Цифровые технологии

КриптоАРМ Gate

Документация продукта

Релиз v1.0

Исходная документация на Trusted.ru

Редакция от 30.10.2026

Правообладатель и разработчик
ООО «Цифровые технологии»

Содержание

О продукте ... 3
Установка и настройка ... 9
Часто задаваемые вопросы ... 16
Глоссарий ... 19
Дополнительные статьи
Index ... 21

Открыть статью на сайте

О продукте КриптоАРМ Gate

КриптоАРМ Gate — программный шлюз безопасности и reverse proxy (обратный прокси-сервер) с поддержкой криптографических алгоритмов ГОСТ. Продукт предназначен для защиты веб-сервисов, безопасного проксирования трафика на внутренний бэкенд, строгой mTLS-аутентификации пользователей по ГОСТ-сертификатам и интеграции с OIDC-провайдерами без модификации целевых приложений.

Содержание:

- О продукте КриптоАРМ Gate
- Поддерживаемые платформы
- Программное окружение
- Типовые сценарии автоматизации
  - Беспарольный вход по mTLS и OIDC-аутентификация
  - Организация гибридного доступа (Двойной серверный TLS)
  - Валидация электронной подписи «на лету» для API (REST/SOAP)
  - Трансляция контекста безопасности на бэкенд
  - Делегирование тяжелых криптоопераций (Маршрутизация на API)
- Функциональные модули
- Интеграция

Поддерживаемые платформы

Программный комплекс **КриптоАРМ Gate** обладает кроссплатформенной архитектурой и адаптирован для развертывания в гетерогенных ИТ-ландшафтах организаций. Продукт поддерживает установку в качестве системного сервиса на следующие семейства операционных систем:

- Отечественные ОС (в рамках импортозамещения):
  - Astra Linux (включая Special Edition и Common Edition)
  - РЕД ОС (версии 7.3, 8 и выше)
  - ОС «Альт» (Альт Сервер, Альт Рабочая станция)
  - РОСА (ROSA Enterprise Desktop, ROSA Server)
  - AlterOS
  - ОС «Основа»
- Зарубежные дистрибутивы Linux:
  - Ubuntu (версии LTS 20.04 / 22.04 и новее)
- Платформа Microsoft:
  - Microsoft Windows Server (начиная с версии 2016 и выше)

Для минимизации времени развертывания и изоляции компонентов доступна поставка в формате готового контейнера:

- Docker-образ: собран на базе доверенной сертифицированной ОС Astra Linux, что гарантирует соответствие повышенным требованиям информационной безопасности при микросервисной архитектуре.
- Совместимость со средами оркестрации: поддерживается функционирование в инфраструктуре Docker Compose, Podman и Kubernetes (включая российские Enterprise-платформы оркестрации).

Программное окружение

Функционирование шлюза безопасности КриптоАРМ Gate обеспечивается строго стандартизированным стеком системного и специализированного программного обеспечения, развернутым внутри изолированных контейнеров:

- **Веб-сервер / Прокси-сервер (Apache 2.4 с модулем mod_ssl):**
  - Применяется в качестве единого фронтенд-компонента независимо от базовой ОС.
  - Отвечает за гибкую маршрутизацию, балансировку нагрузки, проксирование на внутренний бэкенд и инъекцию HTTP-заголовков `X-SSL-Client-...`.
  - Выполняет одновременную терминацию ГОСТ TLS и RSA TLS на одном сетевом порту.
- **СКЗИ «КриптоПро CSP» (версия 5.0 и выше):**
  - Базовый криптографический провайдер системы, встроенный в контейнер.
  - Обеспечивает низкоуровневые операции по стандартам ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012.
  - Отвечает за валидацию клиентских сертификатов при mTLS-соединении в реальном времени с использованием локальных баз CRL и онлайн-серверов OCSP.
- **Среда контейнеризации (Docker & Docker Compose v2):**
  - Обязательная программная среда на хост-машине (минимальная версия Docker Engine 20.10, Docker Compose 2.x).
  - Обеспечивает изоляцию компонентов, сборку шлюза и его развертывание из конфигурационных файлов (`Dockerfile`, `docker-compose.yml`, `.env`).
- **Внешний валидационный модуль (КриптоАРМ OIDC-mTLS):**
  - Опциональный программный компонент.
  - Взаимодействует со шлюзом через API для проведения глубокой проверки «квалифицированности» предъявленного ГОСТ-сертификата пользователя перед его допуском к OIDC-входу.

Типовые сценарии автоматизации

Программный комплекс КриптоАРМ Gate функционирует в качестве интеллектуального шлюза безопасности на границе корпоративной сети. Он автоматизирует процессы защиты каналов связи, строгой аутентификации пользователей и безопасной маршрутизации данных в следующих сценариях:

Беспарольный вход по mTLS и OIDC-аутентификация

- **Проверка на периметре:** Шлюз автоматически запрашивает и перехватывает клиентские ГОСТ-сертификаты (ГОСТ Р 34.10–2012) в момент установления защищенного соединения, блокируя неавторизованный трафик до его попадания во внутреннюю сеть.
- **Интеграция с OIDC-провайдерами:** Gate выступает доверенной входной точкой для систем аутентификации (OpenID Connect), обеспечивая бесшовный и строгий вход пользователей на основе их электронных подписей.
- **Контроль квалифицированности:** В процессе mTLS-авторизации шлюз автоматически обращается к модулю КриптоАРМ OIDC-mTLS для проверки юридического статуса и квалифицированности предъявленного ГОСТ-сертификата.

Организация гибридного доступа (Двойной серверный TLS)

- **Совмещенный порт (ГОСТ + RSA):** Автоматизация публикации одного и того же веб-ресурса с одновременной поддержкой двух типов серверных сертификатов на одном сетевом порту.
- **Маршрутизация по типу клиента:** Внешние государственные системы или пользователи со специализированными браузерами подключаются по ГОСТ TLS. Клиенты без поддержки отечественной криптографии (устаревшие браузеры, смежные ИТ-системы) подключаются по RSA-каналу для построения защищенного HTTPS-соединения. При этом для обоих типов сессий сквозная аутентификация пользователя выполняется строго по его ГОСТ-сертификату.

