невозможен)
                }
            }
        }
    }
} catch {
    // обработка ошибок
}
```

3.3. Добавление (привязка) еще одного устройства к учетной записи пользователя

Процедура добавления (привязки) другого (еще одного) устройства к учетной записи пользователя на сервисе включает в себя следующие шаги:

1. Метод **`addNewDevice`** (см. раздел 4.3.4). Установка ключей на другом устройстве пользователя. Выполняется с устройства 2.
2. Метод **`confirmNewDevice`** (см. раздел 4.3.12). Одобрение / отклонение привязки ключей к учетной записи пользователя. Выполняется с устройства 1.
3. Метод **`checkStatus`** (см. раздел 4.3.13). Проверка статуса привязки ключей к учетной записи пользователя. Выполняется с устройства 2.

4. Использование DSS Client SDK. Классы и функции

DSS Client SDK предоставляет следующие основные классы для работы:

- **Класс CryptoProDSS**: Содержит метод инициализации DSS Client SDK (см. раздел 4.2).
- **Класс Auth**: Управляет устройствами пользователей (см. раздел 4.3).
- **Класс Cert**: Управляет сертификатами пользователей (см. раздел 4.4).
- **Класс Policy**: Содержит методы получения настроек КриптоПро DSS (см. раздел 4.5).
- **Класс Docs**: Управляет передачей документов в КриптоПро DSS при подписи (см. раздел 4.6).
- **Класс Sign**: Получает информацию об операциях, подтверждает операции и подписывает документы (см. раздел 4.7).

4.1. Настройка TLS-соединения

Взаимодействие DSS Client SDK с сервером КриптоПро DSS осуществляется по протоколу TLS с использование только алгоритмов ГОСТ. Для обеспечения доверия клиента к серверу КриптоПро DSS в ресурсы приложения должен быть добавлен корневой сертификат Веб-сервера DSS. Корневой сертификат добавляется через файл CpMyDssRootCerts.json.

Пример содержания файла CpMyDssRootCerts.json:

```
{
    "version":1,
    "root": ["MIIFxzCCBXSgAwIBAgIRAdSExQ … 15ktc8p00v+A9Erolsd5Ig=="],
    "intermediate": []
}
```

В элементе root передаётся один или несколько корневых сертификатов Веб-сервера в кодировке Base64.

Элемент intermediate (опциональный) может содержать список сертификатов подчинённых УЦ.

4.2. Класс CryptoProDSS

4.2.1. Метод init

Инициализирует DSS Client SDK.

```
int _init(view: UIViewController, completion: @escaping (_ result:
CSPInitCode) -> Void)
```

Возвращаемые значения:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **result** — **init_ok** — инициализация прошла успешно.
  init_certs_not_installed — не удалось установить сертификаты в процессе инициализации.
  init_lockScreen_not_installed — экран блокировки не установлен.

init_device_rooted — устройство работает с правами суперпользователя.

4.3. Класс Auth

4.3.1. Метод scanQR

Загружает данные, переданные в виде QR-кода.

```
scanQR(view: UIViewController, base64QR: String? = nil, completion:
@escaping (_ type: String?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **base64QR** — QR-код, закодированный в base64.
- **type** — тип загруженного объекта (см. раздел 5.6).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.2. Метод _init

Создает неподтвержденное мобильное устройство (без привязки) в КриптоПро DSS с получением вектора аутентификации к нему.

```
_init(view: UIViewController, dssUser: DSSUser, registerInfo:
RegisterInfo, keyProtectionType: ProtectionType, password: String? =
nil, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **dssUser** — сведения о пользователе (см. раздел 5.1).
- **registerInfo** — сведения о регистрируемом устройстве пользователя (см. раздел 5.2).
- **keyProtectionType** — способ защиты вектора аутентификации (см. раздел 5.3).
- **password** — ПИН-код для доступа к вектору аутентификации.
- **errorregInfo** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.3. Метод kinit

Создает неподтвержденное мобильное устройство (без привязки) в КриптоПро DSS с получением начального вектора аутентификации к нему с использованием QR-кода.

```
kinit(view: UIViewController, dssUser: DSSUser, registerInfo:
RegisterInfo, keyProtectionType: ProtectionType, activationCode:
String?, password: String? = nil, completion: @escaping (_ error:
Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **dssUser** — сведения о пользователе (см. раздел 5.1).
- **registerInfo** — сведения о регистрируемом устройстве пользователя (см. раздел 5.2).
- **keyProtectionType** — способ защиты вектора аутентификации (см. раздел 5.3).
- **activationCode** — код активации.
- **password** — ПИН-код для доступа к вектору аутентификации.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.4. Метод addNewDevice

Инициализирует новое устройство пользователя.

```
addNewDevice(view: UIViewController, dssUser: DSSUser, registerInfo:
RegisterInfo, keyProtectionType: ProtectionType, uid: String, password:
String? = nil, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **dssUser** — сведения о пользователе (см. раздел 5.1).
- **registerInfo** — сведения о регистрируемом устройстве пользователя (см. раздел 5.2).
- **keyProtectionType** — способ защиты вектора аутентификации (см. раздел 5.3).
- **uid** — идентификатор пользователя КриптоПро DSS, к которому будет привязано мобильное устройство.
- **password** — ПИН-код для доступа к вектору аутентификации.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.5. Метод confirm

Подтверждает установку векторов аутентификации. Данный метод всегда необходимо вызывать после выполнения регистрации нового неподтвержденного мобильного устройства в КриптоПро DSS. Данный метод может быть вызван только для векторов аутентификации, находящихся в состоянии **Created**.

```
confirm(view: UIViewController, kid: String, completion: @escaping (_
error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.6. Метод verify

Подтверждает привязку мобильного устройства к учетной записи пользователя. Данный метод может быть вызван только для векторов аутентификации, находящихся в состоянии **NotVerified**.

Если для данного мобильного устройства требуется осуществлять подтверждение присоединения с использованием **nonce**, то в этом случае перед отправкой запроса мобильное приложение должно отсканировать QR-код, содержащий значение **nonce**.

```
verify(view: UIViewController, kid: String, silent: Bool = false,
completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **silent** — флаг для скрытия/отображения диалоговых окон SDK. Используется только для создания усиленной неквалифицированной электронной подписи.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.7. Метод setPassAuth

Ввод ПИН-кода на вектор аутентификации.

```
setPassAuth(view: UIViewController, kid: String, password: String? =
nil, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **password** — ПИН-код для защиты вектора аутентификации. Параметр используется только для создания усиленной неквалифицированной электронной подписи.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.8. Метод changePassAuth

Изменение ПИН-кода на вектор аутентификации.

```
changePassAuth(view: UIViewController, kid: String, keyProtectionType:
ProtectionType, oldPassword: String? = nil, newPassword: String? = nil,
completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **keyProtectionType** — способ защиты вектора аутентификации (см. раздел 5.3).
- **oldPassword** — существующий ПИН-код (параметр используется только в **silent**-режиме и для создания усиленной неквалифицированной электронной подписи).

- **newPassword** — новый ПИН-код (параметр используется только в **silent**-режиме и для создания усиленной неквалифицированной электронной подписи).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.9. Метод renameAuth

Переименование устройства пользователя.

