Цифровые технологии

КриптоАРМ Server

Документация продукта

Релиз v1.3

Исходная документация на CryptoARM.ru

Редакция от 26.11.2026

Правообладатель и разработчик
ООО «Цифровые технологии»

Содержание

КриптоАРМ Server — интеграция и автоматизация криптоопераций 3
О продукте 7
Установка и запуск 11
Авторизация API 23
API для работы с файлами 27
Настройка текста в PDF-отчетах 34
Варианты использования 39
Часто задаваемые вопросы 50
Глоссарий 53
Введение в стандарты электронной подписи 60

Открыть статью на сайте

О продукте КриптоАРМ Server

КриптоАРМ Server — программный продукт, предназначенный для интеграции и автоматизации криптографических операций в информационных системах организаций. Включает библиотеки и SDK для внедрения функций шифрования, создания и проверки ЭП, а также поддерживает массовую проверку и автоматическую подпись документов по заданным правилам.

Содержание:

- Поддерживаемые платформы
- Программное окружение
- Типовые сценарии автоматизации
- Функциональные модули
- Интеграция

Поддерживаемые платформы

КриптоАРМ Server может быть развернут на следующих операционных системах:

- Astra Linux
- РЕД ОС
- АЛЬТ
- РОСА
- AlterOS
- ОС «Основа»
- Ubuntu
- Microsoft Windows Server

Также доступна поставка Docker-образа (на базе Astra Linux) для контейнеризированного развертывания.

Программное окружение

Для работы **КриптоАРМ Server** используется следующее программное окружение:

- Node.js — сервер приложений
- КриптоПро CSP 5.0 — криптопровайдер
- КриптоАРМ Server — сервис обработки криптографических операций

Типовые сценарии автоматизации

Проверка электронной подписи

Сервис позволяет автоматически проверять подписи входящих документов и формировать отчеты о результатах проверки.

Ключевые возможности:

- проверка действительности электронной подписи
- проверка сертификатов подписанта
- определение квалифицированности сертификата
- поддержка форматов CMS, CAdES, PAdES, XMLDSig
- получение подробного отчета о проверке подписи
- визуализация подписи в PDF-документах
- добавление данных о документе в отчет проверки
- указание причин недействительной подписи

Отчет о проверке может быть **кастомизирован** и дополнен дополнительной информацией.

Автоматизация электронной подписи

КриптоАРМ Server позволяет автоматизировать процесс подписания документов.

Ключевые возможности:

- подписание документов электронной подписью
- поддержка форматов CMS, CAdES, PAdES, XMLDSig
- использование сертификатов с PIN-кодом
- пакетная подпись документов
- объединение нескольких отсоединенных подписей в один файл подписи
- добавление визуального штампа подписи
- создание штампа времени
- получение данных штампа подписи
- усовершенствование подписи до форматов CAdES-T, CAdES-X Long Type 1, CAdES-A

Также доступна подпись документов **обезличенными сертификатами УКЭП**.

Работа с PDF-подписями

Для PDF-документов реализованы дополнительные функции:

- создание и проверка подписи формата **PAdES**
- добавление визуального штампа подписи
- настройка расположения штампа на странице
- отображение штампа только для действительной подписи
- сертификация подписанного документа (защита от изменений)
- пакетная подпись PDF-документов

- конвертация PDF в формат **PDF/A-2b**

Шифрование и защита данных

Сервис поддерживает операции шифрования и расшифрования файлов и почтовых писем.

Ключевые возможности:

- шифрование файлов для одного или нескольких получателей
- расшифрование входящих данных
- получение информации о сертификатах получателей
- отправка защищенных писем контактам и группам контактов
- поддержка различных форматов кодировки

Эти функции позволяют автоматизировать защиту входящих и исходящих документов.

Функциональные модули

КриптоАРМ Server состоит из нескольких функциональных модулей:

Модуль	Описание
trusted-crypto	Формирование и проверка электронной подписи, шифрование и расшифрование данных, получение сведений о сертификатах, генерация ключевых пар, создание запросов на получение сертификатов, подключение защищенных носителей
trusted-pdf	Подпись и проверка электронной подписи в PDF, разметка областей для штампа подписи, добавление визуальных штампов, пакетная подпись PDF, сертификация подписанных документов, конвертация PDF в PDF/A
trusted-mailcore	Сбор, обработка и отправка электронных писем. Поддержка протоколов IMAP, SMTP, POP3, S/MIME, а также работа с контактами и адресными книгами (vCard, CardDAV)
trusted-curl	Обеспечение защищенного соединения с веб-серверами с использованием протокола ГОСТ TLS
trusted-xml-crypto	Проверка электронной подписи формата XMLDSig (XML Digital Signature), используемой в интеграционных системах и государственных сервисах

Интеграция

КриптоАРМ Server предоставляет HTTP API для выполнения криптографических операций из внешних информационных систем.

API позволяет:

- выполнять подписание и проверку электронной подписи;
- шифровать и расшифровывать файлы;
- получать сведения о сертификатах;
- выполнять пакетную подпись документов;
- формировать отчеты о проверке подписи.

Для доступа к API поддерживается авторизация по API-ключам.

Открыть статью на сайте

О продукте КриптоАРМ Server

КриптоАРМ Server — программный продукт, предназначенный для интеграции и автоматизации криптографических операций в информационных системах организаций. Включает библиотеки и SDK для внедрения функций шифрования, создания и проверки ЭП, а также поддерживает массовую проверку и автоматическую подпись документов по заданным правилам.

Содержание:

- Поддерживаемые платформы
- Программное окружение
- Типовые сценарии автоматизации
- Функциональные модули
- Интеграция

Поддерживаемые платформы

КриптоАРМ Server может быть развернут на следующих операционных системах:

- Astra Linux
- РЕД ОС
- АЛЬТ
- РОСА
- AlterOS
- ОС «Основа»
- Ubuntu
- Microsoft Windows Server

Также доступна поставка Docker-образа (на базе Astra Linux) для контейнеризированного развертывания.

Программное окружение

Для работы **КриптоАРМ Server** используется следующее программное окружение:

- Node.js — сервер приложений
- КриптоПро CSP 5.0 — криптопровайдер
- КриптоАРМ Server — сервис обработки криптографических операций

Типовые сценарии автоматизации

Проверка электронной подписи

Сервис позволяет автоматически проверять подписи входящих документов и формировать отчеты о результатах проверки.

Ключевые возможности:

- проверка действительности электронной подписи
- проверка сертификатов подписанта
- определение квалифицированности сертификата
- поддержка форматов CMS, CAdES, PAdES, XMLDSig
- получение подробного отчета о проверке подписи
- визуализация подписи в PDF-документах
- добавление данных о документе в отчет проверки
- указание причин недействительной подписи

Отчет о проверке может быть **кастомизирован** и дополнен дополнительной информацией.

Автоматизация электронной подписи

КриптоАРМ Server позволяет автоматизировать процесс подписания документов.

Ключевые возможности:

- подписание документов электронной подписью
- поддержка форматов CMS, CAdES, PAdES, XMLDSig
- пакетная подпись документов
- объединение нескольких отсоединенных подписей в один файл подписи
- добавление визуального штампа подписи
- усовершенствование подписи до форматов CAdES-T, CAdES-X Long Type 1, CAdES-A

Также доступна подпись документов **обезличенными сертификатами УКЭП**.

Работа с PDF-подписями

Для PDF-документов реализованы дополнительные функции:

- создание и проверка подписи формата PAdES
- добавление визуального штампа подписи
- настройка расположения штампа на странице
- отображение штампа только для действительной подписи
- сертификация подписанного документа (защита от изменений)
- пакетная подпись PDF-документов
- конвертация PDF в формат PDF/A-2b

Шифрование и защита данных

Сервис поддерживает операции шифрования и расшифрования файлов и почтовых писем.

Ключевые возможности:

- шифрование файлов для одного или нескольких получателей
- расшифрование входящих данных
- получение информации о сертификатах получателей
- отправка защищенных писем контактам и группам контактов
- поддержка различных форматов кодировки

Эти функции позволяют автоматизировать защиту входящих и исходящих документов.

Функциональные модули

КриптоАРМ Server состоит из нескольких функциональных модулей:

Модуль	Описание
trusted-crypto	Формирование и проверка электронной подписи, шифрование и расшифрование данных, получение сведений о сертификатах, генерация ключевых пар, создание запросов на получение сертификатов, подключение защищенных носителей
trusted-pdf	Подпись и проверка электронной подписи в PDF, разметка областей для штампа подписи, добавление визуальных штампов, пакетная подпись PDF, сертификация подписанных документов, конвертация PDF в PDF/A
trusted-mailcore	Сбор, обработка и отправка электронных писем. Поддержка протоколов IMAP, SMTP, POP3, S/MIME, а также работа с контактами и адресными книгами (vCard, CardDAV)
trusted-curl	Обеспечение защищенного соединения с веб-серверами с использованием протокола ГОСТ TLS
trusted-xml-crypto	Проверка электронной подписи формата XMLDSig (XML Digital Signature), используемой в интеграционных системах и государственных сервисах

Интеграция

КриптоАРМ Server предоставляет HTTP API для выполнения криптографических операций из внешних информационных систем.

API позволяет:

- выполнять подписание и проверку электронной подписи;
- шифровать и расшифровывать файлы;
- получать сведения о сертификатах;
- выполнять пакетную подпись документов;
- формировать отчеты о проверке подписи.

Для доступа к API поддерживается авторизация по API-ключам.

Открыть статью на сайте

Установка и запуск КриптоАРМ Server

В этой инструкции вы узнаете как установить и настроить сервис **КриптоАРМ Server** с использованием **Docker**.

Содержание:

- Требования к системе
- Технологический стек
- Установка Docker
- Первый запуск сервиса
- Вход в сервис
- Проверка работы сервиса
- Установка сертификатов
- Настройка отчета
- Подпись данных
- Диагностика сетевых подключений

Требования к системе

Характеристика	Минимальное значение	Дополнение
Информационная емкость ОЗУ	4 Гб	
Информационная емкость HDD	50 Гб	
Процессор	x86_64 (или amd64)	
Поддерживаемые ОС	Ubuntu 18.04/20.04, Debian 11, Astra Linux SE 1.8 Орел	Возможна работа на других Linux-дистрибутивах с поддержкой Docker
Дополнительно	Требуется для контейнерного запуска	

Технологический стек

КриптоАРМ Server использует следующий стек технологий:

- Node.js
- NestJS
- КриптоПро CSP 5.0
- КриптоАРМ ГОСТ SDK

- Docker

Установка Docker

Для запуска проекта необходимо установить **Docker** и **Docker Compose**.

> 🔍 Подробная инструкция доступна в официальной документации: https://docs.docker.com/engine/install/.
>
> 📌 Минимально необходимые версии компонентов: Docker Engine: **19.03** или выше

Первый запуск сервиса

1. Создание рабочей директории

Создайте и перейдите в рабочую директорию:

```
mkdir cryptoarm-server && cd cryptoarm-server
```

2. Загрузка конфигурационных файлов

```
curl -O https://git.digtlab.ru/trusted/cryptoarm/server/-/raw/main/docker/docker-compose.yml
curl -O https://git.digtlab.ru/trusted/cryptoarm/server/-/raw/main/docker/Dockerfile
curl -O https://git.digtlab.ru/trusted/cryptoarm/server/-/raw/main/.env
```

3. Подготовка каталога для КриптоПро

Создайте каталог для установки КриптоПро CSP:

```
mkdir cryptopro
```

4. Загрузка КриптоПро CSP

1. Скачайте актуальную версию **КриптоПро CSP 5.0** для Linux (x64, deb) с официального сайта: https://cryptopro.ru/products/csp. Для загрузки требуется авторизация на сайте КриптоПро.
2. Скопируйте файл linux-amd64_deb.tgz в каталог `cryptoarm-server/cryptopro`.