Трансляция контекста безопасности на бэкенд

- **Декодирование данных «на лету»:** Шлюз автоматически разбирает структуру валидного клиентского сертификата (извлекает ФИО, СНИЛС, ИНН, ОГРН, серийный номер и PEM-тело подписи).
- **Обогащение HTTP-трафика:** Очищенный от TLS-сессии трафик проксируется на внутренние бэкенд-системы (СЭД, ERP, CRM). Запросы автоматически обогащаются HTTP-заголовками X-SSL-Client-* (например, X-SSL-Client-Verify: SUCCESS), что позволяет внутренним приложениям мгновенно авторизовывать пользователя без повторных проверок.

Делегирование тяжелых криптоопераций (Маршрутизация на API)

- **Изоляция сетевого контура:** При поступлении запросов на массовое подписание, шифрование или долгосрочное архивное хранение документов (CAdES-T/X) шлюз изолирует хост-систему от ресурсоемких вычислений.

- **Асинхронная маршрутизация:** Шлюз автоматически преобразует и перенаправляет входящие JSON/XML-пакеты и API-запросы (REST/SOAP/gRPC) на внешний кластер КриптоАРМ Server, сохраняя высокую пропускную способность для транзитного сетевого трафика.

Функциональные модули

Архитектура **КриптоАРМ Gate** построена по модульному принципу. Это позволяет отделить задачи сетевой фильтрации и проверки ЭП от ресурсоемких операций создания подписи и шифрования, которые полностью делегируются КриптоАРМ Server. Для глубокого анализа данных и сложных криптографических процедур шлюз может взаимодействовать с внешним компонентом КриптоАРМ Server посредством специализированного API.

1. Модуль сетевого проксирования и маршрутизации (Proxy Module)

Функционирует на базе веб-сервера Apache 2.4 с модулем mod_ssl. Является единой точкой входа для всего входящего трафика.

- **Терминация TLS-сессий:** Обеспечивает одновременную обработку ГОСТ TLS и RSA TLS соединений на одном сетевом порту.
- **Инъекция HTTP-заголовков (Headers Injection):** Извлекает метаданные из проверенных клиентских сертификатов и транслирует их на внутренний бэкенд в виде заголовков `X-SSL-Client-*`.
- **Балансировка и проксирование:** Выполняет обратное проксирование (Reverse Proxy) и интеллектуальное распределение запросов между внутренними целевыми приложениями.

2. Модуль аутентификации и авторизации (AAA Module)

Отвечает за контроль доступа, управление сессиями и идентификацию пользователей на сетевом периметре.

- **mTLS-идентификация:** Обеспечивает строгую двухфакторную аутентификацию пользователей по сертификатам без использования паролей.
- **Интеграция с OIDC и каталогами:** Выступает шлюзом для OIDC-провайдеров (OpenID Connect). Синхронизирует данные пользователей со службами корпоративных каталогов (LDAP / Active Directory / FreeIPA / Samba DC).
- **Разграничение прав:** Сверяет параметры входящей сессии с предустановленными политиками безопасности до момента пропуска трафика во внутренний контур.

3. Модуль интеграции с КриптоАРМ Server (Crypto API Connector)

Специализированный интеграционный модуль для связи шлюза с внешними вычислительными ресурсами.

- **Изоляция шлюза:** Полностью освобождает Gate от выполнения тяжелых криптографических операций (массовое подписание, архивное штампование, шифрование больших массивов данных).
- **API-маршрутизация:** Перехватывает входящие запросы на подпись/шифрование (REST/SOAP/gRPC) и асинхронно перенаправляет их на выделенный кластер КриптоАРМ Server, возвращая клиенту уже готовый результат.

4. Криптографический модуль шлюза (Crypto Core Local)

Низкоуровневое криптографическое ядро шлюза, функционирующее на базе СКЗИ «КриптоПро CSP 5.0».

- **Локальная экспресс-валидация:** Выполняет мгновенную проверку подлинности и цепочек доверия клиентских ГОСТ-сертификатов «на лету» (в момент TLS-рукопожатия).
- **Контроль статуса (CRL/OCSP):** Автоматически проверяет актуальность сертификатов по локальным спискам отзыва (CRL) и через онлайн-протоколы (OCSP).
- **Взаимодействие с OIDC-mTLS:** Передает данные сертификата на внешний модуль КриптоАРМ OIDC-mTLS для расширенной проверки его юридической квалифицированности.

Примечание: модуль не выполняет функции локального создания электронной подписи и шифрования документов.

Интеграция

Программный комплекс Trusted Gate разработан с учетом бесшовной интеграции в существующую ИТ-инфраструктуру предприятия и внешние государственные или коммерческие экосистемы. Взаимодействие со смежными компонентами осуществляется по общепринятым стандартам и защищенным протоколам.

1. Взаимодействие с КриптоАРМ Server

Для выполнения ресурсоемких криптографических процедур и массовой обработки документов КриптоАРМ Gate интегрируется с внешним серверным кластером:

- **REST API:** Шлюз через модуль Crypto API Connector перехватывает и перенаправляет асинхронные и синхронные прикладные запросы к КриптоАРМ Server для создания архивных электронных подписей (CAdES-T/X), массового шифрования или долгосрочного архивного хранения данных.

- **Балансировка и отказоустойчивость:** Поддерживается маршрутизация трафика на отказоустойчивый кластер КриптоАРМ Server для распределения вычислительной нагрузки при пиковом потоке документов.

2. Интеграция с внутренними информационными системами (Бэкэнд)

КриптоАРМ Gate изолирует внутренний контур от внешних угроз на сетевом периметре, транслируя очищенный трафик и контекст безопасности в прикладные системы:

- **Протоколы обмена:** Передача валидированного трафика на внутренние СЭД, ERP, CRM и OIDC-провайдеры по протоколам HTTP/HTTPS, REST, SOAP и gRPC.
- **Инъекция контекста безопасности (HTTP Headers):** Шлюз обогащает заголовки запросов метаданными извлеченного и проверенного ГОСТ-сертификата клиента (X-SSL-Client-Verify, X-SSL-Client-DN, X-SSL-Client-Serial). Это позволяет внутренним приложениям мгновенно авторизовывать пользователей в режиме Single Sign-On (SSO).
- **Службы каталогов:** Интеграция с Active Directory, FreeIPA, ALD Pro и Samba DC по протоколам LDAP/LDAPS для автоматического сопоставления данных клиентского сертификата (СНИЛС, ИНН, Email) с корпоративными учетными записями сотрудников.

