

ДВА ПОДХОДА

АСМЕ V2.0 И ЕДИНАЯ СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ СЕРТИФИКАТАМИ (ЕСАУС)



**Максим
АРТЮХИН**
Product owner,
Clearway
Integration

ВВЕДЕНИЕ

В условиях стремительного развития ИТ-инфраструктур управление технологическими сертификатами становится ключевым элементом безопасности серверов, устройств и корпоративных систем. Чем короче срок действия сертификата, тем меньше времени у злоумышленников для его использования в случае компрометации. Поэтому крупнейшие мировые корпорации, формирующие интернет-стандарты, к 2029 году планируют сократить максимальный срок действия TLS-сертификатов с 398 до 47 дней.

Параллельно с этим компании активно усиливают безопасность своих информационных систем: приоритетом становятся защита данных, устойчивость сервисов и контроль доступа. Это стимулирует переход к современным архитектурным решениям, таким как Zero Trust. Один из его типовых механизмов — внедрение взаимной TLS-аутентификации (mTLS), при которой сертификаты X.509 обязательны и для пользователей, и для сервисов — API, баз данных, микросервисов, прокси, инфраструктурных агентов и других компонентов.

Эти тренды кратно увеличивают нагрузку на инфраструктуру открытых ключей (PKI), требуя частого выпуска, ротации и распространения сертификатов. Без полноценной автоматизации это быстро приводит к ошибкам и простоям. Наибольшие

сложности испытывают классические веб-серверы и приложения, где обновление по-прежнему выполняется вручную, а также динамические среды Kubernetes.

В динамичных контейнерных средах Kubernetes и Service Mesh (например, Istio) с использованием mTLS управление сертификатами чрезмерно усложняется: срок действия сертификата должен соответствовать среднему периоду жизни самого пода (или сервиса) — от нескольких часов до нескольких дней. Такой короткий период позволяет минимизировать риски, поскольку потенциально скомпрометированный закрытый ключ быстро становится недействительным даже без явного отзыва. В этом случае применение списков отзыва теряет смысл — они не успевают вступить в силу до истечения сертификата. Важно так же учитывать, что использование в кластере Kubernetes механизмов проверки отзыва сертификатов (CRL/OCSP) создаёт дополнительные накладные расходы и добавляет еще одну точку отказа, что негативно влияет на надежность кластера.

Kubernetes и аналогичные платформы устанавливают новый стандарт производительности для систем выпуска сертификатов. Тысячи подов и сервисных соединений, каждое из которых использует mTLS, создают нагрузку, измеряемую десятками тысяч запросов в минуту. Традиционные PKI-модели, спроектированные для значительно меньших объемов, не выдерживают подобную нагрузку. Для соответствия новым требованиям индустрии нужна инфраструктура, способная легко масштабироваться и реализовать надежную автоматизацию выпуска десятков тысяч сертификатов за считанные минуты.

Рассмотрим два подхода для ответа на новые вызовы ИТ-отрасли: ряд ре-

шений на базе протокола ACME v2.0 и новую отечественную разработку — Единую Систему Автоматического Управления Сертификатами (ЕСАУС) на платформе ЦУГИ.

АСМЕ: ПРОТОКОЛ С ОГРАНИЧЕНИЯМИ

ACME (Automated Certificate Management Environment, RFC8555) — открытый интернет-стандарт автоматизации выпуска, обновления и отзыва SSL/TLS-сертификатов. Он лежит в основе таких известных сервисов, как Let's Encrypt. Благодаря этому миллионы сайтов и сервисов по всему миру могут быстро внедрять HTTPS без сложных ручных процедур. В последние годы всё больше публичных, корпоративных и частных центров сертификации внедряют поддержку этого стандарта, а число совместимых продуктов и инструментов постоянно растёт.

Чтобы понимать, как работает этот протокол, важно знать: подтверждение права владения доменом — фундаментальный этап выпуска сертификата по ACME, который сохраняется с первых версий и остаётся обязательным в актуальной спецификации ACME v2. Автоматизированный клиент (например, certbot или acme.sh) инициирует запрос на сертификат в удостоверяющий центр (УЦ), указывая доменное имя. В ответ УЦ предлагает выполнить один из стандартных вызовов проверки подлинности (challenges): HTTP-01, DNS-01 или TLS-ALPN-01. Только успешно пройдя проверку, заявитель получает сертификат.