5. Настройка переменных окружения

Лицензии

Откройте файл **.env** и укажите лицензии **КриптоПро CSP** и **КриптоАРМ Server**:

```
CRYPTOPRO_LICENSE=XXXXXXXXXXXXXXXXXXXXXXXXX
TRUSTED_LICENSE=XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

Настройка сервиса штампов времени

Опционально настройте переменные для настройки TSP-сервиса:

```
TSP_SERVICE_ADDRESS=http://qs.cryptopro.ru/tsp/tsp.srf
```

Настройка PDF-отчета

Переменная	Описание	По умолчанию
`REPORT_TITLE`	Заголовок PDF-отчета	`Отчет о проверке электронной подписи`
`DATA_STAMP_TITLE`	Заголовок штампа подписи	`Документ подписан электронной подписью`
`STAMP_COLOR`	Цвет штампа (HEX)	`#4c40d2`
`REPORT_INFO_MESSAGE`	Нижний колонтитул отчета	`{align:left}© {COMPANY_NAME} | | {align:right}Страница {pageNumber} из {totalPages} {WEBSITE_URL}`

Поддерживаемые переменные

Переменная	Описание	По умолчанию
`{pageNumber}`	номер страницы	-
`{totalPages}`	общее количество страниц	-
`{COMPANY_NAME}`	название организации	`ООО Цифровые технологии`
`{WEBSITE_URL}`	адрес сайта	`www.cryptoarm.ru`
`{VARIABLE_NAME}`	любая переменная окружения	-

Поддерживаемые директивы выравнивания

Директива	Описание
`{align:left}`	выравнивание по левому краю

Директива	Описание
`{align:center}`	выравнивание по центру
`{align:right}`	выравнивание по правому краю
`{align:justify}`	выравнивание по ширине

6. Сборка контейнера

Запустите сборку:

```
docker compose build --pull
```

7. Запуск сервиса

Запустите сервис:

```
docker compose up -d
```

Полезные команды

Просмотр логов:

```
docker compose logs -f
```

Остановка:

```
docker compose stop
```

Обновление сервиса

Для обновления версии необходимо загрузить актуальные конфигурационные файлы и пересобрать контейнер.

```
curl -O https://git.digtlab.ru/trusted/cryptoarm/server/-/raw/main/docker/docker-compose.yml
curl -O https://git.digtlab.ru/trusted/cryptoarm/server/-/raw/main/docker/Dockerfile
docker compose build --pull
docker compose up -d
```

Сборка из исходников

1. Клонируйте репозиторий:

   ```
   git clone https://git.digtlab.ru/trustednet/cryptoarm/server
   ```

2. Выполните пункты:
    - Подготовка каталога для КриптоПро
    - Загрузка КриптоПро CSP
    - Лицензии
3. Запустите сборку:

```
docker compose build
```

Вход в сервис

После запуска сервис становится доступен по адресу: `http://SERVER:PORT`

Пример: `http://172.17.1.115:3037`.

Swagger-документация по REST API: `http://SERVER:PORT/docs`

Пример: `http://172.17.1.115:3037/docs`.

Проверка работы сервиса

Проверить работу сервиса можно следующими способами:

1. Через Swagger интерфейс.
2. С помощью утилиты **curl**.

Пример запроса проверки подписи:

```
curl -X 'POST' \
'http://172.17.1.115:3037/cms/verify' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
"cms": "MIAGCSqGSIb3...==,MSMwIQYJKoZI...==",
"data": "dGVzdA==",
"enhancedType": "CAdES-X Long Type 1"
}'
```

Параметры запроса

Параметр	Описание
cms	подпись в `Base64`
data	подписанные данные (`Base64`)

Параметр	Описание
enhancedType	тип усовершенствования подписи

Поддерживаемые форматы:

- CAdES-T
- CAdES-X Long Type 1
- CAdES-A

Пример ответа

```
HTTP/1.1 201 Created
Vary: Origin
Access-Control-Allow-Credentials: true
Content-Type: application/json; charset=utf-8
Content-Length: 136101
ETag: W/"213a5-+1/6qQ+hWCfJ0yGRz0/fQGpbTbk"
Date: Thu, 26 Sep 2024 12:40:04 GMT
Connection: keep-alive
Keep-Alive: timeout=5
{
  "status": 200,
  "message":"OK",
  "isValid": true,
  "isValidSign": true,
  "signs": [
    {
      "cadesTypeName": "CAdES-BES",
      "certificate": {
        "version": 3,
        "status": true,
        "subjectFriendlyName": "esia test3",
        "issuerFriendlyName": "CRYPTO-PRO Test Center 2",
        "subjectName": "CN=esia test3",
        "issuerName": "E=support@cryptopro.ru\r\nC=RU\r\nL=Moscow\r\nO=CRYPTO-PRO
LLC\r\nCN=CRYPTO-PRO Test Center 2",
        "notAfter": "2024-10-26T16:25:37.000Z",
        "notBefore": "2024-08-26T16:15:37.000Z",
        "serialNumber": "120064ECE1352507757C00A08B00020064ECE1",
        "thumbprint": "29063cd83537dbbd0a221705859b8f6e5bec5ad0",
        "signatureAlgorithm": "1.2.643.7.1.1.3.2",
        "signatureDigestAlgorithm": "1.2.643.7.1.1.2.2",
        "publicKeyAlgorithm": "1.2.643.7.1.1.1.1"
      },
      "certificateData": "MIIDEz...",
      "issuerName": "E=support@cryptopro.ru, C=RU, L=Moscow, O=CRYPTO-PRO LLC, CN=CRYPTO-PRO
Test Center 2",
      "serialNumber": "120064ECE1352507757C00A08B00020064ECE1",
      "signatureAlgorithm": "1.2.643.7.1.1.3.2",
      "signatureDigestAlgorithm": "1.2.643.7.1.1.2.2",
      "certs": [
        {
          "status": true,
          "subjectFriendlyName": "esia test3",
          "issuerFriendlyName": "CRYPTO-PRO Test Center 2",
          "subjectName": "CN=esia test3",
```

```
          "issuerName": "E=support@cryptopro.ru\r\nC=RU\r\nL=Moscow\r\nO=CRYPTO-PRO
LLC\r\nCN=CRYPTO-PRO Test Center 2",
          "notAfter": "2024-10-26T16:25:37.000Z",
          "notBefore": "2024-08-26T16:15:37.000Z",
          "serialNumber": "120064ECE1352507757C00A08B00020064ECE1",
          "thumbprint": "29063cd83537dbbd0a221705859b8f6e5bec5ad0",
          "signatureAlgorithm": "1.2.643.7.1.1.3.2",
          "signatureDigestAlgorithm": "1.2.643.7.1.1.2.2",
          "publicKeyAlgorithm": "1.2.643.7.1.1.1.1"
        }
      ],
      "isCertChainValid": true
    }
  ],
  "dataStamp": "JVBERi0xLjcKJ...=",
  "report": "JVBERi0xLjcKJ...="
}
```


Атрибуты ответа

Поле	Тип	Описание
status	`Integer`	HTTP статус ответа.
message	`String`	Сообщение об ошибке (заполняется, если `status` не является успешным).
isValid	`Boolean`	Общий результат проверки подписи (`true`/`false`).
isValidSign	`Boolean`	Результат математической корректности подписи (`true`/`false`).
signs	`Array`	Массив объектов с детальной информацией о сертификатах и подписях.
dataStamp	`String`	Подписанные данные с визуализацией штампов подписей в формате Base64 (для PDF документов).
report	`String`	PDF-отчет проверки в формате Base64.

Пример неуспешного ответа

```
HTTP/1.1 400 Bad Request
Vary: Origin
Access-Control-Allow-Credentials: true
Content-Type: application/json; charset=utf-8
Content-Length: 101
ETag: W/"65-ykHUL2wV5KLzsDntQL70C94heK8"
```

```
Date: Thu, 26 Sep 2024 12:42:04 GMT
Connection: keep-alive
Keep-Alive: timeout=5
{
    "message":
"Запрошена проверка прикрепленной CMS подписи, но подпись не содержит исходного документа",
    "error": "Bad Request",
    "statusCode": 400
}
```

Установка сертификатов

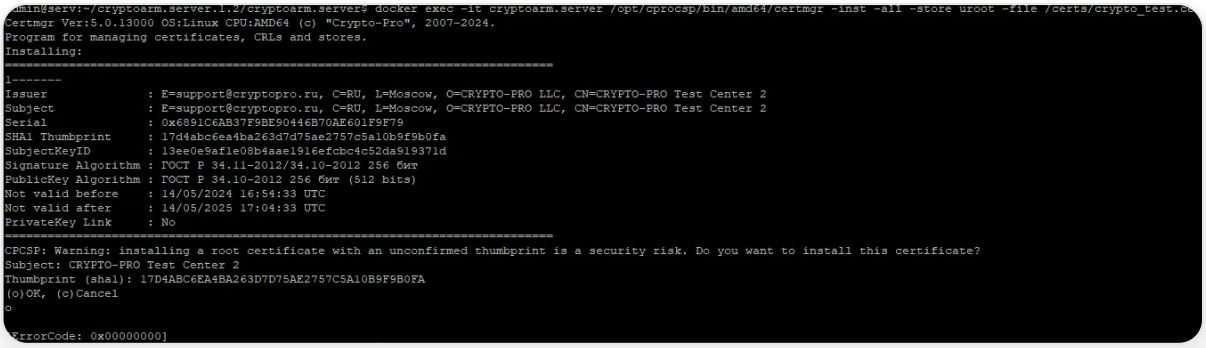
Для работы с подписью необходимо установить сертификаты.

Файлы сертификатов необходимо поместить в каталог: `cryptoarm-server/certs`.

Установка корневого сертификата

Для установки корневого сертификата `certs/crypto.root.test.cer` выполните команду:

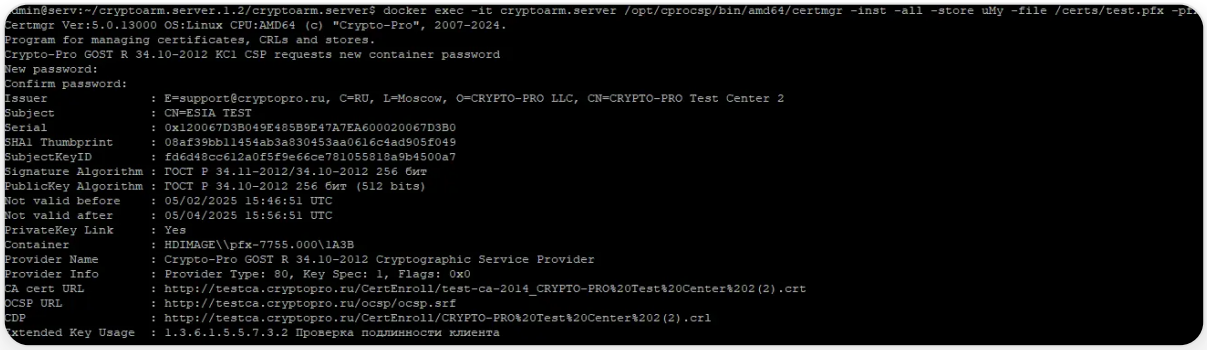
```
docker exec -it cryptoarm.server /opt/cprocsp/bin/amd64/certmgr -inst -all -store uroot -
file /certs/crypto.root.test.cer
```



Установка сертификата пользователя

Для установки пользовательского сертификата `certs/crypto.root.test.pfx` выполните команду:

```
docker exec -it cryptoarm.server /opt/cprocsp/bin/amd64/certmgr -inst -all -store uMy -file /
certs/crypto.root.test.pfx -pfx
```



Настройка отчета

Параметр `REPORT_INFO_MESSAGE` позволяет настроить текст в колонтитуле PDF-отчета.

Поддерживаются:

- переменные `{VARIABLE_NAME}`
- директивы выравнивания `{align:direction}`

Подробная информация приведена в файле:

`docs/VARIABLES_SUPPORT.md

В переменной окружения `REPORT_INFO_MESSAGE` можно настроить сообщение, которое будет отображаться в колонтитуле PDF-отчета. Поддерживается замена переменных в формате `{VARIABLE_NAME}` и выравнивание текста с помощью директив `{align:direction}`.

Подробная документация доступна в инструкции Настройка переменной REPORT_INFO_MESSAGE.

Подпись данных

```
curl -X 'POST' \
  'http://172.17.1.115:3037/cms/sign' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "cert": "MIIDEjCCAr+gAwI...==",
    "data": "dGVzdA==",
    "password": "password"
}'
```

Параметры

Параметр	Описание
`cert`	Сертификат в `Base64`
`data`	Подписываемые данные в `Base64`
`password`	Пароль от контейнера (если требуется)

Пример ответа

```
{
  "cms": "MIAGCSqGSIb3DQEHAq...AAAAAAAA"
}
```

Диагностика сетевых подключений

При проверке электронной подписи сервис может обращаться к внешним ресурсам:

- OCSP — проверка статуса сертификата
- CRL — списки отозванных сертификатов
- TSP — сервисы штампов времени

Если эти ресурсы недоступны из сети сервера, проверка подписи может выполняться некорректно или завершаться ошибкой.

Для диагностики сетевых обращений можно использовать прокси-сервисы.

Диагностика через squid-proxy

`Squid` позволяет увидеть, к каким внешним адресам обращается сервис.

Добавление proxy-сервиса

Добавьте сервис `squid-proxy` в `docker-compose.yml`:

```
services:
  cryptoarm.server:
    build: .
    image: cryptoarm.server
    container_name: cryptoarm.server
    restart: always
    ports:
      - 3037:3037
    env_file:
      - .env
    volumes:
      - ./cert_storage:/var/opt/cprocsp/
      - ./certs:/certs
```

```
  squid-proxy:
    image: ubuntu/squid
    container_name: squid-proxy
    restart: always
```

Настройка прокси

Пропишите адрес прокси в `.env`:

```
http_proxy=http://squid-proxy:3128
```

При необходимости можно также указать:

```
https_proxy=http://squid-proxy:3128
```

Перезапуск сервисов

Перезапустите сервисы:

```
docker compose up -d
```

Просмотр логов прокси

```
docker logs -f squid-proxy
```

В логах будут отображаться все внешние соединения сервиса.

Диагностика через mitmproxy

`mitmproxy` позволяет анализировать HTTP-запросы более детально через web-интерфейс.

Добавление mitmproxy

Добавьте сервис `mitmweb` в `docker-compose.yml`:

```
services:
  cryptoarm.server:
    build: .
    image: cryptoarm.server
    container_name: cryptoarm.server
    restart: always
    ports:
      - 3037:3037
    env_file:
      - .env
    volumes:
      - ./cert_storage:/var/opt/cprocsp/
      - ./certs:/certs

  mitmweb:
    image: mitmproxy/mitmproxy
```

```
    container_name: mitmweb
    ports:
      - 8081:8081   # web UI
      - 8080:8080   # proxy
    volumes:
      - ./logs:/logs
    command: >
      mitmweb
      --set web_password=admin
      --mode reverse:http://cryptoarm.server:3037
      --web-host 0.0.0.0
```

Запуск сервисов

Запустите/перезапустите сервисы:

```
docker compose up -d
```

Доступ к сервису

После запуска:

- API сервиса доступен через proxy: `http://SERVER:8080`
- web-интерфейс `mitmproxy`: `http://SERVER:8081`

Через web-интерфейс можно анализировать все HTTP-запросы сервиса.

Открыть статью на сайте

Авторизация API

В этой инструкции вы узнаете, как настроить авторизацию API в **КриптоАРМ Server**, ознакомитесь с доступными режимами (`none` и `apikey`), научитесь передавать API-ключи различными способами, а также получите практические примеры запросов и рекомендации по безопасной эксплуатации.

Содержание:

- Режимы авторизации
- Настройка авторизации
- Использование API
- Примеры запросов
- Рекомендации по безопасности
- Ошибки авторизации
- Использование API ключей в Swagger UI

Режимы авторизации

Сервис поддерживает два режима авторизации для доступа к REST API:

Режим	Описание
`none`	Авторизация отключена. Все запросы к API выполняются без проверки
`apikey`	Доступ к API разрешен только при наличии корректного API ключа

Выбор режима определяется переменной окружения `AUTH_MODE`.

Настройка авторизации

Переменные окружения

```
# Режим авторизации: "none" или "apikey"
AUTH_MODE=none

# Список API ключей через запятую (используется только при AUTH_MODE=apikey)
API_KEYS=your-api-key-1,your-api-key-2,your-api-key-3
```

Параметры

Переменная	Описание
`AUTH_MODE`	режим авторизации (`none` или `apikey`)
`API_KEYS`	список допустимых API ключей через запятую

Примеры конфигурации

Авторизация отключена

```
AUTH_MODE=none
```

В этом режиме все запросы выполняются без проверки ключей.

Авторизация по API ключу

```
AUTH_MODE=apikey
API_KEYS=abc123def456,xyz789uvw012,test-key-123
```

В этом режиме каждый запрос к API должен содержать корректный API ключ.

Использование API

API ключ можно передать несколькими способами.

Заголовок `X-API-Key` (рекомендуется)

```
curl -H "X-API-Key: your-api-key-1" http://localhost:3037/api/endpoint
```

Этот способ является предпочтительным при интеграции с внешними системами.

Заголовок `Authorization: Bearer`

```
curl -H "Authorization: Bearer your-api-key-1" http://localhost:3037/api/endpoint
```

Подходит для клиентов, использующих стандартный механизм Bearer-токенов.

Query-параметр

```
curl "http://localhost:3037/api/endpoint?apiKey=your-api-key-1"
```

Этот способ можно использовать для тестирования, однако его не рекомендуется применять в production-среде.

Примеры запросов

JavaScript/Node.js

```
// Используя fetch
const response = await fetch('http://localhost:3037/api/endpoint', {
  headers: {
    'X-API-Key': 'your-api-key-1'
  }
});

// Используя axios
const response = await axios.get('http://localhost:3037/api/endpoint', {
  headers: {
    'X-API-Key': 'your-api-key-1'
  }
});
```

Python

```
import requests

headers = {
    'X-API-Key': 'your-api-key-1'
}

response = requests.get('http://localhost:3037/api/endpoint', headers=headers)
```

Рекомендации по безопасности

При использовании API ключей рекомендуется соблюдать следующие меры безопасности:

- используйте **длинные и случайные ключи**;
- не передавайте API ключи в URL параметрах;
- не сохраняйте ключи в публичных репозиториях;
- регулярно выполняйте **ротацию ключей**;
- используйте **HTTPS** при работе в production-среде.

Ошибки авторизации

Код	Описание
`401 Unauthorized`	API ключ отсутствует или указан неверно
`403 Forbidden`	API ключ недействителен или не имеет доступа

Использование API ключей в Swagger UI

Swagger UI автоматически поддерживает авторизацию API ключами.

Если включен режим:

```
AUTH_MODE=apikey
```

в интерфейсе Swagger отображается кнопка **Authorize**.

Чтобы авторизоваться:

1. Нажмите **Authorize** в правом верхнем углу интерфейса.
2. Введите API ключ в поле **X-API-Key** или **Bearer**.
3. Нажмите **Authorize**.

После этого Swagger будет автоматически добавлять ключ ко всем API-запросам.

Если режим авторизации отключен (`AUTH_MODE=none`) или API ключи не настроены, кнопка **Authorize** в Swagger UI не отображается.

Открыть статью на сайте

API для работы с файлами (PAdES и CMS)

В этой инструкции вы узнаете, как интегрировать функции электронной подписи в ваши приложения с помощью **API КриптоАРМ Server**. Мы подробно разберем форматы **PAdES** для PDF-документов и **CMS/CAdES** для произвольных файлов, рассмотрим оба варианта взаимодействия — через загрузку файлов и через JSON (`Base64`), а также приведем практические примеры запросов для подписания, проверки и улучшения подписей.

Содержание:

- Обзор
- PAdES API
- CMS API
- JSON API (Base64)
- Примеры использования
- Группировка методов в Swagger
- Поддерживаемые форматы файлов

> 💡 Полную спецификацию API можно скачать или импортировать в Swagger UI: swagger.yml

---

Обзор

КриптоАРМ Server предоставляет набор API-методов для работы с электронными подписями.

Поддерживаются два формата подписи:

- **PAdES** — электронная подпись PDF-документов
- **CMS / CAdES** — универсальная подпись произвольных файлов

API предоставляет два варианта работы:

Тип API	Назначение
Файловый API	загрузка файлов через `multipart/form-data`
JSON API	передача данных в формате `Base64`

Файловые методы обычно используются при интеграции с веб-формами и внешними системами.

PAdES API

1. Подписание PDF файла

POST

```
/pades/sign-file
```

Метод загружает PDF-документ и создает подпись **PAdES** с визуализацией штампа подписи.

Параметры запроса (multipart/form-data)

Параметр	Тип	Обязательный	Описание
`pdf`	file	да	PDF файл для подписания
`cert`	file	нет	сертификат (`.cer`, `.p7b`, `.crt`)
`password`	string	нет	пароль контейнера ключа
`params`	string	нет	JSON с параметрами штампа

Пример параметров штампа:

```
{
  "stampMargin": 80,
  "stampColor": "#4c40d2",
  "stampScale": 1
}
```

Ответ

Метод возвращает файл:

```
signed.pdf
```

2. Проверка подписи PDF файла

POST

```
/pades/verify-file
```

Метод проверяет подписи в PDF документе.

Параметры запроса (multipart/form-data)

Параметр	Тип	Обязательный	Описание
`pdf`	file	да	PDF файл
`meta`	string	нет	дополнительные параметры проверки

Пример ответа

```
{
  "status": 200,
  "message": "Все подписи действительны",
  "isValid": true,
  "isValidSign": true,
  "signs": [...]
}
```

CMS API

1. Создание подписи файла

POST

```
/cms/sign-file
```

Метод создает CMS подпись для переданного файла.

Параметры запроса (multipart/form-data)

Формат: `multipart/form-data`

Параметр	Тип	Обязательный	Описание
`data`	file	да	файл для подписания
`cert`	file	нет	сертификат (`.cer`, `.p7b`, `.crt`)
`password`	string	нет	пароль контейнера ключа

Ответ

Метод возвращает файл подписи:

```
signed.sig
```

2. Проверка подписи файла

POST

```
/cms/verify-file
```

Метод проверяет подпись CMS.

Параметры запроса (multipart/form-data)

Параметр	Тип	Обязательный	Описание
`cms`	file	да	файл подписи
`data`	file	нет	файл данных (для откреплённой подписи)
`enhancedType`	string	нет	тип усовершенствования подписи
`meta`	string	нет	дополнительные параметры

Ответ

```
{
  "status": 200,
  "message": "Все подписи действительны",
  "isValid": true,
  "isValidSign": true,
  "signs": [...]
}
```

3. Улучшение подписи файла

POST

```
/cms/enhance-file
```

Метод улучшает существующую CMS подпись.

Параметры запроса (multipart/form-data):

Параметр	Тип	Обязательный	Описание
`cms`	file	да	файл подписи
`data`	file	да	файл данных
`enhancedType`	string	да	тип улучшения подписи

Поддерживаемые типы улучшения

Тип	Описание
`CAdES-T`	подпись с временной меткой
`CAdES-X Long Type 1`	долгосрочная подпись
`CAdES-A`	архивная подпись

Ответ

Метод возвращает файл:

```
enhanced.sig
```

JSON API (Base64)

Помимо файловых методов, сервис поддерживает API с передачей данных в `Base64`.

PAdES-методы

Метод	Назначение
`POST /pades/sign`	подписание PDF
`POST /pades/verify`	проверка подписи

CMS методы

Метод	Назначение
`POST /cms/sign`	создание подписи
`POST /cms/verify`	проверка подписи
`POST /cms/enhance`	улучшение подписи

Примеры использования

cURL для подписания PDF файла

```
curl -X POST "http://localhost:3037/pades/sign-file" \
  -H "Content-Type: multipart/form-data" \
  -F "pdf=@document.pdf" \
  -F "cert=@certificate.cer" \
  -F "password=your_password" \
```

```
  -F 'params={"stampMargin": 80, "stampColor": "#4c40d2", "stampScale": 1}' \
  --output signed.pdf
```

cURL для проверки PDF файла

```
curl -X POST "http://localhost:3037/pades/verify-file" \
  -H "Content-Type: multipart/form-data" \
  -F "pdf=@signed_document.pdf"
```

cURL для создания CMS подписи

```
curl -X POST "http://localhost:3037/cms/sign-file" \
  -H "Content-Type: multipart/form-data" \
  -F "data=@document.pdf" \
  -F "cert=@certificate.cer" \
  -F "password=your_password" \
  --output signed.sig
```

cURL для проверки CMS подписи

```
curl -X POST "http://localhost:3037/cms/verify-file" \
  -H "Content-Type: multipart/form-data" \
  -F "cms=@signature.p7b" \
  -F "data=@document.pdf"
```

cURL для улучшения CMS подписи

```
curl -X POST "http://localhost:3037/cms/enhance-file" \
  -H "Content-Type: multipart/form-data" \
  -F "cms=@signature.p7b" \
  -F "data=@document.pdf" \
  -F "enhancedType=CAdES-T" \
  --output enhanced.sig
```

Группировка методов в Swagger

Методы API сгруппированы по тегам:

Тег	Описание
PAdES	работа с подписями PDF
CMS	работа с CMS / CAdES подписями

Это упрощает навигацию по методам в Swagger UI.

Поддерживаемые форматы файлов

Поддерживаемые форматы сертификатов

Формат	Описание
`.cer`	X.509 сертификат
`.crt`	сертификат X.509
`.p7b`	PKCS#7 контейнер сертификатов

Поддерживаемые форматы данных

Поддерживаются любые файлы для создания подписи.

Наиболее распространенные:

- `.pdf`
- `.xml`
- `.docx`
- `.xlsx`

Форматы результирующих файлов

Расширение	Описание
`.pdf`	PDF документ с подписью PAdES
`.sig`	CMS / CAdES подпись

Открыть статью на сайте

Настройка переменной REPORT_INFO_MESSAGE

В этой инструкции вы узнаете, как с помощью переменной окружения `REPORT_INFO_MESSAGE` гибко настроить служебную информацию в PDF-отчетах: добавлять динамические данные (номера страниц, названия компаний), управлять форматированием, выравниванием и отступами текста, а также комбинировать несколько элементов в одной строке для создания профессиональных документов.

Содержание:

- Описание
- Синтаксис
- Поддерживаемые переменные
- Разделение текста на одной строке
- Приоритет замены переменных

Описание

Переменная окружения `REPORT_INFO_MESSAGE` позволяет настроить текст служебной информации, отображаемой в PDF-отчётах.

В строке поддерживаются:

- подстановка переменных
- выравнивание текста
- отступы
- HTML-разметка

Синтаксис

Переменные подставляются в формате:

```
{VARIABLE_NAME}
```

Например:

```
Страница {pageNumber} из {totalPages}
```

Каждая строка может содержать директивы форматирования.

Поддерживаемые переменные

Специальные переменные

Переменная	Описание
`{pageNumber}`	номер текущей страницы
`{totalPages}`	общее количество страниц

Переменные окружения

Любая переменная окружения может быть использована в формате `{VARIABLE_NAME}`.

Предустановленные переменные

Переменная	Значение по умолчанию
`COMPANY_NAME`	ООО Цифровые технологии
`WEBSITE_URL`	www.cryptoarm.ru

Переменные для настройки PDF-отчетов

Переменная	Описание	Значение по умолчанию
`REPORT_TITLE`	заголовок PDF-отчёта	`Отчет о проверке электронной подписи`
`DATA_STAMP_TITLE`	текст штампа на подписанном документе	`Документ подписан электронной подписью`
`STAMP_COLOR`	цвет штампа (HEX)	`#4c40d2`

Переменные для настройки TSP-сервиса

Для работы расширенных подписей используется сервис штампов времени.

Переменная	Описание	Значение по умолчанию
`TSP_SERVICE_ADDRESS`	адрес сервиса TSP	`http://qs.cryptopro.ru/tsp/tsp.srf`

Директивы форматирования

Выравнивание

Директива	Описание
`{align:left}`	выравнивание по левому краю (по умолчанию)
`{align:center}`	выравнивание по центру
`{align:right}`	выравнивание по правому краю
`{align:justify}`	выравнивание по ширине (пока работает как `left`)

Директива должна располагаться **в начале строки**.

Отступы

Можно задавать отступы от краёв страницы.

Директива	Описание
`{left-margin:N}`	отступ слева (в пикселях)
`{right-margin:N}`	отступ справа (в пикселях)

Директивы отступов можно комбинировать с директивами выравнивания на одной строке.

Разделение текста на одной строке

Для размещения текста слева и справа в одной строке используется разделитель:

```
||
```

Пример:

```
REPORT_INFO_MESSAGE="{left-margin:50}{align:left}Текст слева || {right-margin:50}{align:right}
Текст справа"
```

Это создаст эффект: `Текст слева Текст справа`

Примеры использования

Базовый пример

```
REPORT_INFO_MESSAGE="© Цифровые технологии
Страница {pageNumber} из {totalPages}
www.cryptoarm.ru"
```

С использованием переменных окружения

```
# Установка переменных окружения (опционально)
COMPANY_NAME="ООО Цифровые технологии"
WEBSITE_URL="www.cryptoarm.ru"

# Использование в REPORT_INFO_MESSAGE
REPORT_INFO_MESSAGE="© {COMPANY_NAME}
Страница {pageNumber} из {totalPages}
{WEBSITE_URL}"
```

> 💡 **Примечание:** Переменные `COMPANY_NAME` и `WEBSITE_URL` имеют значения по умолчанию, поэтому их установка необязательна.

Настройка PDF-отчетов

```
# Настройка заголовков и цветов
REPORT_TITLE="Отчет о проверке подписи"
DATA_STAMP_TITLE="Документ подписан"
STAMP_COLOR="#0066cc"

# Использование в REPORT_INFO_MESSAGE
REPORT_INFO_MESSAGE="© {COMPANY_NAME}\nСтраница {pageNumber} из {totalPages}\n{WEBSITE_URL}"

# Настройка TSP-сервиса
TSP_SERVICE_ADDRESS="http://your-tsp-server.com/tsp/tsp.srf"
```

С выравниванием текста

```
REPORT_INFO_MESSAGE="{align:center}© Цифровые технологии
{align:left}Страница {pageNumber} из {totalPages}
{align:right}www.cryptoarm.ru"
```

С отступами и выравниванием

```
# Разделенные строки
REPORT_INFO_MESSAGE="{align:center}© Цифровые технологии
{left-margin:50}{align:left}© ООО Цифровые технологии
{right-margin:50}{align:right}Страница {pageNumber} из {totalPages}"

# Одна строка с разделителем ||
REPORT_INFO_MESSAGE="{left-margin:50}{align:left}© ООО Цифровые технологии || {right-margin:
50}{align:right}Страница {pageNumber} из {totalPages}"
```

> 💡 **Примечание:** Директивы должны быть в начале строки. Для создания эффекта "текст слева, текст справа" используйте разделитель `||` в одной строке.

С дополнительной информацией и выравниванием

```
# Дополнительные переменные окружения
VERSION="1.0.0"
BUILD_DATE="2024-01-01"

# Использование в REPORT_INFO_MESSAGE
REPORT_INFO_MESSAGE="{align:center}© {COMPANY_NAME}
{align:left}Страница {pageNumber} из {totalPages}
Версия: {VERSION} | Сборка: {BUILD_DATE}
{align:right}{WEBSITE_URL}"
```

Приоритет замены переменных

Подстановка выполняется в следующем порядке:

1. Специальные переменные
2. Переменные окружения

Если переменная не найдена, текст остаётся без изменений.

HTML-поддержка

Переменные поддерживаются внутри HTML-тегов:

```
REPORT_INFO_MESSAGE="© {COMPANY_NAME}
Страница {pageNumber} из {totalPages}
<a href='{WEBSITE_URL}'>Сайт компании</a>"
```

Комбинирование выравнивания и HTML

Директивы выравнивания можно комбинировать с HTML-тегами:

```
REPORT_INFO_MESSAGE="{align:center}© {COMPANY_NAME}
{align:left}Страница {pageNumber} из {totalPages}
{align:right}<a href='{WEBSITE_URL}'>Сайт компании</a>"
```

Комбинирование отступов, выравнивания и HTML

Все директивы можно комбинировать:

```
REPORT_INFO_MESSAGE="{align:center}© {COMPANY_NAME}
{left-margin:50}{align:left}© {COMPANY_NAME}
{right-margin:50}{align:right}Страница {pageNumber} из {totalPages}
{align:center}<a href='{WEBSITE_URL}'>Сайт компании</a>"
```

Открыть статью на сайте

Варианты использования КриптоАРМ Server

КриптоАРМ Server может использоваться как серверная платформа для обработки электронных подписей и криптографических операций. Сервис предоставляет REST API и SDK, что позволяет интегрировать функции работы с электронной подписью в различные системы.

> 💡 **Примечание**: Ниже приведены демонстрационные сценарии, реализованные на демо-стенде Sign.kloud.one. Это не полный список всех возможностей **КриптоАРМ Server**. В реальных проектах функциональность может быть расширена или интегрирована иначе.

Содержание:

- Сервис проверки и улучшения электронных подписей
- Бот для проверки электронной подписи
- Сервис для настройки рабочего места
- Сервис проверки подлинности сертификата электронной подписи

Сервис проверки и улучшения электронных подписей

С помощью **КриптоАРМ Server** можно реализовать сервис проверки и улучшения электронных подписей, обеспечивающий надежную верификацию документов, автоматическую генерацию отчетов и преобразование подписей в долговременные форматы (CAdES-T, CAdES-X, CAdES-A).

1. Проверка подписи документа

> 📚 Примеры запросов описаны в статье Автоматизация проверки электронной подписи с использованием КриптоАРМ Server.

Типовой сценарий

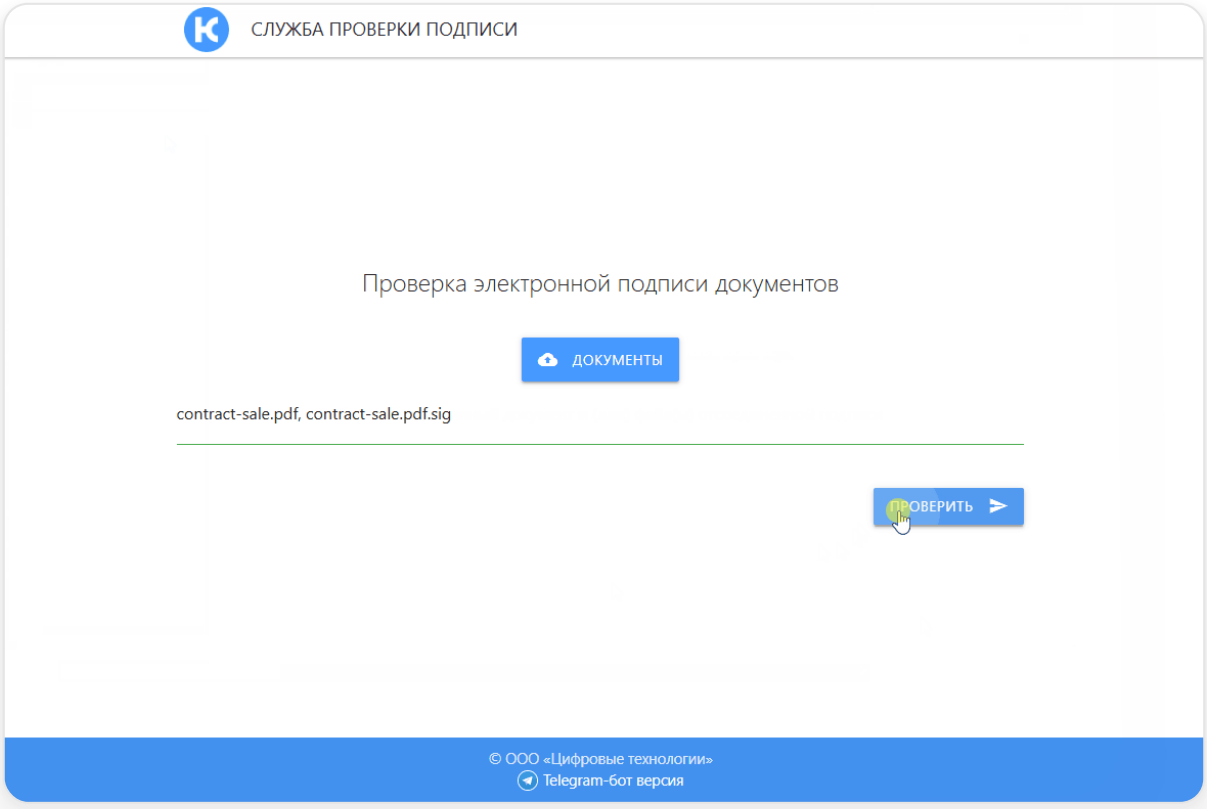
1. Загрузка документа:
    - Для присоединенной подписи - сам документ
    - Для отделенной подписи - документ и файл подписи
2. Автоматический анализ:
    - Определение типа и формата подписи

- Проверка целостности документа
- Верификация сертификата

3. Формирование результатов:

- Отображение статуса проверки (валидна/невалидна)
- Вывод технических параметров подписи и сведений о сертификате

Демонстрация работы **Sign.kloud.one**:



Статичный кадр демонстрации. Открыть анимацию

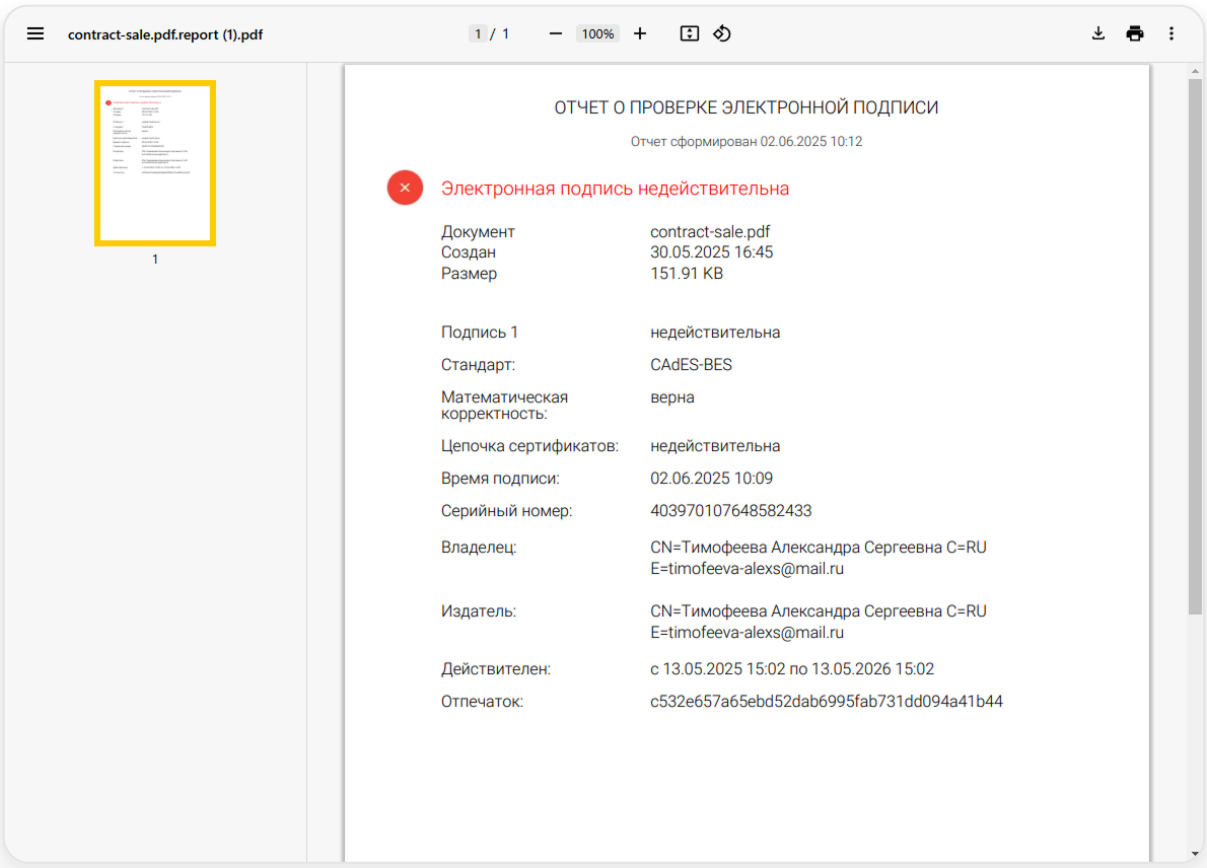
> 💡 При необходимости сертификат подписанта можно экспортировать для дальнейшего использования, например, для шифрования документов.

2. Создание отчета о проверке подписи

Решение автоматически формирует PDF-отчеты, содержащие:

- Технические параметры подписи
- Сведения о сертификате
- Результаты проверки цепочки доверия

Пример отчета:

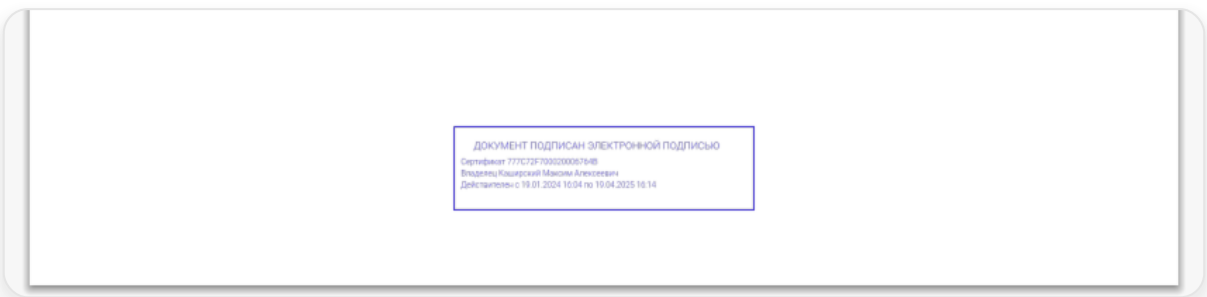


3. Создание печатной формы с визуальным штампом

Печатная форма - копия исходного документа с графическим штампом подписи. В такой копии присутствует только визуальное представление штампа, а сама электронная подпись отсутствует.

> 🚨 **Важно понимать!** Печатная форма предназначена для ознакомления, но не имеет юридической силы оригинала.
>
> 💡 **Ограничение**: В текущей версии поддерживается только для PDF-документов.

Пример печатной формы:



4. Улучшение подписи

Улучшение подписи — это процесс преобразования подписи в более защищенный и долговременный формат, например, CAdES-T, CAdES-X Long Type 1 или CAdES-A.

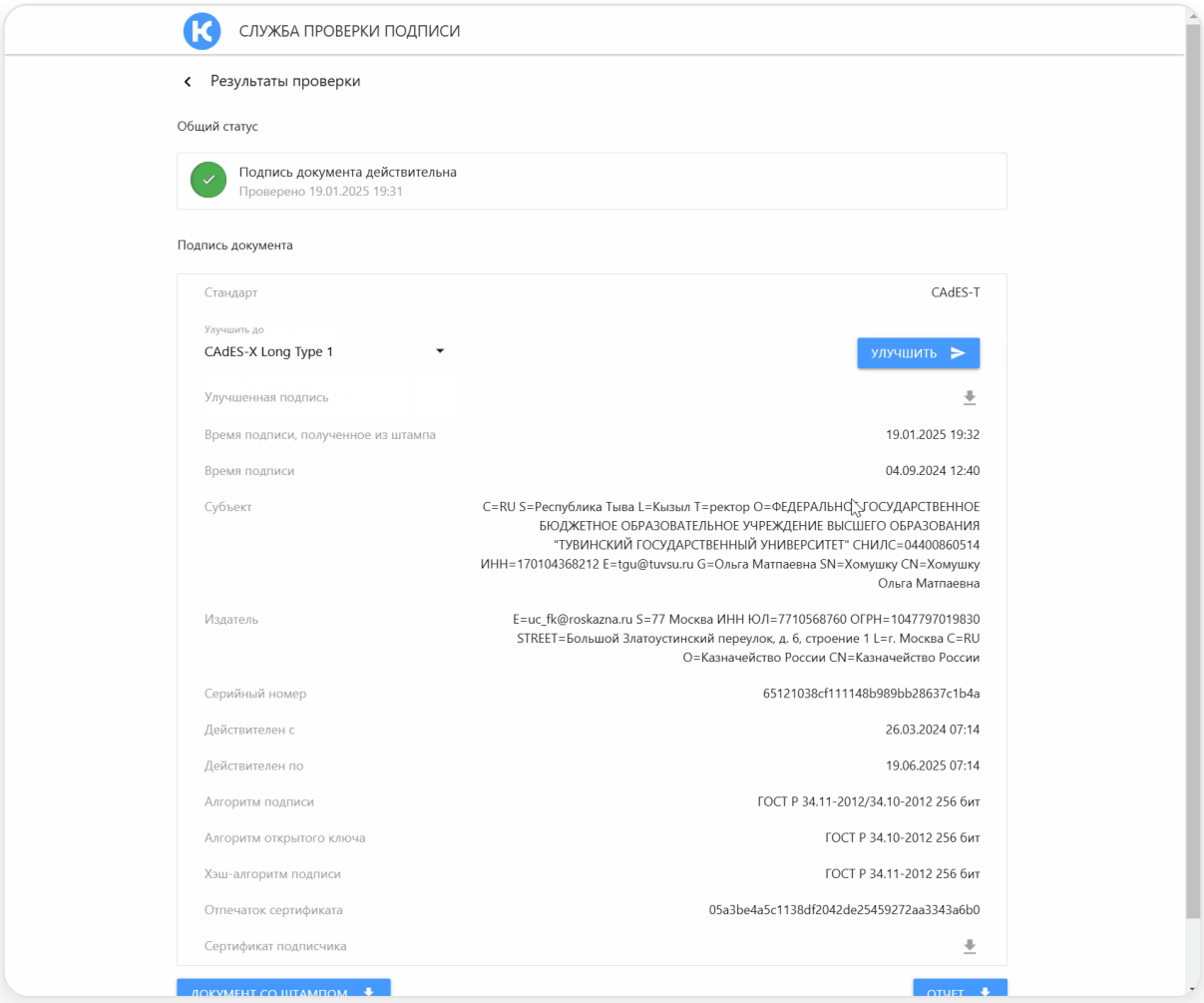
Улучшение подписи до более продвинутых форматов дает следующие преимущества:

- до **CAdES-T** (Подпись со штампом времени). Фиксируется момент подписания документа с привязкой к доверенному источнику времени.
- до **CAdES-X Long Type 1** (Долговременная подпись). Обеспечивает проверяемость подписи на протяжении 10 лет и более. Включает штамп времени, OCSP-штамп, подтверждающий действительность сертификата на момент подписи, и полную цепочку доверия.
- до **CAdES-A** (Архивная подпись). Разработана для долговременного хранения документов. Содержит специальные атрибуты, которые придают юридическую силу подписи на срок более 15 лет. Предусматривает добавление новых архивных меток в процессе хранения.

Типовой сценарий

1. Выбор целевого формата.
2. Добавление необходимых атрибутов:
    - Метки времени
    - Данные о статусе сертификата
    - Архивные атрибуты
3. Сохранение улучшенного документа

Демонстрация работы Sign.kloud.one



Статичный кадр демонстрации. Открыть анимацию

5. Объединение подписи

Функция **Объединение подписей** позволяет объединять несколько отсоединенных подписей (CMS/CAdES) в один файл подписи. Это удобно, когда:

- Необходимо создать единый файл подписи для документа, подписанного разными участниками;
- Требуется оптимизировать хранение подписей;
- Нужно подготовить файл для последующей проверки или архивации.

> 💡 **Примечание**: Объединенные подписи сохраняются отдельно и не изменяют исходные файлы.

Типовой сценарий

1. Выбор исходных файлов с открепёнными подписями.

2. Сервис загружает выбранные подписи.
3. Объединение подписей в один файл CMS/CAdES.
4. Сохранение нового файла на диск или отправка через API.
5. При необходимости — последующая проверка или улучшение объединённой подписи (CAdES-T, CAdES-X, CAdES-A).

Бот для проверки электронной подписи

Бот с интеграцией КриптоАРМ Server — это современное решение для автоматизированной проверки электронных подписей прямо в мессенджере.

Вы создаете собственного бота и подключаете его к **КриптоАРМ Server**, получая мощный инструмент для проверки ЭП без сложных интерфейсов.

Типовой сценарий

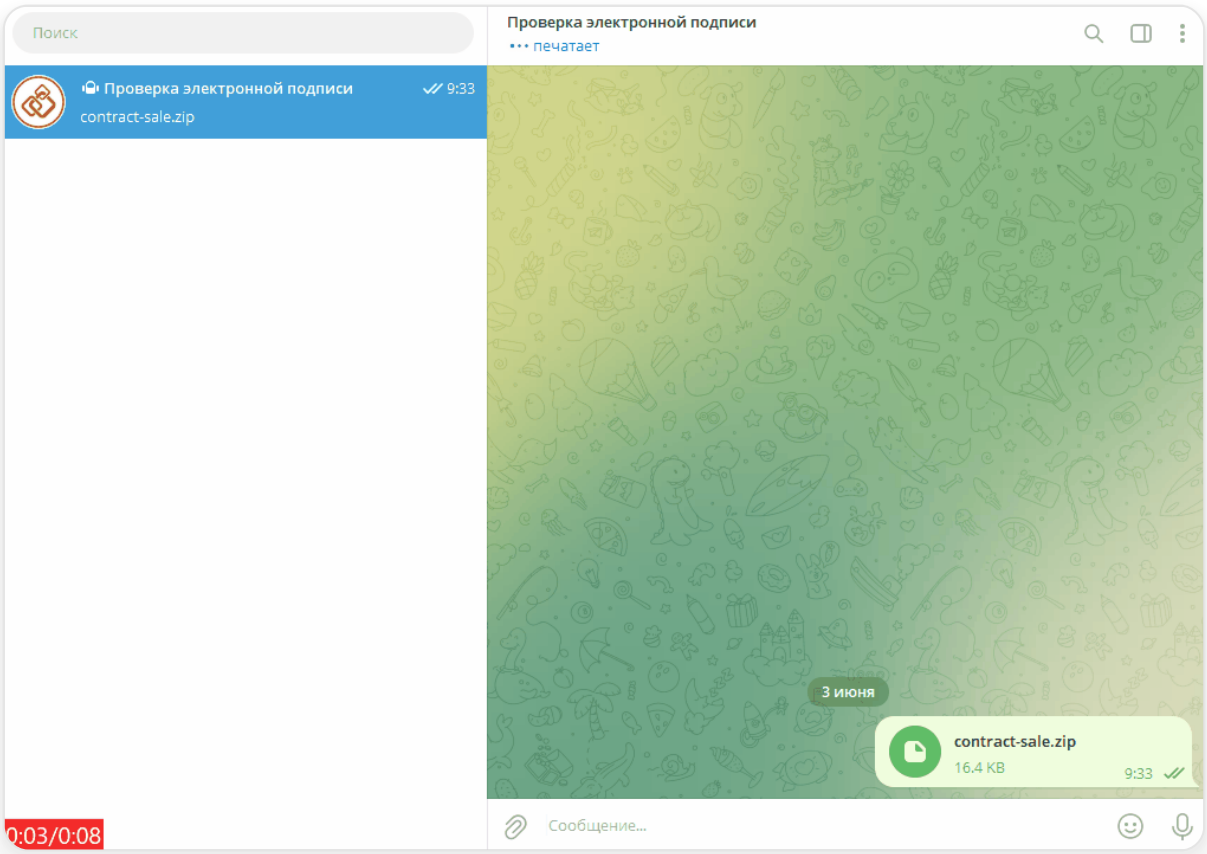
1. Начало работы:
    - Найдите бота через поиск и запустите его.
    - Нажмите кнопку **СТАРТ**, чтобы активировать.
2. Отправка документов:
    - Если подпись отделенная, то отправьте два файла: исходный документ и файл подписи.
    - Если подпись прикрепленная, загрузите файл с расширением `.sig`, `.sgn`, `.p7s`.
    - Если подпись встроенная (формат PAdES), загрузите PDF-документ.

> 💡**Альтернативный способ**: Можно загрузить в бот zip-архив, содержащий исходный документ и файл подписи.

3. Получение результатов:

   КриптоАРМ Server автоматически проверит:
    - Действительность сертификата ЭП, отозван он либо нет.
    - Соответствие подписи документу, не были ли изменения после подписания.
    - Наличие и валидность штампов времени (для CAdES-T, CAdES-A).

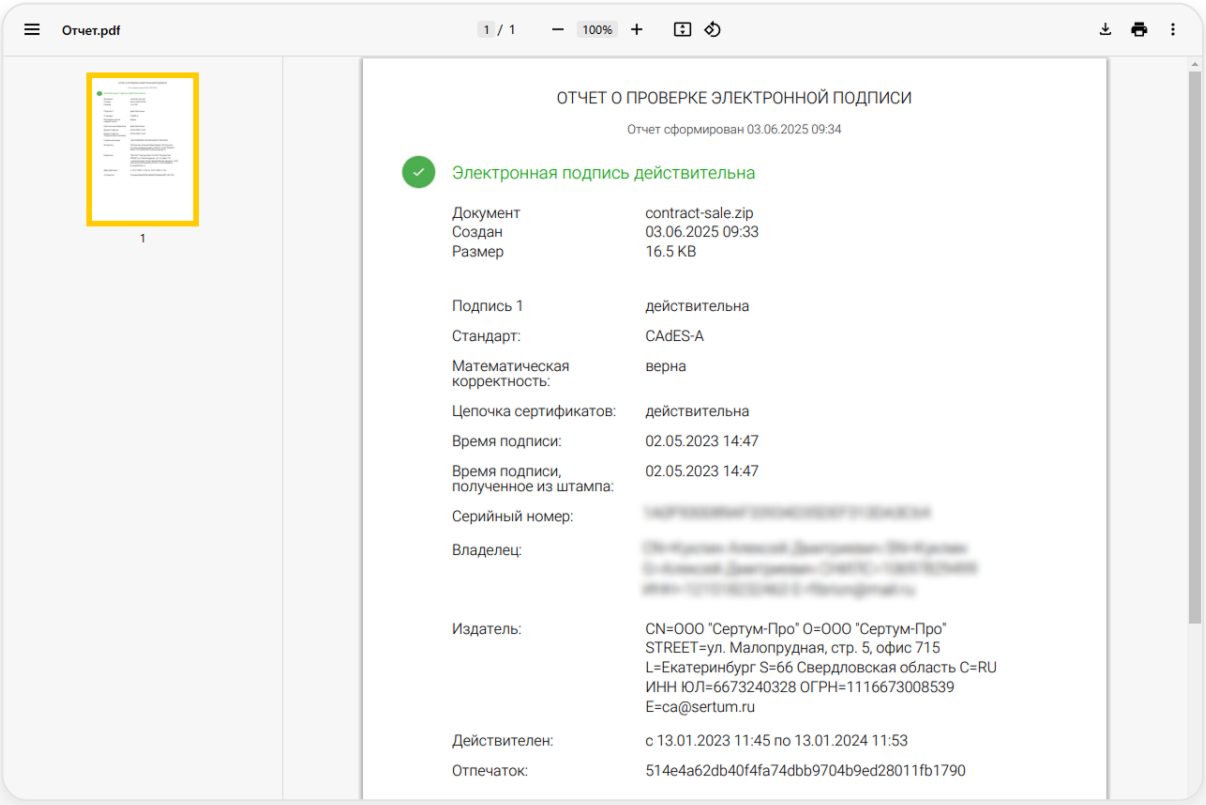
Интерфейс @SignKloudOne_bot



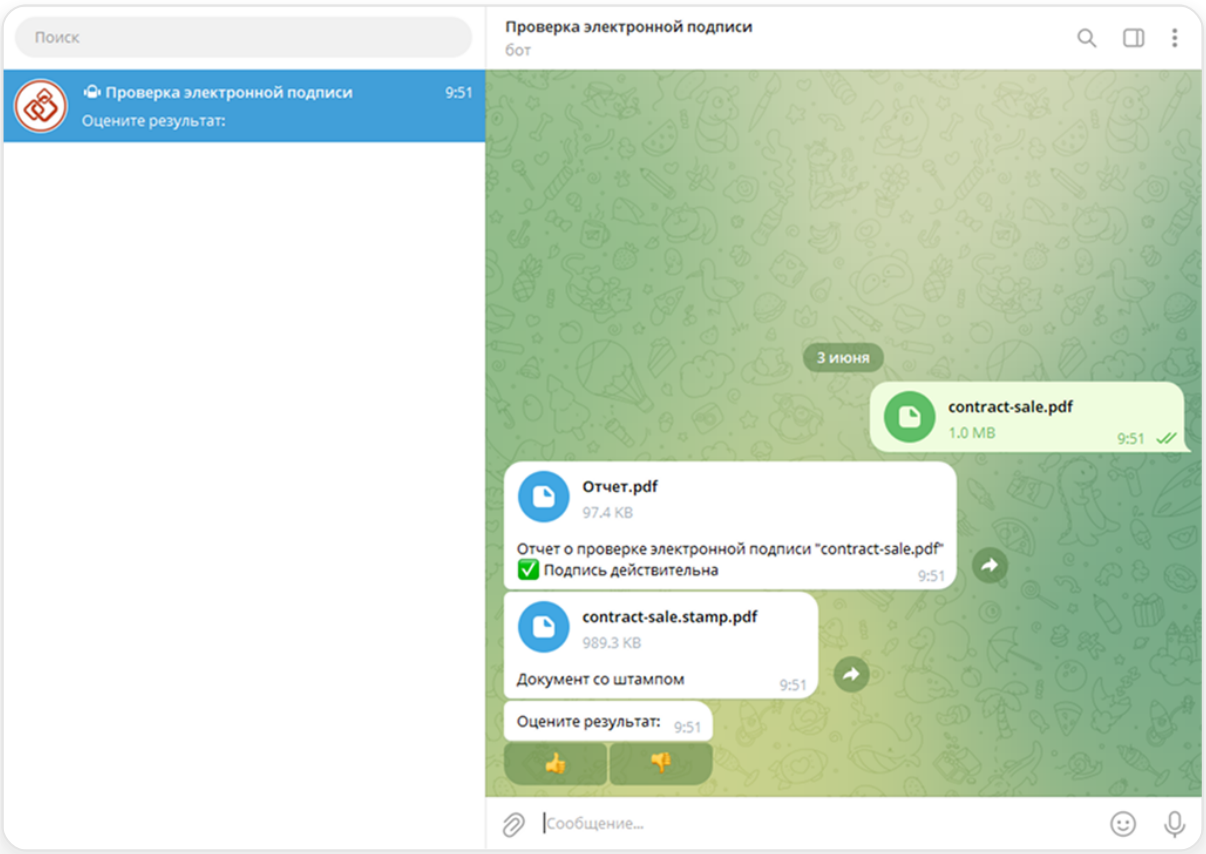
Статичный кадр демонстрации. Открыть анимацию

Результаты проверки в @SignKloudOne_bot

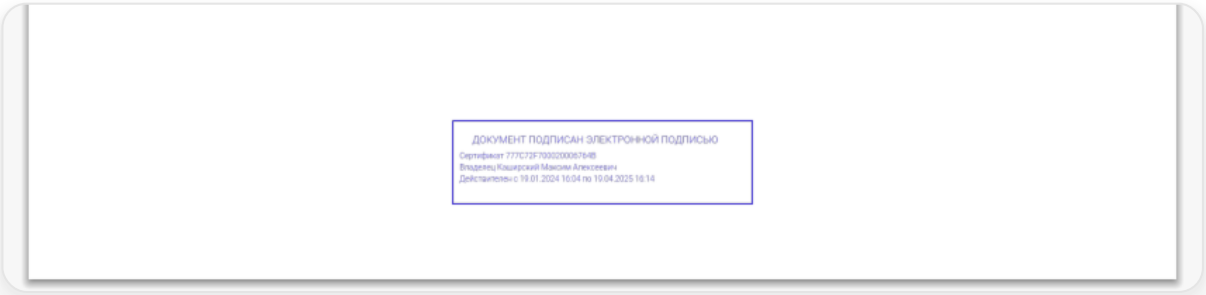
1. Общий статус подписи:
    - ✅ Подпись действительна
    - ❌ Подпись недействительна
2. PDF-отчет с детализацией



3. Для PDF-документов создается печатная форма с визуальным отображением подписи:



Пример визуализации:



Сервис для настройки рабочего места

С помощью **КриптоАРМ Server** можно реализовать сервис для проверки рабочего места. Это проверка позволяет убедиться, что рабочее место пользователя готово для работы с электронной подписью.

Решение предлагает два варианта использования:

- Разработка собственного веб-интерфейса с использованием SDK
- Использование готового сервиса

> 🚀 Готовый сервис: sign.kloud.one

Основные возможности

Доступно выполнение проверок **в разрезе юрисдикции**:

Российская Федерация (63-ФЗ «Об электронной подписи», ГОСТ):

1. Проверка установки и версий **КриптоАРМ** и **КриптоПро CSP**.
2. Проверка действующих лицензий **КриптоАРМ** и **КриптоПро CSP**.
3. Проверка наличия личного сертификата с алгоритмом ГОСТ
4. Предоставление ссылки на скачивание дистрибутива **КриптоАРМ**.
5. Поддержка авторизации через КриптоАРМ ID для журналирования загрузок.

Открытая криптография (RSA/ECDSA, DES/AES):

1. Проверка установки и версии **КриптоАРМ**.
2. Проверка действующей лицензии **КриптоАРМ**.
3. Проверка наличия личного сертификата с алгоритмом **RSA/ECDSA**.
4. Предоставление ссылки на скачивание дистрибутива **КриптоАРМ**.
5. Поддержка авторизации через КриптоАРМ ID для журналирования загрузок.

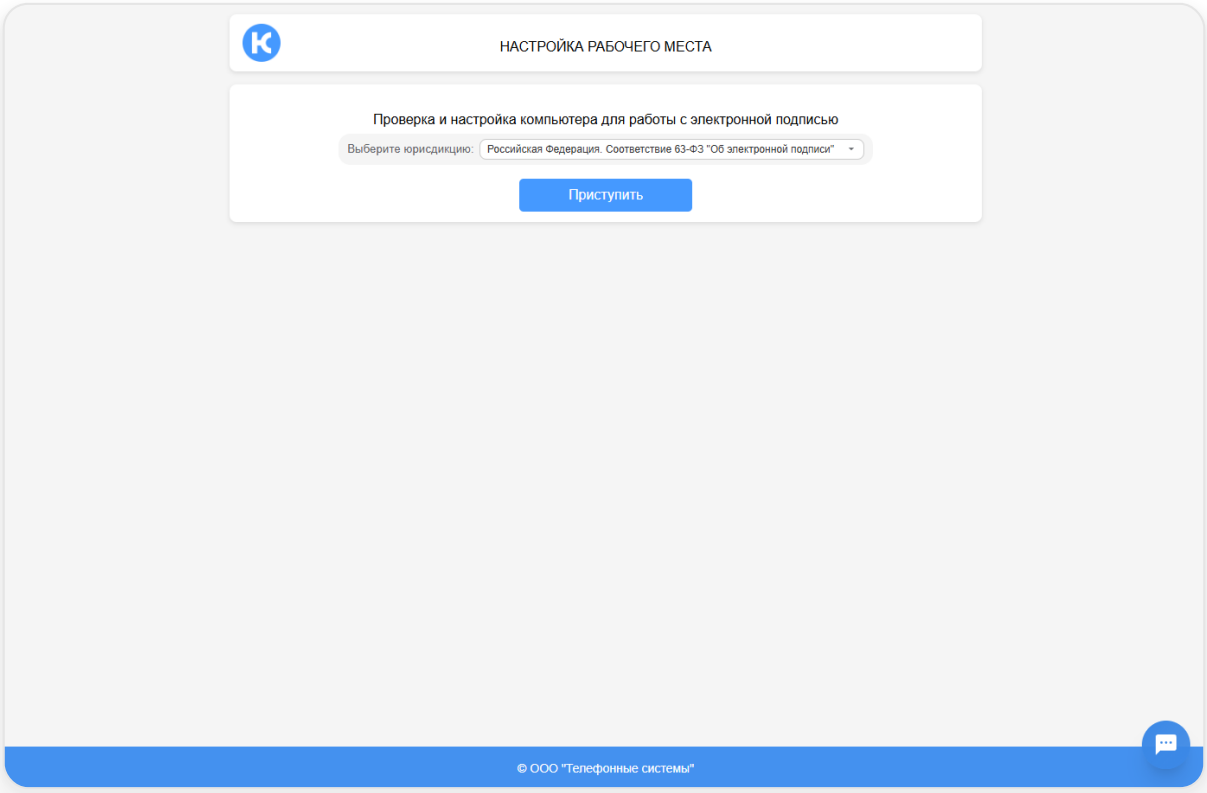
Типовой сценарий

1. Запустите проверку рабочего места.

> 💡 Для проведения проверки требуется установленное приложение КриптоАРМ. В случае его отсутствия, сервис предложит его скачивание.

2. Предоставьте разрешение на открытие приложения в окне браузера.
3. Подтвердите запуск диагностики в окне приложение КриптоАРМ.
4. Дождитесь окончания проверки.

Демонстрация работы Sign.kloud.one



В результате проверки сервис Sign.kloud.one отображает:

1. Общий статус:
   - ✅ Ваше рабочее место может быть использовано для формирования и проверки электронной подписи.