```
renameAuth(kid: String, newName: String)
```

Параметры:

- **kid** — идентификатор устройства пользователя.
- **newName** — новое отображаемое имя вектора аутентификации.

4.3.10. Метод removeAuth

Удаление устройства пользователя и его вектора аутентификации.

```
removeAuth(view: UIViewController, kid: String, deletedKid: String,
forceDelete: Bool, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **deletedKid** — идентификатор удаляемого устройства пользователя.
- **forceDelete** — удаление устройства вне зависимости ответа сервера.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.11. Метод getAuthList

Получение сведений о зарегистрированных пользователях и их устройствах.

```
getAuthList() throws -> [DSSUser]
```

Список полей типа **DSSUser** приведен в разделе 5.1.

4.3.12. Метод confirmNewDevice

Подтверждение запроса на добавление нового устройства.

```
confirmNewDevice(view: UIViewController, kid: String, confirmedKid:
String, silent: Bool = false, completion: @escaping (_ error: Error?) ->
Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **confirmedKid** — идентификатор устройства пользователя, добавление которого требуется подтвердить.

- **silent** — флаг для скрытия/отображения диалоговых окон SDK. Используется только для создания усиленной неквалифицированной электронной подписи.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.3.13. Метод checkStatus

Проверка статуса запроса на добавление нового устройства.

```
checkStatus(view: UIViewController, kid: String, completion: @escaping
(_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4. Класс Cert

4.4.1. Метод getCert

Создание запроса на сертификат ключа проверки электронной подписи.

```
getCert(view: UIViewController, kid: String, caId: Int, tid: String, dn:
[String: String], completion: @escaping (_ cert: Certificate?, _ error:
Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **caId** — идентификатор обработчика УЦ.
- **tid** — идентификатор шаблона сертификата.
- **dn** — различительное имя субъекта в формате {"OID компонента имени", "Значение компонента имени"}.
- **cert** — сведения о созданном сертификате / запросе на сертификат при отсутствии ошибок (см. раздел 5.4).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4.2. Метод setCert

Установка сертификата ключа проверки электронной подписи.

```
setCert(view: UIViewController, kid: String, crt: String, completion:
@escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.

- **crt** —сертификат, закодированный в base64.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4.3. Метод getCertList

Получение списка запросов на сертификаты и списка сертификатов ключей проверки электронной подписи.

```
getCertList(view: UIViewController, kid: String, completion: @escaping
(_ certs: [Certificate], _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **certs** — список сертификатов и запросов на сертификат, возвращаемые при отсутствии ошибок (см. раздел 5.4).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4.4. Метод setNameCert

Установка отображаемого имени сертификата ключа проверки электронной подписи.

```
setNameCert(view: UIViewController, kid: String, cid: String,
friendlyName: String, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **frendlyName** — отображаемое имя сертификата.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4.5. Метод suspendCert

Приостановление действия сертификата ключа проверки электронной подписи.

```
suspendCert(view: UIViewController, kid: String, cid: String, holdDate:
Int64, unholdDate: Int64, completion: @escaping (_ error: Error?) ->
Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.

- **holdDate** — дата приостановления сертификата. 0 — если требуется немедленное приостановление сертификата, конкретная дата — если требуется отложенное приостановление сертификата.
- **unholdDate** — дата возобновления действия сертификата.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4.6. Метод resumeCert

Возобновление действия сертификата ключа проверки электронной подписи.

```
resumeCert(view: UIViewController, kid: String, cid: String, completion:
@escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4.7. Метод revokeCert

Отзыв сертификата ключа проверки электронной подписи.

```
revokeCert(view: UIViewController, kid: String, cid: String, reason:
Int, date: Int32, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **Reason** — причина отзыва:
  0 (CRL_REASON_UNSPECIFIED) — Причина неизвестна.
  1 (CRL_REASON_KEY_COMPROMISE) — Компрометация ключей.
  2 (CRL_REASON_CA_COMPROMISE) — Компрометация Центра Сертификации.
  3 (CRL_REASON_AFFILIATION_CHANGED) — Имя пользователя или другая информация в сертификате изменена, но нет причины полагать, что секретный ключ скомпрометирован.
  4 (CRL_REASON_SUPERSEDED) — Сертификат заменен другим, но нет причины полагать, что секретный ключ скомпрометирован.
  5 (CRL_REASON_CESSATION_OF_OPERATION) — Сертификат более не нужен для целей, которых он выдавался, но нет причины полагать, что секретный ключ скомпрометирован.
- **date** — дата отзыва сертификата. 0 — если требуется немедленный отзыв сертификата, конкретная дата — если требуется отложенный отзыв сертификата.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4.8. Метод setDefaultCert

Метод установки сертификата ключа проверки электронной подписи сертификатом по умолчанию.

```
setDefaultCert(view: UIViewController, kid: String, cid: String,
_default: Bool, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **_default** — флаг использования сертификата как сертификата по умолчанию.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.4.9. Метод deleteCert

Метод удаления сертификата или запроса на сертификат ключа проверки электронной подписи.

```
deleteCert(view: UIViewController, kid: String, cid: String? = nil, rid:
String? = nil, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **cid** — идентификатор сертификата.
- **rid** — идентификатор запроса на сертификат.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.5. Класс Policy

4.5.1. Метод getOperations

Получение списка операций пользователя данного устройства.

```
getOperations(view: UIViewController, kid: String, type: String?, opid:
String?, completion: @escaping (_ operationsInfo: OperationsInfo?, _
error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **type** — тип операции. Типы операций описаны в документе «ЖТЯИ.00096-02 91 02 КриптоПро DSS. Руководство Администратора».
- **opid** — идентификатор операции.
- **operationsInfo** — список операций пользователя данного устройства (см. раздел 5.7).

- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.5.2. Метод getHistoryOperations

Получение записей аудита для определенного пользователя и устройства

```
getHistoryOperations(view: UIViewController, kid: String, count: Int? = nil, bookmark: Int64? = nil, operationCodes: [Int]? = nil, completion: @escaping (_ operationHistory: OperationHistory?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **count** — выводимое количество записей.
- **bookmark** — идентификатор записи, относительно которой осуществляется поиск.
- **operationCodes** — разделенный запятой список кодов событий, которые должны быть включены в выборку (см. «ЖТЯИ.00096-02 91 02 КриптоПро DSS. Руководство Администратора»).
- **operationHistory** — список записей аудита для определенного пользователя и устройства (см. раздел 5.10).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.5.3. Метод getParamDSS

Запрос параметров сервера КриптоПро DSS.

```
getParamsDSS(serviceUrl: String, completion: @escaping (_ policy: PolicyPayload?, _ error: Error?) -> Void)
```

Параметры:

- **serviceURL** — адрес сервера КриптоПро DSS.
- **policy** — параметры сервера КриптоПро DSS (см. раздел 5.12).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.5.4. Метод setPersonalisation

Персонализация элементов интерфейса SDK.