3. Взаимодействие с внешними сервисами актуальных статусов

Для обеспечения непрерывной валидации сертификатов в режиме реального времени (при установлении mTLS-соединений) локальное криптоядро шлюза интегрируется со службами Удостоверяющих Центров:

- **Серверы статусов (OCSP):** Автоматическая отправка запросов по протоколу OCSP (RFC 6960) для экспресс-проверки действительности ключей пользователей «на лету» в момент TLS-handshake.
- **Списки отзыва (CRL):** Автоматическое скачивание по расписанию и обновление локальных баз списков отзыва сертификатов (CRL) по протоколам HTTP/HTTPS.
- **Валидация квалифицированности (OIDC-mTLS):** Интеграция со специализированным внешним модулем КриптоАРМ OIDC-mTLS для расширенной проверки юридического статуса и ограничений предъявленного ГОСТ-сертификата перед допуском пользователя к OIDC-авторизации.

Открыть статью на сайте

Установка и запуск КриптоАРМ Gate

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

Содержание:

- Установка и запуск КриптоАРМ Gate
- Условия применения
  - Требования для установки и настройки
  - Используемый стек технологий
- Установка
  - Установка Docker и Docker Compose
  - Установка и первый запуск
  - Сборка на базе Astra Linux
- Настройка
  - Сертификаты
  - Режим mTLS и OIDC
  - Переменные mTLS и заголовки для бэкенда
  - Проксирование на бэкенд
- Проверка работы

Условия применения

Требования для установки и настройки

Характеристика	Минимальное значение	Дополнение
ОЗУ	4 ГБ	
HDD	10 ГБ	
Процессор	x86_64 (amd64)	
Поддерживаемые ОС	Ubuntu 18.04/20.04, Debian 11, Astra Linux SE 1.8 Орел	Поддерживаются другие Linux-дистрибутивы с поддержкой Docker
Дополнительно	Docker	Средство автоматизации развертывания и контейнеризации

Используемый стек технологий

Apache 2.4, КриптоПро CSP 5.0 (mod_ssl), Docker. Опционально: модуль КриптоАРМ OIDC-mTLS (для проверки квалифицированности сертификата).

Установка

Установка Docker и Docker Compose

Для запуска проекта необходимы **Docker** и **Docker Compose**.
Инструкции: официальная документация Docker.

Минимальные версии: - Docker Engine: **20.10** или выше (для работы Docker Compose v2) - Docker Compose: **2.x** (плагин `docker compose` или отдельная установка)

Установка и первый запуск

1. Создайте и перейдите в рабочую директорию:

```
mkdir cryptoarm-gate && cd cryptoarm-gate
```

1. Загрузите конфигурационные файлы:

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

1. Создайте необходимые директории и загрузите конфигурацию сайта:

```
mkdir -p cryptopro certs/gost certs/rsa certs/root conf html logs
curl -o conf/ssl.conf https://git.digtlab.ru/trusted/cryptoarm/gate/-/raw/main/conf/ssl.conf
```

1. Скачайте актуальную версию КриптоПро CSP 5.0 для Linux (x64, deb) и поместите архив `linux-amd64_deb.tgz` в каталог `cryptoarm-gate/cryptopro/`.
   Для скачивания требуется авторизация на сайте КриптоПро.

2. Отредактируйте настройки в файле **.env**: - `CRYPTO_PRO_LICENSE` — серверная лицензия КриптоПро CSP (при необходимости). - При использовании бэкенда укажите его адрес в конфигурации Apache (см. Проксирование на бэкенд).

3. Разместите сертификаты сервера в соответствующих каталогах (см. Сертификаты; для теста можно скачать примеры из репозитория командами `curl`, указанными в этом разделе).

4. Запустите сборку и сервис:

```
docker compose build --pull
docker compose up -d
```

Дополнительно: - Просмотр логов: `docker compose logs -f` - Остановка: `docker compose stop` - Обновление до актуальной версии: повторите п. 2, затем `docker compose build --pull` и `docker compose up -d`

Сборка на базе Astra Linux

Для развёртывания в средах, где предъявляются требования к использованию ОС Astra Linux, предусмотрена сборка образа на базе универсального базового образа (UBI) Astra Linux.

1. Загрузите файл сборки для Astra Linux (в дополнение к п. 2 раздела Установка и первый запуск):

```
curl -O https://git.digtlab.ru/trusted/cryptoarm/gate/-/raw/main/Dockerfile.astralinux
```

1. Убедитесь, что в каталоге есть все артефакты: `docker-compose.yml`, `.env`, `entrypoint.sh`, каталог `cryptopro/` с архивом КриптоПро CSP, при необходимости `conf/ssl.conf` и сертификаты.

2. Соберите образ с использованием Dockerfile для Astra Linux (базовый образ — Astra Linux SE, реестр registry.astralinux.ru). При необходимости замените тег базового образа в `Dockerfile.astralinux` на актуальный из вашего реестра:

```
docker build -f Dockerfile.astralinux -t cryptoarm.gate .
```

1. Запустите сервис так же, как при сборке на Ubuntu — через Docker Compose (образ уже собран под именем `cryptoarm.gate`):

```
docker compose up -d
```

Состав установленного ПО (Apache 2.4, КриптоПро CSP, mod_ssl, entrypoint) совпадает с вариантом на базе Ubuntu; отличается только базовый слой ОС. Остальные шаги настройки (сертификаты, mTLS, проксирование) выполняются так же, как описано в разделах Настройка и Проверка работы.

Настройка

Сертификаты

- **ГОСТ (обязательно)**: серверные сертификаты в `certs/gost/` (например, `localhost.cer`, `localhost.pfx`). Используются для HTTPS и для приёма клиентских ГОСТ-сертификатов при mTLS.

