ГОСУДАРСТВЕННЫЙ СТАНДАРТ УЗБЕКИСТАНА

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

Структура сертификата открытого ключа ЭЦП и сертификата атрибута

Издание официальное

Узбекское агентство стандартизации, метрологии и сертификации

Ташкент

Предисловие

1 РАЗРАБОТАН Государственным унитарным предприятием Центр научно-технических и маркетинговых исследований - «UNICON.UZ» (ГУП «UNICON.UZ»)

2 ВНЕСЕН Техническим комитетом по стандартизации в сфере связи и информатизации №7

3 УТВЕРЖДЕН постановлением Узбекского агентства стандартизации, метрологии и сертификации (агентство «Узстандарт») от 01.06.2011 № 05-304

4 ВЗАМЕН O‘z DSt 1108:2006

Информация о введении в действие (прекращении действия) настоящего стандарта и изменений к нему на территории Узбекистана публикуется в указателе, издаваемом агентством «Узстандарт». В случае пересмотра или отмены настоящего стандарта соответствующая информация будет опубликована в информационном указателе, издаваемом агентством Узстандарт

Исключительное право официального опубликования настоящего стандарта на территории Узбекистана принадлежит агентству «Узстандарт»

Содержание

1 Область применения 1
2 Нормативная ссылка....1
3 Термины, определения и сокращения 2
3.1 Термины и определения 2
3.2 Сокращения 5
4 Общие положения 6
5 Открытые ключи и сертификаты открытых ключей 7
5.1 Генерация пары ключей 7
5.2 Создание сертификата открытого ключа 8
5.3 Действительность сертификата 16
5.4 Дополнения X.509 версии 3....19
5.5 Отказ от цифровой подписи....27
6 Расширение сертификата открытого ключа и расширение CRL 27
6.1 Обработка политики 27
6.2 Расширения информации о ключе и о политике....33
6.3 Расширения информации о субъекте и издателе 40
6.4 Расширения ограничения пути сертификации....43
6.5 Расширение базового CRL 49
6.6 Расширения пункта распределения CRL и дельта-CRL 59
7 Дельта CRL, имеющий связь с базой 66
8 Процедура обработки пути сертификации 68
8.1 Входные данные обработки пути сертификации 68
8.2 Выходные данные обработки пути сертификации 69
8.3 Переменные величины обработки пути сертификации 69
8.4 Этап инициализации 70
8.5 Обработка сертификата....70
9 Схема директории PKI 74
9.1 Классы объекта директории PKI и формы названия 74
9.2 Атрибуты директории PKI 75
9.3 Правила соответствия директории PKI 79
10 Сертификаты атрибута 85
10.1 Структура сертификата атрибута....85
10.2 Пути сертификата атрибута 90
11 Взаимоотношения органа по присвоению атрибутов, источника полномочий и сертифицирующего органа 91
11.1 Привилегия в сертификатах атрибута 91
11.2 Привилегия в сертификатах открытого ключа 92
12 Модели PMI 93
12.1 Общая модель 93
12.2 Модель управления PMI 95
12.3 Модель делегирования 96
12.4 Модель ролей 98

ЭЛЕКТРОН ЭЛЕМЕНТЛАРИ
СТАНДАРТЛАШТИРИШ
МИЛЛИЙ ТЕХНИК КЕНГАШИ
РАЗОРАТИ
ВА ДАВЛАТ
ТАШКИЛОТЛАРИ

12.5 Признание полномочий модели.....100
12.6 XML привилегии атрибутивной информации.....104
12.7 Атрибут разрешения и согласования правил.....105
13 Расширения сертификата для управления привилегиями 106
13.1 Базовое расширение управления привилегиями 106
13.2 Расширения аннулирования привилегий 110
13.3 Расширения источника полномочий.....112
13.4 Расширения роли 114
13.5 Расширение делегирования 116
13.6 Признание полномочий расширений.....123
14 Процедура обработки пути привилегий.....126
14.1 Процедура простой обработки 126
14.2 Процедура обработки роли 128
14.3 Процедура обработки делегирования 128
15 Схема директории PMI 131
15.1 Класс объекта директории PMI 131
15.2 Атрибуты директории PMI 133
15.3 Общие правила соответствия директории PMI 135
16 Аутентификация директории 137
16.1 Процедура простой аутентификации 137
16.2 Жесткая аутентификация 140
Приложение А (обязательное) Структура сертификата открытого ключа и сертификата атрибута.....146
Приложение В (обязательное) Правила генерации и обработки CRL объекта.....170
Приложение С (обязательное) Эталонное определение алгоритма идентификаторов сертификата атрибута 180
Библиография 181

ГОСУДАРСТВЕННЫЙ СТАНДАРТ УЗБЕКИСТАНА

Ахборот технологияси
ОЧИҚ ТИЗИМЛАР ЎЗАРО БОҒЛИҚЛИГИ
ЭРИ очиқ калити сертификати ва атрибут сертификатининг тузилмаси

Информационная технология
ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ
Структура сертификата открытого ключа ЭЦП и сертификата атрибута

Information technologies
OPEN SYSTEM INTERCONNECTION
Public key and attribute certificate frameworks

Дата введения 01.07.2011
Год от

1 Область применения

Настоящий стандарт предназначен для обеспечения требований безопасности в области аутентификации и других услуг безопасности, посредством установления структур, на которых могут быть основаны другие услуги безопасности. Данный стандарт определяет структуру для сертификатов открытого ключа, сертификатов атрибута, услуг аутентификации.

Данный стандарт распространяется на сертификаты открытого ключа электронной цифровой подписи и сертификаты атрибута органов государственной власти и управления, хозяйствующих субъектов, функционирующих в Республике Узбекистан.

2 Нормативные ссылки

В настоящем стандарте использована ссылка на следующие стандарты:

O‘z DSt 1047: 2003 Информационные технологии. Термины и определения

O‘z DSt 1109:2006 Информационная технология. Криптографическая защита информации. Термины и определения

O‘z DSt ISO/IEC 2382-8:2007 Информационные технологии. Информационная безопасность. Термины и определения

Издание официальное

O'ZSTANDART AGENTLIGI
2031 son bilan
«01» 06 2011 yilda
Davlat ro'yxatiga olindi

Примечание – При пользовании настоящим стандартом целесообразно проверить действие ссылочного стандарта на территории Узбекистана по соответствующему указателю стандартов, составленному по состоянию на 1 января текущего года, и по соответствующим информационным указателям, опубликованным в текущем году. Если ссылочный документ заменен (изменен), то при пользовании настоящим стандартом следует руководствоваться замененным (измененным) стандартом. Если ссылочный документ отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящем стандарте применены термины по O‘z DSt 1047, O‘z DSt 1109 и O‘z DSt ISO/IEC 2382-8, а также следующие термины с соответствующими определениями:

3.1.1 **базовый CRL:** CRL, который используется как основа при генерации дельта-CRL.

3.1.2 **верификатор привилегий:** Объект, удостоверяющий подлинность сертификатов в соответствии с политикой привилегий.

3.1.3 **владелец:** Объект, которому были делегированы некоторые привилегии непосредственно одним источником полномочий (центром регистрации) или косвенно через центр по присвоению атрибутов (центром регистрации, присваивающим атрибуты).

3.1.4 **доверие:** Убежденность первого объекта в том, что второй объект будет вести себя точно так, как ожидает первый объект.

3.1.5 **доверяющая сторона:** Пользователь или агент, который полагается на данные в сертификате при принятии решений.

3.1.6 **делегирование:** Передача привилегий от одного объекта, который владеет определенной привилегией, другому объекту.

3.1.7 **дельта-CRL:** Неполный список аннулирования, который содержит только список, содержащий записи только для тех сертификатов, статус аннулирования которых изменялся с момента составления справочной (реестровой) базы CRL.

3.1.8 **жесткая аутентификация:** Аутентификация посредством верительных грамот, созданных с помощью криптографических методов.

3.1.9 **инфраструктура управления привилегиями:** Инфраструктура, способная поддерживать управление привилегиями в подтверждении комплексной службы авторизации и во время взаимоотношений с PKI.

3.1.10 **источник полномочий:** Центр по присвоению атрибутов, который является верификатором привилегий для отдельного ресурса, ответственного как первичный орган, назначающий набор привилегий.

3.1.11 **конечный объект:** Объект сертификата, который использует его секретный ключ для других целей кроме подписания сертификатов или объект, который является доверяющей стороной.

3.1.12 **косвенный CRL:** Список аннулирования, который содержит минимальную информацию об аннулировании сертификатов, изданных другими органами, отличными от органа выпустившего данный CRL.

3.1.13 **критичность:** Характеристика ресурса, которая подразумевает его значение и важность.

3.1.14 **маркер (признак) аутентификации:** Информация, сообщаемая во время обмена при жесткой аутентификации, которая может быть использована для подтверждения подлинности её отправителя.

3.1.15 **назначитель привилегий:** Владелец привилегий, использующий сертификат атрибута или сертификат открытого ключа для назначения привилегий.

3.1.16 **объектный метод:** Действие, которое может быть совершено над ресурсом (например, файловая система может иметь объектные методы чтения, записи и выполнения).

3.1.17 **орган:** Объект, ответственный за выдачу сертификатов.

3.1.18 **орган по присвоению атрибутов:** Орган (центр регистрации), который назначает привилегии посредством выдачи сертификата атрибута.

3.1.19 **отображение политики:** Признание того, что, когда центр регистрации в одном домене сертифицирует центр регистрации, находящийся в другом домене, политика сертификата, действующая во втором домене, может рассматриваться органом в первом домене как эквивалентная (но не обязательна идентичность во всех отношениях) политике сертификата, действующей в первом домене.

3.1.20 **переменные среды:** Аспекты политики, которые требуются для авторизованных решений, которые не содержатся в статических структурах, но доступны для верификатора привилегий с помощью некоторых локальных средств (например, время суток или текущий баланс отчета).

3.1.21 **политика сертификата:** Установленный набор правил, который указывает применимость сертификата к конкретному семейству и/или классу приложений с общими требованиями безопасности.

3.1.22 **полный CRL:** Полный список аннулирования, который содержит отдельные записи для всех сертификатов, которые отменены в данной области.

3.1.23 **пользователь сертификата:** Объект, которому необходимо знать открытый ключ другого объекта.

3.1.24 **привилегия:** Атрибут или свойство, назначенное объекту соответствующим органом.

3.1.25 **простая аутентификация:** Аутентификация посредством соглашений простых паролей.

3.1.26 **пункт распределения CRL:** Элемент директории (реестра) или другой источник распространения CRL.

3.1.27 **путь делегирования:** Упорядоченная последовательность сертификатов, которые, вместе с идентифицирующей информацией назначителя привилегий могут быть обработаны для подтверждения аутентичности привилегий назначителя привилегий.

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
VA DAVLAT NAZORATI

3.1.28 **путь сертификации:** Упорядоченная последовательность сертификатов объектов в информационном дереве директории, которые вместе с открытым ключом исходного объекта могут быть обработаны, чтобы достичь конечного объекта в пути.

3.1.29 **сертификат назначения роли:** Сертификат, который содержит атрибут роли, назначающий одну или более одной роли субъекту/владельцу.

3.1.30 **серийный номер сертификата:** Целое число, уникальное для выдавшего органа, которое однозначно связано с сертификатом, выпущенным данным центром регистрации.

3.1.31 **сертификат атрибута:** Структура цифровых данных, подписанная органом по присвоению атрибутов (центром регистрации), которая связывает некоторые значения атрибута с идентифицирующей информацией о ее владельце.

3.1.32 **сертификат определения роли:** Сертификат, который содержит назначение привилегий для роли.

3.1.33 **сертификат органа:** Сертификат, выпущенный для органа (например, для центра регистрации).

3.1.34 **сертификат центра регистрации:** Сертификат для одного центра регистрации, изданный органом регистрации.

3.1.35 **система, использующая сертификат:** Система, реализующая те функции, которые определены в данном стандарте и, которые используются пользователем сертификата.

3.1.36 **список аннулированных органов по присвоению атрибута:** Список аннулирования, содержащий список ссылок на сертификаты атрибутов, выданных для органа по присвоению атрибутов, которые больше не считаются действительными при авторизации.

3.1.37 **соглашение о ключах:** Метод онлайнового согласования значения ключа, без передачи ключа, даже в зашифрованном виде (например, метод Диффи-Хеллмана).

3.1.38 **список аннулированных сертификатов:** Подписанный список, указывающий на набор сертификатов, которые больше не считаются действиительными издателем сертификатов.

3.1.39 **список аннулированных сертификатов атрибута:** Список аннулирования, содержащий список ссылок на сертификаты атрибутов, которые больше не считаются действительными издающим органом.

3.1.40 **список аннулированных сертификатов атрибутов конечного объекта:** Список аннулирования, содержащий список сертификатов атрибутов, которые выданы держателям, не являющимися органами по присвоению атрибутов, больше не считающимися действительным органом, издающим сертификаты.

3.1.41 **список аннулированных сертификатов открытых ключей конечного объекта:** Список аннулирования, содержащий список сертификатов открытых ключей, которые изданы субъектом, не являющимся центром регистрации, больше не считающимся действительным органом, выдающим сертификаты.

3.1.42 **список аннулированных центров регистрации:** Список аннулирования, содержащий список сертификатов открытых ключей, изданных центрам регистрации, которые больше не считаются действительными, издателем сертификатов.

3.1.43 **среднее время по Гринвичу:** Время по Гринвичу (Англия), используемое как начало отсчета для вычисления временных зон Земли.

3.2 Сокращения

В данном стандарте применяются следующие сокращения:

ПО – программное обеспечение;

СОС – список отозванных сертификатов;

ЦР – центр регистрации;

ЦС – центр сертификации;

ЭЦП – электронная цифровая подпись;

AA (Attribute Authority) – орган по присвоению атрибутов;

AARL (Attribute Authority Revocation List) – список аннулированных органов по присвоению атрибутов;

ACRL (Attribute Certificate Revocation List) – список аннулированных сертификатов атрибута;

ASN.1 (Abstract Syntax Notation One) – язык описания абстрактного синтаксиса данных;

CA (Certification Authority) – орган регистрации;

CARL (Certification Authority Revocation List) – список аннулированных центров регистрации;

CRL (Certificate Revocation List) – список аннулированных сертификатов;

dCRL (Delta Certificate Revocation List) – дельта список аннулированных сертификатов;

DIB (Directory Information Base) – информационная база директории;

DIT (Directory Information Tree) – информационное дерево директории;

DN (Distinguished Name) – выделенное имя;

DNS (Domain Name System) – система доменных имён;

DP (Distribution point) – пункт распределения;

DSA (Directory System Agent) – системный агент директории;

DUA (Directory User Agent) – агент пользователя директории;

EARL (End-entity Attribute certificate Revocation List) – список аннулированных сертификатов атрибута конечного объекта;

EPRL (End-entity Public-key certificate Revocation List) – список аннулированных сертификатов открытого ключа конечного объекта;

GMT (Greenwich Mean Time) – среднее время по Гринвичу;

HASH – хэш-функция;

IANA (Internet Assigned Numbers Authority) – администрация адресного пространства Интернет;

O'ZSTANDART AGENTLIGI
STANDART AGENTSTVO
MUSTAQILLIGI
VA BARCHA HUQUQLARI
HIMOYALANGAN

iCRL (Indirect Certificate Revocation List) – список аннулированных косвенных сертификатов;

IDP (Issuing distribution point) – издающий пункт распределения;

IP (Internet protocol) – межсетевой протокол;

ISO (International Standardization Organization) – Международная организация по стандартизации;

OID (Object identifier) – идентификатор объекта;

PIN (Personal Identification Number) – личный опознавательный номер;

PKCS (Public-Key Cryptosystem) – криптосистема с открытым ключом;

PKI (Public-Key Infrastructure) – инфраструктура открытых ключей;

PMI (Privilege Management Infrastructure) – инфраструктура управления привилегиями;

RDN (Relative Distinguished Name) – относительное выделенное имя;

SOA (Source of Authority) – источник полномочий;

URI (Uniform Resource Identifier) – унифицированный идентификатор ресурса;

UTCTime (Universal Time Coordinated) – всемирное координированное время;

XML (eXtensible Markup Language) – расширяемый язык разметки;

XSD (eXtensible schema definition) – язык определения схем.

4 Общие положения

Структура сертификата открытого ключа, определенная в данном стандарте, включает определение информационного объекта для PKI, включающего сертификаты открытого ключа и CRL.

Структура сертификата атрибута включает определение информационного объекта для инфраструктуры управления привилегиями, включающего сертификаты атрибута и список аннулированных сертификатов атрибута. Данный стандарт дает возможность структуре выпускать, управлять, использовать и аннулировать сертификаты.

Данный стандарт также включает набор стандартных расширений, которые рассчитаны так, чтобы они были пригодными для большого количества приложений PKI и PMI и механизм дополнений, дополнительной информации, которая может быть помещена в сертификат. В данный стандарт включены компоненты схемы, классы объектов, имеющиеся типы атрибутов и правила согласования для хранения в директории (реестре) объектов PKI и PMI. Другие элементы PKI и PMI, такие как протоколы управления ключом и сертификатом, протокол эксплуатации, дополнительный сертификат и расширение CRL выходят за пределы рассмотрения данного стандарта.

Схема аутентификации, определенная в данном стандарте, является общей, и может быть использована в различных приложениях и средах.

Директория (реестр) ITU-T X.500 [1] использует сертификаты открытого ключа и сертификаты атрибута. Технология открытого ключа включает

сертификаты, используемые директорией (реестром) для активизации жесткой аутентификации, операций подписания и шифрования и для хранения подписанных или зашифрованных данных в директории (реестре).

Сертификаты атрибута могут быть использованы директорией (реестром) для активизации управления доступом, основанного на ролях.

Данный стандарт определяет структуру услуг аутентификации следующим образом:

- установлением формы информации для аутентификации, содержащейся в директории (реестре);
- описанием, как из директории (реестра) может быть получена информация для аутентификации;
- установлением правил формирования и расположения информации для аутентификации в директории (реестре);
- определением трех методов, с помощью которых приложения могут использовать данную информацию для выполнения аутентификации, и описанием, как с помощью аутентификации могут поддерживаться другие сервисы безопасности.

Данный стандарт описывает два уровня аутентификации: простая аутентификация, использующая пароль как подтверждение предъявленного идентификатора, и жесткая аутентификация, включающая в себя оформление верительных грамот, использующих криптографические методы. Простая аутентификация обеспечивает несколько ограниченную защиту от несанкционированного доступа, только жесткая аутентификация может быть использована как основа для обеспечения услуг безопасности.

Аутентификация (и другие услуги безопасности) может быть обеспечена только в пределах контекста определённой политики безопасности. Она предназначена для пользователей приложений, чтобы определить их собственную политику безопасности, которая может быть ограничена услугами, обеспечиваемыми стандартом.

Аутентификация предназначена для стандартно определяемых приложений, которые используют структуру аутентификации для определения протоколов обмена, которые должны быть выполнены для того, чтобы успешно завершить аутентификацию, основанную на информации для аутентификации, полученной из директории (реестра). Протокол, используемый приложениями для получения верительной грамоты, называется протоколом доступа к директории (реестру).

5 Открытые ключи и сертификаты открытых ключей

5.1 Генерация пары ключей

Общая политика управления безопасностью реализации определяется жизненным циклом пары ключей и выходит за область рассмотрения данного стандарта. Однако, крайне необходимо, чтобы общая безопасность

закрытых ключей зависела от того, что они являются известными только пользователю, которому они принадлежат.

Ключевые данные нелегко запомнить простому пользователю, поэтому должны употребляться подходящие методы для их сохранения в удобном для транспортировки оборудовании. Один из используемых механизмов это «Smart Card». Он должен хранить закрытый и открытый ключ пользователя, сертификат пользователя и копию открытого ключа ЦР. Использование этой карты обеспечивает дополнительную безопасность, например, использование PIN увеличивает безопасность системы, в которой работает пользователь. Однако конкретный метод сохранения такого рода данных выходит за пределы данного стандарта.

Существуют три метода изготовления пары ключей:

a) пользователь самостоятельно генерирует пару ключей. Этот метод имеет преимущество тем, что закрытый ключ пользователя никогда не будет передан другим объектам, но при этом требуется определенный уровень знаний пользователя;

b) пара ключей генерируется третьей стороной. Третья сторона должна передавать закрытый ключ пользователю физически безопасным методом, затем уничтожить всю информацию, связанную с созданием пары ключей и с самими ключами. Должны быть применены подходящие критерии соответствующей физической безопасности для гарантирования, что третья сторона и данные о действиях будут защищены от подделки;

c) пара ключей генерируется ЦР. Это частный случай метода a) и b).

5.2 Создание сертификата открытого ключа

Структура сертификата открытого ключа, определенная данным стандартом, предназначена для использования приложениями, требующими аутентификации, проверки целостности, конфиденциальности и безотказности от авторства.

Формат сертификатов открытого ключа, определенный в данном стандарте, включает механизм расширяемости и набор специфических расширений сертификата. Если по некоторым причинам орган аннулирует заранее выданный сертификат открытого ключа, пользователи должны быть уведомлены о том, что произошло аннулирование, чтобы они не использовали ненадежные сертификаты. Списки аннулирования - это одна из структур, которая может быть использована для уведомления пользователей об аннулировании. Формат списков аннулирования, определенный в данном стандарте, включает механизм расширяемости и набор расширений списков аннулирования. В сертификате и списке аннулирования могут быть также определены другие расширения, которые применяются для их специфических сред.

Система, использующая сертификат открытого ключа, должна подтверждать подлинность сертификата до использования этого сертификата в приложениях. Процедуры для выполнения подтверждения,

определенные в данном стандарте, включают проверку целостности сертификата, его статуса аннулирования, и его действительности для использования в определенных приложениях.

Директория (реестр) использует сертификат открытого ключа при обеспечении услуг безопасности, которые включают:

- жесткую аутентификацию между компонентами директории (реестра);
- аутентификацию, целостность и конфиденциальность операций директории (реестра);
- целостность и аутентификацию хранимых данных.

Для аутентификации пользователя открытый ключ должен быть получен из доверенного источника. Таким источником может быть ЦР, удостоверяющий открытый ключ посредством выдачи сертификата открытого ключа, который связывает открытый ключ с соответствующим закрытым ключом объекта. Процедуры, используемые ЦР для обеспечения гарантии того, что объект на самом деле владеет закрытым ключом, и другие процедуры, связанные с изданием сертификата открытого ключа, выходят за пределы рассмотрения данного стандарта.

Сертификат, форма которого определена в данном стандарте, имеет следующие свойства:

- любой пользователь, имеющий доступ к открытому ключу ЦР, может получить из ЦР открытый ключ, который был сертифицирован;
- не существует другого субъекта, кроме ЦР, который мог бы модифицировать сертификат (сертификаты не подделываемы).