   - ❌ Необходимо установить недостающие компоненты.
2. Статус по каждому из проверяемых компонентов.

Сервис проверки подлинности сертификата электронной подписи

С помощью **КриптоАРМ Server** можно реализовать сервис проверки подлинности сертификата электронной подписи. Такой сервис позволяет:

- Проверить статус сертификата (действителен, отозван, просрочен);
- Убедиться, что сертификат выпущен аккредитованным удостоверяющим центром;
- Получить информацию о владельце и атрибутах сертификата;
- Обеспечить автоматическую верификацию сертификатов для корпоративных систем, порталов и электронного документооборота.

> 💡 **Примечание**: Проверка сертификата может выполняться как по отдельному файлу сертификата, так и в составе CMS/CAdES подписи.

Типовой сценарий

1. **Выбор сертификата**: Загружается файл сертификата `.cer`, `.crt`, `.p7b` или извлекается из электронного документа с подписью.
2. **Запрос на проверку**: Сертификат передаётся на сервер через API или веб-интерфейс.
3. **Анализ и верификация**
    - Проверка **цепочки доверия** до корневого сертификата удостоверяющего центра.
    - Проверка **актуальности сертификата** (не отозван, не просрочен).
    - Определение алгоритмов подписи и хэширования.
4. **Формирование отчета**
    - Статус сертификата: ✅ Действителен / ❌ Недействителен / ⚠ Проблемы с цепочкой доверия.
    - Данные владельца и издателя сертификата.
    - В случае недействительности — причина отказа.

Открыть статью на сайте

Часто задаваемые вопросы по КриптоАРМ Server

Для чего нужен КриптоАРМ Server

КриптоАРМ Server — программный продукт, предназначенный для интеграции и автоматизации криптографических операций в информационных системах организаций. Он включает в себя набор криптографических библиотек и средства разработки (SDK). Подробнее читайте в статье О продукте.

Какие криптографические операции поддерживает КриптоАРМ Server

- Формирование электронной подписи, проверка подписи, шифрование и расшифрование, сведения о сертификатах, генерация ключевой пары и создание запросов на получение сертификата, подключение защищенных носителей.
- Подпись и проверка электронной подписи встроенной в pdf, разметка областей для оттисков подписи, добавление графических штампов с информацией о подписи, пакетная подпись, сертификация подписанных документов с защитой от изменений, конвертация pdf-файлов в формат pdf-a/2b.
- Сбор, разбор и отправка электронных писем, поддержка IMAP, SMTP, POP3, S/MIME, контакты и адресные книги vCard, CardDAV.
- Обеспечение защищенного соединения к веб-серверам с поддержкой ГОСТ TLS.
- Проверка электронной подписи формата XMLDSig (XML Digital Signature).

Как интегрировать КриптоАРМ Server в свою систему

КриптоАРМ Server предлагает два инструмента для интеграции и автоматизации: API и SDK. Оба варианта упрощают автоматизацию криптографических операций, но имеют разные области применения.

Где посмотреть документацию и примеры для API или SDK?

1. Документация КриптоАРМ API и КриптоАРМ SDK.
2. Ссылка на видеодемонстрацию работы API от наших партнеров.

Мы готовы помочь вам с интеграцией API и SDK в вашу систему, по всем вопросам обращайтесь на sales@cryptoarm.ru.

Требуется ли оценка влияния?

Для использования КриптоАРМ Server в некоторых сценариях может потребоваться проведение исследований по оценке влияния (СФ).

В КриптоАРМ Server возможны следующие варианты применения:

1. Использование графических интерфейсов СКЗИ. В данном варианте использования СКЗИ дополнительных исследований **не требуется**.
2. Программный вызов КриптоАРМ API. В данном варианте использования СКЗИ **требуется проведение исследований** по оценке влияния СФ на СКЗИ.
3. Использование КриптоАРМ SDK рассматривается как встраивание СКЗИ во вновь разрабатываемое или существующее прикладное программное обеспечение, программно-аппаратные и аппаратные решения. В данном случае **требуется проведение тематических исследований** СФ со встроенным СКЗИ как самостоятельного шифровального (криптографического) средства.

Если вам требуется помощь в оценке влияния, оставьте заявку на sales@cryptoarm.ru.

Какие лицензии нужны

Для использования КриптоАРМ SDK необходима серверная лицензия.

КриптоАРМ API может работать как с клиентскими, так и серверными лицензиями на КриптоАРМ. С актуальными ценами можно ознакомиться в интернет-магазине cryptoarm.ru и в прайс-листе.

Для API будет доступен дополнительный вариант лицензирования — транзакционный. В этом случае оператор информационной системы приобретает пакет лицензий и распространяет своим клиентам клиентские лицензии вместе с параметрами операции. Пользователю не надо покупать КриптоАРМ или вручную вводить лицензии.

Также потребуется лицензия на криптопровайдер КриптоПро CSP.

Как интегрировать проверку подписи в свою систему

С помощью КриптоАРМ Server можно легко интегрировать проверку электронной подписи во внешние системы. КриптоАРМ предлагает два инструмента для интеграции и автоматизации: API и SDK. Оба варианта упрощают автоматизацию криптографических операций, но имеют разные области применения.

Можно ли создать протокол проверки подписи

С помощью КриптоАРМ Server можно создать протокол (отчет) проверки электронной подписи. Кроме того, для PDF-файлов доступно создание копии с визуальным штампом.

Можно ли создать копию документа с визуальным штампом

С помощью КриптоАРМ Server можно проверить электронную подпись и визуализировать штамп на PDF-документе.

Как интегрировать электронную подпись в свою систему

С помощью КриптоАРМ Server можно автоматизировать процессы подписания электронных документов. Для этого КриптоАРМ предлагает два инструмента: API и SDK. Оба варианта упрощают автоматизацию криптографических операций, но имеют разные области применения.

Какие стандарты подписи поддерживаются

В КриптоАРМ Server реализована поддержка стандартов CMS, CAdES, PAdES.

Можно ли улучшить электронную подпись с помощью КриптоАРМ Server

С помощью КриптоАРМ Server можно улучшить любую электронную подпись с формата CAdES-BES до CAdES-A.

Как защитить документы с помощью КриптоАРМ Server

С помощью КриптоАРМ Server можно автоматизировать процессы шифрования и расшифрования файлов.

Кроме того, с помощью КриптоАРМ Server (модуль trusted-mailcore2) можно реализовать интеграцию электронной почты в вашем проекте на Node.js.

Основные преимущества такой интеграции:

- Криптографическая защита писем: поддержка стандарта S/MIME с использованием ГОСТ-алгоритмов,
- Надежные TLS соединения: благодаря интеграции с cpcurl от КриптоПро обеспечена поддержка ГОСТ TLS для безопасных соединений с почтовыми серверами.
- Работа с сертифицированным криптопровайдером КриптоПро CSP.

Открыть статью на сайте

Глоссарий

- Сертификаты
- Подпись и шифрование
- Почта

Сертификаты

Сертификат электронной подписи

Электронный документ, который подтверждает владение ключом электронной подписи

Сертификат усиленной квалифицированной электронной подписи (УКЭП)

Выпускает только аккредитованный удостоверяющий центр. Документ, подписанный УКЭП, действителен для любой организации или ведомства.

Самоподписанный сертификат

Сертификат, изданный самим пользователем, без обращения к Удостоверяющему центру.

Открытый ключ электронной подписи

Ключ проверки электронной подписи. Содержит содержит уникальный код, с помощью которого можно проверить подлинность электронной подписи. Сертификат открытого ключа содержит сведения о подписанте (ФИО, СНИЛС и др.), об удостоверяющем центре, который выдал ЭП и срок действия ключа.

Закрытый ключ электронной подписи

Секретный уникальный набор символов, который также формируется криптопровайдером. Закрытый ключ необходим для формирования электронной подписи и хранится в зашифрованном виде, как правило, на носителе (токене). Доступ к закрытому ключу имеет только владелец.

Запрос на сертификат

Электронный документ, который содержит информацию для получения сертификата электронной подписи в Удостоверяющем центре

Ключевой контейнер

Способ хранения закрытых ключей. Кроме ключей, он содержит служебную информацию, необходимую для обеспечения криптографической защиты ключей, их целостности и т.п. Физическое представление ключевого контейнера зависит от типа ключевого носителя: флешка, жёсткий диск, смарт-карты, реестр.

Статус сертификата электронной подписи

Информация о действительности сертификата

Удостоверяющий центр (УЦ)