- **RSA (опционально)**: серверные сертификаты в `certs/rsa/` (например, `localhost.cer`, `localhost.pfx`). Подключаются только если нужно поднять один веб-ресурс одновременно по ГОСТ и по RSA (совместимость с клиентами без поддержки ГОСТ). Вход по сертификату и mTLS при этом выполняются только по ГОСТ.
- **Корневые и доверенные СА**: в `certs/root/` (файлы `.cer`) — для проверки **клиентских ГОСТ-сертификатов** при mTLS. При старте контейнера устанавливаются в хранилище КриптоПро.

В качестве примеров для тестирования можно скачать сертификаты из репозитория:

```
mkdir -p certs/gost certs/rsa certs/root
curl -o certs/gost/localhost.cer https://git.digtlab.ru/trusted/cryptoarm/gate/-/raw/main/
certs/gost/localhost.cer
curl -o certs/gost/localhost.pfx https://git.digtlab.ru/trusted/cryptoarm/gate/-/raw/main/
certs/gost/localhost.pfx
curl -o certs/rsa/localhost.cer https://git.digtlab.ru/trusted/cryptoarm/gate/-/raw/main/
certs/rsa/localhost.cer
curl -o certs/rsa/localhost.pfx https://git.digtlab.ru/trusted/cryptoarm/gate/-/raw/main/
certs/rsa/localhost.pfx
curl -o certs/root/cryptopro_test.cer https://git.digtlab.ru/trusted/cryptoarm/gate/-/raw/
main/certs/root/cryptopro_test.cer
```

Внимание: примеры предназначены только для проверки развёртывания и тестовых сценариев. В продуктивной среде используйте собственные сертификаты и корневые CA.

Конфигурация SSL задаётся в `conf/ssl.conf`, монтируемом в контейнер.

Режим mTLS и OIDC

Вход по сертификату и допуск к защищённым ресурсам выполняются **по ГОСТ-сертификату** клиента. В конфигурации Apache включают проверку клиентского сертификата и передачу его данных бэкенду:

- В `conf/ssl.conf` раскомментируйте `SSLVerifyClient require` при необходимости обязательной проверки клиентского сертификата (принимаются только ГОСТ-сертификаты, настроенные в КриптоПро).
- Раскомментируйте блоки `RequestHeader set X-SSL-Client-*`, чтобы передавать в бэкенд атрибуты предъявленного сертификата:
- `X-SSL-Client-Verify`, `X-SSL-Client-DN`, `X-SSL-Client-Serial`, `X-SSL-Client-Issuer`, `X-SSL-Client-Cert`, `X-SSL-Client-S-DN-CN`, `X-SSL-Client-I-DN-CN`.

Проверка **квалифицированности** предъявленного ГОСТ-сертификата выполняется с привлечением **модуля КриптоАРМ OIDC-mTLS**: GATE передаёт данные подписи/сертификата в API модуля; по результату проверки принимается решение о допуске пользователя для OIDC-входа.

Переменные mTLS и заголовки для бэкенда

При успешной проверке клиентского ГОСТ-сертификата Apache (mod_ssl) устанавливает переменные окружения `SSL_CLIENT_*` и при включённых `RequestHeader` передаёт их бэкенду в HTTP-заголовках `X-SSL-Client-*`. Ниже — пример значений (из SSI, например `index.shtml`).

Переменные сертификата клиента (mTLS):

Переменная / заголовок	Описание	Пример значения
`SSL_CLIENT_VERIFY` / `X-SSL-Client-Verify`	Результат проверки клиентского сертификата	`SUCCESS`
`SSL_CLIENT_S_DN_CN` / `X-SSL-Client-S-DN-CN`	Common Name субъекта сертификата	`TEST_КРИПТОАРМ`
`SSL_CLIENT_I_DN_CN` / `X-SSL-Client-I-DN-CN`	Common Name издателя (УЦ)	`Тестовый УЦ ООО "КРИПТО-ПРО"`
`SSL_CLIENT_S_DN` / `X-SSL-Client-DN`	Distinguished Name субъекта	`CN=TEST_КРИПТОАРМ`
`SSL_CLIENT_I_DN` / `X-SSL-Client-Issuer`	DN издателя	`CN=Тестовый УЦ ООО "КРИПТО-ПРО",O=ООО "КРИПТО-ПРО",L=Москва,ST=г. Москва,C=RU,...`
`SSL_CLIENT_M_SERIAL` / `X-SSL-Client-Serial`	Серийный номер сертификата	`7C002E6AAA722B2E8F1A3D81930012002E6AAA`
`SSL_CLIENT_CERT` / `X-SSL-Client-Cert`	Сертификат клиента (PEM)	`-----BEGIN CERTIFICATE----- ... -----END CERTIFICATE-----`
`SSL_CLIENT_A_KEY`	Алгоритм ключа	`GOST R 34.10-2012 with 256 bit modulus`
`SSL_CLIENT_A_SIG`	Алгоритм подписи	`GOST R 34.10-2012 with GOST R 34.11-2012 (256 bit)`
`SSL_CLIENT_V_START` / `SSL_CLIENT_V_END`	Срок действия сертификата	`Feb 13 08:58:40 2026 GMT` … `Apr 13 09:08:40 2026 GMT`

Дополнительные переменные SSL (только на стороне GATE, в SSI/скриптах):

```
HTTPS=on
SSL_PROTOCOL=TLSv1.2
SSL_CLIENT_I_DN_C=RU
SSL_CLIENT_I_DN_ST=г. Москва
SSL_CLIENT_I_DN_L=Москва
SSL_CLIENT_I_DN_O=ООО "КРИПТО-ПРО"
SSL_CLIENT_I_DN_CN=Тестовый УЦ ООО "КРИПТО-ПРО"
SSL_VERSION_INTERFACE=mod_ssl/2.4.41
SSL_VERSION_LIBRARY=OpenSSL/1.1.1f
```

Бэкенд (OIDC-провайдер и др.) при проксировании получает заголовки `HTTP_X_SSL_CLIENT_VERIFY`, `HTTP_X_SSL_CLIENT_DN`, `HTTP_X_SSL_CLIENT_SERIAL`, `HTTP_X_SSL_CLIENT_ISSUER`, `HTTP_X_SSL_CLIENT_CERT`, `HTTP_X_SSL_CLIENT_S_DN_CN`, `HTTP_X_SSL_CLIENT_I_DN_CN` (префикс `HTTP_` и замена `-` на `_` задаются правилами сервера/приложения).