Из-за того, что сертификаты не подделываемы, они могут быть опубликованы в директории (реестре) без необходимости дополнительной защиты.

ЦР создает сертификаты пользователей с помощью набора информации, в том числе различительного имени пользователя и открытого ключа, а также необязательного уникального идентификатора, включающего дополнительную информацию о пользователе.

Точный состав уникального идентификатора не определен в данном стандарте и определяется ЦР: например, может быть идентификатор объекта, сертификат, дата или какая-нибудь другая форма сертификации, которая определяет достоверность различительного имени. Сертификат пользователя с именем А и уникальным идентификатором УА, выданный ЦР, и уникальным идентификатором УЦР, имеет следующую форму:

$$ЦР\langle\langle A\rangle\rangle = ЦР\{В, СН, ИА, ЦР, УЦР, А, УА, Ар, ТА\}$$

где:

В - версия сертификата;

СН - серийный номер сертификата;

ИА - идентификатор алгоритма, использованного для подписания сертификата;

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH

УЦР - это необязательный уникальный идентификатор ЦР;

УА - необязательный уникальный идентификатор пользователя;

Ар, ТА - период действия сертификата и содержит две даты, начало и окончание действительности сертификата.

Период действительности сертификата - это интервал времени, в течение которого ЦР гарантирует, что он будет поддерживать информацию о статусе сертификата, т.е. не будет издавать данные об аннулировании. Сразу после получения ТА изменения будут введены не менее чем за 24 часа, здесь предполагается, что система будет использовать GMT как эталонное время. Подпись в сертификате будет проверена на достоверность любым пользователем знающим ЦР. Для представления сертификатов могут быть использованы нижеприведенные типы данных:

```
Certificate                ::= SIGNED { SEQUENCE {
version      [0]           Version DEFAULT v1,
serialNumber               CertificateSerialNumber,
signature                  AlgorithmIdentifier,
issuer                     Name,
validity                   Validity,
subject                    Name,
subjectPublicKeyInfo       SubjectPublicKeyInfo,
issuerUniqueIdentifier  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
--агар келтирилган бўлса, у ҳолда версия v2 ёки v3 бўлиши керак
subjectUniqueIdentifier  [2]  IMPLICIT UniqueIdentifier
OPTIONAL,
--агар келтирилган бўлса, у ҳолда версия v2 ёки v3 бўлиши керак
extensions      [3]  Extensions OPTIONAL
--агар келтирилган бўлса, у ҳолда версия v3 бўлиши керак } }
Version      ::=  INTEGER { v1(0), v2(1), v3(2), v4 (3) }
CertificateSerialNumber  ::=  INTEGER
AlgorithmIdentifier   ::=  SEQUENCE {
algorithm    ALGORITHM.&id ({SupportedAlgorithms}),
parameters   ALGORITHM.&Type ({SupportedAlgorithms}{
@algorithm})
OPTIONAL }
SupportedAlgorithms   ALGORITHM   ::= { ... }
Validity     ::=   SEQUENCE {
notBefore  Time,
notAfter  Time }
SubjectPublicKeyInfo    ::= SEQUENCE {
algorithm    AlgorithmIdentifier,
subjectPublicKey  BIT STRING }
Time    ::= CHOICE {
utcTime    UTCTime,
```

```
generalizedTime  GeneralizedTime }
Extensions ::= SEQUENCE OF Extension
Extension ::= SEQUENCE {
extnId EXTENSION.&id ({ExtensionSet}),
critical BOOLEAN DEFAULT FALSE,
extnValue OCTET STRING
}
ExtensionSet EXTENSION ::= { ... }
```

Перед значением Time используется любая операция сравнения, например, как часть правила согласования при поиске, а если синтаксис Time выбран как тип UTCTime, значение двухцифрового поля будет усовершенствовано в четырёх цифровое значение:

- если двухцифровое значение от 00 до 49 включительно, то значение равно 2000 в дополнение к нему;
- если двухцифровое значение от 50 до 99 включительно, то значение равно 1900 в дополнение к нему;

В сертификате открытого ключа и сертификате атрибута используются следующие типы данных:

1) version - это версия сертификата. Если в сертификате представляется основная часть extensions (расширения), то версия сертификата будет v3, согласно ITU-T X.509 [2]. Если представляется компонента issuerUniqueIdentifier или subjectUniqueIdentifier, то версия сертификата будет v2 или v3. Если появляются неизвестные элементы в пределах расширения и расширение не является критическим, эти неизвестные элементы игнорируются согласно правилам расширяемости;

2) serialNumber целое число, отведенное ЦР для каждого сертификата. Значение серийного номера должно быть уникальным для каждого сертификата, выданного определенным ЦР (например, выданное название и серийный номер определяют уникальность сертификата);

3) signature состоит из идентификатора алгоритма и хеш-функций использованных ЦР для подписания сертификата (например, md5WithRSAEncryption, sha-1WithRSAEncryption, id-dsa-with-sha1, и т.д.);

4) issuer устанавливает объект, который подписывает и выдает сертификат;

5) validity временной интервал, в течение которого ЦР гарантирует, что он будет поддерживать информацию о статусе сертификата;

6) subject устанавливает объект, который связан с открытым ключом получаемого из поля открытого ключа субъекта;

7) subjectPublicKeyInfo используется для сообщения о том, что открытый ключ сертифицирован и для идентификации алгоритма (например rsaEncryption, dhpublicnumber, id-dsa);

8) issuerUniqueIdentifier используется, чтобы уникально идентифицировать выдающий орган в случае повторного использования имени;

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
MA'LUMOTLARI MARKAZI

9) subjectUniqueIdentifier используется, чтобы уникально идентифицировать субъект в случае повторного использования имени.

В случае если различительное имя могло быть передано другому пользователю органом, выдающим имена, ЦР может использовать уникальный идентификатор, чтобы иметь возможность различать случаи повторного использования имён.

Однако если одинаковые пользователи обеспечены сертификатами различных ЦР, необходимо, чтобы ЦР координировали выдачу уникальных идентификаторов как часть их процедуры регистрации пользователей.

Поле extensions позволяет добавлять новые поля в структуру без модификации определения языка ASN.1. Поле расширения содержит идентификатор расширения, флаг критичности, зашифрованное значение данных языка ASN.1, связанного с идентифицированным расширением. Для тех расширений, где установка индивидуальных расширений находится в пределах SEQUENCE, спецификация этих индивидуальных расширений должна включать правила для выражения порядка. Когда реализация обработки сертификата не распознает расширение и если флаг критичности имеет значение FALSE, она может игнорировать его. Если флаг критичности имеет значение TRUE, нераспознанные расширения будут причиной того, что структура будет признана недействительной, например, в сертификате нераспознанное расширение будет причиной утверждения подписи, использующей этот сертификат, как поврежденной. Когда реализация, использующая сертификат распознает и обрабатывает расширения, тогда эта реализация будет обрабатывать расширения вне зависимости от значения флага критичности. Любые расширения, которые обозначены как некритические, будут причиной несовместимости между системами, которые используют сертификат и обрабатывают расширения, системами, использующими сертификат и которые не распознают расширение и игнорируют его.

Если в пределах расширения появились неизвестные элементы и расширение не отмечено как критическое, эти неизвестные элементы будут проигнорированы согласно правилам расширяемости.

ЦР имеют три варианта, касающихся расширения и могут:

1) исключить расширение из сертификата;

2) включить расширение и его флаг как некритические;

3) включить расширение и его флаг как критические.

Процедура подтверждения производится в двух возможных действиях, касающихся расширения:

1) игнорированием расширения и принятием сертификата (все другие действия производятся эквивалентно);

2) обработкой расширения и принятием или отклонением сертификата в зависимости от содержания расширения и от обстоятельств, в которых происходит обработка (например, когда текущие значения обработки пути изменяются).

Некоторые расширения могут быть отмечены только как критические. В таких случаях, процедура подтверждения, которая анализирует расширение, обрабатывает его и признает/отклоняет сертификат, в зависимости от содержания расширения. Процедура подтверждения, которая не может анализировать расширение, отклоняет сертификат.

Некоторые расширения могут быть отмечены только как некритические. В таких случаях, процедура подтверждения, которая понимает расширение, обрабатывает его и признает/отклоняет сертификат, в зависимости от содержания расширения. Процесс подтверждения, который не понимает расширение, признает сертификат (пока другие факторы не повлияют на отклонение).

Некоторые расширения могут быть отмечены как критические или некритические. В таких случаях, процедура подтверждения, которая понимает расширение и обрабатывает его, признает/отклоняет сертификат, в зависимости от содержания расширения, несмотря на наличие флага в сертификате. Процесс подтверждения, не понимающий расширение, принимает сертификат, если расширение отмечено как некритическое (пока другие факторы не повлияют на отклонение) и отклоняет сертификат, если расширение отмечено как критическое.

Когда ЦР рассматривает состав расширения в сертификате, он считает, что его содержание будет поддерживаться при любых обстоятельствах. Если необходимо, чтобы содержание расширения было рассмотрено, прежде чем будут рассмотрены другие данные в сертификате, ЦР должен установить расширение критическим. Это производится при реализации, чтобы любой процесс подтверждения, который не обрабатывает расширение, был отклонен (возможен ограничивающий набор приложений, который может подтвердить сертификат). ЦР может отметить некоторые расширения как некритические для достижения полной совместимости с предыдущими версиями подтверждающих приложений, которые не могут обработать расширения. Там, где необходимость совместимости расширения с более поздними версиями и взаимодействие с подтверждающими приложениями, неспособными обработать расширения, более существенно, чем возможность ЦР вводить в действие расширение, эти необязательно критические расширения должны быть отмечены как некритические. Это означает, что ЦР должны установить необязательные критические расширения как некритические в течение переходного периода пока улучшаются верификаторы приложений обработки сертификата.

Для определения специфических расширений используется следующий класс объекта.

```
EXTENSION ::= CLASS {
&id OBJECT IDENTIFIER UNIQUE,
&ExtnType }
WITH SYNTAX {
SYNTAX      &ExtnType
```

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA
MUVOFIQLIKNI BAHOLASH

IDENTIFIED BY &id }

Существуют два основных типа сертификатов открытого ключа: сертификаты конечного объекта и сертификаты ЦР.

Сертификат конечного объекта выдается со стороны ЦР субъекту, который не издает другие сертификаты открытых ключей.

Сертификат ЦР выдается со стороны ЦР субъекту, который сам является ЦР и поэтому способен издавать сертификаты открытого ключа. Сертификаты ЦР могут быть классифицированы по следующим типам:

- сертификат, изданный самому себе. Это сертификат, где издатель и субъект один и тот же ЦР. ЦР может использовать сертификаты, изданные самому себе, например, в течение операции продления действия ключа для обеспечения достоверности при переходе от старого ключа к новому;

- сертификат, подписанный самим ЦР. Это частный случай сертификата, выданного самому себе, где закрытый ключ используется ЦР для подписания сертификата соответствующего открытому ключу, который сертифицирован посредством сертификата;

- кросс сертификат. Это сертификат, где выдающая сторона и субъект - это разные ЦР.

ЦР выдают сертификаты другим ЦР или как механизм для авторизации наличия субъектного ЦР (например, в точно определённой иерархии) или для признания действительности субъектного ЦР (например, в распределенной модели доверия). Структура кросс сертификата используется в обоих случаях.

Элемент директории (реестра) всех пользователей, которые принимают участие в жесткой аутентификации, содержит сертификат(ы) пользователя А. Такой сертификат генерируется центром регистрации пользователя А, который является объектом в DIT. Центр регистрации пользователя А, который может быть не уникальным, обозначается как ЦР(А) или просто центр регистрации, если подразумевается пользователь А. Открытый ключ пользователя А может быть найден любым пользователем, знающим открытый ключ центра регистрации. Нахождение открытого ключа называется рекурсией.

Если пользователь А пытается приобрести открытый ключ пользователя В, он также достает открытый ключ ЦР(В). Чтобы у пользователя А была возможность достать открытый ключ ЦР(В), элемент директории каждого ЦР содержит некоторое количество сертификатов.

Сертификаты бывают двух типов. Первые - готовые сертификаты X, сгенерированные другими сертифицирующими органами. Вторые – сертификаты, сгенерированные самим X, которые сертифицированы закрытым ключом других сертифицирующих органов. Наличие этих сертификатов дает пользователям возможность создания пути от одной точки к другой.

Список сертификатов, известный как путь сертификации, необходим для предоставления отдельному пользователю возможности

получения открытого ключа другого пользователя. Каждый элемент в списке, это сертификат, сертифицирующего органа следующего элемента в списке. Путь сертификации от А к В (обозначен как А®В):

- начинается с сертификата выданного ЦР(А), т.е. ЦР(А) «Х1» для некоторого объекта Х1;
- далее идут следующие сертификаты Xi«Xi+1»;
- заканчивается сертификатом пользователя В.

Путь сертификации логическая форма неразрывной (непрерывной) цепи доверенных пунктов в DIT между пользователями, желающими произвести аутентификацию. Определенные методы употребляемые пользователями А и В для получения путей сертификации А®В и В®А могут отличаться. Один из путей согласования - это систематизация иерархий ЦР в самой иерархии, которые могут совпадать или не совпадать со всей или с частью иерархии DIT.

Пользователь может получить один или более одного сертификата от одного или нескольких сертифицирующих органов. Каждый сертификат носит (поддерживает) название сертифицирующего органа, который выдал его. Для представления сертификата или пути сертификации могут использоваться следующие типы данных языка ASN.1:

```
Certificates      ::= SEQUENCE {
userCertificate   Certificate,
certificationPath   CertPath OPTIONAL }

CertificationPath    ::= SEQUENCE {
userCertificate   Certificate,
theCACertificates   SEQUENCE OF CertificatePair OPTIONAL }
```

Кроме этого, для представления начального пути сертификации может быть использован следующий тип данных ASN.1. Эта компонента содержит путь сертификации, который может быть точкой возврата для инициатора.

```
CertPath      ::= SEQUENCE OF CrossCertificates
CrossCertificates    ::=   SET OF Certificate
```

Каждый сертификат в пути сертификации должен быть уникальным. Не может быть сертификата, который появляется более чем один раз в значении theCACertificates компоненты CertificationPath или в значении Certificate в CrossCertificates компоненты CertPath.

Сертификат открытого ключа связан с открытым ключом и уникальным различительным именем пользователя, которое описывает его таким образом:

а) сертифицирующий орган должен выполнять идентичность Пользователя перед созданием сертификата для него;

b) сертифицирующий орган не должен выдавать сертификаты с одинаковыми именами двум пользователям.

Важно, что передача информации сертифицирующему органу не должна быть скомпрометирована и должны быть приняты соответствующие критерии физической безопасности. В связи с этим:

a) будет серьезным нарушением безопасности, если ЦР выдаст сертификат для пользователя несмотря на то, что открытый ключ был подделан;

b) если применен метод генерации ключей описанный в 5.1 перечисления b) и c), то закрытый ключ пользователя будет передан пользователю безопасным методом;

c) если применен метод генерации ключей описанный в 5.1 перечисления a) и b), пользователь может использовать различные методы (on-line или off-line) для передачи открытого ключа ЦР безопасным методом. Метод on-line может обеспечивать некоторую гибкость для удаленных операций, выполняемых между пользователем и ЦР.

Сертификат открытого ключа публично доступен и нет определенных уровней безопасности, необходимых для его транспортировки в директорию.

5.3 Действительность сертификата

Орган, который выдаёт сертификаты (открытого ключа и атрибута) также ответственен за индикацию действительности сертификатов, выданных им. Сертификаты это субъекты, которые позднее должны быть аннулированы. Эти аннулирования и уведомления об аннулировании могут быть сделаны непосредственно одним органом, который выдал сертификат или косвенно другим органом, который авторизован органом, выдающим сертификат.

Орган, производящий аннулирование сертификатов, может использовать механизм(ы) доверяющей стороны для получения информации о статусе аннулирования, о сертификатах выданных органом. В данном стандарте определен механизм CRL, но не исключается использование других механизмов. Доверяющие стороны проверяют информацию о статусе аннулирования для утверждения сертификата, как присуще для всех сертификатов рассматриваемых в процессе процедуры обработки пути сертификации описанной в разделе 7, и в процессе обработки передачи привилегий, описанного в разделе 13.

Сертификаты, включающие сертификаты открытого ключа, также как сертификаты атрибута должны иметь время жизни, связанное с ними, в конце которого оно истекает. Для того чтобы обеспечить непрерывность услуг, орган должен гарантировать регулярное обновление работоспособности сертификатов, чтобы заменять сертификаты, срок которых истек/ истекает. Дата извещения об аннулировании - это дата/время, когда извещение об аннулировании сертификата впервые появилось в CRL, вне зависимости от того, оно в базовом или в дельта CRL. В CRL, дата

извещения об аннулировании сертификата содержится в поле thisUpdate. Дата аннулирования - это дата/время фактического аннулирования сертификата со стороны ЦР. В CRL дата аннулирования содержится в компоненте revocationDate.

Возможно, что:

- действие сертификатов может быть рассчитано так, что каждый становится действительным ко времени, когда истекает срок предыдущего, или может быть допущено некоторое наложение их даты;

- сертификаты, сроки которых истекли, удаляются из директории.

Сертификаты могут быть аннулированы до их времени аннулирования, например, если закрытый ключ пользователя скомпрометирован или если был скомпрометирован сертификат ЦР. Аннулирование сертификата пользователя или сертификата органа должно быть произведено при соответствующем оповещении и новый выданный сертификат должен быть действительным. Орган может затем информировать владельца сертификата о его аннулировании с помощью некоторых off-line методов.

Орган, выдающий и аннулирующий сертификаты:

a) может нуждаться в поддержке аудита записей его событий, аннулирования для всех типов сертификатов, выданных органом (например, сертификаты открытого ключа, сертификаты атрибута, выданные конечным объектам также как другим органам);

b) должен предоставлять информацию о статусе аннулирования надёжным сторонам, используя CRL, протокол получения в on-line статуса сертификата или другими механизмами для публикации информации о статусе аннулирования;

c) если используются CRL, должен поддерживать и публиковать CRL, даже если список аннулированных сертификатов пуст.

Доверяющие стороны могут использовать несколько механизмов для определения места расположения информации о статусе аннулирования. Например, в сертификате может быть указатель, который направляет доверяющую сторону к месту, где находится информация об аннулировании. Это может быть указатель в списке аннулирования, который перенаправляет доверяющую сторону в различные места. Доверяющая сторона может найти информацию об аннулировании в архиве (например, в директории).

Поддержание директории затрагивает списки аннулированных органов ответственных за директорию и его пользователей выполняющих обязанности в согласованности с политикой безопасности. Например, пользователь может модифицировать элемент объекта, поменяв его старый сертификат на новый.

Если списки аннулирования опубликованы в директории, они держатся в списке как атрибуты следующих типов:

- список аннулированных сертификатов;

- список аннулированных органов;

- дельта список аннулирования;

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
MA'LUMOTLARI MARKAZI

- список аннулированных сертификатов атрибутов;
- список аннулированных органов по присвоению атрибутов.

Действительность сертификата определяется следующими типами данных:

CertificateList ::= SIGNED { SEQUENCE {
version Version OPTIONAL,
--если представлено то версия будет v3
signature AlgorithmIdentifier,
issuer Name,
thisUpdate Time,
nextUpdate Time OPTIONAL,
revokedCertificates SEQUENCE OF SEQUENCE {
serialNumber CertificateSerialNumber,
revocationDate Time,
crlEntryExtensions Extensions OPTIONAL } OPTIONAL,
crlExtensions [0] Extensions OPTIONAL }}

Ниже приведены описания всех типов данных:

1) version версия зашифрованного списка аннулирования. Если компонента extensions отмечена как критическое, то она представляется в списке аннулирования, и версия сертификата должна быть v3. Если нет компоненты extensions отмеченной как критическое, то оно представляется в списке аннулирования и версия может отсутствовать или может представляться как v3;

2) signature содержит идентификатор алгоритма, для алгоритма использованного органом для подписания списка аннулирования;

3) issuer указывает на объект, который подписал и издал список аннулирования;

4) thisUpdate это дата/время, когда этот список аннулирования был издан;

5) nextUpdate, если указано, то указывает на дату/время, согласно которому будет издан следующий список аннулирования в этой серии;

6) revokedCertificates указывает на сертификаты, которые аннулированы.

Аннулированные сертификаты идентифицируются серийным номером. Если ни один из сертификатов неаннулирован согласно CRL, строго рекомендуется чтобы параметр revokedCertificates не был включен в CRL, лучше, если он не будет включен с пустым параметром SEQUENCE.

Если представлено crlExtension, оно содержит один или более расширений CRL.

5.4 Дополнения X.509 версии 3

Версия 3 описала механизм дополнений, дополнительной информации, которая может быть помещена в сертификат. Рекомендация ITU-T X.509 [2] и рекомендация RFC 2459 [3] описывают набор стандартных дополнений, но вместе с тем не ограничивают возможности использования любых других дополнений путем регистрации идентификатора (ISO или IANA).

Каждое дополнение состоит из трех полей: type, critical, value.

Таким образом, дополнение представляет собой структуру, содержащую:

1) идентификатор дополнения;
2) признак "критичное/не критичное дополнение;
3) собственно значение дополнения, представленное в бинарном виде (OCTET STRING).

В свою очередь само дополнение может являться какой угодно сложной структурой (от простого текстового значения до сложной структуры), формат и интерпретация которого определяется идентификатором дополнения.

Основной целью критичных дополнений является предохранение сертификата, изданного ЦР, от возможности использования его в приложениях, которые не могут обработать такие дополнения. Таким образом, правила обработки дополнений требуют от прикладного ПО отвергнуть сертификат, если дополнение отмечено критичным и прикладное ПО не может его интерпретировать. В свою очередь, требование отвергнуть дополнение прикладным ПО, отмеченное как критичное, при невозможности его интерпретации, требует от прикладного ПО детального разбора дополнений сертификатов и затрудняет процесс модификации как прикладного ПО, так и ПО, обеспечивающего реализацию инфраструктуры открытых ключей.

Базовый идентификатор дополнений, определенный рекомендацией ITU-T X.509 [2]: id-ce OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 29},
где: id-ce обозначает: Object identifier assignments for ISO certificate extensions.

Стандартные дополнения можно разделить на две категории: ограничивающие дополнения и информационные дополнения. Первые ограничивают область применения ключа, определенного сертификатом, или сам сертификат. Вторые содержат дополнительную информацию, которая может быть использована в прикладном ПО пользователем сертификата.