```
setPersonalisation(url: URL) throws
```

Параметры:

- **url** — путь к файлу персонализации (см. Приложения 1–2).

4.5.5. Метод getUserDevices

Получение сведений об устройствах пользователя.

```
getUserDevices( view: UIViewController, kid: String, completion: @escaping (_ devices: Devices?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **devices** — сведения об устройствах пользователя (см. раздел 5.17).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.5.6. Метод getCaParams

Запрос с сервера DSS параметров подписи: список профилей подписи, параметры Удостоверяющих Центров и т.п.

```
getCaParams(view: UIViewController, kid: String, completion: @escaping (_ policy: SignServerPolicy?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **policy** — параметры подписи, полученные от КриптоПро DSS (см. раздел5.14).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.5.7. Метод initBioRng

Открывает окно ДСЧ.

```
initBioRng(provType: Int) throws
```

Параметры:

- **provType** — тип криптопровайдера. Равен 80.

4.6. Класс Docs

4.6.1. Метод uploadDocument

Загрузка документа в КриптоПро DSS.

```
uploadDocument(view: UIViewController, kid: String, doc: UploadFile, completion: @escaping (_ uploadData: UploadDocInfo?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **doc** — информация о документе (см. раздел 5.19).
- **uploadData** — информация о загруженном документе (см. раздел 5.20).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.6.2. Метод downloadDocument

Выгрузка документа из КриптоПро DSS.

```
downloadDocument(view: UIViewController, kid: String, docId: String,
completion: @escaping (_ data: Data?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **docId** — идентификатор документа.
- **data** — дата и время выгрузки документа.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.7. Класс Sign

4.7.1. Метод signMT

Подтверждение операции, созданной на сервере КриптоПро DSS.

```
signMT(view: UIViewController, kid: String, operation: Operation?,
documentSelectedMode: DocumentSelectedMode, confirmationSendingMode:
ConfirmationSendingMode, silent: Bool = false, completion: @escaping (_
approveRequest: ApproveRequestMT?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **operation** — сведения об операции (см. раздел 5.10).
- **documentSelectedMode** — режим выбора документов (по умолчанию «только все» — пользователь может подтвердить/отклонить операцию целиком или может выбрать отдельные документы из списка для подтверждения подписи) (см. раздел 5.22).
- **confirmationSendingMode** — режим отправки подтверждения (по умолчанию «немедленно» - сформированный запрос с подтверждением SDK сразу отправляет на сервер или приложение сохраняет данный запрос для возможности отправить его позднее) (см. раздел 5.23).
- **approveRequest** — запрос на подтверждение/отклонение операции (см. раздел 5.24).
- **silent** — флаг для скрытия/отображения диалоговых окон SDK. Используется только для создания усиленной неквалифицированной электронной подписи.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.7.2. Метод signMO

Подтверждение операции, созданной на клиенте (в мобильном приложении).

```
signMO(view: UIViewController, kid: String, parameters: [String:
String], enableMultiSelection: Bool, inmediateSendConfirm: Bool,
```

```
uploadDocInfo: [UploadDocInfo], silent: Bool = false, completion:
@escaping (_ signatureResult: SignatureResult?, _ approveRequest:
ApproveRequestMO?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **parameters** — сведения об операции (см. раздел 5.21).
- **enableMultiSelection** — режим выбора документов (по умолчанию «только все» — пользователь может подтвердить/отклонить операцию целиком или может выбрать отдельные документы из списка для подтверждения подписи).
- **inmediateSendConfirm** — режим отправки подтверждения (по умолчанию «немедленно» - сформированный запрос с подтверждением SDK сразу отправляет на сервер или приложение сохраняет данный запрос для возможности отправить его позднее).
- **uploadDocInfo** — список документов, переданных на подпись (см. раздел 5.20).
- **signatureResult** — Результат подписи пакета документов (см. раздел 5.25).
- **approveRequest** — запрос на подтверждение/отклонение операции (см. раздел 5.27).
- **silent** — флаг для скрытия/отображения диалоговых окон SDK. Используется только для создания усиленной неквалифицированной электронной подписи.
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.7.3. Метод deferredRequest

Отложенное подтверждение операции, созданной на сервере КриптоПро DSS.

```
deferredRequest(view: UIViewController, kid: String, approveRequest:
ApproveRequestMT, completion: @escaping (_ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.
- **kid** — идентификатор устройства пользователя.
- **approveRequest** — запрос на подтверждение/отклонение операции (см. раздел 5.24).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

4.7.4. Метод deferredRequest

Отложенное подтверждение операции, созданной на клиенте (в мобильном приложении).

```
deferredRequest(view: UIViewController, kid: String, approveRequest:
ApproveRequestMO,        completion: @escaping (_ signatureResult:
SignatureResult?, _ error: Error?) -> Void)
```

Параметры:

- **view** — ссылка на **UIViewController**, поверх которого будут выводиться окна DSS Client SDK.

- **kid** — идентификатор устройства пользователя.
- **approveRequest** — запрос на подтверждение/отклонение операции (см. раздел 5.27).
- **error** — описание ошибки, если таковая возникла. Перечень ошибок приведен в Приложении 3.

5. Типы данных

5.1. Тип DSSUser

Поле	Тип	Описание
kid	String	Идентификатор устройства пользователя.
uid	String	Идентификатор пользователя DSS.
alias	String	Человекочитаемый идентификатор мобильного устройства пользователя.
state	String	Состояние начального вектора аутентификации (см. раздел 5.5).
profile	String	Профиль пользователя в DSS.
notBefore	Int64	Дата начала действия вектора аутентификации. Время в формате Unix Time.
notAfter	Int64	Дата окончания действия вектора аутентификации. Время в формате Unix Time.
serviceUrl	String	URL для взаимодействия с сервером DSS.
name	String	Дружественное устройства пользователя.
keyType	Bool	Тип вектора аутентификации. Для создания усиленной квалифицированной электронной подписи используется значение **`True`**.

5.2. Тип RegisterInfo

Поле	Тип	Описание
pushAddress	String	PUSH-адрес устройства пользователя.
appVersion	String	Версия приложения, в которое встроен DSS Client SDK (опционально).
userName	String	Логин пользователя (опционально)
phone	String	Номер телефона пользователя (опционально)
email	String	Адрес электронной почты пользователя (опционально)
token	String	Токен аутентификации интегрированной

Поле	Тип	Описание
		информационной системы (опционально)

5.3. Тип ProtectionType

Тип **`ProtectionType`** содержит способы защиты вектора аутентификации и может принимать следующие значения:

- PASSWORD — ПИН-код;
- NO_PROTECTION — без защиты;
- BIOMETRIC — биометрические данные (отпечаток пальца, лицо и т.д.).