Каждый тип проверки имеет свои особенности и риски, которые необходимо учитывать при выборе метода.

HTTP-01 — простой метод, при котором на веб-сервере размещается специальный файл, доступный по порту 80. Подходит для классических сайтов, но в защищенных



инфраструктурах неудобен и менее безопасен. В Kubernetes часто возникают ошибки из-за особенностей работы Ingress Controller: несовпадение маршрутов между ресурсами, ошибки 404 при проверках и другие сбои. Кроме того, открытие порта 80 во внешнюю сеть увеличивает поверхность атаки и создаёт дополнительные риски компрометации сервисов.

DNS-01 — универсальный метод, при котором в DNS через API добавляется TXT-запись без необходимости открывать веб-сервер на порт 80. Используется для wildcard-сертификатов, а также любых доменов и сервисов, где невозможно или неудобно использовать HTTP-01. Основные недостатки: сложность разграничения прав, риск компрометации учётных данных DNS-API, уязвимость к атакам типа DNS cache poisoning (отравление DNS-кеша), что

Тысячи подов и сервисных соединений, каждое из которых использует mTLS, создают нагрузку, измеряемую десятками тысяч запросов к PKI в минуту. Традиционные PKI-модели, спроектированные для значительно меньших объёмов, не выдерживают подобную нагрузку

в обоих случаях может привести к подмене сертификатов. Жёсткая политика прав доступа усложняет эксплуатацию, а прокси-решения вроде астерохи облегчают контроль и аудит, но одновременно с этим увеличивают архитектурную сложность и не устраняют фундаментальные уязвимости DNS.

TLS-ALPN-01 — основан на использовании расширения ALPN в процессе TLS-handshake. Для его работы сервис должен принимать входящее соединение на порт 443 и возвращать временный сертификат с ALPN-значением acme-tls/1. В Kubernetes применение этого метода затруднено, поскольку TLS-терминация обыч-

Многие ACME-решения создают иллюзию простоты и легкости внедрения, но на практике сопровождаются существенными ограничениями и скрытыми трудностями

но выполняется на уровне Ingress Controller, который занимает порт 443 и не передаёт ALPN-информацию в поды. В итоге TLS-ALPN-01 остаётся нишевым, уступая по универсальности и удобству HTTP-01 и DNS-01.

Как видно, протокол ACME изначально не ориентирован на динамичные контейнерные среды, такие как Kubernetes и Istio. Его методы проверки (HTTP-01, DNS-01, TLS-ALPN-01) не учитывают специфику распределённой маршрутизации и безопасного хранения секретов. Кроме того, в ACME отсутствует нативная поддержка сервисной идентичности SPIFFE/SPIRE — ключевого механизма безопасности и аутентификации в Istio. Поэтому для полноценного использования ACME в Kubernetes и Istio необходимы дополнительные компоненты и интеграции, обеспечивающие автоматическое выполнение вызовов проверки, безопасное управление ключами, ротацию сертификатов и поддержку сервисной идентичности.

Многие ACME-решения создают иллюзию простоты и легкости внедрения, но на практике сопровождаются существенными ограничениями и скрытыми трудностями. Чтобы разобраться в специфике построения PKI на базе ACME для Kubernetes и Istio, нужно понимать, из каких ключевых компонентов складывается процесс выпуска сертификатов с использованием внутреннего ACME-сервера и как эти элементы взаимодействуют между собой.

Для построения PKI на ACME для контейнерных сред требуются:

- ♦ **ACME-сервер** (УЦ вне Kubernetes) — принимает запросы на выпуск, обновление и отзыв сертификатов.

- ♦ **Оператор cert-manager** — нативный Kubernetes контроллер, ис-

пользующий API и CRD для автоматической ротации сертификатов и их хранения в Kubernetes Secret (секретах). При этом приватные ключи доступны объектам с соответствующими правами доступа. Реальная защита этих данных напрямую зависит от уровня безопасности кластера, включая настройки RBAC, шифрование etcd, контроль доступа к API и аудит. Поэтому подход хранения ключей в секретах не соответствует принципам Zero Trust. Для повышения уровня защиты рекомендуется использовать специализированные CSI-драйверы, интегрирующиеся с внешними системами управления секретами. Кроме того, для cert-manager регулярно выявляются уязвимости (например CVE-2024-12401 и др.) и отмечены проблемы с масштабируемостью в больших кластерах (>10,000 подов).