К ограничивающим дополнениям относятся:

1) базовые ограничения (basicConstraints);
2) область применения ключа (keyUsage);
3) расширенная область применения ключа (extendedKeyUsage);
4) регламенты сертификата (модифицируемые ограничениями регламентов и соответствием регламентов) (certificatesPolicies);
5) ограничения имен (nameConstraints).

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH

Сертификат версии 3 имеет формат, приведенный в таблице 1:

Таблица 1 – Формат сетрификата версии 3

Version	Версия сертификата	3
Certificate Serial Number	Серийный номер сертификата	40:00:00:00:00:00:00:ab:38:1e:8b:e9:00:31:0c:60
Issuer X.500 [1] Name	Имя Издателя сертификата	C=UZ, ST=Tashkent, O=PKI, CN=Certification Authority
Validity Period	Срок действия сертификата	Действителен с : Ноя 2 06:59:00 1999 GMT Действителен по : Ноя 6 06:59:00 2004 GMT
Subject X.500 [1] Name	Имя Владельца сертификата	C=UZ, ST= Tashkent, O=PKI, CN=Alimov
Subject Public Key Info	Открытый ключ Владельца	тип ключа: Открытый ключ O‘z DSt длина ключа: 1024 значение: AF:ED:80:43.....
Issuer Unique ID version 2	Уникальный идентификатор Издателя	
Subject Unique ID version 2	Уникальный идентификатор Владельца	
type	critical value	дополнения (только версия 3)
type	critical value	
type	critical value	
CA Signature	ЭЦП ЦР	

К информационным дополнениям относятся:

1) идентификаторы ключей (subjectKeyIdentifier, authorityKeyIdentifier);

2) альтернативные имена (subjectAltName, issuerAltName);

3) точка распространения СОС (cRLDistributionPoint, issuingDistributionPoint);

4) способ доступа к информации ЦС (authorityAccessInfo).

Идентификатор ключа Издателя

Дополнение Идентификатор ключа Издателя (authorityKeyIdentifier) используется для идентификации открытого ключа, соответствующего секретному ключу ЭЦП, использованному ЦС при подписи издаваемого сертификата (или СОС). Данное дополнение может быть использовано в

случае, когда Издатель сертификата (ЦР) имеет несколько различных секретных ключей ЭЦП (например при плановой их смене), а так же для однозначного построения цепочек сертификатов.

Идентификатор ключа Владельца

Данное дополнение используется для идентификации различных сертификатов, содержащих открытый ключ. Для упрощения процедуры построения цепочки, данное дополнение должно устанавливаться во всех сертификатах ЦС, которые включают дополнение basicConstraints с установленным значением cA TRUE. Во всех издаваемых ЦР сертификатах значение keyIdentifier в дополнении authorityKeyIdentifier должно быть идентично значению subjectKeyIdentifier сертификата ЦР.

Для сертификатов, значение subjectKeyIdentifier должно вырабатываться из открытого ключа или с использованием метода генерации уникальных значений. Рекомендацией RFC 2459 [3] предлагается два метода генерации идентификатора на основе значения открытого ключа:

1) значение keyIdentifier определяется как 160 bit хэш-функции, вычисляемой по алгоритму SHA-1 из значения BIT STRING subjectPublicKey (исключая тэг, длину и неиспользованные биты);

2) значение keyIdentifier определяется как четырех битовое поле со значением 0100 и следующие за ним 60 bit наименьшей значимой части хэш-функции, вычисляемой по алгоритму SHA-1 из значения BIT STRING subjectPublicKey.

Для идентификации без использования открытого ключа, можно также использовать монотонно возрастающую последовательность целых чисел.

Для сертификатов конечного пользователя, данное дополнение используется для идентификации приложением различных сертификатов содержащих определенный открытый ключ. Если конечный пользователь обладает несколькими сертификатами, особенно от разных ЦР, данное дополнение позволяет быстро определить требуемый сертификат. Для этих целей данное дополнение должно добавлять во все сертификаты конечных пользователей.

Данное дополнение служит для определения взаимосвязей между различными объектами на всем сроке существования открытого ключа (запрос, сертификат, сообщение о компрометации, СОС).

Область применения ключа

Данное дополнение определяет область применения секретного ключа, соответствующего открытому, содержащемуся в сертификате.

Используются следующие значения флагов, приведенные в таблице 2.

Расширенная область применения ключа

Данное дополнение (Extended keyUsage) предназначено для задания дополнительных областей применения ключа по требованиям прикладного ПО. Значение Область применения ключа (KeyPurposeId) данного дополнения может быть определена любой организацией в зависимости от конкретных целей.

'ZSTANDART ASSENTLIGI
SIZDA SAQLANTIRISH,
MUTSIRLASHTIRISH
VA BAYLAT HAMRATI
QO'LLANILSIN

Идентификатор объекта, для идентификации области применения должен назначаться в соответствии с IANA или ITU-T X.660 | ISO/IEC/ITU 9834-1 [4].

В реализации PKI рекомендуется использовать данное дополнение для целей, предусмотренных процессом обработки в прикладном ПО.

Таблица 2 - Области применения ключа

Флаг	Применение ключа
digitalSignature	Ключ может быть использован для целей обеспечения целостности и авторства информации, формирования и проверки ЭЦП электронных документов и информации, установления идентичности в процессе аутентификации и т.д.
nonRepudiation	не используется
keyEncipherment	не используется
dataEncipherment	Ключ может быть использован для целей обеспечения конфиденциальности и целостности информации. Шифрование и расшифрование данных и контроль целостности с использованием имитозащиты.
keyAgreement	не используется
KeyCertSign	Ключ может быть использован для целей формирования ЭЦП сертификатов. Может использоваться ЦР.
CRLSign	Ключ может быть использован для целей формирования ЭЦП СОС. Может использоваться ЦР.
EncipherOnly	не используется
DecipherOnly	не используется

Срок действия секретного ключа

Данное дополнение позволяет Издателю сертификата задать различные сроки действия секретного и сертификата. Дополнение содержит два опциональных компонента: notBefore и notAfter. Секретный ключ, соответствующий открытому в сертификате, не должен быть использован до или после времен, указанных соответствующими компонентами. ЦР не должен формировать сертификат, в котором не указан ни один из компонентов.

Регламенты использования сертификата

Данное дополнение представляет собой последовательность, состоящую из одного или нескольких информаторов регламента (PolicyInformation), каждый из которых содержит OID и опциональный квалификатор.

Данный информатор регламента отображает регламент, с учетом которого сертификат был издан и цели, для которых сертификат может быть использован. Опциональный квалификатор, который может присутствовать,

не предусмотрен для целей изменения регламента, определенного информатором.

Приложения, с определенными требованиями функционирования, должны содержать внутренний список регламентов, удовлетворяющих данным требованиям, для целей сравнения идентификаторов объектов в сертификате с имеющимся внутренним списком приложения. Если данное дополнение критичное, ПО производящее обработку должно обладать возможностью интерпретации данного дополнения (включая опциональный квалификатор). В противном случае сертификат должен быть отвергнут.

Используются четыре Регламента (таблица 3) использования сертификата (certificatePolicies).

Таблица 3 - Информация о Регламенте сертификата

Регламент и номер ссылки	Информация о Регламенте сертификата
Registration (n = 1)	Данный сертификат и соответствующий ему секретный ключ предназначен для целей, предусмотренных процессом регистрации пользователя в системе, и не могут быть использованы для обеспечения авторства, целостности и конфиденциальности любой другой передаваемой или хранимой информации.
Class1 (n = 2)	Центр регитсрации не гарантирует принадлежность открытого ключа и дополнений Владельцу сертификата. Использование данного сертификата в приложениях, требующих идентификации Владельца, может привести к фальсификации конфиденциальной информации.
Class2 (n = 3)	Данный сертификат и соответствующий ему секретный ключ предназначен для обеспечения авторства, целостности и конфиденциальности любой передаваемой или хранимой информации, не составляющей государственную тайну.
Class3 (n = 4)	Данный сертификат и соответствующий ему секретный ключ предназначен для обеспечения авторства, целостности и конфиденциальности любой передаваемой или хранимой информации, не составляющей государственную тайну.

Эти регламенты предназначены определять:

1) области использования сертификатов и соответствующих им секретных ключей;

2) комплекс организационно-технических мероприятий, связанный с жизненным циклом объектов PKI (распространение ПО, регистрация, сертификация, компрометация и т. п.).

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASH
VA DAVLAT XIZMATI
AGENTLIGI

Регламенты использования сертификата аналогичны делению сертификатов на различные типы и устанавливаются в соответствующем стандартном дополнении всех сертификатов конечных пользователей.

Соответствие регламентов

Данное дополнение предназначено для использования в сертификатах ЦС. Оно содержит список парных OID. Каждая пара в свою очередь включает Регламент Зоны Издателя (issuerDomainPolicy) и Регламент Зоны Владельца (subjectDomainPolicy). Такая парность означает, что ЦР, выступающий в роли Издателя (issuing CA), принимает Регламент Зоны Издателя эквивалентным Регламенту Зоны Владельца для подчиненного ЦР (subject CA).

Пользователи, относящиеся к Издающему ЦР (issuing CA) могут принимать Регламент Зоны Издателя(issuerDomainPolicy) для соответствующих приложений. Дополнение Соответствие Регламентов ставит в известность пользователей Издающего ЦР о том наборе Регламентов, subject CA) которые сравнимы с регламентами, соответствующими их требованиями.

Альтернативное имя Владельца

Дополнение Альтернативное Имя Владельца может использоваться для двух целей. Во-первых, оно позволяет расширить границы идентификации Владельца сертификата. Для этого используются заранее определенные идентификаторы, которые включают адрес электронной почты Internet, имя в DNS, IP адрес и URI. Во-вторых, оно предоставляет набор дополнительной справочной информации о Владельце сертификата. Для этого используется представление имени в различных видах и множественное представление имен. При необходимости введения такой дополнительной идентификации в сертификат должно использоваться дополнение Альтернативное Имя Владельца или Альтернативное Имя Издателя.

В связи с тем, что альтернативное имя может быть использовано для целей определения соответствия Владельца и открытого ключа, все части идентифицирующие его и входящие в альтернативное имя, должны быть проверены ЦР. Уровень проверки дополнительной информации определяется Регламентом ЦР.

Альтернативное Имя Владельца может быть ограничено тем же способом, что и поле Владелец в сертификате, используя дополнение nameConstraintsExtension.

В соответствии с рекомендацией ITU-T X.681 [5] синтаксис поля определен в следующем виде:

TYPE-IDENTIFIER ::= CLASS { &id OBJECT IDENTIFIER UNIQUE, &Type } WITH SYNTAX {&Type IDENTIFIED BY &id}

Альтернативное имя Издателя

Так же как и дополнение Альтернативное Имя Владельца (таблица 4), дополнение Альтернативное имя Издателя (issuerAltName) служит целям

дополнительной ассоциации Издателя сертификата. Правила использования данного дополнения аналогичны использованию дополнения Альтернативное Имя Владельца.

Таблица 4 - Поля дополнения Альтернативное Имя

Наименование	Тип	Назначение	Идентификатор
rfc822Name	IA5String	Адрес электронной почты rfc 822	CHOICE [1]
dNSName	IA5String	Имя DNS	CHOICE [2]
directoryName	IA5String	X.500 [1] DN (имя)	CHOICE [4]
uniformResource Identifier	IA5String	адрес URI	CHOICE [6]
iPAddress	OCTET STRING	Адрес IP	CHOICE [7]
registeredID	OBJECT IDENTIFIER	Идентификатор ASN.1 объекта	CHOICE [8]
organizationName	DirectoryString	Наименование организации	id-at 10
registredAddress	DirectoryString	Зарегистрирован-ный (юридический адрес) организации	id-at 26
surname	DirectoryString	Фамилия, имя, отчество	id-at 4
businessCategory	DirectoryString	Должность	Должность
physicalDelivery	DirectoryString	Почтовый адрес	id-at 19
telephoneNumber	PrintableString	Номер телефона	id-at 20
description	DirectoryString	Дополнительное описание	id-at 13
accountNumber	DirectoryString	Номер банковского расчетного счета	OBJ_mdprei_extensions,10
bankID	DirectoryString	Банковский идентификацион-ный код	OBJ_mdprei_extensions,11

Атрибуты Справочника Владельца сертификата

Дополнение предусмотрено для хранения дополнительной информации, связанной с атрибутами директории ITU-T X.500 [1].

Основные ограничения

O'ZSTANDART AGENTLIGI
O'ZBEKISTON
MUSTAQILLIGINING
YA. DAVLAT BELGISI

Дополнение Базовые ограничения идентифицирует, является ли Владелец сертификата ЦР, и какова длина цепочки сертификатов для этого ЦР.

Поле pathLenConstraint имеет смысл только при условии, если значение cA установлено в TRUE. В этом случае оно обозначает максимальное количество сертификатов ЦР, которые следуют за данным сертификатом в цепочке. Значение нуль означает, что только сертификаты конечного пользователя могут следовать в цепочке за данным сертификатом. При использовании, значение pathLenConstraint больше или равно нулю. Если значение не установлено, это означает, что лимит на длину цепочки не определен.

Ограничения имени

Дополнение Ограничение имени, должно использоваться только в сертификатах ЦР. Оно указывает пространство имен, внутри которого должны быть расположены все имена Владельцев издаваемых сертификатов. Ограничения могут быть применимы как к имени Владельца (Subject DN), так и к альтернативному имени.

Ограничения определены в терминах допускаемого (permittedSubtrees) или исключаемого (excludedSubtrees) поддерева имен. Любое имя совпадающее с ограничением в исключающем поддереве является некорректным, в независимости от возможного его присутствия в допускаемом поддереве.

Ограничение регламента

Данное дополнение может быть использовано в сертификатах, издаваемых для ЦС. Дополнение Ограничение регламента накладывает ограничения на проверяемую цепочку в двух направлениях. Оно может использоваться для запрещения проверки соответствия регламентов (policy mapping) или требовать, чтобы каждый сертификат в в цепочке содержал приемлемый идентификатор регламента (policy identifier).

Точка распространения СОС

Точка распространения СОС является дополнением, которое определяет механизм и расположение СОС определенного типа в сети, и определяет зону действия СОС как:

1) только для конечных пользователей;
2) только для ЦР;
3) или ограничивает коды мотивации.

Коды мотивировки, ассоциированные с точкой распространения, должны специфицироваться в поле onlySomeReasons. Если поле onlySomeReasons отсутствует, точка распространения должна содержать отзываемые сертификаты для всех кодов. ЦР может использовать Точку распространения СОС как основу для управления потоками данных при компрометации. В этом случае, отзывы сертификатов с кодами keyCompromise и cACompromise располагаются в одной точке распространения, а все остальные в другой.

5.5 Отказ от цифровой подписи

Участник взаимодействия, подписавшийся в электронном документе своей электронной цифровой подписью, не может отказаться от своей электронной цифровой подписи.

Невозможность отказа описана в ITU-T X.813 | ISO / IEC 10181-4 [6] и содержит процесс урегулирования споров, включающий в себя следующие элементы:

1) свидетельство о генерации;

2) свидетельство о передаче, хранении и поиске информации;

3) свидетельство проверки,

4) разрешение споров.

Свидетельство о генерации включает в себя:

- аудит записей, имеющих отношение к событию и утверждение о намерениях;
- заявления, сделанные доверенной третьей стороной;
- политику отчетности;
- цифровую подпись информации;
- временные метки цифровой подписи информации;
- сертификаты поддержки цифровой подписи;
- соответствующую информацию об опубликованных списках отзыва сертификатов, доступных на момент спорных событий.

Целостность данных, которые используются в качестве свидетельства, должна поддерживаться следующими способами:

1) контроль доступа;

2) хранение значений хэш-функции доверенной третьей стороны;

3) подписание цифровой подписью.

Отозванные сертификаты и сертификаты с истекшим сроком действия используются для проверки свидетельства в следующих случаях:

1) если они используются для повторной проверки подписей, созданных в течение срока действия этого сертификата;

2) если доказано, что сертификат был действителен во время спорного случая.

6 Расширение сертификата открытого ключа и расширение CRL

6.1 Обработка политики

6.1.1 Политика сертификата

Расширения сертификатов, определенные в этом разделе, предназначены для использования с сертификатом открытого ключа. Расширения, используемые с сертификатами атрибутов, определены в разделе 12. Расширения CRL, определенные в этом разделе, могут быть использованы в CRL, CARL, а также для ACRL и AARL, определенных в разделе 14 и 15.

Расширения в зависимости от области применения:

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
VA DAVLAT MAZORATI

a) в информации о ключе и политике дают дополнительную информацию о ключах, включающих идентификатор ключей для субъекта и издателя, указатели предназначения или ограничения использования ключа и указатель политики безопасности;

b) в атрибутах субъекта и издателя поддерживают альтернативные названия различных форм имен для субъекта сертификата, для стороны, выдающей сертификат, или для стороны, выдающей CRL;

c) в ограничениях пути сертификации позволяют ограничить детали для включения в сертификаты ЦР или в сертификаты ЦР, выданные другими ЦР, для облегчения автоматизации процесса обработки путей сертификации, когда вовлечена различная политика сертификата. Различная политика сертификата появляется, когда политика различна для различных приложений в среде или когда происходят операции с внешней средой. Ограничения могут ограничивать типы сертификатов, которые могут быть изданы субъектному ЦР или могут впоследствии появляться на пути сертификации;

d) в основных расширениях CRL позволяют включить в CRL указатель причины аннулирования для обеспечения временного приостановления сертификата и для включения последовательности номеров выдачи CRL для возможности обнаружить пользователем сертификата отсутствующий CRL в последовательности исходящего от одного органа издающего CRL;

e) в пунктах распределения CRL и дельта CRL предоставляют полный набор информации об аннулировании исходящего от одного ЦР, чтобы быть разделенными на отдельные CRL и предоставляют информацию об аннулировании исходящих от различных ЦР для комбинирования в одном CRL. Эти расширения также поддерживают использование не полных CRL, указывающих только на изменения с начала выдачи CRL.

Включение какого-либо расширения в сертификат или CRL производится по усмотрению органа, издающего этот сертификат или CRL.

В сертификате или в CRL расширение устанавливается как критическое или некритическое. Если расширение обозначено как критическое и система, использующая сертификат не признает тип поля расширения сертификата или не выполняет семантику расширения, тогда такая система будет рассматривать сертификат как неправильный. Если расширение обозначено как некритическое система, использующая сертификат, признает сертификат и обрабатывает расширение. Типы расширения в этом стандарте указывают, что критичность, или некритичность может быть решена органом, выдающим сертификат или CRL.

Для всех расширений сертификатов, расширений CRL и расширений элементов CRL, определенных в этом стандарте, не должно быть более одного образца каждого типа расширения в любом сертификате, CRL или элементе CRL соответственно.

Данный стандарт содержит в себе три типа объектов: пользователь сертификата, сертифицирующий орган, субъект сертификата (или конечный объект). Каждый объект работает под обязательством двух других объектов, но пользуется ограниченными гарантиями предложенными ими. Эти обязательства и гарантии определены в политике сертификата.

Политика сертификата это документ (обычно на простом и понятном языке). Он может быть ссылкой на уникальный идентификатор, который может быть включен в расширение политики сертификата, выданного сертифицирующим органом конечному объекту и которому пользователь сертификата доверяет. Сертификат может быть выдан в согласованности с более чем несколькими политиками. Определение политики и назначение идентификатора выполняется органом по ведению политики. Набор политик, управляемых органом по ведению политики, называется доменом политики. Данный стандарт не назначает политику безопасности.

Пользователь сертификата может быть обязанным, согласно с политикой сертификата, использовать открытый ключ органа как надежный указатель или полагаться на сертификат, который включает идентификатор объединенной политики. Сертифицирующий орган может быть ограничен в его обязательствах, согласно акту издания сертификата, который включает идентификатор объединенной политики. Конечный объект может быть ограничен в его обязательствах, согласно акту запроса и принятия сертификата, который включает идентификатор объединенной политики, и согласно использованию соответствующего закрытого ключа. Реализации, которые не используют расширения политики сертификата, должны выполнять требуемые ограничения некоторыми другими методами.

Для объекта простое объявление соответствия политике, в целом, не удовлетворяет требованиям доверия других пользователей в структуре. Они требуют некоторых оснований, чтобы доверять, что другие стороны выполняют надежную реализацию политики.

Сертифицирующий орган может установить ограничения на использование его сертификата, чтобы управлять угрозами, которые появляются в результате выдачи сертификата. Например, он может ограничить круг пользователей сертификата, цели для которых они могут использовать, его сертификаты и/или типы и размер ущерба, чтобы он выполнил возмещение в случае ущерба в результате провала на его стороне. Эти вопросы должны быть определены в политике сертификата. Добавочная информация для понятия объектами условия политики может быть включена в расширение политики сертификата в форме спецификатора политики.

6.1.2 Кросс сертификация

Сертифицирующий орган может быть субъектом сертификата, выданного другим ЦР.

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA
MAHSULOT SIFATI

В этом случае сертификат называется кросс сертификатом. Сертифицирующий орган, который является субъектом, называется субъектным ЦР. Сертифицирующий орган, который выдает кросс - сертификат, называется промежуточным ЦР (рисунок 1). Оба сертификата, и кросс - сертификат, и сертификат конечного объекта, могут содержать расширение политики сертификата.

Гарантии и обязательства распределяются субъектным сертифицирующим органом.

Промежуточный Сертифицирующий орган и пользователь сертификата определяются политикой сертификата установленной в кросс - сертификате, согласно которой субъектный сертифицирующий орган может действовать, и от имени конечного объекта. И гарантии и обязательства, распределенные субъекту сертификата, субъектному ЦР и промежуточному ЦР определяются политикой сертификата, указанной в сертификате конечного объекта, согласно которой промежуточный ЦР может действовать как пользователь сертификата и от имени пользователя сертификата.

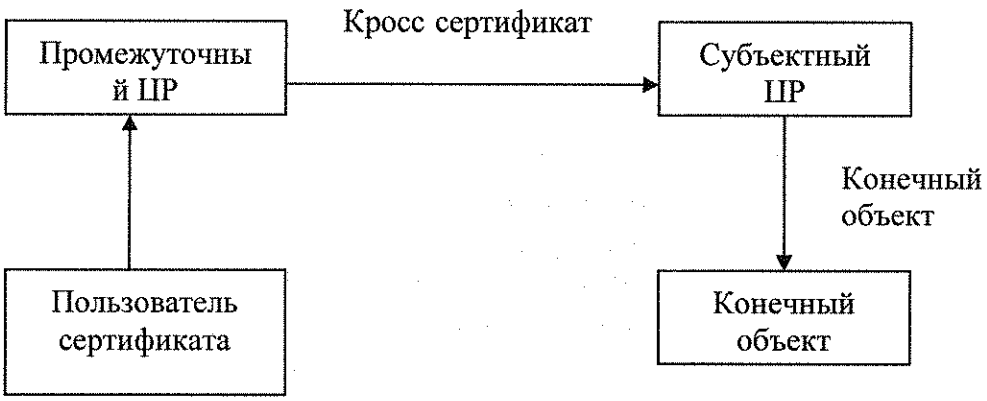


Рисунок 1 – Кросс-сертификация

Промежуточный ЦР может в свою очередь быть субъектом сертификата, выданного другим ЦР, поэтому создается путь сертификации длиною больше чем в два сертификата. За счёт этого при увеличении длины путей сертификации уменьшается достоверность.

