

# Установка центра сертификации на предприятии.

## Часть 2 / Хабр

 [habr.com/ru/companies/microsoft/articles/348956](https://habr.com/ru/companies/microsoft/articles/348956)

Alexander Gureev

February 26, 2018



Мы продолжаем нашу серию статей о центре сертификации на предприятии. Сегодня поговорим о построении диаграммы открытого ключа, планировании имен центра сертификации, планировании списков отзыва сертификатов и еще нескольких моментах. Присоединяйтесь!

## Введение

### Словарь терминов

В этой части серии использованы следующие сокращения и аббревиатуры:

- **PKI** (*Public Key Infrastructure*) — инфраструктура открытого ключа, набор средств (технических, материальных, людских и т. д.), распределённых служб и компонентов, в совокупности используемых для поддержки криптозадач на основе закрытого и открытого ключей. Поскольку аббревиатура ИОК не является распространённой, здесь и далее будет использоваться более знакомая англоязычная аббревиатура PKI.
- **X.509** — стандарт ITU-T для инфраструктуры открытого ключа и инфраструктуры управления привилегиями.
- **ЦС** (*Центр Сертификации*) — служба выпускающая цифровые сертификаты. Сертификат — это электронный документ, подтверждающий принадлежность открытого ключа владельцу.

- **CRL** (*Certificate Revocation List*) — список отзыва сертификатов. Подписанный электронный документ, публикуемый ЦС и содержащий список отозванных сертификатов, действие которых прекращено по внешним причинам. Для каждого отозванного сертификата указывается его серийный номер, дата и время отзыва, а также причина отзыва (необязательно). Приложения могут использовать CRL для подтверждения того, что предъявленный сертификат является действительным и не отозван издателем... Приложения могут использовать CRL для подтверждения, что предъявленный сертификат является действительным и не отозван издателем.
- **SSL** (*Secure Sockets Layer*) или **TLS** (*Transport Layer Security*) — технология обеспечивающая безопасность передачи данных между клиентом и сервером поверх открытых сетей.
- **HTTPS** (*HTTP/Secure*) — защищённый HTTP, является частным случаем использования SSL.
- **Internet PKI** — набор стандартов, соглашений, процедур и практик, которые обеспечивают единый (унифицированный) механизм защиты передачи данных на основе стандарта X.509 по открытым каналам передачи данных.
- **CPS** (*Certificate Practice Statement*) — документ, описывающий процедуры управления инфраструктурой открытого ключа и цифровыми сертификатами.

## Общие вопросы планирования

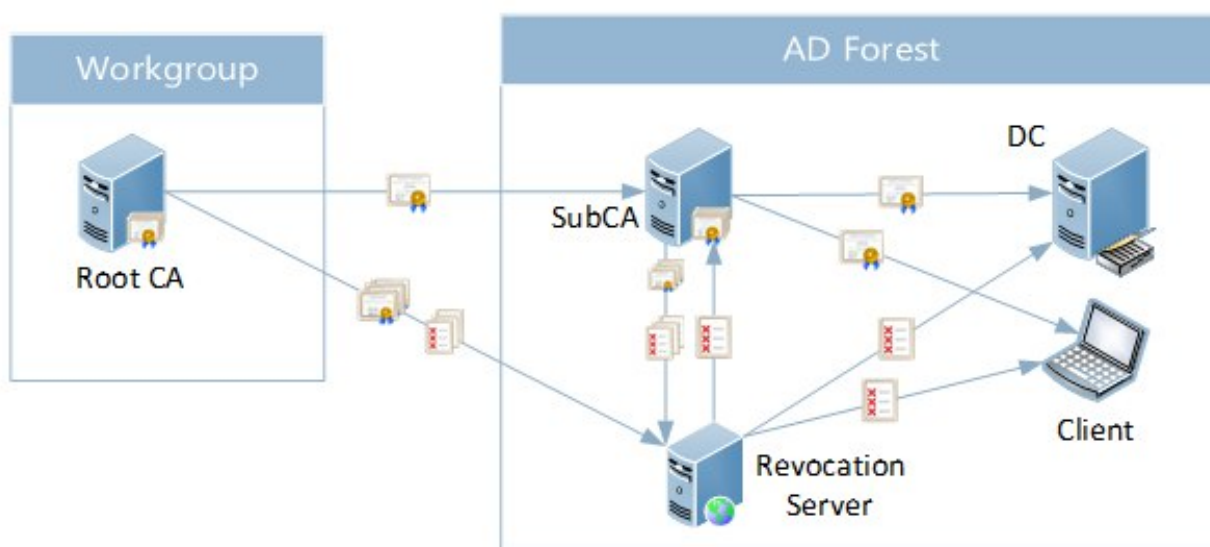
---

Для успешного внедрения любого технического решения необходимо тщательное планирование. Внедрение PKI не является исключением. Более того, если в определённых случаях ошибки изначального планирования могут быть исправлены относительно быстро и легко, то в PKI это однозначно не так. Как я уже отмечал, службы PKI рассчитаны на работу на протяжении многих лет с минимальными (или некритичными) изменениями в ходе работы. Например, срок действия сертификатов CA составляет порядка 10-20 лет. Одна из причин такого долгого срока жизни в том, что перевыпуск этих сертификатов является несколько трудоёмкой операцией и могут потребовать изменений на большом количестве клиентов. Усугубляется это тем, что изменения потребуются и на клиентах, к которым вы можете не иметь доступа. Другой момент заключается в том, что при внесении некоторых изменений в архитектуру PKI вам потребуется поддерживать текущую конфигурацию на всё время жизни уже изданных сертификатов. Иными словами, для новых сертификатов будет действовать новая конфигурация, но параллельно с ней необходимо будет поддерживать и предыдущую конфигурацию, чтобы уже изданные сертификаты могли корректно работать. Это тоже добавляет сложности в поддержке PKI в работоспособном состоянии. Учитывая указанные моменты, к планированию PKI следует подходить самым серьёзным образом. И только тогда PKI будет успешно выполнять свои функции в обеспечении цифровой безопасности в течении продолжительного срока. Многоступенчатый процесс