- ♦ **CSI-драйверы для сертификатов** (csi-driver) — позволяют хранить секреты вне etcd и интегрироваться с внешними системами управления секретами, обеспечивая безопасное и контролируемое монтирование ключей и сертификатов в поды. Они поддерживают автоматическое обновление без необходимости перезапуска подов. Однако, при компрометации пода остается доступ к монтированным секретам, поэтому важно минимизировать привилегии контейнеров и контролировать обновление секретов. Требуется тщательная настройка версионности и мониторинга, особенно в масштабных средах.

Для интеграции Kubernetes с SPIFFE/SPIRE используется специализированный CSI-драйвер csi-driver-spiiffe. Он автоматически выдает короткоживущие сервисные сертификаты X.509-SVID с под-

держкой идентичности рабочей нагрузки (workload identity).

- ♦ **istio-csr** — интеграционный компонент Istio для работы с внешними системами управления сертификатами, включая cert-manager и корпоративные УЦ. Он автоматизирует подпись и ротацию сертификатов для рабочих нагрузок Service Mesh, используя стандарты SPIFFE для workload identity. Однако, ошибки в настройках доступа (RBAC и других механизмов разграничения прав) могут привести к серьезным уязвимостям.

ACME-клиенты (certbot, acme.sh) применимы лишь в специализированных сценариях, например для ручной выдачи сертификатов отдельным сервисам. Они не имеют нативной интеграции с Kubernetes API и CRD, поэтому не умеют автоматически управлять объектами кластера и не синхронизируются с Ingress или внутренними сервисами вроде ClusterIP. Именно это значительно ограничивает их применение в динамичных контейнерных средах.

Пару слов про ACME-серверы. Их условно делят на публичные и внутренние корпоративные для закрытых сетей. Публичные ACME-серверы всегда сопровождаются серьезными ограничениями на взаимодействия для клиентов: лимиты на количество запросов, требование внешней доступности (порт 80 или доступ к DNS-API), фиксированный срок действия сертификатов (обычно 90 дней), отсутствие контроля над средой применения сертификата. Поэтому многие организации рассматривают вариант развёртывания корпоративного ACME-сервера для внутренней инфраструктуры, чтобы сохранить автономность, адаптировать процессы управления сертификатами под бизнес-требования и исключить внешние зависимости.

Рассмотрим популярные ACME-серверы:

- ♦ **OpenACME** — минималистичный ACME-сервер для внутренних сред, подходящий для тестовых задач и базовой автоматизации выпуска сертификатов. Основные недостатки: сла-

бая поддержка, устаревание кода, нет интеграции с HSM, отсутствие масштабируемости и защиты от современных угроз.

♦ **Pebble** – лёгкое решение, созданное для тестирования ACME-клиентов и CI/CD-автоматизации, отличается простотой установки и настройки. Однако не предназначен для промышленной эксплуатации: отсутствуют механизмы масштабирования, безопасности и отказоустойчивости, нет интеграции с HSM.

♦ **Boulder** – мощная open-source платформа, используемая в Let's Encrypt, оптимальна для публичных центров сертификации. Для корпоративных задач избыточна – очень тяжёлая в сопровождении, требовательная к ресурсам и нуждается в доработках для интеграции с HSM и корпоративными системами управления доступом.

♦ **Smallstep CA**: бесплатная версия очень сильно ограничена по функционалу и содержит ряд уязвимостей (поддерживается только один промежуточный УЦ, базовые политики). Коммерческая версия недоступна для российских компаний.

♦ **EJBCA (Keyfactor)**: бесплатная версия имеет существенные ограничения по функционалу (нет механизмов высокой доступности, базовые роли и политики и т.д.), не гарантирует высокий уровень безопасности и оперативной реакции на угрозы. Все обновления и исправления уязвимостей зависят от активности сообщества. Коммерческая версия недоступна в России.

♦ **Venafi Trust Protection Platform и Hashicorp Vault** – коммерческие решения с поддержкой ACME и автоматизацией жизненного цикла сертификатов, обеспечивающие высокий уровень централизованного контроля и безопасности, но также недоступны для российских компаний.