Требуется контроль обеспечения того, что сертификаты конечного объекта с неприемлемым низким ассоциированным уровнем достоверности будут отклонены пользователем сертификата. Это часть функции процедуры обработки пути сертификации.

В дополнение к ситуации, описанной выше, существуют два случая:

а) сертифицирующий орган не использует расширение политики сертификата для сообщения его требований политики пользователю сертификата;

b) пользователь сертификата или промежуточный сертифицирующий орган делегирует привилегии контроля политики следующему органу в пути.

В первом случае сертификат вообще не должен содержать расширение политики сертификата. Результатом является набор политик, согласно которым путь имеет незначительную силу. В этом случае путь может быть действительным. Пользователи сертификата должны быть уверены, что они используют сертификат в соответствии с политикой органов указанных в пути.

Во втором случае пользователь сертификата или промежуточный ЦР должен включать специальное значение any-policy в initial policy set или в кросс сертификат. В случае, когда сертификат включает в себя значение any-policy он не должен включать любой другой идентификатор политики сертификата. Идентификатор any-policy не должен иметь ассоциированный спецификатор политики.

Пользователь сертификата может быть уверенным, что все его обязанности выражаются в соответствии со стандартом, устанавливаемым индикатором initial-explicit-policy. Таким образом, только органы, которые используют стандартное расширение политики сертификата как их метод успешного выполнения ограничений, признаются в пути сертификации и пользователи сертификата не имеют дополнительных обязанностей. Так как органы также влекут за собой обязанности, когда они действуют, как пользователь сертификата или от имени пользователя сертификата, они могут гарантировать, что все их обязанности передаются согласно стандартно устанавливаемому значению requireExplicitPolicy в кросс сертификате.

6.1.3 Преобразование политики

Некоторые пути сертификации могут пересекаться между границами доменов политики. Гарантия и обязательства, согласно которым выдан кросс сертификат, могут быть эквивалентными некоторым или всем гарантиям и обязательствам, согласно которым субъектный сертифицирующий орган выдает сертификаты конечным объектам, несмотря на то, что политика органов, согласно которыми работают два ЦР, может иметь выделенные различные уникальные идентификаторы для этих эквивалентных политик. В этом случае промежуточный сертифицирующий орган может включать расширение преобразования политики в кросс сертификат. В расширении схемы политики промежуточный сертифицирующий орган удостоверяет пользователя сертификата о том, что он будет продолжать обладать схожими обязательствами, а также он должен продолжать выполнять схожие обязательства, несмотря на то, что последующие объекты в пути сертификации работают в другом домене политики. Промежуточный сертифицирующий орган должен включать один или несколько преобразований для каждого подмножества политик, согласно которым он издает кросс сертификат и он не должен включать в себя преобразования каких-либо других политик. Если один или

ZSTANDART AGENTLIGI
STANDARTASHTIRISH,
MUVOFIQLASHTIRISH
VA BAHOLAT NAZORATI

несколько политик сертификата, согласно которым работает субъектный сертифицирующий орган, идентичны тем политикам, согласно которым работает промежуточный сертифицирующий орган (т.е. он имеет одинаковый уникальный идентификатор), тогда эти идентификаторы должны быть исключены из расширения преобразования политики, кроме тех, которые включены в расширение политики сертификата.

Преобразование политики имеет свойство преобразования всех идентификаторов политики в сертификате на эквивалентную политику идентификатора пути сертификации также, как признает пользователя сертификата.

Политика не должна быть преобразована в специальное значение или из специального значения any-policy.

Пользователи сертификата, которые могут определить, что сертификаты выданы в домене политики, отличающемся от их собственных, не должны быть уверенными, даже если что доверенный промежуточный ЦР может установить его политику, чтобы она была эквивалентной политике пользователей. Промежуточный ЦР может сделать это, установлением initial-policy-mapping-inhibit в процедуре установления пути. Кроме того, промежуточный сертифицирующий орган может сделать похожее определение от имени его пользователя сертификата. Для того чтобы гарантировать, что пользователи сертификата правильно выполняют это требование, он может установить inhibitPolicyMapping в расширении ограничений политики.

6.1.4 Обработка пути сертификации

Пользователь сертификата имеет возможность выбора одной из следующих стратегий:

a) он может требовать, что путь сертификации будет удовлетворять как минимум одному из заранее установленных наборов политики;

b) он может просить сообщение у модуля подтверждения пути о наборе политик, для которых действителен путь сертификации.

Первая стратегия может быть самой подходящей, когда пользователь сертификата заранее знает набор политики, которая доступна для использования им.

Вторая стратегия может быть самой подходящей, когда пользователь сертификата не знает заранее набора политик, которые доступны для его использования.

В первом случае процедура утверждения пути сертификации указывает на путь, который действителен, только если он имеют силу согласно одной или более политик, определенных в initial-policy-set, и он возвращает поднабор initial-policy-set, согласно которому путь имеет силу.

Во втором случае процедура утверждения пути сертификации может указывать, что путь не имеет силу согласно initial-policy-set, но имеет силу согласно расчлененному набору: authorities-constrained-policy-set. Тогда пользователь сертификата должен определить его намерения использовать

сертификат согласно с одной или более политиками сертификата, согласно которым путь имеет силу. Но, устанавливая значение initial-policy-set на any-policy, пользователь сертификата может использовать процедуру для получения действительного результата, если путь имеет силу согласно любой политике.

6.1.5 Сертификаты выданные самому себе

Существует три типа сертификата, которые сертифицирующий орган может издавать самому себе:

a) сертификат, используемый для удобного шифрования открытого ключа, связи с пользователями сертификата и хранения пользователями сертификата;

b) сертификат для удостоверения использования ключа при подписании сертификатов пользователей и CRL;

c) сертификат для замены своих собственных сертификатов, срок которых истек.

Эти типы сертификатов называются сертификатами выданными самому себе, и они могут быть признаны действительными согласно тому факту, что имя издателя и субъекта, представленные в них, являются идентичными. Для утверждения действительности пути сертификации сертификаты типа a), выданные самому себе, проверяются открытым ключом, содержащимся в нем, и если они встречаются в пути сертификации, они должны быть игнорированы.

Сертификат типа b), выданный самому себе, может использоваться только как конечный сертификат в пути и должен быть обработан как последний сертификат.

Сертификат типа c), выданный самому себе, (также известный как промежуточный сертификат, выданный самому себе) может использоваться в пути как промежуточный сертификат. Если сертификаты, выданные самому себе, встречаются в пути, они должны быть обработаны как промежуточные сертификаты, но со следующим исключением: они не действуют на длину пути во время обработки расширения basicConstraints компоненты pathLenConstraint и значений skip-certificates, связанных с индикаторами policy-mapping-inhibit-pending и explicit-policy-pending.

6.2 Расширения информации о ключе и о политике

6.2.1 Необходимые условия

Установлены следующие условия, связанные с информацией о ключе и о политике:

a) ключевая пара ЦР может обновляться в одинаковые интервалы или при специальных обстоятельствах. В сертификате необходимо поле для выражения идентификатора открытого ключа, чтобы использовать его для верификации подписи сертификата. Система, использующая сертификат,

O‘ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
DAVLAT NAZORATI
QO‘MITASI

может использовать такие идентификаторы при обнаружении правильного сертификата ЦР для проверки открытого ключа издателя сертификата;

b) вообще, субъект сертификата имеет различные открытые ключи и соответственно различные сертификаты для разных целей, например для цифровой подписи и для согласования ключей шифрования. Поле сертификата необходимо для способствования пользователю сертификата при выборе правильного сертификата для данного субъекта для особых целей или чтобы позволить ЦР ставить условие, что сертифицированный ключ можно использовать только в особых целях;

c) обновление ключевой пары субъекта производится регулярно или при специальных обстоятельствах. Необходимо поле в сертификате для сообщения идентификатора, для определения различия между открытыми ключами для одинаковых субъектов, используемых в различные моменты времени. Система, использующая сертификат, может использовать такие идентификаторы при обнаружении правильного сертификата;

d) закрытый ключ, соответствующий сертифицированному открытому ключу, типично используется в различных периодах времени, исходя из действительности открытого ключа.

Период использования для подписывающего закрытого ключа короче, чем для проверяющего открытого ключа. Период действительности сертификата указывает на период, в течение которого может быть использован открытый ключ, для которого не обязательно соответствовать периоду использования закрытого ключа. В случае, когда закрытый ключ скомпрометирован, период использования сертификата может быть ограничен, если сторона, верифицирующая подпись, знает период использования закрытого ключа. Поэтому существуют условия, дающие возможность указания в сертификате периода использования закрытого ключа;

e) необходимо включить в сертификат информацию о политике сертификата из-за того, что сертификат может быть использован в среде, где применяется множество политик сертификата;

f) при кросс-сертификации одной организации со стороны другой иногда могут быть соглашения о том, что определенные политики двух организаций могут быть эквивалентными. Сертификат ЦР должен позволить издателю сертификата указать, что одна из его собственных политик сертификата эквивалентна политике сертификата другого субъектного ЦР. Это называется преобразованием политики;

g) пользователь системы шифрования или цифровой подписи, который использует сертификаты, определенные в этом стандарте, должен иметь возможность заранее определить алгоритмы, поддерживаемые другими пользователями.

6.2.2 Поля расширения сертификата открытого ключа и расширения CRL

6.2.2.1 Расширение идентификатора ключа органа

Определены следующие поля расширения:

a) идентификатора ключа органа;

b) идентификатора ключа субъекта;

c) использования ключа;

d) расширенного использования ключа;

e) периода использования закрытого ключа;

f) политики сертификата;

j) преобразования политики.

Эти поля расширения используются только как расширения сертификата, за исключением идентификатора ключа органа, который может быть также использован как расширение CRL. Расширения могут быть использованы и в сертификатах ЦР и в сертификатах объекта.

Расширение идентификатора ключа органа может быть использовано и как расширение сертификата и как расширение CRL. Оно идентифицирует открытый ключ, используемый для проверки подписи на данном сертификате или CRL. Оно даёт возможность отличать различные ключи, используемые одним и тем же ЦР. Это поле определяется следующим образом:

authorityKeyIdentifier EXTENSION ::= {
SYNTAX AuthorityKeyIdentifier
IDENTIFIED BY id-ce-authorityKeyIdentifier }
AuthorityKeyIdentifier ::= SEQUENCE {
keyIdentifier [0] KeyIdentifier OPTIONAL,
authorityCertIssuer [1] GeneralNames OPTIONAL,
authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL }
(WITH COMPONENTS {..., authorityCertIssuer PRESENT,
authorityCertSerialNumber PRESENT} |
WITH COMPONENTS {..., authorityCertIssuer ABSENT,
authorityCertSerialNumber ABSENT})
KeyIdentifier ::= OCTET STRING

Ключ может быть установлен определением идентификатора ключа в компоненте keyIdentifier идентифицированием сертификата для ключа (указанием издателя сертификата в компоненте authorityCertIssuer и серийном номере сертификата в компоненте), или определением и идентификатора ключа идентификацией сертификата для ключа. Если используются обе формы идентификации, тогда издатель сертификата или CRL должен гарантировать, что они совместимы. Идентификатор ключа должен быть уникальным по отношению всех идентификаторов ключей для издающего органа для сертификата или CRL, содержащих расширение. Реализация обработки сертификата, которая поддерживает

O‘ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
DAVLAT QO‘MITASI

это расширение, должна иметь возможность обработать все формы имен в компоненте authorityCertIssuer.

Сертифицирующие органы должны назначать серийные номера сертификата, так что каждая пара (издатель, серийный номер сертификата) уникально идентифицировала один сертификат.

Форма keyIdentifier может быть использована для выбора сертификатов сертифицирующего органа при построении пути. Пара authorityCertIssuer и authoritySerialNumber может быть использована для предоставления преимущественного права над другими сертификатами в пути. Это расширение всегда имеет некритическое значение.

6.2.2.2 Расширение идентификатора ключа субъекта

Это поле идентифицирует сертифицируемый открытый ключ. Оно дает возможность различать различные ключи, используемые одним субъектом. Это поле определяется следующим образом:

subjectKeyIdentifier EXTENSION ::= {
SYNTAX SubjectKeyIdentifier
IDENTIFIED BY id-ce-subjectKeyIdentifier }
SubjectKeyIdentifier ::= KeyIdentifier

Идентификатор ключа должен быть уникальным по отношению всех идентификаторов ключа для субъекта, вместе с которыми он используется. Это расширение всегда имеет некритическое значение.

6.2.2.3 Расширение использования ключа

Это поле, указывает цели, для которых используется сертифицированный открытый ключ. Это поле определяется следующим образом:

keyUsage EXTENSION ::= {
SYNTAX KeyUsage
IDENTIFIED BY id-ce-keyUsage }
KeyUsage ::= BIT STRING {
digitalSignature (0),
nonRepudiation (1),
keyEncipherment (2),
dataEncipherment (3),
keyAgreement (4),
keyCertSign (5),
cRLSign (6),
encipherOnly (7),
decipherOnly (8) }

Это расширение может быть выбрано издателем сертификата, и оно может быть критическим или некритическим.

6.2.2.4 Расширение расширенного использования ключа

Это поле указывает на цели, для которых может быть использован сертификат открытого ключа, в дополнение к (или вместо) основным целям, указанным в поле расширения использования ключа. Это поле определяется следующим образом:

extKeyUsage EXTENSION ::= {
SYNTAX SEQUENCE SIZE (1..MAX) OF KeyPurposeId
IDENTIFIED BY id-ce-extKeyUsage }
KeyPurposeId ::= OBJECT IDENTIFIER

Цели использования ключа могут быть определены любой организацией при необходимости.

Это расширение по выбору издателя сертификата может быть критическим или некритическим.

Если это расширение отмечено как критическое, тогда этот сертификат будет использован только для одной из указанных целей.

Если расширение отмечено как некритическое, тогда это означает цель или цели использования ключа и может быть использовано для нахождения правильного ключа/сертификата объекта, который имеет множество ключей/ сертификатов. Если это расширение представлено и система, использующая сертификат, распознает и обрабатывает расширение типа keyUsage, тогда эта система будет гарантировать, что сертификат будет использован только для одной из установленных целей.

Если сертификат содержит и критическое поле использования ключа, и критическое поле расширенного использования ключа, тогда оба поля должны быть обработаны независимо и сертификат должен быть использован в целях, соответствующих этим полям.

Если в сертификате нет цели, соответствующей этим полям, тогда сертификат не должен быть использован ни в каких целях.

6.2.2.5 Расширение периода использования закрытого ключа

Это поле указывает на период использования закрытого ключа, соответствующего сертифицированному открытому ключу. Оно применимо только для ключей ЭЦП. Это поле определяется следующим образом:

privateKeyUsagePeriod EXTENSION ::= {
SYNTAX PrivateKeyUsagePeriod
IDENTIFIED BY id-ce-privateKeyUsagePeriod }
PrivateKeyUsagePeriod ::= SEQUENCE {
notBefore [0] GeneralizedTime OPTIONAL,
notAfter [1] GeneralizedTime OPTIONAL }
(WITH COMPONENTS {..., notBefore PRESENT} |
WITH COMPONENTS {..., notAfter PRESENT})

O'ZBEKISTON RESPUBLIKASI
STANDARTLARI
MUVOFIQLIGI
VA MAVLAN NATURATI
BORDONOSI

Компонента notBefore указывает дату и время, начиная с которого закрытый ключ может быть использован для подписания. Если компонента notBefore не существует, тогда не существует информации о начале действительного периода использования закрытого ключа.

Компонента notAfter указывает на дату и время окончания использования закрытого ключа для подписания. Если компонента notAfter не существует, то не имеется информация об окончании периода использования закрытого ключа.

Это расширение всегда некритическое.

6.2.2.6 Расширение политик сертификата

Это поле перечисляет политики сертификата, признанные издающим ЦР, которые применяются к сертификату вместе с дополнительным (необязательным) спецификатором информации, соответствующим политикам сертификата, приведенным в этом поле. Список политик сертификата применяется для определения действительности пути сертификации, как описано в разделе 7. Дополнительный спецификатор информации не используется в процедуре обработки пути сертификации, но релевантные спецификаторы представляются как результат (выходной параметр) этого процесса для приложения, использующего сертификат, для содействия при определении действительно ли путь соответствует для данной транзакции. Различные политики сертификата могут иметь отношение к различным приложениям, которые могут использовать сертифицированный ключ. Наличие этого расширения в сертификате конечного объекта указывает на политику сертификата, для которой этот сертификат является действительным. Наличие этого расширения в сертификате, изданном одним ЦР для другого ЦР указывает политики сертификата, для которых могут быть действительными пути сертификации, содержащие этот сертификат. Это поле определяется следующим образом:

certificatePolicies EXTENSION ::= {
SYNTAX CertificatePoliciesSyntax
IDENTIFIED BY id-ce-certificatePolicies }
CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation
PolicyInformation ::= SEQUENCE {
policyIdentifier CertPolicyId,
policyQualifiers SEQUENCE SIZE (1..MAX) OF
PolicyQualifierInfo OPTIONAL }
CertPolicyId ::= OBJECT IDENTIFIER
PolicyQualifierInfo ::= SEQUENCE {
policyQualifierId CERT-POLICY-QUALIFIER.&id
({SupportedPolicyQualifiers}),
qualifier CERT-POLICY-QUALIFIER.&Qualifier

({SupportedPolicyQualifiers}{@policyQualifierId})
OPTIONAL }
SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }

Значение типа PolicyInformation указывает и сообщает о спецификаторе информации для одной политики сертификата. Компонента policyIdentifier содержит идентификатор политики сертификата, а компонента policyQualifiers содержит значения спецификатора политики для данного элемента.

Это расширение может быть, по выбору издателя сертификата, критическим или некритическим.

Если расширение отмечено как критическое, то это означает, что сертификат должен быть использован только в целях и согласно с правилами, подразумеваемыми в одной из установленных политик сертификата. Правила отдельной политики могут требовать у системы, использующей сертификат, обработки значения спецификатора отдельными методами.

Если расширение отмечено как некритическое, то не обязательно ограничивать использование сертификата в рамках перечисленных политик. Однако, пользователь сертификата может потребовать определенную политику для того, чтобы использовать сертификат (см. раздел 7). Спецификатор политики по выбору пользователя сертификата может быть обработанным или игнорироваться.

Политики сертификата и типы спецификаторов политики сертификата могут быть определены какими-либо организациями по необходимости. ЦР может утверждать любую политику, используя идентификатор anyPolicy для того, чтобы можно было использовать сертификат для всех возможных политик. Из-за необходимости идентификации этого специального значения, чтобы применять его, не обращая внимания на приложение или среду, этот идентификатор объекта определен в этом стандарте. Этот стандарт не устанавливает идентификатор объекта для конкретной политики сертификата. Установление идентификатора объекта является обязанностью объекта, который определяет политику сертификата.

anyPolicy OBJECT IDENTIFIER ::= { 2 5 29 32 0 }

Идентификатор anyPolicy не должен иметь никаких связанных спецификаторов политики.

Следующий класс языка ASN.1 используется для определения типов спецификаторов политики сертификата:

CERT-POLICY-QUALIFIER ::= CLASS {
&id OBJECT IDENTIFIER UNIQUE,
&Qualifier OPTIONAL }

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA
MUVOFIQLASHTIRISH
DAVLAT NAZORATI
BO'YICHA

WITH SYNTAX {
POLICY-QUALIFIER-ID &id
[QUALIFIER-TYPE &Qualifier] }

Определение типа спецификатора политики должно включать:

- формулировку семантики возможных значений;
- знак идентификатора спецификатора, который может появиться в расширении политики сертификата без сопровождающего значения.

6.2.2.7 Расширение преобразования политики

Это поле, которое должно быть использовано только в сертификате ЦР, позволяет издателю указать, что для целей пользователя пути сертификации, содержащего этот сертификат, одна из политик сертификата издателя может рассматриваться эквивалентной другой политике сертификата, используемой в домене субъектного ЦР. Это поле определяется следующим образом:

policyMappings EXTENSION ::= {
SYNTAX PolicyMappingsSyntax
IDENTIFIED BY id-ce-policyMappings }
PolicyMappingsSyntax::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
issuerDomainPolicy CertPolicyId,
subjectDomainPolicy CertPolicyId }

Компонента issuerDomainPolicy указывает политику сертификата, которая признается в домене издающего ЦР и может приниматься эквивалентной политике сертификата, указанной в компоненте subjectDomainPolicy, которая представлена в домене субъектного ЦР.

Политика не должна преобразовываться к специальному значению anyPolicy.

Это расширение может быть по выбору издателя сертификата или критическим или некритическим. Рекомендуется, чтобы оно было критическим, иначе пользователь сертификата может некорректно интерпретировать условия издающего ЦР.

6.3 Расширения информации о субъекте и издателе сертификата

6.3.1 Необходимые условия

Следующие условия относятся к субъектам сертификата и атрибутам издателя сертификата:

а) сертификаты должны быть пригодны для использования приложениями, которые используют различные формы названий, включающие названия электронной почты, название домена Интернет, адреса источника/получателя ITU-T X.400 [7] и названия сторон электронного документооборота. Это необходимо чтобы была

возможность безопасно связывать множество названий различных форм названий с субъектом сертификата или с издателем сертификата или CRL;

b) пользователь сертификата может нуждаться в том, чтобы безопасно узнать, надёжную идентифицирующую информацию о субъекте, чтобы быть уверенным, что субъект действительно подразумеваемая личность или объект. Например, может быть необходима такая информация как почтовый адрес, положение в корпорации, или фотография. Такая информация может быть легко представлена как атрибуты директории, но эти атрибуты не обязательная часть различительного имени.

6.3.2 Поля сертификата и расширения CRL

6.3.2.1 Расширение альтернативного имени субъекта

Определены следующие поля расширений:

a) альтернативное имя субъекта;

b) альтернативное имя издателя;

c) директория атрибутов субъекта.

Эти поля должны быть использованы только как расширения сертификата, кроме альтернативного имени издателя, которое также может быть использовано как расширение CRL. Также как расширение сертификата они могут быть представлены в сертификате сертифицирующего органа или в сертификате конечного объекта.