планирования опирается на логическую диаграмму выбранной модели. На каждой ступени элементы диаграммы разворачиваются (детализуется) и для него формализуются связи, задачи и требования. При необходимости детализация продолжается до тех пор, когда будет получена полностью формализованная система. В этой статье демонстрируется пример такого подхода к планированию.

## Построение диаграммы PKI

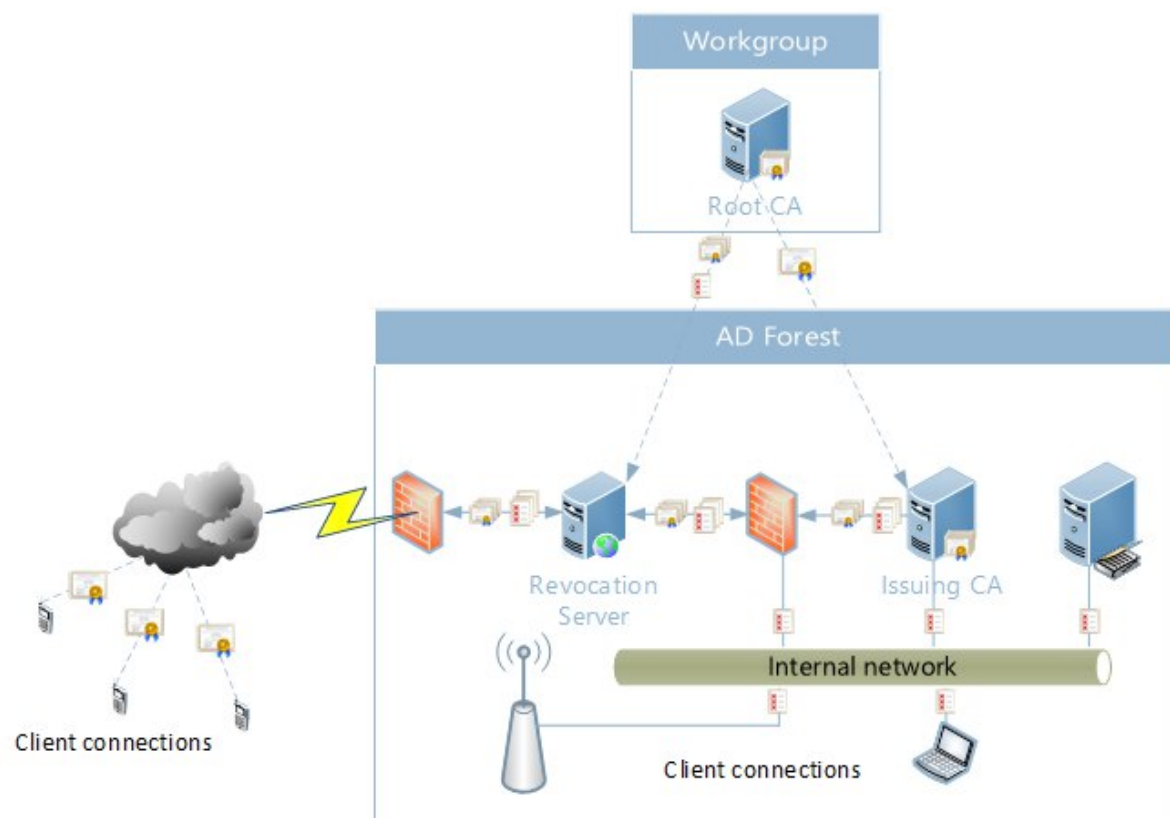
Как я уже говорил, всё начинается с логической диаграммы выбранной модели. Логическая диаграмма отображает все компоненты PKI и она должна быть переложена на физическую топологию. В случае применения двухуровневой модели PKI такая диаграмма может иметь следующий вид:



На диаграмме представлены следующие компоненты и их логические связи:

- **Корневой ЦС** — выдаёт сертификат только подчинённому ЦС и публикует свой сертификат и списки отзыва на сервер отзыва (Revocation Server);
- **Подчинённый (промежуточный) ЦС** — выдаёт сертификаты конечным потребителям и публикует свой сертификат и списки отзыва на сервер отзыва. При этом сам скачивает список отзыва корневого ЦС с сервера отзыва;
- **Сервер отзыва** — является хранилищем сертификатов ЦС и их списков отзыва, которые может скачать любой клиент;
- **Клиентские подключения** — получают свои сертификаты у подчинённого ЦС и скачивают списки отзыва с сервера отзыва.

Физическая топология будет несколько отличаться и иметь следующий вид:



В физической топологии в явном виде выделено, что сервер отзыва доступен для всех клиентов, как внутри, так и снаружи сети, благодаря чему клиенты могут проверять сертификаты в любом месте.