Мы рассмотрели набор компонентов для построения PKI на базе ACME. На практике такие решения редко работают как единое целое. Каждый модуль создавался под отдельные задачи и развивался в собственном темпе.

При объединении их в единый стек возникает фрагментированная архитектура: разные методы управления, несинхронные обновления, отсутствие централизованной модели сопровождения.

Рассмотрим подробнее ключевые проблемы и вызовы, с которыми придется столкнуться при внедрении подобных стеков в изолированных и корпоративных инфраструктурах.

1. Импортзамещение и нормативные ограничения.

♦ Ограниченная доступность зарубежных продуктов и сервисов, отсутствие адаптации под отечественные регуляторные требования.

♦ Нет поддержки российских криптоалгоритмов (ГОСТ, Российских аппаратных СКЗИ) и отсутствие решений в Едином реестре российского ПО, что ограничивает применение в критически важных информационных инфраструктурах (КИИ) и госорганах.

2. Архитектурная сложность и отсутствие централизованной наблюдаемости.

♦ Логи компонентов разрозненные, форматы не согласованы, штатная интеграция с SIEM и централизованным аудитом, как правило, отсутствует.

♦ Нет сквозной прослеживаемости запросов между разнородными модулями.

♦ Отсутствие необходимых метрик (health-check, ошибки выпуска и продления и т.п.), что усложняет мониторинг состояния PKI.

3. Несовместимость с концепцией Zero Trust.

♦ Модель доверия ACME-решений основана на подтверждении владения доменом, а не на аттестации устройств или сервисов.

♦ Большинство реализаций протокола не предусматривают встроенных механизмов оценки контекста окружения и устройства, которому выдается сертификат – это ограничивает возможность реализации полноценного Zero Trust.

4. Эксплуатационные трудности и высокая зависимость от квалификации персонала.

♦ Поддержка ACME-стека требует глубокого понимания всех его компонентов, работы механизмов Kubernetes и методов безопасного управления ключами.

♦ Разные компоненты поддерживаются разными подразделениями, что затрудняет координацию обновлений, аудит и плановое обслуживание.

♦ Отсутствует централизованная поддержка, из-за чего выявление и устранение ошибок занимает больше времени.

5. Проблемы совместимости и жизненного цикла компонентов.

♦ Разные темпы развития компонентов и несинхронизированные обновления повышают риск несовместимостей и эксплуатации уязвимостей.

♦ Поддержка отдельных open-source проектов может быть приостановлена без предупреждения.

6. Расширение поверхности атаки и уязвимости цепочки поставок.

♦ Каждый дополнительный модуль увеличивает количество потенциальных точек атаки.

♦ Используемые библиотеки могут не проходить централизованную проверку безопасности и содержать скрытые уязвимости.

7. Недостатки в управлении доступом и разграничении ответственности.

♦ Отсутствует делегирование полномочий и чёткое разделение ролей; контроль за выпуском сертификатов фактически сосредоточен у системных администраторов или DevOps-инженеров.

♦ Размыта ответственность за выпуск, продление и отзыв сертификатов, затруднён аудит.

♦ Автоматизация на базе ACME часто обходится без участия службы информационной безопасности, что нарушает внутренние политики и снижает общий уровень контроля.

8. Сложности миграции и технологическая зависимость.

♦ Переход на УЦ другого производителя или национальные алгоритмы (ГОСТ) может потребовать переработки всей архитектуры и замены ключевых компонентов.

9. Масштабируемость и производительность.

♦ Современные Kubernetes-кластеры могут содержать до 5000 узлов и 150 000 подов, что накладывает высокие требования на производительность PKI.

♦ Частая ротация сертификатов (каждые 24–48 часов) генерирует тысячи запросов в минуту, что может вызвать тайм-ауты, сбои в cert-manager и перегрузку УЦ.

Что в итоге?