Расширение альтернативного имени субъекта содержит одно или несколько альтернативных имен, использующих несколько разных форм имен для объекта, который ограничивается сертифицирующим органом посредством сертифицированного открытого ключа. Это поле определяется следующим образом:

```
subjectAltName EXTENSION ::= {
SYNTAX  GeneralNames
IDENTIFIED BY  id-ce-subjectAltName }
GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName
GeneralName ::= CHOICE {
otherName                    [0]  INSTANCE OF OTHER-NAME,
rfc822Name                   [1]  IA5String,
dNSName                      [2]  IA5String,
x400Address                  [3]  ORAddress,
directoryName                [4]  Name,
ediPartyName                 [5]  EDIPartyName,
uniformResourceIdentifier    [6]  IA5String,
iPAddress                    [7]  OCTET STRING,
registeredID                 [8]  OBJECT IDENTIFIER }
OTHER-NAME ::= TYPE-IDENTIFIER
EDIPartyName ::= SEQUENCE {
nameAssigner                 [0]  DirectoryString {ub-name}
OPTIONAL,
```

O'ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
DAVLAT QO'MITASI

partyName **[1] DirectoryString {ub-name} }**

Значения в альтернативах типа GeneralName это названия различных форм, которые определяются следующим образом:

- otherName - это название любой формы, определенной как образец OTHER-NAME класса информационного объекта;
- rfc822Name - это адрес электронной почты в Интернете;
- dNSName - это имя домена в Интернете;
- x400Address - это адрес источника/получателя;
- directoryName - это название директории;
- ediPartyName - это название формы, согласованное между связанными партнерами электронного документооборота; компонента nameAssigner идентифицирует орган, который назначает уникальное значение имени в компоненте partyName;
- uniformResourceIdentifier уникальный идентификатор ресурсов для всемирной сети (WWW);
- iPAddress адрес Интернет-протокола, определяемый как двоичная последовательность;
- registeredID идентификатор какого либо зарегистрированного объекта.

Для каждой формы имени, использованной в типе GeneralName, будет существовать система регистрации имен, которая будет гарантировать, что любое использованное имя однозначно идентифицирует один объект и для издателя сертификата, и для пользователя сертификата.

Это расширение по выбору издателя сертификата может быть либо критическим, либо некритическим. Системе, использующей сертификат, которая поддерживает это расширение, не обязательно иметь возможность обрабатывать все формы имен. Если расширение отмечено как критическое, то хотя бы одна из существующих форм имени должна быть опознана и обработана, иначе сертификат будет определен как недействительный. Кроме вышеперечисленных ограничений системе, использующей сертификат, позволено игнорировать любое название с нераспознанной или неподдерживаемой формой имени.

Рекомендуется, чтобы предоставленное поле субъекта сертификата, содержащее название директории, которое однозначно идентифицирует субъект, было установлено как некритическое.

Если это расширение представлено и установлено как критическое, поле subject может содержать нулевое название (например, последовательность нулей соответствующих определенному названию), если субъект идентифицирован в этом расширении только названием или названиями.

6.3.2.2 Расширение альтернативного имени издателя и расширение директории атрибутов субъекта

Это поле содержит одно или несколько альтернативных имен, использующих несколько разных форм имен, для издателя сертификата и CRL. Это поле определяется следующим образом:

issuerAltName EXTENSION ::= {
SYNTAX GeneralNames
IDENTIFIED BY id-ce-issuerAltName }

Это расширение по выбору издателя сертификата или CRL может быть критическим или некритическим. Системам, использующим сертификат, которые поддерживают это расширение, не обязательно иметь возможность обработки всех форм имени. Если расширение отмечено как критическое, хотя бы одна из существующих форм имен должна быть опознана и обработана, иначе сертификат должен быть определен как недействительный. Кроме вышеперечисленных ограничений системе, использующей сертификат, позволено игнорировать любое имя с нераспознанной или неподдерживаемой формой имени. Рекомендуется, чтобы предоставленное поле издателя сертификата, содержащее название директории, которое однозначно идентифицирует издающий орган, было установлено как некритическое.

Расширение директории атрибутов субъекта сообщает любое необходимое значение атрибута директории для субъекта сертификата. Это поле определяется следующим образом:

subjectDirectoryAttributes EXTENSION ::= {
SYNTAX AttributesSyntax
IDENTIFIED BY id-ce-subjectDirectoryAttributes }
AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute

Это расширение всегда некритическое.

В этом расширении представляется открытый ключ, могут также быть представлены некоторые другие расширения.

6.4 Расширения ограничения пути сертификации

6.4.1 Необходимые условия

Для обработки пути сертификации:

а) сертификаты конечного объекта должны быть отличны от сертификатов ЦР для предотвращения того, чтобы конечные объекты не могли без авторизации объявить себя как ЦР. Это также необходимо, чтобы была возможность для ЦР установить лимит на длину последующей цепи сертификатов, получаемой из сертифицированных субъектных ЦР;

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA
SERTIFIKATLASHTIRISH
VA DAVLAT NAZORATI

b) для ЦР должна иметься возможность указать ограничения, которые позволяют пользователю сертификата проверять то, что менее доверенные ЦР в пути сертификации (т.е. самые отдаленные ЦР от ЦР, с которого начинается сертификат открытого ключа пользователя) не нарушают их доверия, выдавая сертификаты субъектам в не соответствующем пространстве имен. Строгое соблюдение этих ограничений необходимо, чтобы иметь возможность автоматической проверки пользователем сертификата;

c) обработка пути сертификации должна быть реализуема в автоматизированном, автономном модуле. Это необходимо, чтобы допустить применение достоверных программных или аппаратных модулей при выполнении функций обработки пути сертификации;

d) должна быть возможность реализации обработки пути сертификации независимо от взаимодействия с локальным пользователем в реальном масштабе времени;

e) должна иметься возможность реализации обработки пути сертификации независимо от использования достоверной локальной базы данных информации об описании политики.

Необходима некоторая достоверная локальная информация, такая как исходный открытый ключ, для обработки пути сертификации, но количество такой информации должно быть минимизировано;

f) пути сертификации должны действовать в средах, где признаются многочисленные политики сертификата. ЦР должен иметь возможность ставить условия ЦР в других доменах, которым он доверяет. Должно поддерживаться формирование цепочки посредством многочисленных доменов политик;

g) требуется полная гибкость в моделях доверия. Для отдельной организации достаточна строгая иерархическая модель, но не достаточна, когда учитываются нужды многочисленных связанных предприятий. Гибкость необходима при выборе первого доверенного ЦР в пути сертификации. В частности, должна быть возможность требовать, чтобы путь сертификации начинался в локальном домене безопасности в системе открытого ключа пользователя;

h) структура присвоения имен не должна быть ограничена необходимостью использования имен в сертификате, т.е. структуры имен директории, учтенных для организаций или географических областей, и имена не должны устанавливаться для приспособления к требованиям сертифицирующего органа;

i) сертифицирующий орган должен иметь возможность запрещать использовать преобразование политики и требовать определенных идентификаторов политики сертификата, чтобы представлять в последующих сертификатах в пути сертификации.

В любой системе, использующей сертификат, обработка пути сертификации требует выбора соответствующего уровня доверия. Этот стандарт определяет функции, которые могут быть использованы при

реализациях, которые требуют приспосабливаемости к специфическим требованиям доверия. Уровень доверия должен быть соизмерим с коммерческими угрозами. Функции обработки пути сертификации должны быть реализуемы в аппаратном криптографическом модуле или в криптографических средствах, как одна из опций;

k) ЦР должен иметь возможность предотвращать (приостанавливать) особые значения какой-либо политики, полученной из обоснованной действительной политики, в последующих сертификатах пути сертификации.

6.4.2 Поля расширения сертификата

6.4.2.1 Расширение основных ограничений

Определены следующие поля расширения:

a) основные ограничения;

b) ограничение названия;

c) ограничения политики;

d) запрет какой-либо политики.

Эти расширения должны быть использованы только как расширения сертификата.

Ограничения в названиях и ограничения политики должны быть использованы только в сертификатах ЦР, простые ограничения могут быть также использованы в сертификатах конечного объекта.

Поле расширения основных ограничений указывает, что субъект работает как ЦР с сертифицированным открытым ключом, используемым для проверки подписей сертификата.

Если это так, то также может быть определено ограничение длины пути сертификации. Это поле определяется следующим образом:

basicConstraints EXTENSION ::= {
SYNTAX BasicConstraintsSyntax
IDENTIFIED BY id-ce-basicConstraints }
BasicConstraintsSyntax ::= SEQUENCE {
cA BOOLEAN DEFAULT FALSE,
pathLenConstraint INTEGER (0..MAX) OPTIONAL }

Компонента cA указывает, что сертифицированный открытый ключ может быть использован для проверки подписи сертификата.

Компонента pathLenConstraint должна быть представлена, только если cA установлен на значении TRUE. Оно показывает максимальное количество сертификатов ЦР, которые могут следовать за этим сертификатом в пути сертификации. Значение 0 означает, что субъект данного сертификата может издавать сертификаты только для конечных объектов, но не для последующих ЦР. Если ни в одном из сертификатов, находящихся в пути сертификации нет поля pathLenConstraint, то в этом случае не существует лимита, ограничивающего длину пути сертификации.

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA DAVLAT GUVOHNOMASI

Это расширение может быть по выбору издателя сертификата критическим или некритическим. Рекомендуется, чтобы он был установлен как критический, иначе объект, который не уполномочен быть ЦР, может выдавать сертификаты и система, использующая сертификат, может случайно использовать такой сертификат.

Если это расширение представлено и установлено как критическое или установлено как некритическое, но признано системой использующей сертификат, тогда:

- если значение cA не установлено на TRUE, то сертификат открытого ключа не должен быть использован для проверки подписи сертификата;
- если значение cA установлено на TRUE и представлено pathLenConstraint, тогда система, использующая сертификат должна проверить, что обрабатываемый путь сертификации согласован со значением pathLenConstraint.

Если это расширение не представлено или установлено как некритическое и признаётся системой, использующей сертификат, тогда сертификат будет рассматриваться, как сертификат конечного объекта, и не может быть использован для проверки подписи сертификата.

Для ограничения субъекта сертификата, чтобы он был только конечным объектом, т.е. не ЦР, издатель может включить это поле расширения, содержащее только пустое значение SEQUENCE.

6.4.2.2 Расширение ограничения названия

Это поле, которое будет использовано только в сертификате ЦР, указывает на пространство имен, в которые должны быть помещены все имена субъекта в последующих сертификатах в пути сертификации. Это поле определяется следующим образом:

```
nameConstraints EXTENSION ::= {
SYNTAX    NameConstraintsSyntax
IDENTIFIED BY   id-ce-nameConstraints }
NameConstraintsSyntax::= SEQUENCE {
permittedSubtrees   [0]   GeneralSubtrees OPTIONAL,
excludedSubtrees   [1]   GeneralSubtrees OPTIONAL }
GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree
GeneralSubtree ::= SEQUENCE {
base      GeneralName,
minimum     [0]  BaseDistance DEFAULT 0,
maximum     [1]  BaseDistance OPTIONAL }
BaseDistance ::= INTEGER (0..MAX)
```

Если существуют компоненты permittedSubtrees и excludedSubtrees, то каждая из них определяет одно или несколько поддеревьев названий, каждое из которых определяется именем корневого поддерева и в пределах того поддерева, область которого ограничивается верхним и/или нижним

уровнями. Если представлена компонента permittedSubtrees, из всех сертификатов, изданных субъектным ЦР и последующими ЦР, указанными в пути сертификации, допустимыми являются только те сертификаты, которые имеют имена субъекта в пределах этих поддеревьев. Если представлена компонента excludedSubtrees, то любой сертификат, изданный субъектным ЦР или последующими в пути сертификации ЦР, не допустим, если он имеет имя субъекта в пределах этих поддеревьев. Если обе компоненты существуют, и пространства имен частично совпадают, то выбирается та часть поддерева, которая имеет преимущество.

Формы имен доступны благодаря типу GeneralName, в этом поле могут быть использованы только те формы имен, которые имеют хорошо определенную иерархическую структуру. Форма имени directoryName удовлетворяет этому требованию, когда поддерево имен, использующее эту форму имен, соответствует поддереву DIT. Если расширение установлено как критическое и система, использующая сертификат не признает форму имен, использованную в какой либо компоненте base, сертификат должен быть обработан как в случае появления непризнанного критического расширения. Если расширение установлено как некритическое и система, использующая сертификат, не признает форму имени, использованную в какой либо компоненте base, тогда спецификация этого поддерева может быть игнорирована.

Когда проверяются имена субъекта сертификата на соответствие с ограничением имени, некритические расширения альтернативного имени субъекта должны быть обработаны, а не игнорированы.

Поле minimum определяет верхнюю границу области в пределах поддерева. Все имена, чьи конечные компоненты имени выше определенного уровня, не входят в пределы области.

Значение minimum равное нулю (значению по умолчанию) соответствует базовому, т.е. наивысшему узлу поддерева. Например, если minimum установлено на 1, тогда поддерево имен исключает начальный узел, но включает все подчиненные узлы.

Поле maximum определяет нижнюю границу области в пределах поддерева. Все имена, чьи конечные компоненты имени ниже определенного уровня, не входят в пределы области.

Значение maximum равное нулю соответствует начальному, т.е. верхушке поддерева.

Отсутствие компоненты maximum указывает, что не должно быть нижней границы, накладываемой на область в пределах поддерева. Например, если значение maximum равно 1 тогда поддерево имен исключает все узлы кроме базового поддерева и его непосредственных подчиненных. Это расширение по выбору издателя может быть критическим или некритическим.

Рекомендуется, чтобы оно устанавливалось как критическое, иначе пользователь сертификата может не проверять, что последующий

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
DAVLAT QO'MITASI

сертификат в пути сертификации расположен в пространстве имен, сокращенного органом сертификации.

Если это расширение представлено и установлено как критическое или как некритическое, но признано системой, использующей сертификат, тогда система, использующая сертификат, будет проверять обрабатываемый путь сертификации на согласованность со значением в этом расширении.

6.4.2.3 Расширение ограничений политики

Это поле определяет ограничения, которые могут требовать идентификации политики сертификата или запрет политики преобразования для остатка пути сертификации. Это поле определяется следующим образом:

```
policyConstraints EXTENSION ::= {
SYNTAX    PolicyConstraintsSyntax
IDENTIFIED BY   id-ce-policyConstraints }
PolicyConstraintsSyntax ::= SEQUENCE {
requireExplicitPolicy    [0] SkipCerts OPTIONAL,
inhibitPolicyMapping   [1] SkipCerts OPTIONAL }
SkipCerts ::= INTEGER (0..MAX)
```

Если представлена компонента requireExplicitPolicy и путь сертификации включает в себя сертификат, изданный назначенным ЦР, то необходимо для всех сертификатов в пути в расширении политики сертификата содержать идентификатор допустимой политики. Идентификатор допустимой политики это идентификатор политики сертификата, требуемый пользователем пути сертификации, идентификатор политики, которого объявлен эквивалентным одной из этих политик посредством преобразования политики или any policy.

Назначенный ЦР это или ЦР, издающий сертификат, содержащий это расширение (если значение requireExplicitPolicy равно 0) или ЦР, который является субъектом последующего сертификата в пути сертификации.

Если существует компонента inhibitPolicyMapping, это означает, что во всех сертификатах, начиная от назначенного ЦР в пути сертификации до конца пути сертификации, не разрешается преобразование политики. Назначенный ЦР это или субъектный ЦР сертификата, содержащего это расширение (если значение inhibitPolicyMapping равно 0), или ЦР, который является субъектом последующего сертификата в пути сертификации.

Значение типа SkipCerts указывает количество сертификатов в пути сертификации, которые должны быть пропущены, перед тем как ограничения станут действительными.

Это расширение может быть по выбору издателя сертификата критическим или некритическим. Рекомендуется, чтобы оно устанавливалось, как критическое, в противном случае пользователь

сертификата может некорректно интерпретировать условия соглашения издающего ЦР.

6.4.2.4 Расширение запрета, какой либо политики

Это поле определяет ограничение, которое указывает any policy, не рассматриваемое как точное совпадение для других политик сертификата сертификатов в пути сертификации, начиная с назначенного ЦР. Назначенный ЦР это или субъектный ЦР сертификата, содержащего это расширение (если значение inhibitAnyPolicy равно 0), или ЦР, который является субъектом последующего сертификата в пути сертификации.

inhibitAnyPolicy EXTENSION ::= {
SYNTAX SkipCerts
IDENTIFIED BY id-ce-inhibitAnyPolicy

Это расширение может быть по выбору издателя сертификата либо критическим, либо некритическим. Рекомендуется, чтобы оно устанавливалось, как критическое, в противном случае пользователь сертификата может некорректно интерпретировать условия соглашения издающего ЦР.

6.5 Расширение базового CRL

6.5.1 Необходимые условия

Следующие условия относятся к CRL:

a) пользователь сертификата должен иметь возможность составить путь CRL, изданных издателем CRL или пунктом распределения CRL (см. 5.6), и должен иметь возможность определить отсутствующий CRL в последовательности. Поэтому необходимо указывать порядковый номер CRL;

b) некоторые пользователи CRL могут реагировать на аннулирование, в зависимости от причины аннулирования. Поэтому существует требование к элементам CRL, чтобы они указывали причину аннулирования;

c) существует требование к органу, чтобы он имел возможность временно приостановить действительность сертификата и в дальнейшем либо аннулировать, либо восстановить его. Возможные причины для таких действий могут быть:

- желание уменьшить обязательства ошибочного аннулирования, когда запрос на аннулирование не аутентифицирован и существует неадекватная информация для определения его действительности;
- другие правовые необходимости, такие как временное приостановление сертификата объекта, ожидающего аудит или расследование;

d) для каждого аннулированного сертификата CRL содержит дату, когда было объявлено аннулирование. Дата аннулирования не достаточна для разрешения некоторых споров, так как в худшем случае подписи выданные в течении периода действительности сертификата должны быть

рассмотрены как недействительные. Однако, для пользователя может быть важным, что подписанный документ будет признан как действительный даже, несмотря на то, что ключ, использованный для подписания документа, был скомпрометирован после того как была поставлена подпись. Для облегчения разрешения этой проблемы элемент CRL может включать в себя вторую дату, указывающую на то, когда стало известно или подозревалось, что закрытый ключ был скомпрометирован;

e) пользователи сертификата должны иметь возможность определить из своих CRL добавочную информацию, включающую область применения сертификатов, охватываемых этим списком (выпиской из извещения об аннулировании);

f) издатели должны иметь возможность динамически изменять разбиение CRL и отсылать сертификаты пользователя в новое место для релевантных CRL, если изменено разделение;

g) дельта CRL могут быть также доступны для обновления базовой CRL, соответствующей данной дельта CRL. Пользователи сертификата должны иметь возможность определить из данного CRL доступны ли данные дельта CRL, где они расположены или когда будет издан следующий дельта CRL.

6.5.2 Поля расширения CRL и элементы CRL

6.5.2.1 Расширение номера CRL

Определены следующие поля расширения:

a) номер CRL;

b) код причины;

c) код содержащий инструкции;

d) область применения CRL;

e) ссылка на состояние;

f) идентификатор группы CRL;

g) упорядоченный список;

h) дельта информация.

Номер CRL, область применения CRL, ссылка на состояние, идентификатор потока CRL, упорядоченный список, дельта информация должны быть использованы только как поле расширения CRL, а другие поля должны быть использованы только как поля расширения элемента CRL.

Расширение номера CRL дает монотонно возрастающее последовательное число для каждого CRL изданного издателем данного CRL посредством директории атрибута органа или, посредством пункта распределения CRL. Оно позволяет пользователю CRL определить, издавались ли раньше эти CRL. Это поле определяется следующим образом:

cRLNumber EXTENSION ::= {
SYNTAX CRLNumber
IDENTIFIED BY id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)

Это расширение всегда имеет некритическое значение.

6.5.2.2 Расширение кода причины

Это поле расширения элемента CRL указывает причины аннулирования сертификата.

Код причины может быть использован приложениями для принятия решения, основываясь на локальной политике, как реагировать на отложенное аннулирование. Это поле определяется следующим образом:

```
reasonCode EXTENSION ::= {
SYNTAX      CRLReason
IDENTIFIED BY   id-ce-reasonCode }
CRLReason ::= ENUMERATED {
unspecified             (0),
keyCompromise           (1),
cACompromise            (2),
affiliationChanged      (3),
superseded              (4),
cessationOfOperation    (5),
certificateHold         (6),
removeFromCRL           (8),
privilegeWithdrawn      (9),
aaCompromise            (10) }
```

Следующие значения кода причины указывают на то, почему сертификат был аннулирован:

- keyCompromise используется для аннулирования сертификата конечного пользователя; оно указывает на то, что известно, или подозревается, что закрытый ключ субъекта или другие аспекты субъекта, утвержденные в сертификате, скомпрометированы;

- cACompromise используется для аннулирования сертификата ЦР; оно указывает на то, что известно, или подозревается, что закрытый ключ субъекта или другие аспекты субъекта, утвержденные в сертификате, скомпрометированы;

- affiliationChanged указывает, что название субъекта или другая информация в сертификате модифицирована, но нет основания для подозрения, что закрытый ключ был скомпрометирован;

- superseded указывает на то, что сертификат подменен, но нет основания для подозрения, что закрытый ключ был скомпрометирован;

- cessationOfOperation указывает на то, что сертификат больше не подлежит применению в целях, для которых он был издан, но нет основания для подозрения, что закрытый ключ был скомпрометирован;

- privilegeWithdrawn указывает на то, что сертификат (открытый ключ или сертификат атрибута) был аннулирован потому, что привилегия, содержащаяся в этом сертификате, отменена;

- aaCompromise указывает на то, что известно или подозревается, что аспекты атрибута органа по присвоению атрибутов, утвержденные в сертификате атрибута, скомпрометированы.

Сертификат может быть помещен на хранение, изданием элемента CRL с кодом причины certificateHold. Извещение о хранении сертификата может включать не обязательный код инструкции хранения для сообщения о добавочной информации пользователю сертификата в соответствии с 6.5.2.3.

Код причины removeFromCRL предназначен только для использования в дельта- CRL и указывает, что существующий элемент CRL должен быть удален из-за окончания срока действия сертификата или из-за выхода из хранения.

6.5.2.3 Расширение кода содержащего инструкции

Это поле расширения элемента CRL предусматривает включение зарегистрированного идентификатора инструкций для указания действия, предпринимаемого при обнаружении сертификата, находящегося на хранении. Оно применимо только в элементах, имеющих код причины certificateHold. Это поле определяется следующим образом:

holdInstructionCode EXTENSION ::= {
SYNTAX HoldInstruction
IDENTIFIED BY id-ce-instructionCode }
HoldInstruction ::= OBJECT IDENTIFIER

Это расширение всегда является некритическим. В данном стандарте не определяются стандартные коды, содержащие инструкции.