5.4. Тип Certificate

Поле	Тип	Описание
type	String	Тип объекта: "crt" - сертификат, "req" - запрос на сертификат.
cid	String	Идентификатор сертификата.
rid	String	Идентификатор запроса на сертификат.
content	String	Содержимое сертификата.
caId	Int	Идентификатор обработчика УЦ.
dn	[String: String]	Различительное имя субъекта в формате {"OID компонента имени", "Значение компонента имени"}.
notBefore	Int32	Дата начала действия сертификата.
notAfter	Int32	Дата окончания действия сертификата.
state	String	Статус сертификата. Допустимые значения: **`Active`**, **`Not_valid`**, **`Revoked`**, **`Out_of_order`**.
friendlyName	String	Отображаемое имя сертификата.
isDefault	Bool	Флаг, определяющий, является ли данный сертификат сертификатом по умолчанию

5.5. Тип state

Поле	Описание
Created	Вектора аутентификации пользователя созданы.
Installed	Вектора аутентификации пользователя установлены на устройство пользователя.
NotVerified	Требуется подтверждение учетной записи пользователя.
Active	Устройство пользователя привязано и готово к использованию.
ApproveRequired	Требуется подтверждение привязки нового устройства пользователя.

5.6. Тип type (тип загруженного объекта)

Тип **`type`** указывает на информацию внутри QR-кода.

Поле	Описание
kinit	Начальный вектор аутентификации
verification	Сведения для подтверждения привязки устройства пользователя к учетной записи.
transaction	Сведения для подтверждения операции.
newdev	Сведения о новом добавляемом устройстве пользователя.

5.7. Тип OperationsInfo

Тип представляет собой список типа **`OperationInfo`** (см. раздел 5.10).

5.8. Тип OperationDescription

Поле	Тип	Описание
type	String	Тип операции (см. документ «ЖТЯИ.00096-02 91 02 КриптоПро DSS. Руководство Администратора»).
caption	String	Краткое описание операции.
description	String	Описание операции.

5.9. Тип Document

Поле	Тип	Описание
id	String	Идентификатор документа.
documentInfo	String	Сопровождающий текст о документе.
documentHash	String	Хэш-значение от документа.
snippet	String	Краткая информация о документе в формате html.
snippetHash	String	Хэш-значение от краткой информации о документе.
fileSize	Int64	Размер документа в мегабайтах.
pageCount	Int	Количество страниц документа.
isPrintableViewAvailable	Bool	Флаг, определяющий, доступность в мобильном приложении печатной формы документа.
isSnippetViewAvailable	Bool	Флаг, определяющий доступность в мобильном приложении краткой информации о документе.
isRawViewAvailable	Bool	Флаг, определяющий доступность в мобильном приложении исходного документа.

5.10. Тип Operation

Поле	Тип	Описание
description	OperationDescription	Описание операции (см. раздел 5.8).
createdAt	Int64	Дата и время создания операции Unix Time.
expiresAt	Int64	Дата и время истечения операции Unix Time.
documentCount	Int	Количество документов в операции.
transactionId	String	Идентификатор транзакции.
parameters	[String: String]	Параметры операции (см. раздел 5.21).
documents	[Document]	Один или несколько документов, с которыми совершается операция (см. раздел 5.9).

5.11. Тип OperationHistory

Поле	Тип	Описание
records	[AuditRecord]	Список записей аудита. Записи аудита приводятся в документе «ЖТЯИ.00096-02 91 02 КриптоПро DSS. Руководство Администратора».
totalCount	Int	Количество выведенных записей аудита.
bookmark	Int	Идентификатор записи, относительно которой осуществляется поиск.

5.12. Тип PolicyPayload

Поле	Тип	Описание
selfRegistrationEnabled	Bool	Флаг, позволяющий/запрещающий самостоятельную регистрацию пользователя.
externalLoginRequired	Bool	Флаг, позволяющий/запрещающий «прозрачную» регистрацию пользователей.
keyActivationRequired	Bool	Флаг, определяющий, требуется ли ввод кода активации.
keyProtectionFlags	KeyProtectionFlags	Требования к защите вектора аутентификации (см. раздел 5.13).
keyActivationTypes	[String]	Список способов доставки кода активации. Допустимые значения: `SMS`, `Email`.

5.13. Тип keyProtectionFlags

Поле	Тип	Описание
fingerprintRequired	Bool	Требуется привязка векторов аутентификации к устройству.
collectEvents	Bool	Зарезервирован для использования в будущем.
collectDeviceInfo	Bool	Зарезервирован для использования в будущем.
collectSimInfo	Bool	Зарезервирован для использования в будущем.

Поле	Тип	Описание
collectLocation	Bool	Зарезервирован для использования в будущем.
passwordPolicy	Int	Требования к паролю. Допустимые значения: 0 – нет требований, 1 – только цифры, 2 – цифры и буквы, 3 – цифры, буквы и специальные символы.
denyOSProtection	Bool	Разрешена / запрещена биометрия для защиты векторов аутентификации.
scoringEnabled	Bool	Зарезервирован для использования в будущем.
strongKeyProtectionType	Bool	Тип векторов аутентификации. Для создания усиленной квалифицированной электронной подписи используется значение **True**.

5.14. Тип SignServerPolicy

Поле	Тип	Описание
CAPolicy	[CAPolicy]	Параметры обработчика УЦ (см. раздел 5.15).
ProcessingTemplateInfo	[ProcessingTemplateInfo]	Список шаблонов подписи (см. раздел 5.16)

5.15. Тип CAPolicy

Поле	Тип	Описание
cryptoProviderInfos	[String: [CryptoProviderInfo]]	Идентификаторы криптопровайдера. Допустимые значения: 80, 81.
showInUi	Bool	Зарезервирован для использования в будущем.
extensionsPolicy	String	Расширения сертификата.
id	Int	Идентификатор обработчика УЦ.
name	String	Отображаемое имя обработчика УЦ.
active	Bool	Статус обработчика УЦ.

Поле	Тип	Описание
allowUserMode	Bool	Зарезервирован для использования в будущем.
snChangesEnable	Bool	Разрешить изменять имя субъекта в сертификате.
namePolicy	[NamePolicy]	Конфигурация компонентов имени пользователя.
ekuTemplates	[String: [String]]	Конфигурация шаблонов сертификатов пользователя.
caType	String	Тип обработчика УЦ.
validationMode	String	Режим проверки сертификата ЭП перед использованием. Возможные значения: **`ChainOffline`** - для локально установленного CRL, **`ChainOnline`** - для локально установленного или загруженного по сети CRL ИЛИ при помощи OCSP-службы, **`NoCheck`** - не проверять.

5.16. Тип ProcessingTemplateInfo

Поле	Тип	Описание
id	Int	Идентификатор шаблона подписи.
description	String	Отображаемое имя шаблона подписи.

5.17. Тип Devices

Тип представляет собой список типа **`deviceinfo`** (см. раздел 5.18).

5.18. Тип Deviceinfo

Поле	Тип	Описание
uid	String	Идентификатор пользователя DSS.
kid	String	Идентификатор устройства пользователя.
userName	String	Логин пользователя.
profile	String	Профиль пользователя в DSS.