Проксирование на бэкенд

В `conf/ssl.conf` задаётся проксирование на внутренний бэкенд (в т.ч. OIDC-провайдер):

```
ProxyPass        /api http://backend/
ProxyPassReverse /api http://backend/
```

Измените `http://backend/` на фактический адрес бэкенда (в Docker Compose — имя сервиса и порт, например `http://cryptoarm-id-back:3005/`).

Пример бэкенда для проверки. В `docker-compose.yml` можно использовать тестовый бэкенд kennethreitz/httpbin: он возвращает в ответе все заголовки запроса, в том числе переданные GATE заголовки mTLS (`X-Ssl-Client-*`). Сервис уже описан в составе проекта:

```
  backend:
    image: kennethreitz/httpbin
    container_name: backend
```

После запуска `docker compose up -d` запрос к `https://SERVER:4443/api/headers` (с предъявлением ГОСТ-сертификата при включённом mTLS) вернёт JSON с заголовками, среди которых будут `X-Ssl-Client-Verify`, `X-Ssl-Client-Dn`, `X-Ssl-Client-Serial`, `X-Ssl-Client-Issuer`, `X-Ssl-Client-Cert`, `X-Ssl-Client-S-Dn-Cn`, `X-Ssl-Client-I-Dn-Cn` и др.

Проверка работы

- Доступ по HTTPS (порт 4443 по умолчанию): `https://SERVER:4443`
- Статическая раздача из `/var/www/html` (монтируется из `html/`).
- Проксирование запросов по путям, указанным в `ProxyPass` (например, `/api`), на бэкенд.

Проверка mTLS и передачи данных сертификата бэкенду. Подключитесь к GATE с **клиентским ГОСТ-сертификатом** и выполните запрос к примеру бэкенда (httpbin):

```
https://SERVER:4443/api/headers
```

В ответе (JSON) в объекте `headers` будут переданные GATE заголовки, например:

```
{
  "headers": {
    "Host": "172.17.1.115:14443",
    "X-Forwarded-Host": "172.17.1.115:14443",
    "X-Ssl-Client-Verify": "SUCCESS",
    "X-Ssl-Client-Dn": "CN=TEST_КРИПТОАРМ",
    "X-Ssl-Client-S-Dn-Cn": "TEST_КРИПТОАРМ",
    "X-Ssl-Client-I-Dn-Cn": "Тестовый УЦ ООО \"КРИПТО-ПРО\"",
    "X-Ssl-Client-Serial": "7C002E6AAA722B2E8F1A3D81930012002E6AAA",
    "X-Ssl-Client-Issuer": "CN=Тестовый УЦ ООО \"КРИПТО-ПРО\",O=ООО \"КРИПТО-
ПРО\",L=Москва,ST=г. Москва,C=RU,...",
    "X-Ssl-Client-Cert": "-----BEGIN CERTIFICATE----- MIIE5DCCBJGgAwIBAgITfAAuaqpy ... -----
```

```
END CERTIFICATE-----"
  }
}
```

По наличию и значениям этих заголовков можно убедиться, что бэкенд получает данные клиентского сертификата для OIDC-входа по сертификату.

Открыть статью на сайте

Часто задаваемые вопросы по КриптоАРМ Gate

Для чего нужен КриптоАРМ Gate?

КриптоАРМ Gate (Trusted Gate) — КриптоАРМ Gate — это программный шлюз безопасности, который функционирует как обратный прокси-сервер (reverse proxy) на базе Apache 2.4. Он предназначен для защиты веб-сервисов организации, организации безопасного удаленного доступа по ГОСТ-алгоритмам и строгой беспарольной аутентификации пользователей непосредственно на периметре сети. Подробнее архитектура решения описана в статье О продукте.

Какие криптографические и сетевые операции поддерживает КриптоАРМ Gate?

- Терминация ГОСТ TLS и RSA TLS: Обеспечение одновременного защищенного соединения клиентов к веб-серверам организации по отечественным стандартам (ГОСТ Р 34.10-2012 / ГОСТ Р 34.11-2012) и зарубежным алгоритмам на одном сетевом порту.
- Строгая аутентификация (mTLS): Двусторонняя проверка подлинности пользователей по их сертификатам электронной подписи в момент TLS-рукопожатия.
- Проверка статуса сертификатов «на лету»: Мгновенный контроль актуальности ключей пользователей через онлайн-серверы OCSP и локальные списки отзыва (CRL).
- Инъекция контекста безопасности: Извлечение реквизитов владельца сертификата и автоматическая передача этих данных во внутренние системы в виде стандартизированных HTTP-заголовков (X-SSL-Client-*).
- Делегирование тяжелых операций: Интеграция по REST API с внешним компонентом КриптоАРМ Server для выполнения ресурсоемких задач (массового серверного подписания документов, шифрования или визуализации штампов подписи в PDF).

Как КриптоАРМ Gate встраивается в существующую сеть организации?

Шлюз устанавливается на периметре сети (в DMZ-зоне) перед целевыми веб-сервисами и OIDC-провайдерами. Он принимает внешний входящий трафик, полностью терминирует TLS-сессию, выполняет проверку пользователя и только после этого перенаправляет очищенные запросы во внутренний контур безопасности.

Обязательно ли устанавливать КриптоПро CSP на конечные серверы приложений (бэкэнд)?

Нет, не обязательно. КриптоАРМ Gate полностью берет на себя функции терминации ГОСТ TLS и валидации клиентских сертификатов на периметре. Внутренние СЭД и ERP-системы могут получать трафик по обычному протоколу HTTP/HTTPS без использования криптопровайдера.

Может ли шлюз работать в контейнерах Docker?