Примерами инструкций могут быть сообщения со следующим содержанием «пожалуйста, свяжитесь с ЦР» или «восстановить признак пользователя».

6.5.2.4 Расширение области применения CRL

Область применения CRL указывается в этом расширении. Если представляется поле расширения, то оно должно быть критическим чтобы предотвратить подмену CRL приложением, которое не поддерживает поле расширения.

Это расширение может быть использовано для предоставления сообщения об области применения различных типов CRL, включающих:

- простые CRL, которые предоставляют информацию об аннулировании сертификатов, выданных одним органом;

- косвенные CRL, которые предоставляют информацию об аннулировании сертификатов выданных различными органами;

- дельта CRL, которые обновляют прежде изданную информацию об аннулировании;

- косвенные дельта CRL, которые предоставляют информацию об аннулировании, которая обновляет несколько базовых CRL, выданных одним или несколькими органами.

```
crlScope EXTENSION ::= {
SYNTAX CRLScopeSyntax
IDENTIFIED BY id-ce-cRLScope }
CRLScopeSyntax ::= SEQUENCE SIZE (1..MAX) OF PerAuthorityScope
PerAuthorityScope ::= SEQUENCE {
authorityName [0] GeneralName OPTIONAL,
distributionPoint [1] DistributionPointName OPTIONAL,
onlyContains [2] OnlyCertificateTypes OPTIONAL,
onlySomeReasons [4] ReasonFlags OPTIONAL,
serialNumberRange [5] NumberRange OPTIONAL,
subjectKeyIdRange [6] NumberRange OPTIONAL,
nameSubtrees [7] GeneralNames OPTIONAL,
baseRevocationInfo [9] BaseRevocationInfo OPTIONAL
}
OnlyCertificateTypes ::= BIT STRING {
user (0),
authority (1),
attribute (2) }
NumberRange ::= SEQUENCE {
startingNumber [0] INTEGER OPTIONAL,
endingNumber [1] INTEGER OPTIONAL,
modulus INTEGER OPTIONAL }
BaseRevocationInfo ::= SEQUENCE {
cRLStreamIdentifier [0] CRLStreamIdentifier OPTIONAL,
cRLNumber [1] CRLNumber,
baseThisUpdate [2] GeneralizedTime }
```

Если CRL это косвенный CRL, который предоставляет информацию о статусе аннулирования для нескольких органов, то расширение будет включать несколько конструкций PerAuthorityScope, один или несколько для каждого органа, для которых включена информация об аннулировании. Каждый экземпляр PerAuthorityScope, который связан с органом, отличающимся от того, который издает CRL, должен содержать компоненту authorityName.

Если CRL является dCRL, который предоставляет информацию о статусе дельта аннулирования для множества базовых CRL, выданных одним органом, то расширение включает множество конструкций PerAuthorityScope, одну для каждых базовых CRL, для которых dCRL

обеспечивает обновления. Если представлено значение компоненты authorityName, несмотря на то, что существует множество экземпляров конструкций PerAuthorityScope, оно будет одинаковым для всех случаев.

Если CRL это косвенный dCRL, который предоставляет информацию о статусе дельта аннулирования для множества базовых CRL изданных разными органами, то расширение включает множество конструкций PerAuthorityScope, один для каждых базовых CRL, для которых dCRL обеспечивает обновления. Каждый экземпляр PerAuthorityScope, который связан с другим органом, не выдающим этот dCRL, должен включать компоненту authorityName.

Для каждого экземпляра PerAuthorityScope, представляемого в расширении, поля используются следующим образом. В случае косвенных dCRL и CRL каждый экземпляр PerAuthorityScope может содержать различные комбинации этих полей и различных значений, если:

1) представлено поле authorityName оно указывает на орган, который издал сертификаты, для которых представлена информация об аннулировании. Если это поле не включено, оно по умолчанию указывается в названии издателя CRL;

2) представлено поле distributionPoint, оно используется так, как описано в расширении issuingDistributionPoint;

3) представлено поле onlyContains оно указывает какой тип(ы) сертификатов CRL содержит информацию, касающуюся статуса аннулирования. Если это поле отсутствует, CRL содержит информацию о всех типах сертификатов;

4) представлено поле onlySomeReasons оно используется, как описано в расширении issuingDistributionPoint;

5) представлен элемент serialNumberRange, то он используется следующим образом. Когда представлено значение модуля, серийный номер, перед проверкой для представления в области уменьшается по модулю данного значения. Затем сертификат с уменьшенным серийным номером рассматривается в пределах CRL если он:

- эквивалентен или больше чем startingNumber и меньше чем endingNumber, где они оба представлены;
- эквивалентен или больше чем startingNumber, когда представляется endingNumber;
- меньше чем endingNumber, когда не представлен startingNumber;

6) представлен элемент subjectKeyIdRange, то он интерпретируется одинаково с serialNumberRange за исключением того, что использованный номер это значение в расширении subjectKeyIdentifier сертификата. Кодирование по DER (Distinguished Encoding Riles - поднабор правил кодирования) BIT STRING (пропускающие тег, отрезок и не используемые биты октета) будет рассматриваться как значение кодирования по DER INTEGER. Если в BIT STRING установлен на 0, тогда добавочный нулевой октет должен быть представлен так, чтобы обеспечить положительный результат шифрования INTEGER, например:

03 02 01 f7
преобразуются в
02 02 00 f7

7) представлено поле nameSubtrees, оно использует одинаковые соглашения для форм имен как определено в расширении nameConstraints;

8) представлено поле baseRevocationInfo, оно указывает, что CRL это dCRL, касающийся сертификатов предусмотренных конструкцией PerAuthorityScope.

Использование расширения crlScope для определения CRL как dCRL отличается от использования расширения deltaCRLIdentifier следующим. В случае crlScope информация в компоненте baseRevocationInfo указывает на время, начиная с которого CRL, содержащий это расширение обеспечивает обновления. Хотя это выполняется посредством ссылки к CRL, сосланный CRL может быть полным для области применения или не может быть, в то время как расширение deltaCRLIdentifier ссылается на изданный CRL, который полный для области применения. Однако, обновленная, содержащая расширение crlScope, информация представленная в dCRL это обновления для информации об аннулировании. Этот механизм более гибкий, чем расширение deltaCRLIndicator, поскольку пользователи могут создавать полные CRL локально, и они (CRL) будут сконструированы, основываясь на времени, а не на издании базовых CRL, которые полные для применимой области. В обоих случаях, dCRL всегда предоставляет обновления для статуса аннулирования для сертификатов в данной области, начиная с определенного момента.

В зависимости от политики ответственного органа, перед тем как будет опубликован новый базовый CRL, могут быть опубликованы несколько dCRL. Если dCRL содержат расширение crlScope, указывающее их точку основания, тогда не обязательно указывать cRLNumber в поле BaseRevocationInfo самого последнего изданного базового CRL.

Однако, cRLNumber, указанный в поле BaseRevocationInfo dCRL, будет меньше или равно cRLNumber самого последнего изданного CRL, который полный для области применения.

Заметим, что расширение issuingDistributionPoint и crlScope могут противоречить друг другу и не предназначаться для использования вместе. Однако, если CRL содержит оба эти расширения, тогда сертификат находится в области CRL, только если он соответствует критерию обоих расширений. Если CRL не содержит ни расширения issuingDistributionPoint, ни расширения crlScope, тогда областью является вся область органа и CRL может быть использован для любого сертификата, исходящего из этого органа.

Когда система, использующая сертификат, использует CRL, который содержит расширение crlScope для проверки статуса сертификата, то он должен проверять сертификат и интересующий код причины на то, что попадают ли они в пределы области CRL как это определено расширением crlScope, следующим образом:

O‘ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
DAVLAT QO‘MITASI
AXBOROTNOMASI

a) система, использующая сертификат, должна проверять, что сертификат попадает в область, получаемую в результате пересечения областей serialNumberRange, subjectKeyIdRange и nameSubtrees, и, если представлены distributionPoint и onlyContains, то они должны быть совместимы с соответствующей конструкцией PerAuthorityScope;

b) если CRL содержит компоненту onlySomeReasons в расширении crlScope, тогда система, использующая сертификат, должна проверять, что коды причины, охватываемые этим CRL, соответствуют целям приложения. Если нет, то могут быть востребованы дополнительные CRL. Заметим, что если CRL содержит оба расширения crlScope и issuingDistributionPoint, и оба содержат компоненту onlySomeReasons, тогда только эти коды причины, включенные в компоненту onlySomeReasons обоих расширений, охватываются этим CRL.

6.5.2.5 Расширение ссылки на состояние

Это расширение CRL предназначено для использования в структуре CRL в качестве способа сообщения информации об уведомлении аннулирования пользователям сертификата. Оно должно быть представлено в структуре CRL, которая сама не содержит уведомления об аннулировании сертификата. Структура CRL, содержащая это расширение, не должна быть использована пользователями сертификата или доверяющими сторонами как источник уведомления об аннулировании, она должна использоваться скорее как инструмент для обеспечения гарантии того, что используется соответствующая информация об аннулировании.

Это расширение служит для обеспечения двух первичных функций:

- это расширение представляет механизм для публикации доверенного «списка CRL», включая всю информацию, относящуюся к ней, для того, чтобы помочь доверяющим сторонам определить имеют ли они достаточно информации об аннулировании. Например, орган может периодически издавать новый достоверный CRL, с относительно высокой частотой переиздания. Список должен включать время/дату последнего обновления для каждого ссылаемого CRL. Пользователь сертификата при получении этого списка может быстро определить, не устарели ли кэшированные копии CRL. Этим можно устранить много ненужных исправлений CRL. Кроме этого, используя этот механизм, пользователи сертификата могут узнать о CRL изданных органом, в течении обычного периода обновления, таким образом улучшается своевременность системы CRL;
- это расширение также предоставляет механизм для переадресации доверяющей стороны от предварительных мест (например, на место указывающее на расширение пункта распределения CRL или элемент директории издающего органа) в различные места для получения информации об аннулировании. Это свойство позволяет органам изменять схему разделения CRL, которую они используют без конфликтов с

существующими сертификатами или пользователями сертификата. Для достижения этого орган должен включать каждое новое местонахождение и область CRL, который должен быть найден на этом месте.

Доверяющая сторона должна сравнить интересующий сертификат с формулировками области и следовать по указателю на соответствующее новое местонахождение для получения информации об аннулировании, касающейся этого сертификата.

Это расширение само по себе является растяжимым и в дальнейшем другие non-CRL, базируемые на схемах аннулирования, могут также указываться, используя это расширение.

```
statusReferrals EXTENSION ::= {
SYNTAX    StatusReferrals
IDENTIFIED BY   id-ce-statusReferrals }
StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral
StatusReferral ::= CHOICE {
cRLReferral   [0]   CRLReferral,
otherReferral   [1]   INSTANCE OF OTHER-REFERRAL}
CRLReferral ::= SEQUENCE {
issuer      [0]   GeneralName OPTIONAL,
location    [1]   GeneralName OPTIONAL,
deltaRefInfo     [2]   DeltaRefInfo OPTIONAL,
cRLScope      CRLScopeSyntax,
lastUpdate     [3]   GeneralizedTime OPTIONAL,
lastChangedCRL   [4]   GeneralizedTime OPTIONAL}
DeltaRefInfo ::= SEQUENCE {
deltaLocation     GeneralName,
lastDelta     GeneralizedTime OPTIONAL }
OTHER-REFERRAL ::= TYPE-IDENTIFIER
```

Поле issuer указывает на объект, который подписывает CRL. По умолчанию это имя издателя, охватывающего CRL.

Поле location представляет информацию о местонахождении, на которое нужно перенаправиться и по умолчанию имеет такое же значение как имя issuer.

Поле deltaRefInfo предоставляет необязательное альтернативное местонахождение, из которого будет получен dCRL, и необязательное дату предыдущей дельты.

Поле cRLScope представляет область применения CRL, которую можно найти в сосланном месте.

Поле lastUpdate является значением thisUpdate в самом последнем изданном сосланном CRL.

Поле lastChangedCRL является значением поля thisUpdate в самом последнем изданном CRL, который имеет изменения в содержании.

OTHER-REFERRAL обеспечивает возможность растяжимости других non-CRL, основанных на схемах аннулирования, чтобы в дальнейшем была возможность приспособления.

Это поле всегда установлено как критическое для обеспечения гарантии того, что CRL, содержащий это расширение, преднамеренно установлен системой, использующей сертификат, как источник информации о статусе аннулирования сертификата.

Если это расширение представляется и признается системой, использующей сертификат, то эта система не будет использовать CRL как источник информации о статусе аннулирования. Система должна использовать или информацию, содержащуюся в этом расширении, или другие методы, выходящие за пределы данного стандарта, чтобы определить место соответствующей информации о статусе аннулирования.

Если это расширение представляется и не признано системой, использующей сертификат, то система не будет использовать CRL как источник информации о статусе аннулирования. Система не должна использовать другие методы, выходящие за пределы этого стандарта.

6.5.2.6 Расширение идентификатора группы CRL

Поле идентификатор группы CRL используется для идентификации контекста, в котором номер CRL является уникальным.

```
cRLStreamIdentifier EXTENSION ::= {
SYNTAX    CRLStreamIdentifier
IDENTIFIED BY   id-ce-cRLStreamIdentifier }
CRLStreamIdentifier ::= INTEGER (0..MAX)
```

Это расширение всегда является некритическим.

Каждое значение этого расширения для каждого органа уникально. Идентификатор группы CRL объединен с номером CRL, который служит уникальным идентификатором для каждого CRL, выданного каким либо органом, несмотря на тип CRL.

6.5.2.7 Расширение упорядоченного списка

Это расширение указывает, что последовательность аннулированных сертификатов в поле revokedCertificates CRL расположено либо в возрастающем порядке по серийному номеру сертификата либо по дате аннулирования. Это поле определяется следующим образом:

```
orderedList EXTENSION ::= {
SYNTAX    OrderedListSyntax
IDENTIFIED BY   id-ce-orderedList }
OrderedListSyntax ::= ENUMERATED {
ascSerialNum    (0),
ascRevDate    (1) }
```

Это расширение всегда некритическое. Поля расширения выполняют следующие функции:

- ascSerialNum указывает, что последовательность аннулированных сертификатов в CRL в возрастающем порядке серийного номера сертификата, основанного на значении компоненты serialNumber каждого элемента в списке;

- ascRevDate указывает, что последовательность аннулированных сертификатов в CRL в возрастающем порядке даты аннулирования, основанного на значении компоненты revocationDate каждого элемента в списке.

Если не представлен orderedList, то не существует информации, обеспечивающей упорядочение списка аннулированных сертификатов в CRL.

6.5.2.8 Расширение дельта информации

Это расширение предназначено для использования в CRL, который не является dCRL, и оно используется для указания доверяющим сторонам того dCRL, который также действителен для CRL, содержащего это расширение. Это расширение предоставляет местонахождение, на котором могут быть найдены связанные dCRL и время, в котором будет издан следующий dCRL.

deltaInfo EXTENSION ::= {
SYNTAX DeltaInformation
IDENTIFIED BY id-ce-deltaInfo }
DeltaInformation ::= SEQUENCE {
deltaLocation GeneralName,
nextDelta GeneralizedTime OPTIONAL

Это расширение всегда некритическое.

6.6 Расширения пункта распределения CRL и дельта-CRL

6.6.1 Необходимые условия

Так как списки аннулирования могут быть большими и громоздкими, то необходимо представлять CRL в частичном (неполном) формате. Необходимы различные решения для реализаций двух различных типов, которые обрабатывают CRL.

Первый тип реализации это реализация на индивидуальной рабочей станции, возможно в прикрепленном криптографическом жетоне. Эти реализации, вероятно, будут иметь, ограниченную способность достоверного хранения. Поэтому необходимо проверить весь CRL для того, чтобы определить его действительность, и после этого смотреть является ли сертификат действительным. Если CRL является длинным, то обработка тоже может быть очень длинной. Расчленение CRL необходимо для того, чтобы устранить эту проблему.

Второй тип реализации осуществляется на серверах с высокой производительностью, где осуществляется обработка сообщений больших объемов, например сервер обработки транзакций. В таких средах CRL обрабатываются в фоновом режиме, где после удостоверения CRL содержание CRL сохраняется локально в том представлении, в котором ускоряется их проверка. Это представление хранится в достоверных хранилищах. Серверу такого типа необходима самая последняя обновленная версия CRL для большого количества органов. Так как он уже имеет список прежде аннулированных сертификатов, ему только необходимо извлечь список новых (недавно) аннулированных сертификатов. Этот список, который называется dCRL, будет меньше и требует малых ресурсов для его извлечения и обработки, чем полный список CRL. Поэтому следующие требования относятся и к пунктам распределения CRL и dCRL:

a) для того чтобы контролировать размеры CRL, необходимо иметь возможность назначать подмножества набора всех сертификатов, изданных одним органом для различных CRL. Это может быть выполнено посредством привязывания каждого сертификата с пунктом распределения CRL, который является либо:

- элементом директории, чей атрибут CRL будет содержать элемент аннулирования для этого сертификата, если он аннулирован;
- местонахождение, такое как адрес электронной почты или единый идентификатор ресурса Интернета (Internet URI), из которого можно получить соответствующий CRL;

b) в целях увеличения производительности, при утверждении действительности сложных сертификатов желательно уменьшить количество CRL, которые необходимо проверить. Например, это может быть выполнено при наличии одной подписи издателя CRL и выделением CRL, содержащих аннулирования со стороны многих органов;

c) существует требование для разделения CRL, охватывающих аннулированные сертификаты органа и аннулированные сертификаты конечного объекта. Это способствует обработке путей сертификации и они могут предполагаться очень короткими (обычно пустыми) так же как CRL для аннулированных сертификатов органа. Для этой цели определены атрибуты authorityRevocationList и certificateRevocationList. Однако, чтобы обеспечить безопасность посредством этого разделения необходимо иметь индикатор, который является списком при идентификации CRL. Иначе незаконная подмена одного списка другим может быть не обнаружена;

d) необходимо предостережение для неполных CRL (известных как dCRL), которые содержат только элементы для сертификатов, которые аннулированы с момента издания базового CRL;

e) для dCRL также необходимо указание даты/времени, после которой список будет содержать обновления;

f) существует требование об указании в сертификате мест, где можно найти самый новый CRL (например, дельта CRL).

6.6.2 Расширения пункта распределения CRL и дельта-CRL

6.6.2.1 Расширение пункта распределения CRL

Определены следующие поля расширений:

a) пункты распределения CRL;

b) издающий пункт распределения;

c) издатель сертификата;

d) индикатор дельта CRL;

e) обновление базы;

f) самый новый CRL.

Расширения пункта распределения CRL и самого нового CRL должны быть использованы только как расширения сертификата. Расширения, издающего пункта распределения, индикатора дельта CRL и обновления базы, должны быть использованы только как расширение CRL. Расширение издателя сертификата должно использоваться только как расширение элемента CRL.

Расширение пункта распределения CRL должно использоваться только, как расширение сертификата, и может быть использовано в сертификатах органа, сертификатах открытого ключа конечного объекта и в сертификатах атрибута. Это поле указывает на пункты распределения CRL или точки, к которым должен обращаться пользователь сертификата, чтобы убедиться аннулирован ли сертификат. Пользователь сертификата может достать CRL из подходящего пункта распределения или у него может быть возможность достать текущий полный CRL из элемента директории органа.

Это поле определяется следующим образом:

```
cRLDistributionPoints EXTENSION ::= {
SYNTAX    CRLDistPointsSyntax
IDENTIFIED BY  id-ce-cRLDistributionPoints }
CRLDistPointsSyntax  ::=  SEQUENCE  SIZE  (1..MAX)  OF
DistributionPoint
DistributionPoint ::= SEQUENCE {
distributionPoint  [0]  DistributionPointName OPTIONAL,
reasons     [1]  ReasonFlags OPTIONAL,
cRLIssuer    [2]  GeneralNames OPTIONAL }
DistributionPointName ::= CHOICE {
fullName     [0]  GeneralNames,
nameRelativeToCRLIssuer  [1]  RelativeDistinguishedName }
ReasonFlags ::= BIT STRING {
unused                  (0),
keyCompromise           (1),
cACompromise            (2),
```

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

affiliationChanged (3),
superseded (4),
cessationOfOperation (5),
certificateHold (6),
privilegeWithdrawn (7),
aACompromise (8) }

Компонента distributionPoint указывает на место, из которого может быть взят CRL.

Если компонента отсутствует, пункт распределения имеет по умолчанию имя издателя CRL.

Когда используется альтернатива fullName или когда применяется по умолчанию, имя пункта распределения может иметь множественную форму имен. Одинаковое имя будет представлено, по крайней мере, в одной форме имен, в поле distributionPoint расширения, издающего пункта распределения CRL. Системе, использующей сертификат не обязательно иметь возможность обработки всех форм имен. Она может использовать пункт распределения при условии, если может быть обработана хотя бы одна форма имен. Если не одна из форм имен для пункта распределения не может быть обработана, то система, использующая сертификат, и в этом случае может использовать сертификат, при условии, что реквизиты информации об аннулировании могут быть извлечены из другого источника, например, другого пункта распределения или элемента директории органа.

Компонента nameRelativeToCRLIssuer может быть использована, только если пункт распределения CRL назначается названием директории, т.е. непосредственно зависит от названия директории издателя CRL. В этом случае компонента nameRelativeToCRLIssuer выражает взаимосвязанные различительные имена, касающиеся имени директории издателя CRL.

Компонента reasons указывает причины аннулирования охваченные этим CRL. Если эта компонента отсутствует, соответствующий пункт распределения CRL распределяет CRL, который будет содержать элемент для сертификата, если этот сертификат аннулирован не смотря на причины аннулирования. В противном случае значение reasons указывает, какие причины аннулирования охватываются соответствующим пунктом распределения CRL.

Компонента cRLIssuer указывает орган, который издает и подписывает CRL. Если эта компонента отсутствует, то имя издателя сертификата подразумевает имя издателя CRL.

Это расширение может быть по выбору издателя сертификата либо критическим, либо некритическим. В целях совместимости рекомендуется, чтобы это расширение было отмечено как некритическое.

Если это расширение отмечено как критическое, то система, использующая сертификат, не должна использовать сертификат без извлечения и проверки CRL из одного из назначенных пунктов распределения, покрывающего интересующий код причины. Там, где

используются пункты распространения, для того чтобы распространять информацию о CRL для всех кодов причины аннулирования и всех сертификатов, изданных ЦР, включают расширение cRLDistributionPoints как критическое расширение и ЦР не обязательно также публиковать полный CRL в элементах ЦР.