Поле	Тип	Описание
nonceRequired		Для подтверждения учетной записи требуется сканирование QR-кода типа, указанного в типе **type** (см. раздел 5.6).
deviceName	String	Отображаемое имя устройства пользователя.
notBefore	Int64	Дата начала действия вектора аутентификации. Время в формате Unix Time.
notAfter	Int64	Дата окончания действия вектора аутентификации. Время в формате Unix Time.
state	String	Состояние начального вектора аутентификации (см. раздел 5.5).

5.19. Тип UploadFile

Поле	Тип	Описание
documentInfo	String	Сопровождающий текст о документе.
snippetTemplate	String	Html-шаблон для формирования краткой информации о документе.
previewTemplate	String	Html-шаблон для формирования печатной формы документа
url	URL	Идентификатор документа.

5.20. Тип UploadDocInfo

Поле	Тип	Описание
docId	String	Идентификатор документа.

5.21. Тип parameters

Поле	Тип	Описание
TemplateId	String	Идентификатор шаблона подписи.

5.22. Тип DocumentSelectedMode

Поле	Описание
allDocuments	Режим выбора документов «только все».
withSelected	Режим выбора отдельных документов для подписи.

5.23. Тип ConfirmationSendingMode

Поле	Описание
online	Режим онлайн-отправки кода подтверждения.
offline	Режим оффлайн-отправки кода подтверждения.

5.24. Тип ApproveRequestMT

Поле	Тип	Описание
approvedOperation	String	Данные операции, для которых вычисляется код подтверждения.
hmac	String	Код подтверждения.

5.25. Тип SignatureResult

Список **`OperationResultInfo (см. раздел 5.26).`**

5.26. Тип OperationResultInfo

Поле	Тип	Описание
refId	String	Идентификатор подписанного документа.
originalRefId	String	Идентификатор исходного документа.
status	String	Результат операции.
error	String	Код ошибки взаимодействия с DSS.
errorDescription	String	Описание ошибки взаимодействия с DSS.

5.27. Тип approveRequestMO

Поле	Тип	Описание
approvedOperation	ApprovedOperation	Структурированные данные операции, для которых вычисляется код подтверждения (см. раздел 5.28).

5.28. Тип ApprovedOperation

Поле	Тип	Описание
id	String	Идентификатор транзакции.
type	String	Тип операции (возвращается только в методе **`SignMT`**).
caption	String	Краткое описание операции (возвращается только в методе **`SignMT`**).
parameters	[String: String]	Параметры операции (см. раздел 5.21).
confirmedDocuments	[ConfirmedDocument]	Список подтверждаемых документов.
declinedDocuments	[DeclinedDocument]	Список отклоняемых документов.
timeStamp	Int64	Метка времени, полученная на момент вычисления кода подтверждения.

5.29. Тип ConfirmedDocument

Поле	Тип	Описание
id	String	Идентификатор документа.
documentHash	String	Хэш-значение от документа.
snippetHash	String	Хэш-значение от краткой информации о документе.

5.30. Тип DeclinedDocument

Поле	Тип	Описание
id	String	Идентификатор документа.
documentHash	String	Хэш-значение от документа.

Поле	Тип	Описание
snippetHash	String	Хэш-значение от краткой информации о документе.
reason	String	Причина отклонения операции с документом.

5.31. Тип Error

Содержит информацию об ошибках. Полный перечень ошибок приведен в Приложении 3.

Приложение 1. Файл стилизации графического интерфейса DSS Client SDK

Задание параметров стилей графического интерфейса DSS Client SDK производится при помощи файла стилизации **SDKStyles.json**. При необходимости разработчик может переопределять стили, тем самым персонализируя фрагменты графического интерфейса под дизайн своего приложения.

Значения по умолчанию файла стилизации **SDKStyles.json** представлены в Приложении 2. Ниже приведены пояснения по применению этих параметров.

При задании значений цветов используется цветовая модель **#RGBA**, где параметр A является альфа-каналом.

1. **header.** Стилизация заголовка окна.
   - 1.1. **backgroundColor.** Цвет фона хэдера. Применяется для всех окон sdk.
   - 1.2. **rightIcon.** Картинка (png) в формате base64. Данная картинка используется во view отображения списка документов в операции, ожидающий подтверждения пользователя (при signMT).
   - 1.3. **iconsTintColor**. Цвет вышеуказанной картинки.
   - 1.4. **title**. Стилизация заголовка окна (цвет, размер текста, шрифт).
2. **body.** Стилизация основной части окна.
   - 2.1. **backgroundColor.** Цвет основной части окна. Применяется для всех окон sdk.
   - 2.2. **signCell.** Стилизация ячейки, отображающая информацию об подписываемом документе. Используется при signMT и signMO.
     - 2.2.1. **cell.** Стилизация контура ячейки.
       - 2.2.1.1. **backgroundColor**. Цвет фона ячейки.
       - 2.2.1.2. **shadowStyle.** Наложение тени на ячейку.
       - 2.2.1.3. **borderStyle**. Стилизация границ ячейки.
     - 2.2.2. **textColor.** Цвет основного текста ячейки.
     - 2.2.3. **moreImage.** Картинка (png) в формате base64.
     - 2.2.4. **imageTintColor.** Цвет вышеуказанной картинки.
     - 2.2.5. **subMenu.** Стилизация всплывающего подменю.
       - 2.2.5.1. **backgroundColor**. Цвет фона подменю.
       - 2.2.5.2. **shadowStyle.** Наложение тени на подменю.
       - 2.2.5.3. **borderStyle**. Стилизация границ подменю.
     - 2.2.6. **subMenuText.** Стилизация текста подменю (цвет, размер текста, шрифт).
   - 2.3. **modalView.** Стилизация окна - предупреждения об отклонении операции.
     - 2.3.1. **backgroundColor**. Цвет фона окна.
     - 2.3.2. **text.** Стилизация текста - предупреждения (цвет, размер текста, шрифт).
     - 2.3.3. **saveChoiceText.** Стилизация текста - предупреждения о небезопасном использовании sdk (цвет, размер текста, шрифт).
     - 2.3.4. **switchOnTintColor**. Цвет тумблера.
     - 2.3.5. **switchThumbTintColor**. Цвет активного состояния тумблера.
     - 2.3.6. **shadowStyle.** Наложение тени на окно – предупреждение.
     - 2.3.7. **borderStyle**. Стилизация границ окна – предупреждения.
     - 2.3.8. **confirm.** Стилизация кнопки – подтверждения.
       - 2.3.8.1. **backgroundColor**. Цвет фона кнопки.
       - 2.3.8.2. **shadowStyle.** Наложение тени на кнопку.
       - 2.3.8.3. **borderStyle**. Стилизация границ кнопки.
       - 2.3.8.4. **title.** Стилизация текста кнопки (цвет, размер текста, шрифт).
     - 2.3.9. **reject.** Стилизация кнопки – отклонения.
       - 2.3.9.1. **backgroundColor**. Цвет фона кнопки.
       - 2.3.9.2. **shadowStyle.** Наложение тени на кнопку.
       - 2.3.9.3. **borderStyle**. Стилизация границ кнопки.

2.3.9.4. title. Стилизация текста кнопки (цвет, размер текста, шрифт).