Да, продукт поставляется исключительно в формате Docker-контейнеров. Для микросервисных сред подготовлены оптимизированные образы на базе Ubuntu и доверенной ОС Astra Linux (UBI-образ). СКЗИ «КриптоПро CSP 5.0» развертывается и функционирует непосредственно внутри изолированного контейнера шлюза.

Каким образом шлюз проверяет электронную подпись и сертификат пользователя?

Шлюз осуществляет проверку в момент установления защищенного соединения (TLS Handshake). Локальный криптографический модуль через СКЗИ «КриптоПро CSP» строит цепочку доверия для предъявленного пользователем ГОСТ-сертификата и проверяет его валидность через OCSP/CRL. Дополнительно для проверки квалифицированности и юридического статуса сертификата шлюз может обращаться к внешнему модулю КриптоАРМ OIDC-mTLS.

Что происходит, если пользователь пытается войти с отозванным или поврежденным сертификатом?

КриптоАРМ Gate блокирует установление защищенного соединения непосредственно на сетевом периметре. Невалидный трафик не доходит до внутренней сети организации, а внешнему клиенту веб-сервер автоматически возвращает стандартную ошибку авторизации (например, 403 Forbidden).

Как внутренние системы узнают, чей сертификат был проверен шлюзом?

Шлюз использует механизм инъекции заголовков (Header Injection). После успешной mTLS-аутентификации модуль mod_ssl добавляет в HTTP-пакет служебные заголовки (такие как X-SSL-Client-Verify: SUCCESS, X-SSL-Client-DN, X-SSL-Client-Serial). Внутренний бэкэнд считывает эти данные и автоматически авторизует пользователя.

Для чего нужен КриптоАРМ Server?

КриптоАРМ Server — это специализированное серверное приложение (криптографический веб-сервис), предназначенное для выполнения ресурсоемких операций с электронными документами. Он осуществляет массовое создание и проверку электронных подписей, шифрование файлов, работу со штампами времени и визуализацию подписей на стороне бэкенда.

Чем КриптоАРМ Server отличается от КриптоАРМ Gate?

- КриптоАРМ Gate — это «сетевой щит» (reverse proxy) на периметре контура. Он отвечает за безопасность каналов связи, mTLS-вход пользователей по ГОСТ-сертификатам и проксирование трафика. Он не создает электронные подписи.
- КриптоАРМ Server — это «вычислительное ядро» внутри сети. Он принимает документы от внутренних систем (или через Gate) по API и выполняет над ними непосредственные криптооперации (подписание, шифрование, создание архивов).

Открыть статью на сайте

Глоссарий

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.

Обратный прокси-сервер (Reverse Proxy)

Сетевой компонент, который принимает внешние запросы и перенаправляет их на один или несколько серверов во внутренней сети организации.

Терминация TLS / ГОСТ TLS

Процесс перехвата и расшифрования защищенного сетевого трафика на шлюзе безопасности до его передачи во внутреннюю сеть в открытом виде.

Двусторонняя аутентификация (mTLS)

Механизм защиты, при котором и клиент, и сервер проверяют подлинность сертификатов друг друга перед установкой защищенного соединения.

Инъекция HTTP-заголовков (Header Injection)

Метод передачи данных, при котором шлюз безопасности автоматически добавляет в HTTP-запрос метаданные о результатах проверки электронной подписи для бэкэнд-систем.

Валидация ЭП «на лету»

Автоматический процесс извлечения и проверки подлинности электронной подписи из транзитного сетевого трафика без задержки передачи данных.

Протокол OCSP (Online Certificate Status Protocol)

Онлайн-протокол для оперативного получения статуса действительности сертификата электронной подписи без скачивания полных списков отзыва.

Список отзыва сертификатов (CRL)

Электронный документ со списком серийных номеров сертификатов, которые были аннулированы удостоверяющим центром до истечения срока их действия.

OpenID Connect (OIDC)

Современный протокол аутентификации, построенный поверх OAuth 2.0, который позволяет сторонним приложениям (клиентам) проверять личность пользователей на основе аутентификации, выполненной сервером авторизации.

Модуль Apache mod_ssl

Специализированный компонент веб-сервера Apache, обеспечивающий поддержку протоколов криптографической защиты TLS. В КриптоАРМ Gate он модифицирован для интеграции с СКЗИ «КриптоПро CSP».

JSON Web Token (JWT)

Строго структурированный компактный формат безопасной передачи данных между сторонами в виде JSON-объекта, подписанного цифровой подписью; часто используется OIDC-провайдерами для управления сессиями пользователей.

CAdES

Формат электронной подписи, расширяющий базовый стандарт CMS возможностью добавления штампов времени и доказательств актуальности сертификатов для долгосрочного хранения.

XAdES

Формат электронной подписи для XML-документов, обеспечивающий проверку целостности структурных элементов веб-запросов и API-пакетов.

Открыть статью на сайте

- Trusted Gate
- криптографический шлюз
- шлюз безопасности
- электронная подпись
- проверка ЭП
- шифрование данных
- защита веб-сервисов author: Команда КриптоАРМ Gate date: 2026-04-30 updated: 2026-05-13 menu_title: "О продукте"

О продукте КриптоАРМ Gate

КриптоАРМ Gate — программный шлюз безопасности и reverse proxy (обратный прокси-сервер) с поддержкой криптографических алгоритмов ГОСТ. Продукт предназначен для защиты веб-сервисов, безопасного проксирования трафика на внутренний бэкенд, строгой mTLS-аутентификации пользователей по ГОСТ-сертификатам и интеграции с OIDC-провайдерами без модификации целевых приложений.