Доверенная организация, которая имеет право выпускать сертификаты электронной подписи (ЭП) юридическим и физическим лицам

Корневой сертификат

Цифровой документ, в котором содержится информация об удостоверяющем центре. Корневые сертификаты выдаются только организациям и могут иметь разный уровень. В Российской Федерации главный корневой сертификат принадлежит Министерству цифрового развития. На основе этого сертификата Минцифры РФ выдаёт удостоверяющим центрам собственные сертификаты, подписывая их своим корневым сертификатом.

Промежуточный сертификат

Сертификат, выданный удостоверяющим центрам

Сертификат другого пользователя

Сертификат, принадлежащий другому пользователю. Предназначен для шифрования документов в его адрес.

Список отзыва сертификатов

Список сертификатов, которые удостоверяющий центр пометил как отозванные.

Цепочка сертификации или цепочка доверия

Иерархическая коллекция сертификатов. Для квалифицированных сертификатов ЭП, как правило, состоит из 3 звеньев. Верхнее звено - корневой сертификат. Среднее звено - сертификат удостоверяющего центра. Конечное звено - сертификат пользователя.

Сервис DSS

Сервис электронной подписи. Предназначен для шифрования и расшифрования электронных документов, формирования запросов на создание и управление сертификатами ключей проверки электронной подписи, создания и проверки электронной подписи различного формата криптографических сообщений.

Подпись и шифрование

Электронная подпись (ЭП)

Информация в электронно-цифровой форме для определения и подтверждения личности автора подписи

Усиленная неквалифицированная электронная подпись (УНЭП)

Цифровой аналог собственноручной подписи. Применяется, если стороны заключили соглашение об этом или так указано в законе. В таких случаях документы, подписанные от руки или с помощью УНЭП, имеют равную юридическую силу.

Усиленная квалифицированная электронная подпись (УКЭП)

Цифровой аналог собственноручной подписи. Ей можно подписывать любые документы без предварительной договорённости с другой стороной.

Профили подписи в КриптоАРМ

Заранее подготовленные наборы настроек, которые можно применять для выполнения подписи, архивирования и шифрования.

Алгоритмы подписи

Криптографические алгоритмы, которые используются для создания электронной цифровой подписи

Алгоритмы шифрования

Криптографические алгоритмы, которые используются для шифрования файлов

Криптопровайдер

Провайдер криптографических услуг, криптопровайдер специализированный программный модуль или набор модулей, предоставляющих средства шифрования и дешифрования данных, цифровой подписи, а также генерации и хранения ключей.

Присоединенная подпись

Подпись, в процессе создания которой формируется специальный контейнер, в котором находятся как подписываемый документ, так и файл подписи.

Отсоединенная подпись

Файл подписи формируется и хранится отдельно от подписываемого документа.

Стандарт подписи CAdES-BES (Basic Electronic Signature)

Основной и простейший формат электронной подписи, описываемый в стандарте CAdES. Он обеспечивает базовую проверку подлинности данных и защиту их целостности

Стандарт подписи CAdES-T (Timestamp)

Формат электронной подписи, в которую автоматически добавляется штамп времени на подпись.

Стандарт подписи CAdES-X Long Type 1

Формат электронной подписи, включающий дополнительные атрибуты certificate-values и revocation-values, которые представляют собой полные данные сертификатов и списки отзыва сертификатов. А также штампы времени на подпись и ссылки на сертификаты и списки отзыва сертификатов.

Стандарт подписи CAdES-A (Archival)

Формат электронной подписи, сформированный на основе CAdES-X Long Type 1 или Type 2 путём добавления одного или нескольких атрибутов archive-time-stamp, представляющих собой архивные штампы времени.

Машиночитаемая доверенность

Аналог бумажной доверенности в электронном виде. Она создаётся в формате XML, который позволяет компьютеру прочитать её содержание и подтвердить личность того, кто подписал документ

Сертификация PDF документа

Позволяет защитить PDF документ от изменений путем его подписания электронной подписью, с указанием типа изменений, разрешенных для документа.

Конвертация PDF документа

При конвертации PDF документа происходит удаление скриптов, форм и прочих элементов, влияющих на безопасность и целостность документа.

Формат подписи CadES

Стандарт электронной подписи, представляющий собой расширенную версию стандарта электронной подписи CMS. Эту электронную подпись можно прикрепить к любому типу файла : текстовым файлам (.doc, .docx), электронным таблицам (.xls, .xlsx), изображениям (.jpg, .png, .gif) и PDF-файлам.

Формат подписи PadES

Формат, при котором подпись встраивается в PDF-документ. «Встроенная» подпись позволяет подписывать электронные документы без изменения формата файла и без создания отдельного файла подписи, как это происходит при использовании присоединенной и отсоединенной подписи соответственно. Подпись встраивается в электронный документ, и получатель может проверить документ на действительность электронной подписи.

Видимая подпись PDF документа

«Встроенная» подпись, которая отображается на странице документа в виде штампа.

Невидимая подпись PDF документа

«Встроенная» подпись, которая визуально на страницах не видна.

Область для штампа подписи в PDF документе

Область на странице PDF документа, размеченная для видимой подписи.

Произвольная область в PDF документе

Область для графического изображения или текста. В области можно разместить свой комментарий, статус или отметку о конфиденциальности документа.

Почта

Адрес электронной почты

Название электронного почтового ящика, который может получать (и отправлять) электронные письма в сети (например, в интернете или в локальной сети, не подключённой к интернету).

Почтовый сервер

Это программа, работающая у интернет-провайдеров и на крупных сайтах и используемая для передачи почты. Обычно пользователи не взаимодействуют с почтовыми серверами напрямую: электронная почта отправляется с помощью почтового клиента на почтовый сервер, который доставляет её в почтовый клиент получателя.

Почтовый клиент

Это программа (например, на компьютере или мобильном устройстве), которая используется для чтения и отправки электронных сообщений.

IMAP

Протокол, который используется для получения сообщений электронной почты с сервера. IMAP позволяет почтовым программам получать доступ не только к новым сообщениям, но и к папкам на сервере. Действия синхронизируются между несколькими почтовыми программами, подключёнными через IMAP.

POP

Протокол, который используется для получения сообщений электронной почты с сервера. В отличие от IMAP, POP позволяет почтовому клиенту загружать только последние сообщения, которыми можно управлять в клиенте и на устройстве.

SMTP

Протокол, который используется для передачи электронной почты. Он определяет формат сообщений и процедуру их передачи через Интернет от отправителя к получателю с помощью почтовых серверов.

S/MIME

Стандарт для шифрования и подписи электронных писем с помощью открытого ключа.

SSL

Протокол, который обеспечивает безопасное соединение между клиентом и сервером.

Логин (имя пользователя)

Имя почтового ящика — комбинация из букв и цифр, которая, как правило, находится в адресе перед именем домена.

Пароль

Набор символов, предназначенный для подтверждения личности.

Скрытая копия

Копия электронного письма, отправленного получателю, адрес электронной почты которого не указан (в качестве получателя) в сообщении

Копия

Копия электронного письма, отправленного получателю, адрес электронной почты которого указан в поле заголовка «Копия».

Тело письма

Основная часть электронного письма, которая содержит текст, изображения и другие данные (например, прикреплённые файлы).

LDAP

Протокол быстрого доступа к каталогам. Он позволяет получать и принимать через сеть данные из каталогов с иерархической структурой.

ALD Pro

Служба каталога для Linux, программный комплекс централизованного управления компьютерами с помощью групповых политик, а также прикладными сервисами для ИТ-инфраструктур организаций различного масштаба. Комплекс создан «Группой Астра».

CardDAV

Протокол адресной книги, предназначенный для предоставления пользователям доступа к данным на сервере.

vCard

текстовый формат для обмена электронными визитными карточками. В карточке может содержаться имя, номера телефонов, адрес электронной почты, адрес проживания, дата рождения, фотография, сертификат и другие данные контакта.