2.4. profileCell. Стилизация ячейки, отображающая профиль пользователя. Используется в функции verify.

2.4.1. cell. Стилизация контура ячейки.

2.4.1.1. backgroundColor. Цвет фона ячейки.

2.4.1.2. shadowStyle. Наложение тени на ячейку.

2.4.1.3. borderStyle. Стилизация границ ячейки.

2.4.2. title. Стилизация заголовка ячейки (цвет, размер текста, шрифт).

2.4.3. subTitle. Стилизация основного текста ячейки (цвет, размер текста, шрифт).

2.5. textField. Стилизация textField. Используется в окнах ввода пароля.

2.5.1. backgroundColor. Цвет фона textField.

2.5.2. lineColor. Цвет линии.

2.5.3. shadowStyle. Наложение тени на textField.

2.5.4. borderStyle. Стилизация границ textField.

2.5.5. title. Стилизация заголовка textField (цвет, размер текста, шрифт).

2.5.6. text. Стилизация основного текста textField (цвет, размер текста, шрифт).

2.6. errorLabel. Стилизация текста - описания ошибки (цвет, размер текста, шрифт).

2.7. attemptsDescriptionLabel. Стилизация текста - описания количества попыток для ввода пароля (цвет, размер текста, шрифт).

2.8. hintLabel. Стилизация текста - подсказки об используемой сложности пароля (цвет, размер текста, шрифт).

2.9. operationInfoCell. Стилизация окна, отображающую подробную информацию об операции. Используется в signMT.

2.9.1. cell. Стилизация контура ячейки.

2.9.1.1. backgroundColor. Цвет фона ячейки.

2.9.1.2. shadowStyle. Наложение тени на ячейку.

2.9.1.3. borderStyle. Стилизация границ ячейки.

2.9.2. header. Стилизация заголовка окна (цвет, размер текста, шрифт).

2.9.3. title. Стилизация заголовка ячейки (цвет, размер текста, шрифт).

2.9.4. subTitle. Стилизация основного текста ячейки (цвет, размер текста, шрифт).

2.10. qrReader. Стилизация окна, сканирования qr кода.

2.10.1. description. Стилизация текста - подсказки (цвет, размер текста, шрифт).

2.10.2. error. Стилизация текста - описания ошибки (цвет, размер текста, шрифт).

2.10.3. galleryIcon. Картинка (png) в формате base64.

2.10.4. imageTintColor. Цвет вышеуказанной картинки.

3. footer. Стилизация футера окна.

3.1. backgroundColor. Цвет фона футера. Применяется для всех окон sdk.

3.2. shadowStyle. Наложение тени на футер.

3.3. borderStyle. Стилизация границ футера.

3.4. confirm. Стилизация кнопки – подтверждения.

3.4.1. backgroundColor. Цвет фона кнопки.

3.4.2. shadowStyle. Наложение тени на кнопку.

3.4.3. borderStyle. Стилизация границ кнопки.

3.4.4. title. Стилизация текста кнопки (цвет, размер текста, шрифт).

3.5. reject. Стилизация кнопки – отклонения.

3.5.1. backgroundColor. Цвет фона кнопки.

3.5.2. shadowStyle. Наложение тени на кнопку.

3.5.3. borderStyle. Стилизация границ кнопки.

3.5.4. title. Стилизация текста кнопки (цвет, размер текста, шрифт).

3.6. slider.

3.6.1. minimumTrackTintColor. Цвет минимального изображения дорожки.

3.6.2. maximumTrackTintColor. Цвет максимального изображения дорожки.

3.6.3. thumbTintColor. Цвет ползунка

3.7. pageCounter. Стилизация текста - количества страниц в pdf (цвет, размер текста, шрифт).

4. dialogString. Текстовка окон.