Содержание:

- О продукте КриптоАРМ Gate
- Поддерживаемые платформы
- Программное окружение
- Типовые сценарии автоматизации
  - Беспарольный вход по mTLS и OIDC-аутентификация
  - Организация гибридного доступа (Двойной серверный TLS)
  - Валидация электронной подписи «на лету» для API (REST/SOAP)
  - Трансляция контекста безопасности на бэкенд
  - Делегирование тяжелых криптоопераций (Маршрутизация на API)
- Функциональные модули
- Интеграция

Поддерживаемые платформы

Программный комплекс **КриптоАРМ Gate** обладает кроссплатформенной архитектурой и адаптирован для развертывания в гетерогенных ИТ-ландшафтах организаций. Продукт поддерживает установку в качестве системного сервиса на следующие семейства операционных систем:

- Отечественные ОС (в рамках импортозамещения):
  - Astra Linux (включая Special Edition и Common Edition)
  - РЕД ОС (версии 7.3, 8 и выше)

    - ОС «Альт» (Альт Сервер, Альт Рабочая станция)
    - РОСА (ROSA Enterprise Desktop, ROSA Server)
    - AlterOS
    - ОС «Основа»
- Зарубежные дистрибутивы Linux:
    - Ubuntu (версии LTS 20.04 / 22.04 и новее)
- Платформа Microsoft:
    - Microsoft Windows Server (начиная с версии 2016 и выше)

Для минимизации времени развертывания и изоляции компонентов доступна поставка в формате готового контейнера:

- Docker-образ: собран на базе доверенной сертифицированной ОС Astra Linux, что гарантирует соответствие повышенным требованиям информационной безопасности при микросервисной архитектуре.
- Совместимость со средами оркестрации: поддерживается функционирование в инфраструктуре Docker Compose, Podman и Kubernetes (включая российские Enterprise-платформы оркестрации).

Программное окружение

Функционирование шлюза безопасности КриптоАРМ Gate обеспечивается строго стандартизированным стеком системного и специализированного программного обеспечения, развернутым внутри изолированных контейнеров:

- **Веб-сервер / Прокси-сервер (Apache 2.4 с модулем mod_ssl):**
    - Применяется в качестве единого фронтенд-компонента независимо от базовой ОС.
    - Отвечает за гибкую маршрутизацию, балансировку нагрузки, проксирование на внутренний бэкенд и инъекцию HTTP-заголовков `X-SSL-Client-...`.
    - Выполняет одновременную терминацию ГОСТ TLS и RSA TLS на одном сетевом порту.
- **СКЗИ «КриптоПро CSP» (версия 5.0 и выше):**
    - Базовый криптографический провайдер системы, встроенный в контейнер.
    - Обеспечивает низкоуровневые операции по стандартам ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012.
    - Отвечает за валидацию клиентских сертификатов при mTLS-соединении в реальном времени с использованием локальных баз CRL и онлайн-серверов OCSP.
- **Среда контейнеризации (Docker & Docker Compose v2):**
    - Обязательная программная среда на хост-машине (минимальная версия Docker Engine 20.10, Docker Compose 2.x).
    - Обеспечивает изоляцию компонентов, сборку шлюза и его развертывание из конфигурационных файлов (`Dockerfile`, `docker-compose.yml`, `.env`).
- **Внешний валидационный модуль (КриптоАРМ OIDC-mTLS):**
    - Опциональный программный компонент.

- Взаимодействует со шлюзом через API для проведения глубокой проверки «квалифицированности» предъявленного ГОСТ-сертификата пользователя перед его допуском к OIDC-входу.

Типовые сценарии автоматизации

Программный комплекс КриптоАРМ Gate функционирует в качестве интеллектуального шлюза безопасности на границе корпоративной сети. Он автоматизирует процессы защиты каналов связи, строгой аутентификации пользователей и безопасной маршрутизации данных в следующих сценариях:

Беспарольный вход по mTLS и OIDC-аутентификация

- **Проверка на периметре:** Шлюз автоматически запрашивает и перехватывает клиентские ГОСТ-сертификаты (ГОСТ Р 34.10–2012) в момент установления защищенного соединения, блокируя неавторизованный трафик до его попадания во внутреннюю сеть.
- **Интеграция с OIDC-провайдерами:** Gate выступает доверенной входной точкой для систем аутентификации (OpenID Connect), обеспечивая бесшовный и строгий вход пользователей на основе их электронных подписей.
- **Контроль квалифицированности:** В процессе mTLS-авторизации шлюз автоматически обращается к модулю КриптоАРМ OIDC-mTLS для проверки юридического статуса и квалифицированности предъявленного ГОСТ-сертификата.

Организация гибридного доступа (Двойной серверный TLS)

- **Совмещенный порт (ГОСТ + RSA):** Автоматизация публикации одного и того же веб-ресурса с одновременной поддержкой двух типов серверных сертификатов на одном сетевом порту.
- **Маршрутизация по типу клиента:** Внешние государственные системы или пользователи со специализированными браузерами подключаются по ГОСТ TLS. Клиенты без поддержки отечественной криптографии (устаревшие браузеры, смежные ИТ-системы) подключаются по RSA-каналу для построения защищенного HTTPS-соединения. При этом для обоих типов сессий сквозная аутентификация пользователя выполняется строго по его ГОСТ-сертификату.

Трансляция контекста безопасности на бэкенд

- **Декодирование данных «на лету»:** Шлюз автоматически разбирает структуру валидного клиентского сертификата (извлекает ФИО, СНИЛС, ИНН, ОГРН, серийный номер и PEM-тело подписи).
- **Обогащение HTTP-трафика:** Очищенный от TLS-сессии трафик проксируется на внутренние бэкенд-системы (СЭД, ERP, CRM). Запросы автоматически обогащаются HTTP-заголовками X-SSL-Client-* (например, X-SSL-Client-Verify: SUCCESS), что позволяет внутренним приложениям мгновенно авторизовывать пользователя без повторных проверок.

Делегирование тяжелых криптоопераций (Маршрутизация на API)

- **Изоляция сетевого контура:** При поступлении запросов на массовое подписание, шифрование или долгосрочное архивное хранение документов (CAdES-T/X) шлюз изолирует хост-систему от ресурсоемких вычислений.
- **Асинхронная маршрутизация:** Шлюз автоматически преобразует и перенаправляет входящие JSON/XML-пакеты и API-запросы (REST/SOAP/gRPC) на внешний кластер КриптоАРМ Server, сохраняя высокую пропускную способность для транзитного сетевого трафика.