## Планирование имён ЦС

Имя ЦС – это имя, которое будет отображено в поле **Subject** конкретного ЦС. Не путать с именем хоста, на котором работает служба сертификатов. Полное имя ЦС будет состоять из двух компонентов, самого имени (атрибут CN или Common Name) и опционального суффикса в формате X.500. По умолчанию ADCS назначает имя в следующем формате: Для Standalone CA: **<ComputerName>-CA** Для Enterprise CA: **<DomainShortName>-<ComputerName>-CA, <X500DomainSuffix>** Хорошо это или плохо? Технически, вы можете выбрать любое имя, функционально оно ни на что влиять не будет. Есть мнение, что имя вашего ЦС является в некотором роде визитной карточкой вашей PKI, отражая ваше отношение к деталям, которые не имеют непосредственного отношения к функциональности, но обеспечивают достаточный уровень информативности и открытости. Поэтому при выборе полного имени сертификата следует руководствоваться несколькими рекомендациями:

- Имя должно отражать название организации (можно и сокращённое) и роль конкретного ЦС в иерархии (атрибут CN, Common Name);
- Суффикс должен отражать название отдела или подразделения, которое отвечает за его управление в атрибуте OU (Organizational Unit);
- Дублировать полное название организации (атрибут O, Organization);

- Юридическое место дислокации ЦС. Для этого достаточно использовать атрибуты L (Locality) и C (Country). Как правило, это название города и страны, где юридически зарегистрирована организация. Если необходимо, можно указать штат/область посредством атрибута S (State).

Предположим, что вы подбираете имя для корневого ЦС компании Contoso Pharmaceuticals Ltd., которая находится в городе Рига, Латвия и управление обеспечивается отделом информационных технологий. В этом случае имя ЦС может иметь следующий вид:

CN=Contoso Pharm Root Certification Authority, OU=Division Of IT (DoIT), O=Contoso Pharmaceuticals Ltd., L=Riga, C=LV

Следует помнить, что атрибут Country поддерживает только двухбуквенный индекс страны. Например, LV, GB, RU, US и т.д. В качестве дополнительных примеров, можете обратиться к сертификатам ЦС коммерческих провайдеров, как VeriSign/Symantec, DigiCert и т.д. Для подчинённого ЦС это имя будет похожим, за исключением того, что слово Root в имени будет заменено на Subordinate или Issuing. В случае трёхуровневой иерархии, где явно выделяется ЦС политик, слово Root будет заменено на Policy. Как я выше отмечал, в вашей компании могут применяться другие правила, и вы можете их внедрить в имена ЦС, на функциональность это влиять не будет. При этом следует избегать:

- Чрезмерно длинных имён в атрибуте CN (не более 50 символов). При длине атрибута CN свыше 51 символа, оно будет укорочено с пристыковкой хэша отброшенного фрагмента имени в конец имени. Это называется процессом «санитизации» имени, который описан в [§3.1.1.4.1.1](#) протокола [\[MS-WCCE\]](#). Т.е. может случиться так, что при слишком длинном имени слово оборвётся на середине и будет иметь неприглядный вид.
- Использовать буквы, которые не входят в состав латинского алфавита, т.е. никакой кириллицы или диактрических букв (например, ā, ž, Ü, ß). ADCS поддерживает только однобайтовые кодировки для атрибута CN и для ограниченного набора символов. Неподдерживаемые символы будут преобразованы в другую кодировку и станут нечитаемыми. Полный список запрещённых символов представлен в [§3.1.1.4.1.1.2](#) протокола [\[MS-WCCE\]](#). Здесь работает принцип «лучшее – враг хорошему», поэтому имена должны быть достаточно лаконичными и информативными.

## Планирование списков отзыва (CRL)

---

В соответствии с логической диаграммой, каждый ЦС будет публиковать свой

список отзыва. Списки отзыва у нас будут характеризоваться двумя основными категориями:

1. Точки публикации и распространения списков отзыва;
2. Состав и срок действия списков отзыва.

## **Точки публикации и распространения списков отзыва**

---

Для публикации списков отзыва используются два типа точек распространения CRL: точка публикации (куда физический файл будет записываться) и точка распространения (получения) файла. Первый тип точек указывает локальный или сетевой путь (в формате UNC) куда будет записываться файл. Второй тип точек будет регистрироваться в издаваемых сертификатах с указанием пути, по которому клиенты могут скачать список отзыва. Эти пути публикуются в расширении сертификата CRL Distribution Points. Эти пути в общем случае не будут совпадать (кроме протокола LDAP, где пути публикации и распространения совпадают). При определении точек публикации следует руководствоваться следующими правилами:

- Для корневого ЦС указывается строго локальный путь, поскольку этот сервер будет изолирован от сети. Копирование файла на сервер распространения (IIS) будет производиться вручную. Это не проблема, поскольку периодичность публикации списков отзыва для корневого ЦС будет измеряться месяцами (об этом см. далее).
- Для издающих ЦС указывается сетевой путь. Я рекомендую создать общую папку в DFS, которую можно легко определить как виртуальную директорию в IIS. В этом случае процесс публикации физического файла в точку распространения будет полностью автоматизирован.
- Протокол LDAP не должен использоваться для публикации и распространения списков отзыва.

Более детально о планировании расширений CRL Distribution Points и Authority Information Access и практиками вы можете ознакомиться в статье моего блога: [Designing CRL Distribution Points and Authority Information Access locations](#).