Если это расширение отмечено как некритическое и система, использующая сертификат не распознает тип поля расширения, то эта система должна использовать сертификат только если:

- она может получить и проверить полный CRL у органа;
- проверка аннулирования не требуется согласно с локальной политикой;
- проверка аннулирования завершена другими способами.

Возможно, что будет иметься несколько CRL для одного сертификата, которые были изданы более чем одним издателем CRL. Координация этих издателей CRL и издающих органов решается аспектами политики органа.

Значение (смысл) каждого кода причины является таким, как это определено в поле кода причины в 6.5.2.2 данного стандарта.

6.6.2.2 Расширение издающего пункта распределения

Это поле расширения CRL указывает на пункт распределения CRL для отдельного CRL и указывает на то, что CRL ограничен для аннулирования только сертификатов конечных объектов, только сертификатов органов или только для ограниченного набора причин. CRL подписывается только ключом издателя CRL – сами пункты распределения не имеют своих пар ключей. Однако, для CRL, которые были распространены через директорию, CRL хранится в элементе пункта распределения CRL, который в свою очередь может и не быть элементом директории издателя CRL.

Это поле определяется следующим образом:

issuingDistributionPoint EXTENSION ::= {
SYNTAX IssuingDistPointSyntax
IDENTIFIED BY id-ce-issuingDistributionPoint }
IssuingDistPointSyntax ::= SEQUENCE {
distributionPoint [0] DistributionPointName OPTIONAL,
onlyContainsUserCerts [1] BOOLEAN DEFAULT FALSE,
onlyContainsAuthorityCerts [2] BOOLEAN DEFAULT FALSE,
onlySomeReasons [3] ReasonFlags OPTIONAL,
indirectCRL [4] BOOLEAN DEFAULT FALSE,
onlyContainsAttributeCerts [5] BOOLEAN DEFAULT FALSE }

Компонента distributionPoint содержит название пункта распределения в одной или более чем в одной форме имен. Если это поле отсутствует, CRL будет содержать элементы для всех аннулированных сертификатов, изданных издателем CRL. После появления сертификата в CRL, он, может

быть, вычеркнут из последующего CRL после истечения срока сертификата.

Если onlyContainsUserCerts имеет значение TRUE, то CRL содержит только аннулирования для сертификатов конечного объекта. Если onlyContainsAuthorityCerts имеет значение TRUE, то CRL содержит только аннулирования для сертификатов органа.

Если представлено onlySomeReasons CRL содержит только установленную причину или причины, иначе CRL содержит аннулирования для всех причин.

Если indirectCRL имеет значение TRUE, то CRL может содержать извещения об аннулировании от другого издателя CRL.

Если onlyContainsAttributeCerts имеет значение TRUE, то CRL содержит аннулирования для сертификатов атрибута.

Следующие правила касаются использования атрибутов для CRL, распределенных через директорию. CRL, который имеет набор onlyContainsAuthorityCerts, будет распределен посредством атрибута authorityRevocationList взаимодействующего пункта распределения, или если нет установленного пункта распределения посредством атрибута authorityRevocationList элемента издателя CRL. Иначе CRL будет распределен через атрибут certificateRevocationList взаимодействующего пункта распределения или, если пункт распределения не установлен, через атрибут certificateRevocationList элемента издателя CRL.

Это расширение всегда критическое. Пользователь сертификата, который не может понять это расширение не может предположить, что сертификат содержит полный список аннулированных сертификатов идентифицированных органом. CRL, не содержащие критическое расширение содержат все текущие элементы CRL для издающего органа, включая элементы для всех сертификатов пользователя и органа.

6.6.2.3 Расширение издателя сертификата

Это расширение элемента CRL указывает издателя сертификата, связанного с элементом в косвенном CRL, т.е. CRL который имеет индикатор indirectCRL, установленный в его расширении издающего пункта распределения. Если это расширение не представлено в первом элементе косвенного CRL, то издателем сертификата по умолчанию становится издатель CRL. Если это расширение не представлено, то относительно последующего элемента в косвенном CRL издатель сертификата для элемента такой же, как для предшествующего элемента.

Это поле определяется следующим образом:

certificateIssuer EXTENSION ::= {
SYNTAX GeneralNames
IDENTIFIED BY id-ce-certificateIssuer }

Это поле всегда критическое. Если система игнорирует это расширение, она не может правильно определить элементы CRL в сертификатах.

6.6.2.4 Расширения указателя дельта CRL

Поле индикатора дельта CRL устанавливает CRL, являющийся dCRL, который обеспечивает обновление для сосланных базовых CRL. Сосланный базовый CRL - это CRL, который точно был издан как CRL, который полный для данной области. CRL, содержащей в себе расширение указателя dCRL, содержит обновления к статусу аннулирования сертификата для той же области.

Это поле определяется следующим образом:

deltaCRLIndicator EXTENSION ::= {
SYNTAX BaseCRLNumber
IDENTIFIED BY id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber

Значение типа BaseCRLNumber устанавливает номер CRL базового CRL, который был использован как базовый при генерации данного dCRL. Сосланным CRL будет CRL, который является полным для области применения.

Это расширение всегда критическое. Пользователь сертификата, который не понимает использование dCRL, не должен использовать CRL содержащий это расширение, в виду того, что CRL может быть не таким полным, как ожидает пользователь.

6.6.2.5 Расширение базового обновления

Поле базового обновления существует для использования в dCRL и для установления даты/времени, после которой эта дельта обеспечивает обновление для статуса аннулирования. Это расширение должно быть использовано только в dCRL, который содержит расширение deltaCRLIndicator. dCRL, который вместо этого расширения содержит расширение crlScope, не требует этого расширения в качестве поля baseThisUpdate расширения crlScope, которое может быть использовано в одинаковых целях:

baseUpdateTime EXTENSION ::= {
SYNTAX GeneralizedTime
IDENTIFIED BY id-ce-baseUpdateTime }

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA DAVLAT NAZORATI
BOSHQARMASI

Это расширение всегда некритическое.

6.6.2.6 Расширение самого нового CRL

Это расширение используется только, как расширение сертификата и может быть использовано в сертификатах, изданных органу, также как в сертификатах, изданных пользователю. Это поле указывает CRL, на который пользователь сертификата должен ссылаться для получения последней информации об аннулировании (например, dCRL).

Это поле определяется следующим образом:

freshestCRL EXTENSION ::= {
SYNTAX CRLDistPointsSyntax
IDENTIFIED BY id-ce-freshestCRL }

Это поле по выбору издателя сертификата может быть либо критическим, либо некритическим. Если это расширение установлено как критическое, система, использующая сертификат, будет использовать сертификат без начального извлечения и проверки последнего CRL. Если расширение установлено как некритическое, система, использующая сертификат, может использовать локальные устройства для определения необходимости проверки последнего CRL.

7 Дельта CRL, имеющий связь с базой

dCRL содержит в себе или расширение deltaCRLIndicator или crlScope для указания базовой информации об аннулировании, т.е. обновляемый этим dCRL.

Если в dCRL представлено расширение deltaCRLIndicator, обновляемая базовая информация об аннулировании является базовым CRL, сосланным в этом расширении.

Базовый CRL, сосланный посредством расширения deltaCRLIndicator, будет являться CRL, который издан как полный для этой области (т.е. это не dCRL изданный самому себе).

Если представлено расширение crlScope и оно содержит компоненту baseRevocationInfo для ссылки на базовую информацию об аннулировании, которая обновляется, то это ссылка на точный момент времени, начиная с которого dCRL обеспечивает обновление. Компонента baseRevocationInfo, ссылается на CRL, который может или не может быть издан как тот, который является полным для этой области (т.е. сосланный CRL может быть издан только как dCRL). Однако, dCRL, содержащий компоненту baseRevocationInfo, обновляет эту информацию об аннулировании, которая полная для области сосланного CRL к моменту времени, к которому был издан сосланный CRL. Пользователь сертификата может применить dCRL к CRL, который полный для данной области и который был издан, как только или после ссылки на CRL в dCRL, содержащем компоненту baseRevocationInfo.

Из-за возможности появления конфликтной информации CRL не содержит одновременно и расширение deltaCRLIndicator и расширение crlScope с компонентой baseRevocationInfo. CRL может содержать оба расширения и deltaCRLIndicator и расширения crlScope только если компонента baseRevocationInfo не представлена в расширении crlScope.

dCRL также может быть косвенным CRL, в котором он может содержать обновленную информацию об аннулировании, связанную с базовыми CRL, изданными одним или более чем одним органом. Расширение crlScope должно быть использовано как средство установления CRL как косвенного CRL. Расширение crlScope содержит один экземпляр компоненты PerAuthorityScope для каждого базового CRL, для которого dCRL предоставляет обновленную информацию.

dCRL для сосланной базовой информации об аннулировании точно отражает текущий статус аннулирования.

Извещение об аннулировании сертификата с причиной аннулирования certificateHold, может появиться или в dCRL или в CRL, который полный для данной области. Этот код причины предназначен для указания временного аннулирования сертификата, ожидающего дальнейшее решение о перманентном аннулировании сертификата или восстановления его как неаннулированного.

Извещение об аннулировании сертификата может сначала появиться в dCRL и возможно, что период действительности сертификата может истечь до того как будет издан следующий CRL, который полный для применимой области. В этом случае, это извещение об аннулировании будет включаться во все дальнейшие dCRL до тех пор пока это извещение об аннулировании не будет включено, по крайней мере, в один изданный CRL, который полный для данной области.

CRL, который в текущее время полон для данной области, может быть создан локально одним из следующих путей:

- извлечением текущего dCRL для данной области и объединением его с изданным CRL, который полный для данной области, и который имеет cRLNumber больше чем или равный cRLNumber базового CRL, сосланного в dCRL;
- извлечением текущего dCRL для данной области и объединением его с локально созданным CRL, который полный для данной области, и который был создан с dCRL, который имеет cRLNumber больше чем или равный cRLNumber базового CRL, сосланного в dCRL.

O‘ZBEKISTON AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA
VA SIFAT NAZORATI

8 Процедура обработки пути сертификации

8.1 Входные данные обработки пути сертификации

Обработка пути сертификации выполняется в системе, которая должна использовать открытый ключ удаленного конечного объекта, например система, которая проверяет цифровую подпись сгенерированную удаленным объектом. Политики сертификата, базовые ограничения, ограничения имени и расширения ограничений политики предназначены для способствования автоматизированию, автономной реализации логики обработки пути сертификации.

Ниже описана схема процедуры проверки действительности путей сертификации.

Реализация будет функционально эквивалентна внешней характеристике, вытекающей из этой процедуры.

Входными данными процедуры обработки пути сертификации являются:

a) набор сертификатов, входящих в путь сертификации. Каждый сертификат в пути сертификации является уникальным. Путь сертификации, содержащий две или более одинаковых сертификатов, не является действительным путем сертификации;

b) достоверное значение открытого ключа или идентификатора ключа (если ключ хранится внутри модуля обработки пути сертификации), предназначенное для подтверждения первого сертификата в пути сертификации;

c) значение initial-policy-set, включающее в себя один или более идентификаторов политик, указывающих, что любая из этих политик должна быть допустимой для пользователя сертификата в целях обработки пути сертификации; также этот входной параметр может принимать специальное значение any-policy;

d) значение индикатора initial-explicit-policy, которое указывает, необходимо ли допустимому идентификатору политики точно показываться в поле расширения политик сертификата во всех сертификатах в пути сертификации;

e) значение индикатора initial-policy-mapping-inhibit, которое показывает, является ли запрещенным преобразование политик в пути сертификации;

f) значение индикатора initial-inhibit-policy, которое показывает, что если специальное значение anyPolicy указано в расширении политик сертификата, оно принимается равносильным для какого либо специфического значения политики сертификата в ограниченном наборе;

g) текущая дата/время.

Следует заметить, что из-за того, что все, перечисленные выше, входные параметры являются индивидуальными входными параметрами для обработки достоверности пути сертификации, пользователь сертификата

может ограничить доверие (которое было возложено) к любому данному доверенному открытому ключу для данного набора политик сертификата. Это может быть достигнуто посредством гарантирования того, что данный открытый ключ является входным параметром для обработки только тогда, когда входной параметр initial-policy-set включает политику, на основе которой пользователь сертификата доверяет этому открытому ключу.

8.2 Выходные данные обработки пути сертификации

Выходными параметрами обработки являются:

a) успешное или неудачное утверждение пути сертификации;

b) если утверждение завершено не успешно, код диагностики указывает на причину неудачной обработки;

c) набор политик, ограничивающих привилегии органов и их связанных спецификаторов, в соответствии с которыми утверждается путь сертификации, или специальное значение any-policy;

d) набор политик, ограничивающих привилегии пользователей образованных из пересечения authorities-constrained-policy-set и initial-policy-set;

e) explicit-policy-indicator, указывающий на пользователя сертификата или орган в пути и требующий, что допустимая политика в каждом сертификате в пути будет идентифицирована;

f) детали какого-либо преобразования политики, которые происходят в процессе обработки пути сертификации.

8.3 Переменные величины обработки пути сертификации

Процедура использует следующий набор переменных состояний:

a) authorities-constrained-policy-set: Таблица идентификаторов и спецификаторов политики из пути сертификации (ряды представляют политики, их спецификаторы и историю преобразования, а столбцы, представляют сертификаты в пути сертификации);

b) permitted-subtrees: Набор спецификаций поддерева определяющих поддеревья, в которых все названия субъектов в последующих сертификатах в пути сертификации должны быть опущены или могут принимать специальное значение unbounded;

c) excluded-subtrees: Набор (возможно пустой) спецификаций поддерева, определяющих поддеревья, в которых нет названия субъекта, который может быть опущен в последующем сертификате в пути сертификации;

d) explicit-policy-indicator: Указывает на то, что должна ли быть допустимая политика идентифицированной в каждом сертификате в пути;

e) path depth: Целое число эквивалентное более чем одному экземпляру сертификата в пути сертификации, для которого обработка была завершена;

STANDARDLASHTIRISH, METROLOGIYA VA DAVLAT NAZORATI

f) policy-mapping-inhibit-indicator: Указывает на то, что преобразование политики запрещено;

g) inhibit-any-policy-indicator: Указывает на то, что соответствует ли рассматриваемое значение anyPolicy любой особой политике сертификата;

h) pending-constraints: Элементы точной политики ограничения запрет преобразования политики и/или запрета какой-либо политики, которые были оговорены, но ещё имеют силу.

8.4 Этап инициализации

Процедура обработки пути сертификации включает в себя этап инициализации, следующий за серией шагов обработки сертификата. Этап инициализации включает:

a) запись any-policy в нулевой или первый столбец нулевого ряда таблицы authorities-constrained-policyset;

b) инициализацию переменной permitted-subtrees на unbounded;

c) инициализацию переменной excluded-subtrees на пустое множество;

d) инициализацию переменной explicit-policy-indicator на значение initial-explicit-policy;

e) инициализацию path-depth на path-depth;

f) инициализацию policy-mapping-inhibit-indicator на значение initial-policy-mapping-inhibit;

g) инициализацию inhibit-any-policy-indicator на значение initial-inhibit-policy;

h) инициализацию трех индикаторов pending-constraints на исходное положение.

8.5 Обработка сертификата

8.5.1 Базовая проверка сертификата

Каждый сертификат обрабатывается по очереди, начиная с сертификата, подписанного, используя исходный доверенный открытый ключ. Последний сертификат рассматривается как конечный сертификат; любые другие сертификаты рассматриваются как промежуточные.

В сертификате применяются следующие виды проверки:

a) проверка того, что подпись подтверждает, что дата действительна, что цепь названий субъекта сертификата и издателя сертификата правильны и что сертификат неаннулирован;

b) если поле расширения базовых ограничений представлено в промежуточном сертификате, проверяется, что представлена компонента cA и установлена на значение TRUE. Если представлена компонента pathLenConstraint, то проверяется, что текущий путь сертификации не нарушил это ограничение (игнорируя промежуточный сертификат изданный самому себе);

c) если расширение политики сертификата не представлено, тогда таблица authorities-constrained-policy-set устанавливается на нуль удалением всех рядов из таблицы;

d) если расширение политики сертификата представлено, тогда к каждой политике в расширении, отличающемся от anyPolicy, накладываются спецификаторы, связанные с P для каждого ряда в таблице authorities-constrainedpolicy-set, чей элемент столбца [path-depth] содержит значение P. Если в таблице authorities-constrained-policy-set нет ряда, содержащего значение P в его элементе столбца [path-depth], но значение в authoritiesconstrained-policy-set[0, path-depth] установлено на any-policy, тогда добавляется новый ряд в таблицу дублицированием нулевых рядов и записью идентификатора политики P вместе с его спецификаторами в элементе столбца [path-depth] нового ряда;

e) если расширение политики сертификата представлено и не включает значение anyPolicy или если установлен inhibitany-policy-indicator, тогда удаляется любой ряд, для которого входной столбец [path-depth] содержит значение any-policy вместе с любым рядом, для которого элемент столбца [path-depth] не содержит одно из значений в расширении политики сертификата;

f) если расширение политики сертификата представлено и включает значение anyPolicy и не установлен inhibit-any-policy-indicator, тогда оно стыкует спецификатор политики, связанный с anyPolicy в каждом ряду в таблице authorities-constrained-policy-set, во входном столбце [path-depth] которого содержится значение any-policy или содержится значение которое не появляется в расширении политики сертификата;

g) если сертификат не промежуточный сертификат, выданный самому себе, проверяется то, что название субъекта в пределах пространства названий, установленного значением permitted-subtrees, и не в пределах пространства названий установленного значением excluded-subtrees.

8.5.2 Обработка промежуточного сертификата

Следующие ограничения регистрируют действия для промежуточного сертификата, выполняемые, чтобы правильно установить переменные состояния для обработки последующего сертификата, если:

a) в сертификате представлено расширение nameConstraints с компонентой permittedSubtrees, то переменная состояния permitted-subtrees устанавливается на пересечении с его предыдущим значением и значением, установленным в расширении сертификата;

b) в сертификате представлено расширение nameConstraints с компонентой excludedSubtrees, то переменная состояния excluded-subtrees устанавливается вместе с его предыдущим значением и значением, установленным в расширении сертификата;

c) policy-mapping-inhibit-indicator установлен, то обработка расширения преобразования политики для каждого преобразования, установленного в расширении, выполняется определением всех рядов в таблице authorities-

O‘ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
AGENTLIGI

constrained-policy-set, элементы столбца [path-depth] которого эквивалентны значению доменной политики издателя в расширении;

d) policy-mapping-inhibit-indicator не установлен, то процесс преобразования расширения политик для каждого преобразования, установленного в расширении, выполняется определением всех рядов в таблице authorities-constrained-policy-set, чей элемент столбца [path-depth], эквивалентный значению доменной политики издателя в расширении. Элемент столбца [path-depth] записывает значение доменной политики субъекта из расширения в элемент столбца [path-depth+1] того же ряда. Если расширение отображает доменную политику издателя на более чем одну политику домена субъекта, тогда поврежденный ряд копируется и в каждый ряд добавляется новый элемент. Если значение в authorities-constrained-policy-set[0, path-depth] установлено на any-policy, тогда каждый идентификатор доменной политики издателя записывается из расширения преобразования политики в столбец [path-depth] и записывает значение доменной политики субъекта из расширения в элемент столбца [path-depth+1] того же ряда.

e) policy-mapping-inhibit-pending установлен и сертификат не выдан самому себе уменьшается значение соответствующего skip-certificates и, если это значение обнуляется, устанавливается индикатор policy-mapping-inhibit-indicator.

f) в сертификате представлено ограничение inhibitPolicyMapping, то выполняется следующее. Для значения SkipCerts равного 0, устанавливается policy-mapping-inhibit-indicator. Для любых других значений устанавливается индикатор policy-mapping-inhibit-pending и устанавливается соответствующее значение, меньшее значения SkipCerts и предыдущего значения skip-certificates (если индикатор policy-mapping-inhibit-pending уже был установлен);

g) для любого ряда, измененного одним из двух шагов, либо c) или d) приведенных выше (и каждый ряд в случае, когда в сертификате представлено расширения преобразования), идентификатор политики записывается из столбца [path-depth] в столбец [pathdepth+1] этого ряда;

h) inhibit-any-policy-indicator не установлен, индикатор inhibit-any-policy-pending установлен и сертификат не выдан самому себе, уменьшается значение соответствующего skip-certificates. Если значение обнуляется, то устанавливается индикатор policy-mapping-inhibit-indicator;

i) в сертификате представлено ограничение inhibitAnyPolicy, то для значения SkipCerts равного 0 устанавливается inhibit-any-policy-indicator. Для любых других значений SkipCerts устанавливается индикатор policy-mapping-inhibit-pending и устанавливается соответствующее значение skip-certificates меньшее значения SkipCerts и предыдущего значения skip-certificates (если индикатор inhibit-any-policy-pending уже был установлен);

j) увеличивается [path-depth].

8.5.3 Обработка индикатора точной политики

Для всех сертификатов выполняются следующие действия, если:

- установлен индикатор explicit-policy-indicator и сертификат не является промежуточным сертификатом, выданным самому себе, то увеличивается соответствующее значение skip-certificates и если это значение обнуляется, то устанавливается индикатор explicit-policy-indicator;

- в сертификате представлено ограничение requireExplicitPolicy то выполняется следующее. Для значения SkipCerts равного 0 устанавливается индикатор explicit-policy-indicator. Для любых других значений SkipCerts устанавливается индикатор explicit-policy-pending и устанавливается соответствующее значение skip-certificates, меньшее значения SkipCerts и предыдущее значение skip-certificates (если индикатор explicit-policy-pending уже был установлен);

- представлена компонента requireExplicitPolicy и путь сертификации включает сертификат, изданный назначенным ЦР, то необходимо, чтобы все сертификаты в пути содержали в расширении политик сертификата идентификатор допустимой политики.

Идентификатор допустимой политики это: идентификатор политики сертификата, требуемый пользователем пути сертификации, идентификатор политики, который объявлен эквивалентным благодаря политике преобразования или any-policy. Назначенный сертифицирующий орган это либо издатель сертификата ЦР, содержащий это расширение (если значение requireExplicitPolicy равно 0), либо ЦР, который является субъектом последующего сертификата в пути сертификации.

8.5.4 Завершающая обработка

Для конечного сертификата выполняются следующие действия, если:

- установлен explicit-policy-indicator проверяется, что таблица authorities-constrained-policy-set не пуста.

- какие-либо из вышеупомянутых видов проверки были не удачными, тогда процедура завершается, возвращая сигнал неисправности соответствующего кода причины explicit-policy-indicator и нулевые значения в user-constrained-policy-set и таблицу authorities-constrained-policy-set.