Приложение 2. Значения по умолчанию файла стилизации графического интерфейса DSS Client SDK

```
{
   "header": {
       "backgroundColor": "#311b92ff",
       "rightIcon": "",
       "iconsTintColor": "#ffffffff",
       "title": {
           "color": "#ffffffff",
           "size": 17,
           "font": ""
       }
   },
   "body": {
       "backgroundColor": "#ffffffff",
       "signCell": {
           "cell": {
               "backgroundColor": "#ffffffff",
               "shadowStyle": {
                   "shadowColor": "#000000ff",
                   "shadowOffsetWidth": 0,
                   "shadowOffsetHeight": 0,
                   "shadowRadius": 0,
                   "shadowOpacity": 0
               },
               "borderStyle": {
                   "borderColor": "#191919FF",
                   "borderWidth": 1.5
               }
           },
           "textColor": "#000000ff",
           "moreImage": "",
           "imageTintColor": "#000000ff",
           "subMenu": {
               "backgroundColor": "#ffffffff",
               "shadowStyle": {
                   "shadowColor": "#434343ff",
                   "shadowOffsetWidth": 0,
                   "shadowOffsetHeight": 0,
                   "shadowRadius": 3,
                   "shadowOpacity": 1
               },
               "borderStyle": {
                   "borderColor": "#464646ff",
                   "borderWidth": 0
               }
           },
           "subMenuText": {
               "color": "#000000ff",
               "size": 15,
               "font": ""
           }
       },
       "modalView": {
           "backgroundColor": "#ffffffff",
           "text": {
               "color": "#000000ff",
               "size": 20,
               "font": ""
           },
           "saveChoiceText": {
               "color": "#000000ff",
               "size": 13,
               "font": ""
           },
           "switchOnTingColor": "#906ff2ff",
           "switchThumbTingColor": "#4527a0ff",
           "shadowStyle": {
               "shadowColor": "#000000ff",
               "shadowOffsetWidth": 0,
               "shadowOffsetHeight": 0,
               "shadowRadius": 0,
               "shadowOpacity": 0
           },
           "borderStyle": {
               "borderColor": "#000000ff",
               "borderWidth": 0
           },
           "confirm": {
               "backgroundColor": "#4527a0ff",
               "shadowStyle": {
                   "shadowColor": "#4527a000",
                   "shadowOffsetWidth": 0,
                   "shadowOffsetHeight": 0,
                   "shadowRadius": 0,
                   "shadowOpacity": 0
               },
               "borderStyle": {
                   "borderColor": "#4527a0ff",
                   "borderWidth": 0.3
               },
               "title": {
                   "color": "#ffffffff",
                   "size": 17,
                   "font": ""
               }
           },
           "reject": {
               "backgroundColor": "#ffffffff",
               "shadowStyle": {
                   "shadowColor": "#bdbdbd00",
                   "shadowOffsetWidth": 0,
                   "shadowOffsetHeight": 0,
                   "shadowRadius": 0,
                   "shadowOpacity": 0
               },
               "borderStyle": {
                   "borderColor": "#dcdcdcFF",
                   "borderWidth": 0.3
               },
               "title": {
                   "color": "#512da8ff",
                   "size": 17,
                   "font": ""
               }
           }
       },
       "profileCell": {
           "cell": {
               "backgroundColor": "#00000000",
               "shadowStyle": {
                   "shadowColor": "#000000ff",
                   "shadowOffsetWidth": 0,
                   "shadowOffsetHeight": 0,
                   "shadowRadius": 0,
                   "shadowOpacity": 0
               },
               "borderStyle": {
                   "borderColor": "#000000ff",
                   "borderWidth": 0
               }
           },
           "title": {
               "color": "#000000ff",
               "size": 15,
               "font": ""
           },
           "subTitle": {
               "color": "#000000ff",
               "size": 15,
               "font": ""
           }
       },
       "textField": {
           "backgroundColor": "#eeeeeeff",
           "lineColor": "#512da8ff",
           "shadowStyle": {
               "shadowColor": "#000000ff",
```

```
            "shadowOffsetWidth": 0,
            "shadowOffsetHeight": 0,
            "shadowRadius": 0,
            "shadowOpacity": 0
        },
        "borderStyle": {
            "borderColor": "#000000ff",
            "borderWidth": 0
        },
        "title": {
            "color": "#512da8ff",
            "size": 15,
            "font": ""
        },
        "text": {
            "color": "#000000ff",
            "size": 17,
            "font": ""
        }
    },
    "errorLabel": {
        "color": "#c62828ff",
        "size": 15,
        "font": ""
    },
    "attemptsDescriptionLabel": {
        "color": "#512da8ff",
        "size": 15,
        "font": ""
    },
    "hintLabel": {
        "color": "#000000ff",
        "size": 11,
        "font": ""
    },
    "operationInfoCell": {
        "cell": {
            "backgroundColor": "#ffffffff",
            "shadowStyle": {
                "shadowColor": "#000000ff",
                "shadowOffsetWidth": 0,
                "shadowOffsetHeight": 0,
                "shadowRadius": 0,
                "shadowOpacity": 0
            },
            "borderStyle": {
                "borderColor": "#000000ff",
                "borderWidth": 0
            }
        },
        "header": {
            "color": "#000000ff",
            "size": 20,
            "font": ""
        },
        "title": {
            "color": "#000000ff",
            "size": 17,
            "font": ""
        },
        "subTitle": {
            "color": "#000000ff",
            "size": 15,
            "font": ""
        }
    },
    "biometric": {
        "shadowStyle": {
            "shadowColor": "#000000ff",
            "shadowOffsetWidth": 0,
            "shadowOffsetHeight": 0,
            "shadowRadius": 0,
            "shadowOpacity": 0
        },
        "borderStyle": {
            "borderColor": "#000000ff",
            "borderWidth": 0
        },
        "title": {
           "color": "#bdbdbdFF",
            "size": 17,
            "font": ""
        },
        "errorTitle": {
            "color": "#512da8ff",
            "size": 17,
            "font": ""
        },
        "description": {
            "color": "#000000ff",
            "size": 15,
            "font": ""
        },
        "errorDescription": {
            "color": "#000000ff",
            "size": 15,
            "font": ""
        },
        "cancelButtonTitle": {
            "color": "#000000ff",
            "size": 15,
            "font": ""
        },
        "fingerprintIcon": "",
        "fingerprintIconColor": "#512da8ff",
        "errorIcon": "",
        "errorIconColor": "#512da8ff"
    },
    "qrReader": {
        "description": {
            "color": "#000000ff",
            "size": 17,
            "font": ""
        },
        "error": {
            "color": "#ff0000ff",
            "size": 13,
            "font": ""
        },
        "galleryIcon": "",
        "imageTintColor": "#000000ff"
    }
},
"footer": {
    "backgroundColor": "#ffffffff",
    "shadowStyle": {
        "shadowColor": "#000000ff",
        "shadowOffsetWidth": 0,
        "shadowOffsetHeight": 0,
        "shadowRadius": 0,
        "shadowOpacity": 0
    },
    "borderStyle": {
        "borderColor": "#000000ff",
        "borderWidth": 0
    },
    "confirm": {
        "backgroundColor": "#4527a0ff",
        "shadowStyle": {
            "shadowColor": "#4527a000",
            "shadowOffsetWidth": 0,
            "shadowOffsetHeight": 0,
            "shadowRadius": 0,
            "shadowOpacity": 0
        },
        "borderStyle": {
            "borderColor": "#4527a0ff",
            "borderWidth": 0.3
        },
        "title": {
            "color": "#ffffffff",
            "size": 17,
            "font": ""
        }
    },
    "reject": {
        "backgroundColor": "#ffffffff",
        "shadowStyle": {
            "shadowColor": "#bdbdbd00",
            "shadowOffsetWidth": 0,
```

```
                "shadowOffsetHeight": 0,
                "shadowRadius": 0,
                "shadowOpacity": 0
            },
            "borderStyle": {
                "borderColor": "#dcdcdcff",
                "borderWidth": 0.3
            },
            "title": {
                "color": "#512da8ff",
                "size": 17,
                "font": ""
            }
        },
        "slider": {
            "minimumTrackTintColor":
"#512da8ff",
            "maximumTrackTintColor":
"#bdbdbdff",
            "thumbTintColor": "#512da8ff"
        },
        "pageCounter": {
            "color": "#000000ff",
            "size": 17,
            "font": ""
        }
    },
    "dialogString": {
        "passwordHeader": "Ввод пароля",
        "passwordHeaderWithConfirm":    "Введите
новый пароль",
        "passwordTitle": "Пароль",
        "passwordPlaceholder": "Введите пароль",
        "passwordConfirmTitle":   "Подтверждение
пароля",
        "passwordConfirmPlaceholder":
"Подтвердите пароль",
        "passwordContinue": "ПОДТВЕРДИТЬ",
        "passwordEasy": "Пароль должен быть не
менее 6 символов",
        "passwordMedium": "Пароль должен быть не
менее 8 символов и содержать строчные и
прописные буквы",
        "passwordStrong": "Пароль должен быть не
менее 8 символов и содержать строчные и
прописные буквы, цифры",
        "passwordEmpty": "Поле не может быть
пустым",
        "passwordsNotMatch":     "Пароли     не
совпадают",
        "incorrectPassword": "Неверный пароль",
        "passwordEasyHint": "Буквы a-z, A-Z, не
менее 6 символов",
        "passwordMediumyHint": "Буквы a-z, A-Z,
не менее 8 символов",
        "passwordStrongHint": "Цифры 0-9, буквы
a-z, A-Z, не менее 8 символов",
        "passwordTouchIDHint": "Введите пароль
для использования в случае ошибки TouchID",
        "numberOfAttempts": "Количество попыток:
",
        "numberOfAttemptsExceeded":          "Вы
превысили число попыток. Попробуйте снова через
минуту",
        "biometricAuthReason":   "Аутентификация
для доступа к персональным данным",
        "biometricAuthReasonSave":   "Сохранение
ключа",
        "biometricErrorTitle": "Ошибка",
        "biometricDescription": "Прикоснитесь к
сенсору",
        "biometricErrorDescription": "Не удалось
распознать отпечаток. Прикоснитесь еще раз",
        "incorrectQRCode": "Некорректные данные
в QR коде",
        "qrDescription": "Наведите камеру на QR-
код",
        "settingsTitle":   "Доступ   к   камере
отключен",
        "settingsSubtitle": "Перейти в Настройки
для разрешения доступа к камере?",
        "settingsCancel": "Отмена",
        "settingsSettings": "Настройки",
        "operationHeader":           "Содержимое
операции",
        "operationCancel": "ОТМЕНА",
        "operationConfirm": "ПОДТВЕРДИТЬ",
        "operationRefuse": "ОТКАЗАТЬСЯ",
        "printPdfHeader":   "Версия   для   печати
(pdf)",
        "rawPdfHeader":    "Оригинал    документа
(pdf)",
        "profileHeader": "Профиль пользователя",
        "profileConfirm": "ПОДТВЕРДИТЬ",
        "confirmedDeviceInfo":      "Добавляемое
устройство",
        "confirmedDeviceName": "Имя устройства",
        "confirmedDeviceConfirm": "ПОДТВЕРДИТЬ",
        "confirmedDeviceRefuse": "ОТКАЗАТЬСЯ",
        "moreAboutOfOperation":   "Подробнее   об
операции",
        "modalViewDescription": "Вы уверены, что
хотите отправить запрос 'Отмена'?",
        "modalViewConfirm": "ПОДТВЕРДИТЬ",
        "modalViewRefuse": "ОТКАЗАТЬСЯ",
        "pdfPageNotLoaded":      "Не      удалось
загрузить страницу",
        "securityHeader":   "Вы   уверены,   что
хотите продолжить работу?",
        "securityNotInstalledLockScreen":    "*
экран блокировки не установлен",
        "securityDeviceIsRooted": "* устройство
рутовано",
        "securityKasperskyNotInstalled":     "*
касперский не установлен",
        "securityRememberChoice":    "Запомнить
выбор?",
        "securityHasOverlayApps":    "*    список
возможных шпионских приложений",
        "securityConfirm": "ПОДТВЕРДИТЬ",
        "securityRefuse": "ОТКАЗАТЬСЯ"
    }
}
```