## **Состав списков отзыва**

---

Перед планированием состава и срока действия списков отзыва необходимо понять назначение списков отзыва и оптимальные параметры в зависимости от условий их эксплуатации. Как известно, каждый ЦС периодически публикует списки отзывов, которые включают списки всех отозванных конкретным ЦС сертификатов. Причём каждый список включает все отозванные сертификаты за всё время жизни ЦС. При сроке жизни ЦС, например, в 10 лет этот список может вырасти до внушительных размеров (порядка нескольких мегабайт). Даже при наличии высокоскоростных

подключений трафик списков отзыва будет существенным по размеру, т.к. все потребители сертификатов нуждаются в актуальной версии списка отзыва. Для уменьшения трафика списков отзыва предусматривают публикацию двух типов CRL: базовый (Base CRL) и дифференциальные (Delta CRL). Базовый список включает полный список отзыва. Дифференциальный список включает в себя только список отозванных сертификатов, которые были отозваны с момента последней публикации базового CRL. Это позволяет вам публиковать базовый список реже и на более длительный срок, а для ускорения времени реакции клиентов на отозванные сертификаты в промежутке выпускать несколько короткоживущих дифференциальных CRL. Подбор параметров зависит от нескольких факторов. Например, планируемый объём издаваемых сертификатов и планируемый объём отзыва. Рассмотрим типовые сценарии. **Корневой ЦС** Корневой ЦС выписывает сертификаты только промежуточным ЦС, количество которых обычно в пределах десятка. Срок действия промежуточных ЦС сопоставим со сроком жизни сертификата корневого ЦС. Также предполагается, что риск отзыва нижестоящих ЦС весьма низкий, поскольку они управляются обученным персоналом и в отношении них обеспечиваются надлежащие меры безопасности. Поэтому можно утверждать, что объём списка отзыва может включать в себя лишь небольшое количество отозванных сертификатов и, соответственно, файл CRL будет гарантированно маленьким по размеру. *Справка: как посчитать планируемый размер файла CRL исходя из объёмов отзыва? Типичный пустой CRL занимает примерно 600-800 байт. Каждая запись об отозванном сертификате занимает 88 байт. Исходя из этих значений можно высчитать размер CRL в зависимости от количества отозванных сертификатов.* Отсюда следует, что на протяжении всей жизни корневого ЦС список отзыва будет в пределах 1кб и смысла в дифференциальном CRL нет. **Издающий ЦС** Для издающего ЦС картина меняется. Объём издаваемых сертификатов уже высок, он может составлять тысячи и миллионы штук. Потребителями являются пользователи и устройства, которые обладают высоким риском отзыва, поскольку они не находятся под постоянным контролем квалифицированного персонала, и невозможно обеспечить надлежащие меры. Как следствие, список отзыва может достигать серьёзных размеров. Например, если заложить 10% риск отзыва, то на миллион изданных сертификатов приходится порядка 100к отозванных. 100к записей по 88 байт будет составлять немногим меньше 10мб. Очень часто обновлять файл на 10мб не очень практично, целесообразней его публиковать реже, а в интервале между публикациями основного CRL распространять несколько облегчённых дифференциальных Delta CRL. Т.е. если для корневого ЦС достаточно только базового списка отзыва, то для ЦС, выпускающих сертификаты конечным потребителям, следует применять и дельты.

## Планирование сроков действия CRL

---

Это всё было о составе списков отзыва для каждого ЦС. Теперь следует определить сроки:



- На какой срок следует публиковать список отзыва?
- Как долго информация в нём может считаться достоверной и достаточно актуальной?

Здесь тоже можно применить подход в зависимости от условий эксплуатации. Риск отзыва промежуточного ЦС весьма низкий, следовательно, нет смысла слишком часто публиковать пустой CRL. В современной практике применяются следующие типовые значения по сроку действия CRL для ЦС, которые выписывают сертификаты только другим ЦС: 3, 6 или 12 месяцев. Здесь следует отталкиваться от степени риска и административных расходов на обслуживание списков отзыва. Если нет никаких особых условий, то я рекомендую выбирать что-то среднее, порядка 6 месяцев. Для подчиненных ЦС схема такая же. Поскольку риск отзыва клиентских сертификатов высокий, то можно предположить и высокую частоту отзыва. Следовательно, таким ЦС следует выполнять публикацию списков отзыва гораздо чаще, а для экономии трафика комбинировать базовые и дифференциальные CRL. По умолчанию Microsoft CA публикует списки отзыва со следующей периодичностью: базовый CRL раз в неделю, дельты – ежедневно. В этой ситуации клиенты будут оповещены о последних отозванных сертификатах в пределах суток. Можно понять желание администраторов уменьшить это время (в идеале – мгновенно), чтобы клиенты не признавали отозванный сертификат действительным. Однако, уменьшение одного риска приводит к увеличению другого риска. Представьте, что по какой-то причине отказал сервер ЦС в момент, когда предыдущий CRL близок к истечению срока действия, а новый CRL невозможно опубликовать. Тогда начнутся проблемы с проверкой отзыва сертификатов и их остановке, пока сервер ЦС не будет восстановлен в работе. Этот момент необходимо обязательно учитывать при настройке сроков действия списков отзыва. Microsoft CA по умолчанию уже закладывает некоторый резерв по времени на непредвиденные случаи и когда распространение списков отзыва по всем точкам публикации занимает некоторое время (например, вызваны латентностью репликации). Этот резерв в английской терминологии называется CRL overlap. Идея защитного механизма заключается в том, что ЦС генерирует и публикует списки отзыва до истечения действия предыдущего опубликованного списка. Это достигается использованием двух полей в списке отзыва: Next CRL Publish и Next Update. Поле Next CRL Publish указывает на время, когда ЦС опубликует обновлённый список отзыва (автоматически). Next Update указывает на время, когда срок действия текущего списка истечёт. Поле Next Update будет всегда выставлен на несколько позднее время, чем Next CRL Publish. Другими словами, ЦС опубликует обновлённый список отзыва до истечения срока предыдущего. Алгоритм вычисления автоматических значений для этих полей нетривиален и описан в следующей статье: [How ThisUpdate, NextUpdate and NextCRLPublish are calculated \(v2\)](#). Если значения по умолчанию вас не устраивают по тем или иным причинам, их можно отредактировать. Необходимо учитывать, что запас по времени имеет нижние и верхние границы. Например, верхняя граница не может превышать срока действия самого CRL. Так, если срок действия CRL составляет 1 день, то запас