Качественные коммерческие продукты зарубежных вендоров недоступны российским компаниям, а построение PKI на базе open-source АСМЕ-решений в корпоративных и изолированных инфраструктурах — трудная задача с огромным количеством подводных камней. При попытке объединить разные компоненты в единое и цельное решение будьте готовы к последствиям:

♦ **фрагментированная архитектура с разнотипными механизмами управления;**

♦ **проблемы с несовпадающими жизненными циклами компонентов ПО;**

♦ **отсутствие единой модели сопровождения.**

ЕСАУС

На этом фоне Единая Система Автоматического Управления Сертификатами (ЕСАУС) отечественного производителя Clearway Integration — достойный ответ на современные вызовы. ЕСАУС — это комплексное централизованное решение, которое обеспечивает аудит, объединяет работу с политиками безопасности, поддерживает отечественные криптоалгоритмы, управление ролями и гибкую автоматизацию всех этапов жизненного цикла сертификатов и ключей. Предусмотрена поддержка как классических серверов, так и кластеров Kubernetes и Istio.

ЕСАУС построена на платформе ЦУГИ (№ 13033 в реестре отечественного ПО) из модульных компонентов, легко интегрируется с другими системами на платформе: отечественным УЦ MiniCA,

Система Учета и Управления СКЗИ (СУУ СКЗИ 2.0) и Личным Кабинетом Пользователя Сертификатов (ЛКПС). В основе ЕСАУС — многоуровневая архитектура, разработанная специально для комфортной работы со сложными и сегментированными сетями:

♦ Принцип Zero Trust: специализированные агенты передают не только запрос на сертификат (CSR), но и сведения об окружении, в котором он был сформирован. Эта информация сверяется с утвержденными политиками выпуска сертификатов.

♦ Централизованный аудит: система ведёт полный журнал всех операций, включая сертификаты, которые УЦ обычно не регистрируют в своей базе.

♦ Архитектурные паттерны: постоянная проверка доступности компонентов друг другом; автоматический возврат к восстановившемуся компоненту после его отказа; механизмы коммуникации между компонентами для управления пропускной способностью системы и др.

♦ Гетерогенная инфраструктура: агенты для классических приложений (OS Windows, Linux, macOS), Операторы и Цитадель для кластеров Kubernetes и Istio.

Отдельно отметим, что на волне современных трендов технологический суверенитет от зарубежных решений становится все более важным для корпоративных и государственных ИТ-инфраструктур. Особую ценность приобретает возможность гибкой миграции между разными УЦ (например, с Microsoft CA на отечественный MiniCA или другие отечественные УЦ). Благодаря поддержке широкого спектра УЦ (Microsoft CA, MiniCA, КриптоПро PKI-кластер, SafeTech CA, Aladdin eCA) ЕСАУС позволяет балансировать нагрузку и легко переключаться между УЦ разных вендоров, сохраняя непрерывность работы и надежность бизнес-процессов.

Все это дает возможность с помощью ЕСАУС решать следующие задачи:

♦ Выносить логику УЦ за пределы микросервисных платформ, возвращая контроль службе информационной безопасности, разделяя функций ИБ и администраторов Kubernetes/Istio.

♦ Выдерживать пиковые нагрузки по выпуску сертификатов в больших кластерах Kubernetes и Istio, предотвращая перегрузки на УЦ.

♦ Автоматизировать настройку информационных систем для использования обновленных сертификатов, включая координированное обновление в кластерах и в распределённых системах.

♦ Актуализировать и распространять списки доверенных корневых и промежуточных УЦ по всей организации.

♦ Изолировать УЦ от прямого доступа по сети клиентов и потребителей сертификатов.

ЗАКЛЮЧЕНИЕ

PKI в современных средах Kubernetes и Istio перестала быть просто вспомогательным инструментом и превратилась в стратегическую компетенцию. Удобные на первый взгляд АСМЕ-решения обещают «один клик — и готово», но за этим стоит хрупкая архитектура: разрозненные компоненты, разные жизненные циклы ПО, отсутствие единого управления. Только комплексный подход, сочетающий автоматизацию, контроль, аудит и интеграцию с корпоративными системами, позволит создать по-настоящему надёжную и масштабируемую инфраструктуру доверия для современных облачных платформ.

В этом контексте ЕСАУС — это не просто инструмент, а перспективное решение, которое позволяет перейти от фрагментированного и реактивного управления сертификатами к проактивной и централизованной системе. Автоматизируя рутинные операции, минимизируя человеческий фактор и повышая уровень безопасности, ЕСАУС помогает сократить финансовые затраты за счёт экономии времени высококвалифицированных специалистов и направить освободившиеся ресурсы на решение ключевых задач бизнеса.