Приложение 3. Сообщения об ошибках

Приложение содержит идентификаторы и тексты сообщений об ошибках, возвращаемых в методах DSS Client SDK.

Идентификатор	**Сообщение об ошибке**
functionFailed	Key function failed. CSP: Работа с ключами невозможна.
invalidParameters	Input parameters are incorrect. CSP: Входные параметры неверны.
noMemory	No memory. CSP: Недостаточно памяти.
moreDataNeeded	More data needed. CSP: Требуется больше данных.
invalidBlobHeader	Invalid blob header. CSP: Неверный заголовок блоба.
invalidPassword	Invalid password. CSP: Неверный пароль.
functionNotImplemented	Function not implemented. CSP: Функция не реализована.
base64EncodedFailed	Can't convert base64 String to Data. Не удалось преобразовать String в Data.
decodeStruct2String	Failed to decode struct in string. Ошибка при десериализации структуры.
nameIsExist	An authentication vector with the same name already exists. Вектор аутентификации с таким именем уже существует.
nameIsEmpty	Name can't be empty. Имя не может быть пустым.
authVectorNotFound	Authentication vector not found. Вектор аутентификации не найден.
kidNotFound	kid not found. kid не найден.
incorrectKeyStatus	Incorrect key status. Некорректный статус ключа.
keyExpired	Keys are expired. Ключ недействительный.
selfRegistrationImpossible	Self-registration on this server disabled. Саморегистрация на сервере запрещена.
needKeyActivationCode	Need key activation code to the server. Требуется код активации.
biometricKeyProtectionImpossible	Key protection policy does not allow using TouchId / FaceId parameter. Политика защиты ключей не позволяет использовать тип защиты 'TouchId / FaceId.

noPolicyKeyProtectionError	The key protection policy does not allow the use of the 'No policy' parameter. Политика защиты ключей не позволяет использовать тип защиты 'No policy'.
encryptSetKeyError	Failed to set a key. Не удалось задать ключ шифрования.
encryptSetVectorError	Failed to set an initial vector. Не удалось задать вектор инициализации для шифрования.
biometryNotAvailable	Biometry is not available on the device. Биометрическая защита недоступна на устройстве.
biometryLockout	Biometry is locked because there were too many failed attempts. Биометрия заблокирована, потому что было слишком много неудачных попыток.
biometryNotEnrolled	The user has no enrolled biometric identities. У пользователя нет зарегистрированных биометрических идентификаторов.
biometricUndefinedError	Did not find error code on LAError object. Не удалось найти код ошибки для объекта LAError.
touchIDLockout	Touch ID is locked because there were too many failed attempts. Touch ID заблокирован, потому что было слишком много неудачных попыток.
touchIDNotAvailable	Touch ID is not available on the device. Touch ID недоступен на устройстве.
touchIDNotEnrolled	The user has no enrolled Touch ID fingers. У пользователя нет зарегистрированных пальцев Touch ID.
authenticationFailed	The user failed to provide valid credentials. Пользователь не смог предоставить действительные учетные данные.
appCancel	The app canceled authentication. Приложение отменило аутентификацию
invalidContext	The context was previously invalidated. Контекст был ранее признан недействительным.
notInteractive	Displaying the required authentication user interface is forbidden. Отображение необходимого интерфейса пользователя для аутентификации запрещено.
passcodeNotSet	A passcode isn’t set on the device. Пароль не установлен на устройстве.
systemCancel	The system canceled authentication. Система отменила аутентификацию.
userCancel	Operation has been cancelled by user. Операция была отменена пользователем.
userFallback	The user tapped the fallback button in the authentication dialog, but no fallback is available for the authentication policy. Пользователь нажал кнопку возврата в диалоге аутентификации, но для политики аутентификации нет запасного варианта.
invalidData	Invalid data. Не удалось считать пароль пользователя.

biometricPwdSaveError	Saving on password failed for biometric authentication. Не удалось сохранить пароль пользователя для биометрической аутентификации.
avCaptureDeviceError	AVCaptureDevice.default(for: .video) error. Ошибка AVCaptureDevice.
deviceError	Your device does not support scanning a code from an item. Please use a device with a camera. Ваше устройство не поддерживает сканирование кода из элемента. Пожалуйста, используйте устройство с камерой.
imageError	Expected a dictionary containing an image, but was provided the following: Ожидается словарь, содержащий изображение, но было предоставлено следующее:
invalidQrCode	Wrong qr code. Некорректный qr код.
qrNotContainOperation	Qr code does not contain an operation. Операция недействительна.
operationExpired	Operation has expired. Операция не загружена в оперативную память.
operationNotLoaded	Operation not loaded into RAM.
reasonIncorrect	The var 'reason' must be number between 0 and 5 inclusive. Переменная 'reason' должна быть числом от 0 до 5 включительно.
cidOrRidFillingError	One of var 'cid' or 'rid' need to fill. Необходимо заполнить один из переменных 'cid' или 'rid'.
sessionFailed	Session not opened. Сессия не открыта.
deniedOfUseSDKMethod	Denied of use sdk method Отказано в использовании метода SDK