может составлять максимум 1 день, и тогда ЦС будет публиковать списки отзыва ежедневно, но срок действия будет составлять 2 дня. Тем самым достигается запас времени на восстановление ЦС в случае непредвиденных обстоятельств. *На практике я достаточно часто наблюдал желание администраторов закрутить настройки сроков действия CRL до минимального предела с таким обоснованием: «пользователь уволился и не должен иметь возможность аутентифицироваться с отозванным сертификатом».* Мотивация понятна, но решение задачи через списки отзыва не совсем правильно. В случае, если пользователю необходимо прекратить доступ к корпоративным системам, необходимо отключить учётную запись пользователя или компьютера. При планировании сроков действия CRL и периодичности следует руководствоваться следующими рекомендациями:

- Все ЦС, которые выдают сертификаты только другим ЦС (не конечным потребителям), должны публиковать CRL сроком действия от 3-х до 12 месяцев с запасом в один месяц.
- Все ЦС, которые выдают сертификаты конечным потребителям (пользователям и устройствам), должны публиковать базовые CRL не реже одного раза в неделю и дифференциальные списки не реже 3-х дней (желательно, ежедневно). Запас по времени не следует корректировать (используйте тот, который будет автоматически высчитан внутренней логикой ЦС).

## Online Certificate Status Protocol

---

В рамках данного цикла статей я не буду использовать OCSP серверы для дополнительного метода распространения информации об отозванных сертификатах. При желании вы можете обратиться к исчерпывающей статье на сайте TechNet: [Online Responder Installation, Configuration, and Troubleshooting Guide](#). Как показывает практика, в большинстве случаев установка и поддержка OCSP не оправдывает себя по ряду причин. Основная задача OCSP: разгрузка трафика скачивания CRL. Как известно, CRL содержит список всех отозванных сертификатов за всё время жизни ЦС, и в какой-то момент при интенсивном отзыве сертификатов его размер может достичь внушительных размеров (несколько мегабайт). Выше уже отмечалось, что 100к отозванных сертификатов составит порядка 9МБ в CRL файле. В то время как проверка отзыва любого сертификата при использовании OCSP будет занимать фиксированный размер ~2.5КБ. Есть ощутимая разница. На практике же, зачастую интенсивность отзыва гораздо ниже. Если говорить о корневых ЦС или ЦС политик, у них отзыв будет штучный, и размер их списка отзыва едва ли превысит 1КБ. Следует отметить, что OCSP может быть эффективным в ситуации, когда есть один проверяемый сертификат и много клиентов, которые его хотят проверить. Это типичный сценарий сертификата SSL/TLS. В этом случае каждый клиент вместо скачивания условного 9МБ списка отзыва потратит 2.5КБ трафика OCSP. Но в обратной ситуации (один сервер

проверяет множество клиентских сертификатов) OCSP может вызвать значительную нагрузку на сеть. К этому можно отнести типичные сценарии корпоративных сетей: аутентификация клиентов при помощи сертификатов, такие как аутентификация EAP-TLS в беспроводных сетях и VPN, аутентификация Kerberos на контроллерах домена. Предположим, сотрудники пришли на работу и используют сертификаты для аутентификации в сети (смарт-карты, сертификаты на мобильных устройствах) и контроллер домена, Серверы RADIUS вынуждены проверять каждый клиентский сертификат. Для проверки только 1K сертификатов будет затрачено 2.5МБ трафика. В этой ситуации пользы от OCSP никакой, даже наоборот. Этот аспект учтен в логике продуктов Microsoft. Если за определённый промежуток времени клиент Crypto API проверяет 50 (это значение можно настроить) сертификатов от одного издателя при помощи OCSP, тогда на этом работа с OCSP заканчивается, и клиент скачивает и кэширует CRL для этого издателя. Более подробно с этим поведением можно ознакомиться в разделе [Optimizing the Revocation Experience](#) документа [Certificate Revocation Checking in Windows Vista and Windows Server 2008](#).

## Планирование политик выдачи сертификатов

---

Политики выдачи сертификатов являются одним из самых сложных для понимания аспектов в работе сертификатов и зачастую полностью игнорируется администраторами при планировании и развёртывании PKI на предприятии. Однако понимание и умение управлять политиками выдачи даёт нам более гибкую систему, дополнительный уровень контроля и, в конце концов, как метод описания и документирования PKI.