Функциональные модули

Архитектура **КриптоАРМ Gate** построена по модульному принципу. Это позволяет отделить задачи сетевой фильтрации и проверки ЭП от ресурсоемких операций создания подписи и шифрования, которые полностью делегируются КриптоАРМ Server. Для глубокого анализа данных и сложных криптографических процедур шлюз может взаимодействовать с внешним компонентом КриптоАРМ Server посредством специализированного API.

1. Модуль сетевого проксирования и маршрутизации (Proxy Module)

Функционирует на базе веб-сервера Apache 2.4 с модулем mod_ssl. Является единой точкой входа для всего входящего трафика.

- **Терминация TLS-сессий:** Обеспечивает одновременную обработку ГОСТ TLS и RSA TLS соединений на одном сетевом порту.
- **Инъекция HTTP-заголовков (Headers Injection):** Извлекает метаданные из проверенных клиентских сертификатов и транслирует их на внутренний бэкенд в виде заголовков `X-SSL-Client-*`.
- **Балансировка и проксирование:** Выполняет обратное проксирование (Reverse Proxy) и интеллектуальное распределение запросов между внутренними целевыми приложениями.

2. Модуль аутентификации и авторизации (AAA Module)

Отвечает за контроль доступа, управление сессиями и идентификацию пользователей на сетевом периметре.

- **mTLS-идентификация:** Обеспечивает строгую двухфакторную аутентификацию пользователей по сертификатам без использования паролей.
- **Интеграция с OIDC и каталогами:** Выступает шлюзом для OIDC-провайдеров (OpenID Connect). Синхронизирует данные пользователей со службами корпоративных каталогов (LDAP / Active Directory / FreeIPA / Samba DC).
- **Разграничение прав:** Сверяет параметры входящей сессии с предустановленными политиками безопасности до момента пропуска трафика во внутренний контур.

3. Модуль интеграции с КриптоАРМ Server (Crypto API Connector)

Специализированный интеграционный модуль для связи шлюза с внешними вычислительными ресурсами.

- **Изоляция шлюза:** Полностью освобождает Gate от выполнения тяжелых криптографических операций (массовое подписание, архивное штампование, шифрование больших массивов данных).
- **API-маршрутизация:** Перехватывает входящие запросы на подпись/шифрование (REST/SOAP/gRPC) и асинхронно перенаправляет их на выделенный кластер КриптоАРМ Server, возвращая клиенту уже готовый результат.

4. Криптографический модуль шлюза (Crypto Core Local)

Низкоуровневое криптографическое ядро шлюза, функционирующее на базе СКЗИ «КриптоПро CSP 5.0».

- **Локальная экспресс-валидация:** Выполняет мгновенную проверку подлинности и цепочек доверия клиентских ГОСТ-сертификатов «на лету» (в момент TLS-рукопожатия).
- **Контроль статуса (CRL/OCSP):** Автоматически проверяет актуальность сертификатов по локальным спискам отзыва (CRL) и через онлайн-протоколы (OCSP).
- **Взаимодействие с OIDC-mTLS:** Передает данные сертификата на внешний модуль КриптоАРМ OIDC-mTLS для расширенной проверки его юридической квалифицированности.

Примечание: модуль не выполняет функции локального создания электронной подписи и шифрования документов.

Интеграция

Программный комплекс Trusted Gate разработан с учетом бесшовной интеграции в существующую ИТ-инфраструктуру предприятия и внешние государственные или коммерческие экосистемы. Взаимодействие со смежными компонентами осуществляется по общепринятым стандартам и защищенным протоколам.

1. Взаимодействие с КриптоАРМ Server

Для выполнения ресурсоемких криптографических процедур и массовой обработки документов КриптоАРМ Gate интегрируется с внешним серверным кластером:

- **REST API:** Шлюз через модуль Crypto API Connector перехватывает и перенаправляет асинхронные и синхронные прикладные запросы к КриптоАРМ Server для создания архивных электронных подписей (CAdES-T/X), массового шифрования или долгосрочного архивного хранения данных.

- **Балансировка и отказоустойчивость:** Поддерживается маршрутизация трафика на отказоустойчивый кластер КриптоАРМ Server для распределения вычислительной нагрузки при пиковом потоке документов.

2. Интеграция с внутренними информационными системами (Бэкэнд)

КриптоАРМ Gate изолирует внутренний контур от внешних угроз на сетевом периметре, транслируя очищенный трафик и контекст безопасности в прикладные системы:

- **Протоколы обмена:** Передача валидированного трафика на внутренние СЭД, ERP, CRM и OIDC-провайдеры по протоколам HTTP/HTTPS, REST, SOAP и gRPC.
- **Инъекция контекста безопасности (HTTP Headers):** Шлюз обогащает заголовки запросов метаданными извлеченного и проверенного ГОСТ-сертификата клиента (X-SSL-Client-Verify, X-SSL-Client-DN, X-SSL-Client-Serial). Это позволяет внутренним приложениям мгновенно авторизовывать пользователей в режиме Single Sign-On (SSO).
- **Службы каталогов:** Интеграция с Active Directory, FreeIPA, ALD Pro и Samba DC по протоколам LDAP/LDAPS для автоматического сопоставления данных клиентского сертификата (СНИЛС, ИНН, Email) с корпоративными учетными записями сотрудников.

3. Взаимодействие с внешними сервисами актуальных статусов

Для обеспечения непрерывной валидации сертификатов в режиме реального времени (при установлении mTLS-соединений) локальное криптоядро шлюза интегрируется со службами Удостоверяющих Центров:

- **Серверы статусов (OCSP):** Автоматическая отправка запросов по протоколу OCSP (RFC 6960) для экспресс-проверки действительности ключей пользователей «на лету» в момент TLS-handshake.
- **Списки отзыва (CRL):** Автоматическое скачивание по расписанию и обновление локальных баз списков отзыва сертификатов (CRL) по протоколам HTTP/HTTPS.
- **Валидация квалифицированности (OIDC-mTLS):** Интеграция со специализированным внешним модулем КриптоАРМ OIDC-mTLS для расширенной проверки юридического статуса и ограничений предъявленного ГОСТ-сертификата перед допуском пользователя к OIDC-авторизации.