Gravatar

Сервис компании Automattic, позволяющий интернет-пользователям хранить свой аватар на специальном сервере.

Открыть статью на сайте

Введение в стандарты электронной подписи

- Обзор форматов PadES, CadES и XadES
- Стандарт и формат подписи CAdES
- Политика подписи
- Дополнительные данные проверки

Обзор форматов PadES, CadES и XadES

1. Формат PAdES (PDF Advanced Electronic Signature)

PAdES - специализированный стандарт для электронного подписания PDF-документов, разработанный с учетом особенностей этого формата.

Технические особенности:

- Встраивание подписи непосредственно в структуру PDF-файла
- Сохранение всех свойств исходного документа после подписания

Ключевые преимущества:

- Совместимость - подписанные документы открываются в любом PDF-ридере без дополнительного ПО
- Многоуровневое подписание - возможность последовательного добавления подписей без потери предыдущих
- Визуализация - поддержка графического представления подписи (штамп, подпись, печать)
- Юридическая сила - соответствие требованиям ГОСТ Р и другим регуляторным нормам

Ограничения применения:

- Работает только с PDF-файлами
- Ограниченные возможности для сложных бизнес-процессов, требующих работы с разными типами файлов

2. Формат CAdES (CMS Advanced Electronic Signature)

CAdES - универсальный стандарт электронной подписи на основе криптографического синтаксиса сообщений (CMS).

Особенности реализации:

- Создание контейнера, который содержит:
  - Исходный документ в неизменном виде
  - Криптографические данные подписи
  - Метаданные и атрибуты подписания
- Поддержка различных уровней защиты (BES, EPES, T, C, X, A, LT)

Области применения:

- Офисные документы (Word, Excel)
- Графические файлы (JPG, PNG)
- Мультимедийный контент (MP3, MP4)
- PDF-документы
- Исполняемые файлы и архивы

Основные недостатки:

- Необходимость специального ПО для просмотра файлов
- Сложности при множественном подписании ("эффект матрешки")
- Отсутствие встроенной визуализации подписи в документе

3. Формат XAdES

XAdES (XML Advanced Electronic Signature (расширенная электронная подпись XML)) - стандарт для подписания XML-документов.

XAdES считывается как человеком, так и компьютером и используется для подписи электронных документов, таких как изображения (. jpeg, .png), звуковые файлы .mp3 , двоичные данные (.exe), PDF-файлы.

Варианты реализации:

1. Отделенная подпись:
   - Подпись хранится в отдельном файле
   - Исходный документ остается неизменным
2. Включенная подпись:
   - Подпись встраивается в XML-структуру документа
   - Создается единый файл, содержащий и данные, и подпись

Преимущества:

- Машинно-читаемый формат - удобен для автоматической обработки
- Поддержка множественных подписей - без эффекта матрешки

Стандарт и формат подписи CAdES

Стандарт **CAdES** (CMS Advanced Electronic Signature) определяет **8 профилей**, обеспечивающих различные уровни защиты для подписанных данных в формате CMS. Эти профили гарантируют, что документы с электронной подписью остаются **действительными в течение длительного времени**.

Согласно спецификации **ETSI TS 101 733 V2.2.1 (2013-04)**, подписывающая сторона должна создавать подпись в одном из двух форматов:

- **CAdES-BES (Basic Electronic Signature)** – базовая электронная подпись.
- **CAdES-EPES (Explicit Policy Electronic Signature)** – подпись с явной политикой.

Подпись **CAdES-BES** включает:

1. Обязательные элементы:
    - **Подписанные данные** (документ или хеш документа).
    - **Обязательные подписанные атрибуты:**
    - `Тип контента (Content-Type)`
    - `Хеш сообщения (Message Digest)`
    - `Сертификат подписи ESS (ESSCertID) или ESS-v2`
2. Дополнительные подписанные атрибуты (опционально):
    - `Время подписания (Signing-Time)`
    - `Комментарии (Content-Hints)`
    - `Ссылки (Content-Reference)`
    - `Идентификатор (Content-Identifier)`
    - `Тип обязательства (Commitment-Type-Indication)`
    - `Местоположение подписавшего (Signer-Location)`
    - `Атрибуты подписчика (Signer-Attributes)`
    - `Штамп времени (Content-Time-Stamp)`
    - `Тип кодировки MIME (MIME-Type)`

Формат **CAdES-EPES** расширяет **CAdES-BES**, добавляя:

- **Атрибут** `sigPolicyID` – явное указание политики подписи (например, соответствие eIDAS, ГОСТ или корпоративным стандартам).
- **Дополнительные атрибуты проверки**, обеспечивающие соответствие нормативным требованиям.

Политика подписи

Политики подписи, связанные с **CAdES**, могут использоваться для установления согласованности проверенных электронных подписей. Когда проверяющий использует комплексную политику подписи, указанную подписывающим или подразумеваемую подписанными данными, он получит согласованный результат при попытке проверки электронной подписи.

Однако, если подписывающий или подписанные данные не указывают, какая политика подписи была использована, или если политика подписи выглядит неполной, результаты, достигнутые проверяющими, могут быть непоследовательными. Рекомендуется, чтобы для поддержания согласованности и подписывающий, и проверяющий согласовали одну и ту же всеобъемлющую политику подписи.

Дополнительные данные проверки

Дополнительные данные проверки могут быть добавлены в форматы **CAdES-BES** и **CAdES-EPES** для дальнейшей проверки электронной подписи. Эти данные собираются подписывающим лицом или проверяющим лицом, или обоими. Эти дополнительные данные включают:

- Сертификаты открытых ключей (PKC);
- Информация о статусе отзыва для отдельных PKC;
- Доверенная временная метка в дополнение к временной метке по умолчанию;
- Если применимо, информация о политике подписи, используемой для проверки электронной подписи.

В зависимости от выбранных выше проверочных данных при использовании форматов **CAdES-BES** или **CAdES-EPES** для создания электронной подписи применяются форматы **CAdES**:

CAdES-T

CAdES-T – электронная подпись со временем. Этот формат применяется, когда доверенное время связано с электронной подписью либо путем добавления неподписанного атрибута временной метки, либо поставщик доверенных услуг предоставляет временную метку.

CAdES-C

CAdES-C – электронная подпись с полными ссылками на данные проверки. Этот формат добавляет атрибуты `complete-certificate-references` и `complete-revocation-references` к **CAdES-T**.

CAdES-C– расширенные форматы электронной подписи

Форматы этой категории расширяют проверку **CAdES-C**, добавляя неподписанные атрибуты к электронной подписи, которые используются, когда требуется проверка подписанного документа в течение очень длительного периода.

- **CAdES-X Long** – расширенная длинная электронная подпись. Эта опция добавляет атрибуты `certificate-values` и `revocation-values`, которые предоставляют информацию о сертификате и отзыве для проверки и средства предотвращения потери этой информации.
- **CAdES-X Type 1** – расширенная электронная подпись с типом времени 1. Эта опция добавляет атрибут временной метки, основанный на токене временной метки, созданном с помощью `CAdES-C`, и может защитить ключи от компрометации.
- **CAdES-X Type 2** — расширенная электронная подпись с типом времени 2. Этот параметр добавляет атрибут `CAdES-C-time-stamped-cert-crls-reference`, который предоставляет токен временной метки в отношении пути сертификации и информации об отзыве для защиты ключей от будущей компрометации.
- **CAdES-X Long Type 1** или **2** – расширенная электронная подпись со временем. Этот вариант объединяет `CAdES-X Long` либо с `CAdES-X Type 1`, либо с `CAdES-X Type 2`.

CAdES-A

- **CAdES-A** – архивная электронная подпись. Форматы этой категории обеспечивают обратную совместимость и сохранение долгосрочной проверки подписи для архивных документов.
- **CAdES-A** – архивная электронная подпись с атрибутом `archive-time-stamp (ATSv2)`. Эта опция добавляет один или несколько атрибутов `archive-time-stamp` к **CAdES-X Long** или **CAdES-X Long Type 1** или **2**.
- **CAdES-A** – архивная электронная подпись с атрибутом archive-time-stamp (ATSv2). Может применяться к любой существующей CAdES подписи (включая BES, EPES, T, C, X, XL или даже другую CAdES-A).

> 🔍 Полное техническое описание расширенных электронных подписей CMS описано в документе ETSI TS 101 733 Электронная подпись и инфраструктура (ESI) – Расширенная электронная подпись CMS (CAdES).