### Определение политик

---

Для начала необходимо ввести определение политик выдачи сертификатов. Любой процесс выдачи/получения сертификата по сути является контрактом между получателем сертификата и издающим ЦС. Этот контракт определяет множество аспектов, таких как порядок выдачи, использования и зоны ответственности. В каждой компании могут существовать различные методы проверки заявок и выдачи сертификатов. Рассмотрим несколько типовых случаев:

- Сертификаты для подписи электронной почты могут выдаваться автоматически с минимальной проверкой заявителя (только на основании успешной аутентификации пользователя в Active Directory). Никаких дополнительных действий для выпуска этих сертификатов не проводится.
- Сертификаты для цифровой подписи документов могут выдаваться только после согласования с непосредственным руководителем и предоставления письменной заявки со всеми необходимыми подписями.
- Сертификаты для смарт-карт могут выдаваться только при личном присутствии работника наряду с инструктажем по правилам использования карт, подписанием соответствующих регулирующих документов.

- Все пользователи могут получить сертификат аутентификации для доступа к беспроводной сети с мобильных устройств, но доступ к критическим системам будет разрешён, если аутентификация была выполнена только с помощью смарт-карт.

Все эти сценарии имеют чётко выраженное различие в порядке выдачи сертификатов. В одном случае достаточно зарегистрироваться в системе, и можно сразу получить сертификат. В другом случае нужно пройти процедуру согласования заявки, в третьем — необходимо личное присутствие заявителя и т.д. Также, политика определяет правила использования полученного сертификата, процедуры обновления или аннулирования сертификата и действия в нестандартных ситуациях (например, при утере или краже сертификата с ключами). Вот эти различия в процедурах и являются политиками выдачи, и эти политики должны регистрироваться в сертификатах. Конечные приложения могут использовать политики выдачи для определения доступа к ресурсу. Наиболее известными примерами таких приложений являются **Network Policy Server (NPS)** и **Active Directory Dynamic Access Control**. Например, все сотрудники компании могут подключаться при помощи сертификатов к общей беспроводной сети компании. Но могут быть беспроводные сети, доступ к которым будет разрешён только при использовании сертификатов на смарт-картах. В NPS можно настроить правило, что будет приниматься не просто сертификат входа, а тот, который был выдан в соответствии с политикой выдачи сертификатов для смарт-карт. Поскольку эта информация отражена в сертификате, NPS может различить два похожих сертификата (оба для аутентификации пользователя) по политикам выдачи. Если сертификат не содержит свидетельств, что он был выдан в соответствии с указанной политикой выдачи, то доступ к сети не будет разрешён. Похожий принцип заложен и в Active Directory Dynamic Access Control, где можно указать критерии для различных уровней доступа.

## Описание политик

---

Политики выдачи просто характеризуют в общих чертах набор процедур и процессов, выполняемых при выдаче тех или иных сертификатов. Следует понимать, что в программном коде никак не проверяется, соблюдались эти процедуры на самом деле или нет. Если на уровне кода невозможно проверить их выполнение, то зачем они? На этот вопрос есть два ответа. Первый заключается в том, что ряд ИТ-процессов невозможно отследить на программном уровне. Они проверяются на соответствие принятым правилам аудитом, который проводится людьми. Чаще всего в качестве аудиторов выступают сторонние организации, имеющие компетенцию в рассматриваемых вопросах. Это касается и политик выдачи сертификатов. В частности, при создании доверия (на уровне PKI и сертификатов) между организациями, они предоставляют документы, описывающие процессы и заказывают сторонний аудит для проверки того, что эти процессы соблюдаются. Второй ответ вытекает из первого: описание политик выдачи

сертификатов в конечном итоге станет документацией на всю PKI и будет иметь своё фирменное название – **Certificate Practice Statement** или CPS (к сожалению, не нашёл подходящего термина на русском языке, поэтому буду использовать англоязычную аббревиатуру). Для унификации описания политик выдачи существует рекомендованный (но не обязательный) шаблон, [описанный в RFC 3647](#). По сути, этот документ является фреймворком по написанию политик и освещает всевозможные аспекты работы PKI. Совершенно не обязательно описывать все имеющиеся разделы в документе. Можно документировать только то, что применимо к вашей ситуации, или добавлять что-то своё. В общем случае CPS будет состоять из двух частей:

1. Описание иерархий PKI, общих для всех процедур и положений, которые будут общими для всех конкретных политик выдачи.
2. Описание положения специфичные для конкретной политики выдачи. В зависимости от размерности PKI, её особенностей, на каждую политику может составляться отдельный CPS, но чаще всего это сводится к составлению единого документа, который будет описывать всё.









## Программирование политик

---

В процессе составления CPS будет определена одна или несколько уникальных политик выдачи (как минимум одна будет обязательно). Каждая политика выдачи должна иметь уникальный идентификатор и указатель на CPS (гиперссылка или текстовая строка). Идентификаторы политик должны быть представлены в формате ITU-T и ISO. В своё время я написал небольшую вводную статью про объектные идентификаторы: [Что в OID'е тебе моём?](#) В ней есть информация о том, как получить в IANA (Internet Assigned Numbers Authority) свою ветку идентификаторов. Получив свою ветку, вы внутри неё выделяете подмножество идентификаторов для политик выдачи, например: 1.3.6.1.4.1.x.1, где x – уникальный идентификатор компании, зарегистрированный в IANA. И в нём вы регистрируете конкретные политики выдачи:

- 1.3.6.1.4.1.x.1.1
- 1.3.6.1.4.1.x.1.2
- 1.3.6.1.4.1.x.1.3
- 1.3.6.1.4.1.x.1.4
- ...

Далее вы назначаете конкретные политики выдачи для ЦС, которые будут их поддерживать. Например, у вас может быть несколько издающих ЦС, каждый из которых будет поддерживать только определённые политики. Список поддерживаемых политик будет содержаться в расширении Certificate Policies сертификата ЦС, а также в сертификатах конечных потребителей.

Field	Value
 Enhanced Key Usage	Server Authentication (1.3.6....
 CRL Distribution Points	[1]CRL Distribution Point: Distr...
 Certificate Policies	[1]Certificate Policy:Policy Ide...
 Authority Information Access	[1]Authority Info Access: Acc...
 1.3.6.1.4.1.11129.2.4.2	04 82 01 6b 01 69 00 76 00 a4...
 Key Usage	Digital Signature, Key Encipher...
 Basic Constraints	Subject Type=End Entity, Pat...
 Thumbprint algorithm	sha 1

```
[1]Certificate Policy:
  Policy Identifier=2.16.840.1.114412.2.1
  [1,1]Policy Qualifier Info:
    Policy Qualifier Id=CPS
    Qualifier:
      https://www.digicert.com/CPS
[2]Certificate Policy:
  Policy Identifier=2.23.140.1.1
```

Например, на картинке показан сертификат DigiCert, выданный в соответствии с политикой выдачи под идентификатором 2.16.840.1.114412.2.1 (в данном случае это **Extended Validation**) и 2.23.140.1.1 (указывает, что ЦС выпускает сертификаты согласно правилами CAB/Forum) и с ссылкой на CPS. По этой ссылке можно скачать самую последнюю версию их CPS и ознакомиться с условиями получения и использования сертификатов. Когда все политики определены, им присвоены идентификаторы и определены указатели, эта информация помещается в конфигурационный файл установки ЦС, чтобы эти сведения попали в сертификат. Для включения информации о политиках выдачи конечных сертификатов следует настроить шаблоны сертификатов и для каждого шаблона указать конкретные политики выдачи. Причём, если какая-то политика включена в конечный сертификат, её идентификатор должен быть представлен как в самом конечном сертификате, так и во всех промежуточных сертификатах цепочки (кроме корневого ЦС). Более подробно об этом можно прочесть в статьях моего блога: [Certificate Policies extension – all you should know \(part 1\)](#) и [Certificate Policies extension – all you should know \(part 2\)](#). Эти статьи освещают общие вопросы, связанные с политиками выдачи, правилами их проверки в цепочках сертификатов и способы их включения в сертификаты на платформе Windows. Здесь есть один тонкий момент: сертификат ЦС выдаётся на довольно большой срок (10 лет и более) и при добавлении или удалении политик выдачи на конкретном ЦС необходимо перевыпускать сертификат самого ЦС, что усложняет процесс управления ЦС и увеличивает административные расходы. Не существует способа описать группу политик одним идентификатором (например, идентификатором компании), поскольку маски не

поддерживаются. В [RFC 5280 §4.2.1.4](#) предусмотрен глобальный идентификатор (global wildcard) anyPolicy = 2.5.29.32.0, который покрывает любые политики выдачи. Технически, можно использовать один этот идентификатор в сертификате ЦС, тогда этот ЦС может выдавать сертификаты по любым политикам. Его использование не рекомендовано, т.к. при аудите невозможно определить, по каким конкретно политикам происходит выдача сертификатов на конкретном ЦС и, если будет проводиться внешний аудит, вполне возможно, что к идентификатору anyPolicy могут быть претензии, что повлечет необходимость указывать политики в явном виде. Но это очень сильно зависит от местных условий, поэтому на раннем этапе можно использовать и этот идентификатор в сертификате ЦС и уже указывать конкретные политики выдачи в конечных сертификатах. В рамках данных статей я буду использовать anyPolicy в сертификате издающего ЦС.

## Планирование аппаратных требований

---

Службы сертификатов AD CS в целом нетребовательны к аппаратным ресурсам (на фоне других серверных служб). Основная нагрузка ложится на центральный процессор для выполнения криптографических операций (хэширование, шифрование и подпись). Кроме того, есть определённая нагрузка на диски для работы базы данных сертификатов (AD CS использует JET Database Engine). Это в теории. На практике аппаратных ресурсов даже бюджетных линеек серверов будет более чем достаточно для функционирования служб сертификатов, поскольку реальные потребности весьма малы на фоне требований ОС к аппаратным ресурсам. В эпоху Windows Server 2003 была написана статья [Evaluating CA Capacity, Performance, and Scalability](#) (ссылка на архивную копию, т.к. оригинал удалён с сайта TechNet), освещающую общие моменты, которые нужно учитывать при планировании аппаратных ресурсов. Статья несколько устарела (например, лимит количества выдаваемых сертификатов в разрезе БД значительно увеличен), но общие характеристики мало изменились. В 2010 году, Windows PKI Team провела тесты производительности с использованием более современного оборудования (образца 2007 года) под управлением Windows Server 2008. С результатами можно ознакомиться в их блоге: [Windows CA Performance Numbers](#). Данные в этих статьях, говорят о том, что сервер AD CS на аппаратном сервере выпуска 2007 года позволяет выпускать порядка 150 сертификатов в секунду. Реальная же потребность в скорости выдачи сертификатов на порядки ниже. Поэтому для ЦС общего назначения советую ориентироваться на рекомендуемые аппаратные требования к самой ОС. Это касается как корневого, так и издающих ЦС. Поэтому перечислю здесь требования для Windows Server 2016 ([Windows Server 2016 System Requirements](#)):

- CPU — dual-core 1.4 GHz;
- Память — 1GB RAM;
- Диск — 48 GB для системного диска и не менее 48 GB для локальных резервных копий. Системный раздел желательно разместить на дисках с RAID1.

- Дисплей — SVGA (800\*600);
- Сеть — стандартный сетевой интерфейс.

Здесь следует отметить, что в конфигурации для корневого ЦС будет отсутствовать (или должен быть отключен) сетевой интерфейс (поскольку он не будет подключен к сети) и присутствовать хотя бы один интерфейс для съёмных носителей.

## Итоговая конфигурация

---

По итогам планирования весьма полезно логически сгруппировать и наглядно представить основные компоненты (и их значения), которые потребуется сконфигурировать в процессе развёртывания. Можно использовать различные форматы, например, таблицы. Данные из этих таблиц будут использоваться в конфигурационных файлах и скриптах.

### Корневой ЦС

---

Для установки и создания корневого ЦС вам потребуется определить и сконфигурировать параметры сертификата и сервера ЦС в соответствии с представленными в следующей таблице значениями.

Название параметра	Значение параметра
<b>Сервер ЦС</b>	
Класс ЦС	Standalone CA
Тип ЦС	Root CA
Срок действия издаваемых сертификатов	15 лет
Публикация в AD (контейнеры)	Certification Authorities AIA
Точки публикации CRT	1) По-умолчанию 2) C:\CertData\contoso-rca<CertificateName>.crt 3) IIS:\inetPub\PKIdata\contoso-rca<CertificateName>.crt*
Точки распространения CRT	1) URL=http://cdp.contoso.com/pki/contoso-rca<CertificateName>.crt
Точки публикации CRL	1) По-умолчанию 2) C:\CertData\contoso-rca<CRLNameSuffix>.crt 3) IIS:\inetPub\PKIdata\contoso-rca<CRLNameSuffix>.crt*



Точки распространения CRL	1) URL=http://cdp.contoso.com/pki/contoso-rca<CRLNameSuffix>.crl
<b>Сертификат</b>	
Имя сертификата	Contoso Lab Root Certification authority
Дополнительный суффикс	OU=Division Of IT, O=Contoso Pharmaceuticals, C=US
Провайдер ключа	RSA#Microsoft Software Key Storage Provider
Длина ключа	4096 бит
Алгоритм подписи	SHA256
Срок действия	15 лет
Состав CRL	Base CRL
<b>Base CRL</b>	
Тип	Base CRL
Срок действия	6 месяцев
Расширение срока действия	1 месяц
Алгоритм подписи	SHA256
Публикация в AD	Нет

\* — копируется на сервер IIS вручную.

### Издающий ЦС

Аналогичная таблица составляется и для издающего ЦС.

Название параметра	Значение параметра
<b>Сервер ЦС</b>	
Класс ЦС	Enterprise CA
Тип ЦС	Subordinate CA
Срок действия издаваемых сертификатов	Максимально: 5 лет (остальное контролируется шаблонами сертификатов)
Автоматическая загрузка шаблонов	Нет

Публикация в AD (контейнеры)	AIA NTAuthCertificates
Состав CRL	Base CRL Delta CRL
Точки публикации CRT	1) По-умолчанию 2) \\IIS\PKI\contoso-pica<CertificateName>.crt
Точки распространения CRT	1) URL=http://cdp.contoso.com/pki/contoso-pica<CertificateName>.crt
Точки публикации CRL	1) По-умолчанию 2) \\IIS\PKI\contoso-pica<CRLNameSuffix><DeltaCRLAllowed>.crl
Точки распространения CRL	1) URL=http://cdp.contoso.com/pki/contoso-pica<CRLNameSuffix><DeltaCRLAllowed>.crl
<b>Сертификат</b>	
Имя сертификата	Contoso Lab Issuing Certification authority
Дополнительный суффикс	OU=Division Of IT, O=Contoso Pharmaceuticals, C=US
Провайдер ключа	RSA#Microsoft Software Key Storage Provider
Длина ключа	4096 бит
Алгоритм подписи	SHA256
Срок действия	15 лет (определяется вышестоящим ЦС)
Политики выдачи	1) Имя: All Issuance Policies OID=2.5.29.32.0 URL=http://cdp.contoso.com/pki/contoso-cps.html
Basic Constraints	isCA=True (тип сертификата — сертификат ЦС) PathLength=0 (запрещается создание других промежуточных ЦС под текущим ЦС).
<b>Base CRL</b>	
Тип	Base CRL
Срок действия	1 неделя
Расширение срока действия	По умолчанию
Алгоритм подписи	SHA256
<b>Delta CRL</b>	
Тип	Delta CRL

Срок действия	1 день
Расширение срока действия	По-умолчанию
Алгоритм подписи	SHA256

## Сервер IIS

Название параметра	Значение параметра
<b>Сервер ЦС</b>	
Веб-сайт	cdp
Заголовок хоста	cdp.contoso.com
Виртуальные директории	PKI=C:\InetPub\wwwroot\PKI\data
Double Escaping	Включен

Как я отмечал, эти таблицы будут использоваться для составления конфигурационных файлов и скриптов. После установки их можно использовать для проверки того, что все фактические значения конфигурации соответствуют запланированным.

## Об авторе



**Вадим Поданс** — специалист в области автоматизации PowerShell и Public Key Infrastructure, Microsoft MVP: Cloud and Datacenter Management с 2009 года и автор модуля PowerShell PKI. На протяжении 9 лет в своём блоге освещает различные вопросы эксплуатации и автоматизации PKI на предприятии. Статьи Вадима о PKI и PowerShell можно найти на его [сайте](#).