- не одна из вышеуказанных проверок не была неудачной в конечном сертификате, тогда user-constrained-policy-set будет рассчитан образованием пересечения authorities-constrained-policy-set и initial-policy-set.

- authorities-constrained-policy-set [0, path-depth] является any-policy, тогда authorities-constrained-policy-set является any-policy. Иначе, authorities-constrained-policy-set является для каждого ряда в таблице значением самой левой ячейки, которая не содержит идентификатор any-policy. Затем процедура завершается возвращением сигнала об успешном завершении вместе с таблицей explicit-policy-indicator, authorities-constrained-policy-set и user-constrained-policy-set.

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA BAHOLASH NAZORATI
BOSHQARMASI

- пересечение authority constrained и user constrained set равно нулю, путь действителен согласно ограничивающей политике органа, но не совсем подходит для пользователя.

9 Схема директории PKI

9.1 Классы объекта директории PKI и формы названия

Этот подраздел определяет элементы схемы директории, использованные для представления информации о PKI в директории. Этот подраздел включает определение соответствующих классов объекта, атрибуты и правила согласования значения атрибута.

Классы объекта директории PKI и формы названия используются для представления объектов PKI в директории:

1) Класс объекта-пользователя PKI используется для определения элементов для объектов, которые могут быть субъектом сертификата открытого ключа.

pkiUser OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {userCertificate}
ID id-oc-pkiUser }

2) Класс объекта ЦР PKI используется для определения элементов для объектов, которые действуют как ЦР.

pkiCA OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {cACertificate |
certificateRevocationList |
authorityRevocationList |
crossCertificatePair }
ID id-oc-pkiCA}

3) Класс объекта пункта распределения CRL используется для определения элементов для объектов, которые действуют как пункт распределения CRL.

cRLDistributionPoint OBJECT-CLASS ::= {
SUBCLASS OF { top }
KIND structural
MUST CONTAIN { commonName }
MAY CONTAIN { certificateRevocationList |
authorityRevocationList |
deltaRevocationList }
ID id-oc-cRLDistributionPoint

Форма названия пункта распределения CRL показывает, как могут быть названы элементы класса объекта cRLDistributionPoint.

cRLDistPtNameForm NAME-FORM ::= {
NAMES cRLDistributionPoint
WITH ATTRIBUTES { commonName}
ID id-nf-cRLDistPtNameForm

4) Класс объекта дельта CRL используется в определении элементов для объектов, которые владеют списками дельта аннулирования (например, ЦР, Орган по присвоению атрибутов и т.д.).

deltaCRL OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {deltaRevocationList}
ID id-oc-deltaCRL }

5) Класс объекта CPS политики сертификата используется в определении элементов для объектов, которые содержат политику сертификата и/или информацию о применении сертификата.

cpCps OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {certificatePolicy |
certificationPracticeStmt}
ID id-oc-cpCps

6) Класс объекта пути сертификации PKI используется в определении элементов для объектов, которые содержат пути PKI, для связи с элементами структурного класса объекта pkiCA.

pkiCertPath OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN { pkiPath }
ID id-oc-pkiCertPath }

9.2 Атрибуты директории PKI

9.2.1 Пользователь сертификата атрибута

Этот пункт содержит определение атрибутов директории, используемых для хранения информации об элементах PKI в этой директории.

Пользователь может получить один или более одного сертификата атрибута от одного или более одного ЦР. Пользователь должен получить у одного или более одного ЦР тип атрибута userCertificate, содержащий сертификаты открытого ключа.

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
VA DAVLAT NAZORATI
BOSHQARMASI

userCertificate ATTRIBUTE ::= {
WITH SYNTAX Certificate
EQUALITY MATCHING RULE certificateExactMatch
ID id-at-userCertificate}

9.2.2 Сертификат атрибута сертифицирующего органа

Атрибут cACertificate элемента директории ЦР используется для хранения сертификатов, изданных самому себе, и сертификатов, изданных этому ЦР другим ЦР, находящимся в той же области, что и данный ЦР. В случае сертификата версии 3 этот сертификат включает в себя расширение basicConstraints со значением cA, установленным на true. Определение этой области исключительно вопрос локальной политики.

cACertificate ATTRIBUTE ::= {
WITH SYNTAX Certificate
EQUALITY MATCHING RULE certificateExactMatch
ID id-at-cAcertificate }

9.2.3 Атрибут парного кросс сертификата

Элементы issuedToThisCA атрибута crossCertificatePair элемента директории ЦР должны быть использованы для хранения всех видов сертификатов, за исключением сертификата выданного самому себе, для данного ЦР. Элементы issuedByThisCA атрибута crossCertificatePair элементов директории ЦР могут содержать подмножество сертификатов, изданных данным ЦР другим ЦР, но это не обязательно. Если ЦР издает сертификат другим ЦР и субъектный ЦР не подчиняется издающему ЦР в иерархии, тогда издающий ЦР поместит этот сертификат в элемент issuedByThisCA атрибута crossCertificatePair его собственных элементов директории. Когда оба элемента issuedToThisCA и issuedByThisCA представлены в одном значении атрибута, название издателя в одном сертификате соответствует названию субъекта в другом и наоборот, открытый ключ в одном сертификате должен допускать проверку цифровой подписи на других сертификатах и наоборот.

Когда представлен элемент issuedByThisCA, значение элемента issuedToThisCA и элемента issuedByThisCA не должны храниться в одинаковых значениях атрибута; другими словами, они могут храниться либо в отдельном значении атрибута, либо в двух значениях атрибута.

В случае сертификата ITU-T X.509 [2] версии v.3, они должны содержать расширение basicConstraints со значением cA, установленным на TRUE.

crossCertificatePair ATTRIBUTE ::= {
WITH SYNTAX CertificatePair
EQUALITY MATCHING RULE certificatePairExactMatch
ID id-at-crossCertificatePair }

```
CertificatePair      ::=  SEQUENCE {
issuedToThisCA       [0]  Certificate OPTIONAL,
issuedByThisCA       [1]  Certificate OPTIONAL
}
(WITH COMPONENTS {..., issuedToThisCA PRESENT} |
WITH COMPONENTS {..., issuedByThisCA PRESENT})
```

9.2.4 Атрибут списка аннулированных сертификатов

Следующий атрибут содержит список аннулированных сертификатов:

```
certificateRevocationList       ATTRIBUTE ::= {
WITH SYNTAX       CertificateList
EQUALITY MATCHING RULE    certificateListExactMatch
ID          id-at-certificateRevocationList
```

9.2.5 Атрибут списка аннулированных сертификатов органа

Следующий атрибут содержит список аннулированных сертификатов органа:

```
authorityRevocationList         ATTRIBUTE ::= {
WITH SYNTAX         CertificateList
EQUALITY MATCHING RULE      certificateListExactMatch
ID          id-at-authorityRevocationList
```

9.2.6 Атрибут списка дельта аннулирования

Следующий тип атрибута определен для содержания dCRL в элементе директории:

```
deltaRevocationList         ATTRIBUTE ::= {
WITH SYNTAX       CertificateList
EQUALITY MATCHING RULE    certificateListExactMatch
ID          id-at-deltaRevocationList
```

9.2.7 Атрибут поддерживаемых алгоритмов

Атрибут директории определён для поддержки выбора алгоритма для использования при связи с удаленным конечным объектом, используя сертификат, как определено в этом стандарте. Следующий код, написанный на языке ASN.1 определяет этот (многозначный) атрибут:

```
supportedAlgorithms         ATTRIBUTE ::= {
WITH SYNTAX         SupportedAlgorithm
EQUALITY MATCHING RULE      algorithmIdentifierMatch
ID          id-at-supportedAlgorithms }
SupportedAlgorithm ::= SEQUENCE {
```

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

algorithmIdentifier AlgorithmIdentifier,
intendedUsage [0] KeyUsage OPTIONAL,
intendedCertificatePolicies [1] CertificatePoliciesSyntax OPTIONAL }

Каждое значение многозначного атрибута должно иметь отличие от значения algorithmIdentifier. Значение компоненты intendedUsage обеспечивает индикацию предназначения алгоритма (см. 5.2.2.3). Значение компоненты intendedCertificatePolicies указывает политику сертификата, с которым может быть использован идентифицированный алгоритм.

9.2.8 Атрибут сообщения о применении сертификата

Атрибут certificationPracticeStmt используется для хранения сообщения о применении сертификата органа.

certificationPracticeStmt ATTRIBUTE ::= {
WITH SYNTAX InfoSyntax
ID id-at-certificationPracticeStmt }
InfoSyntax ::= CHOICE {
content DirectoryString {ub-content},
pointer SEQUENCE {
name GeneralNames,
hash HASH { HashedPolicyInfo } OPTIONAL } }
POLICY ::= TYPE-IDENTIFIER
HashedPolicyInfo ::= POLICY.&Type({Policies})
Policies POLICY ::= {...} – Определяется разработчиками –

Если представлен content, то включается полное содержание сообщения о применении сертификата органа.

Если представлен pointer, компонента name ссылается на одно или более мест, где могут быть расположены копии сообщения о применении сертификата органа. Если представлена компонента hash, то она содержит HASH-содержание сообщения о применении сертификата, которое должно быть найдено по ссылке. Этот хэш может быть использован для полной проверки сосланного документа.

9.2.9 Атрибут политики сертификата

Атрибут certificatePolicy используется для хранения информации о политике сертификата.

certificatePolicy ATTRIBUTE ::= {
WITH SYNTAX PolicySyntax
ID id-at-certificatePolicy }
PolicySyntax ::= SEQUENCE {
policyIdentifier PolicyID,

policySyntax InfoSyntax
}
PolicyID ::= CertPolicyId

Компонента policyIdentifier включает идентификатор объекта, зарегистрированный на основе отдельной политики сертификата.

Если представлена компонента content, то включается полное содержание политики сертификата.

Если представлена компонента pointer, то компонента name ссылается на одно или более мест, где могут быть расположены копии политики сертификата. Если представлена компонента hash, она содержит HASH-содержание политики сертификата, которая должна быть найдена по ссылке. Этот хэш может быть использован для полной проверки сосланного документа.

9.2.10 Атрибут пути PKI

Атрибут пути PKI используется для хранения путей сертификации, каждый из которых состоит из кросс-сертификатов.

pkiPath ATTRIBUTE ::= {
WITH SYNTAX PkiPath
ID id-at-pkiPath }
PkiPath ::= SEQUENCE OF CrossCertificates

Этот атрибут может быть сохранен в элементе директории ЦР и включает некоторые пути сертификации от данного ЦР к другому ЦР. Если этот атрибут используется, он дает возможность более эффективно восстанавливать кросс-сертификаты, форма, которых часто используется путями сертификации. По существу, нет точно определенных требований для использования этого атрибута и набора значений, которые хранятся в атрибуте и вероятно не представляют полный набор последующих путей сертификации для какого-либо данного ЦР.

9.3 Правила соответствия директории PKI

9.3.1 Точное соответствие сертификата

Правило точного соответствия сертификата сравнивает на равенство представленного значения со значением атрибута типа Certificate. Оно уникально выбирает один сертификат.

certificateExactMatch MATCHING-RULE ::= {
SYNTAX CertificateExactAssertion
ID id-mr-certificateExactMatch }
CertificateExactAssertion ::= SEQUENCE {
serialNumber CertificateSerialNumber,
issuer Name }

SERTIFIKAT ASENTLIGI
TA'LIM SIFATINI
NAZORAT QILISH
VA DAVLAT NAZORATI
BO'LIMIDAN

Это правило согласования возвращает true, если компоненты в значении атрибута соответствуют представленному значению.

9.3.2 Соответствие сертификата

Правило соответствия сертификата сравнивает представленное значение со значением атрибута типа Certificate. Оно выбирает один или более сертификатов на основе различных характеристик.

```
certificateMatch MATCHING-RULE ::= {
SYNTAX  CertificateAssertion
ID    id-mr-certificateMatch }
CertificateAssertion ::= SEQUENCE {
serialNumber      [0]  CertificateSerialNumber OPTIONAL,
issuer       [1]  Name OPTIONAL,
subjectKeyIdentifier   [2]  SubjectKeyIdentifier OPTIONAL,
authorityKeyIdentifier    [3]  AuthorityKeyIdentifier OPTIONAL,
certificateValid      [4]  Time OPTIONAL,
privateKeyValid    [5]  GeneralizedTime OPTIONAL,
subjectPublicKeyAlgID  [6]  OBJECT IDENTIFIER OPTIONAL,
keyUsage      [7]  KeyUsage OPTIONAL,
subjectAltName      [8]  AltNameType OPTIONAL,
policy       [9]  CertPolicySet OPTIONAL,
pathToName      [10]  Name OPTIONAL,
subject      [11]  Name OPTIONAL,
nameConstraints    [12]  NameConstraintsSyntax OPTIONAL
}
AltNameType ::= CHOICE {
builtinNameForm ENUMERATED {
rfc822Name      (1),
dNSName      (2),
x400Address      (3),
directoryName      (4),
ediPartyName       (5),
uniformResourceIdentifier    (6),
iPAddress       (7),
registeredId       (8) },
otherNameForm    OBJECT IDENTIFIER }
CertPolicySet ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId
```

Это правило соответствия возвращает TRUE, если все компоненты, представляемые в данных, соответствуют аналогичным компонентам значения атрибута как приведено ниже:

- serialNumber соответствует, если значение этой компоненты в значении атрибута эквивалентно тому, которое приведено в представленном значении;

- issuer соответствует, если значение этой компоненты в значении атрибута эквивалентно тому, которое приведено в представленном значении;

- subjectKeyIdentifier соответствует, если значение этой компоненты в хранимом значении атрибута эквивалентно тому, которое приведено в представленном значении. Соответствия не будет, если хранимое значение атрибута не содержит расширение идентификатора ключа субъекта;

- authorityKeyIdentifier соответствует, если значение этой компоненты в хранимом значении атрибута эквивалентно тому, которое приведено в представленном значении. Соответствия не будет, если хранимое значение атрибута не содержит расширение идентификатора ключа органа или если не все компоненты в представленном значении представлены в хранимом значении атрибута;

- certificateValid соответствует, если представленное значение потеряло силу в течении периода действительности хранимого значения атрибута;

- privateKeyValid соответствует, если представленное значение потеряло силу в течении периода указанного расширением периода использования открытого ключа хранимого значения атрибута или если нет расширения периода использования открытого ключа хранимого значения атрибута;

- subjectPublicKeyAlgID соответствует, если он эквивалентен компоненте algorithm algorithmIdentifier компоненты subjectPublicKeyInformation хранимого значения атрибута;

- keyUsage соответствует, если набор битов в представленном значении такой же, как в расширении использования ключа в хранимом значении атрибута, или если нет расширения использования ключа в хранимом значении атрибута;

- subjectAltName соответствует, если хранимое значение атрибута содержит расширение альтернативного имени субъекта с компонентой одинакового типа названия как указано в представленном значении;

- policy соответствует, если хотя бы один из представленных элементов CertPolicySet появляется в расширении политики сертификата в хранимом значении атрибута или если представленный или хранимый сертификат содержит специальное значение anyPolicy в компоненте policy. Соответствия не будет, если в хранимом значении атрибута нет расширения политики сертификата;

- pathToName соответствует за исключением сертификата, имеющего расширение ограничивающего название, которое запрещает построение пути сертификации в значении представленного названия;

- subject соответствует, если значение этого компонента в значении атрибута эквивалентно тому, которое приведено в представленном значении;

- nameConstraints соответствует, если название субъекта в хранимом значении атрибута находится в пределах области названия, данного значением компоненты permitted-subtrees представленных данных, и за пределами области названия, данного значением компоненты excluded-subtrees представленных данных.

O'ZSTANDART AGENTLIGI
STANDARDLASHTIRISH,
METROLOGIYA VA DAVLAT NAZORATI
BOSHQARMASI

9.3.3 Точное соответствие пары сертификатов

Правило точного соответствия пары сертификатов сравнивает равенство представленного значения со значением типа CertificatePair. Оно уникально выбирает одну пару кросс-сертификатов.

```
certificatePairExactMatch MATCHING-RULE  ::= {
SYNTAX  CertificatePairExactAssertion
ID    id-mr-certificatePairExactMatch }
CertificatePairExactAssertion ::= SEQUENCE {
issuedToThisCAAssertion   [0] CertificateExactAssertion OPTIONAL,
issuedByThisCAAssertion   [1] CertificateExactAssertion OPTIONAL }
( WITH COMPONENTS    {..., issuedToThisCAAssertion PRESENT} |
WITH COMPONENTS    {..., issuedByThisCAAssertion PRESENT} )
```

Это правило согласования возвращает значение TRUE, если компоненты, которые представлены в компонентах issuedToThisCAAssertion и issuedByThisCAAssertion представленного значения соответствуют аналогичным компонентам соответствующих компонент issuedToThisCA и issuedByThisCA в хранимом значении атрибута.

9.3.4 Соответствие пары сертификатов

Правило согласования пары сертификатов сравнивает представленное значение со значением типа CertificatePair. Оно выбирает одну или более одной пары кросс-сертификатов на основе различных характеристик либо пары сертификатов issuedToThisCA либо issuedByThisCA.

```
certificatePairMatch MATCHING-RULE ::= {
SYNTAX    CertificatePairAssertion
ID      id-mr-certificatePairMatch }
CertificatePairAssertion ::= SEQUENCE {
issuedToThisCAAssertion   [0] CertificateAssertion OPTIONAL,
issuedByThisCAAssertion   [1] CertificateAssertion OPTIONAL }
( WITH COMPONENTS    {..., issuedToThisCAAssertion PRESENT} |
WITH COMPONENTS    {..., issuedByThisCAAssertion PRESENT} )
```

Это правило согласования возвращает TRUE, если все компоненты, которые представлены в компонентах issuedToThisCAAssertion и issuedByThisCAAssertion представленного значения, соответствуют аналогичным компонентам соответствующих компонент issuedToThisCA и issuedByThisCA в хранимом значении атрибута.

9.3.5 Точное соответствие списка сертификатов

Правило точного соответствия списка сертификатов сравнивает эквивалентность представленного значения со значением атрибута типа CertificateList. Оно уникально выбирает один CRL.

certificateListExactMatch MATCHING-RULE ::= {
SYNTAX CertificateListExactAssertion
ID id-mr-certificateListExactMatch }
CertificateListExactAssertion ::= SEQUENCE {
issuer Name,
thisUpdate Time,
distributionPoint DistributionPointName OPTIONAL }

Это правило возвращает TRUE, если компоненты в хранимом значении атрибута соответствуют тем, которые приведены в представленном значении. Если компонента distributionPoint представлена, то она совпадает как минимум в одной форме имен.

9.3.6 Соответствие списка сертификата

Правило соответствия списка сертификата сравнивает представленное значение со значением атрибута типа CertificateList. Оно выбирает один или более CRL на основе различных характеристик.

certificateListMatch MATCHING-RULE ::= {
SYNTAX CertificateListAssertion
ID id-mr-certificateListMatch }
CertificateListAssertion ::= SEQUENCE {
issuer Name OPTIONAL,
minCRLNumber [0] CRLNumber OPTIONAL,
maxCRLNumber [1] CRLNumber OPTIONAL,
reasonFlags ReasonFlags OPTIONAL,
dateAndTime Time OPTIONAL,
distributionPoint [2] DistributionPointName OPTIONAL,
authorityKeyIdentifier [3] AuthorityKeyIdentifier OPTIONAL }

Правило согласования возвращает true, если все компоненты, которые представлены в представленном значении, соответствуют аналогичным компонентам хранимого значения атрибута:

- issuer соответствует, если значение этой компоненты в значении атрибута эквивалентно тому, которое дано в представленном значении;

- minCRLNumber соответствует, если его значение меньше или эквивалентно значению в расширении номера CRL хранимого значения атрибута; соответствия не будет, если хранимое значение атрибута не содержит расширение номера CRL;

- maxCRLNumber соответствует, если его значение больше чем или эквивалентно значению в расширении номера CRL хранимого значения атрибута; соответствия не будет, если хранимое значение атрибута не содержит расширение номера CRL;

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA
MUVOFIQLIKNI BAHOLASH
DAVLAT HAZORATI

- reasonFlags соответствует, если какие либо биты, которые установлены в представленном значении, также установлены в компонентах расширения издающего пункта распределения хранимого значения атрибута; соответствие также будет, если хранимое значение атрибута не содержит reasonFlags в расширении издающего пункта распределения или если хранимое значение атрибута не содержит расширение издающего пункта распределения;

- dateAndTime соответствует, если значение эквивалентно значению или позднее значения в компоненте thisUpdate хранимого значения атрибута и раньше значения компоненты nextUpdate хранимого значения атрибута; соответствия не будет, если хранимое значение атрибута не содержит компоненты nextUpdate;

- distributionPoint соответствует, если хранимое значение атрибута содержит расширение издающего пункта распределения и значение этой компоненты в представленном значении эквивалентно соответствующему значению, как минимум, в одной форме названия в этом расширении;

- authorityKeyIdentifier соответствует, если значение этой компоненты в хранимом значении атрибута эквивалентно тому, которое дано в представленном значении; Соответствия не будет, если хранимое значение атрибута не содержит расширение идентификатора ключа органа или если не все компоненты в представленном значении представлены в хранимом значении атрибута.

9.3.7 Соответствие идентификатора алгоритма

Правило соответствия идентификатора алгоритма сравнивает равенство представленного значения со значением атрибута типа SupportedAlgorithms.

algorithmIdentifierMatch MATCHING-RULE ::= {
SYNTAX AlgorithmIdentifier
ID id-mr-algorithmIdentifierMatch }

Правило соответствия возвращает значение TRUE, если представленное значение эквивалентно компоненте algorithmIdentifier хранимого значения атрибута.

9.3.8 Соответствие политики

Правило соответствия политики сравнивает эквивалентность представленного значения со значением атрибута типа CertificatePolicy или со значением атрибута типа privPolicy.

policyMatch MATCHING-RULE ::= {
SYNTAX PolicyID
ID id-mr-policyMatch }

Это правило соответствия возвращает значение TRUE, если представленное значение эквивалентно компоненте policyIdentifier хранимого значения атрибута.

9.3.9 Соответствие пути PKI

Правило соответствия pkiPathMatch сравнивает эквивалентность представленного значения со значением атрибута типа pkiPath. Система, использующая сертификат, может использовать это правило соответствия для выбора пути, начинающегося с сертификата, выданного со стороны ЦР, который он удостоверяет и заканчивающегося сертификатом, выданным для ЦР, который выдал конечному объекту утвержденный сертификат.

pkiPathMatch MATCHING-RULE ::= {
SYNTAX PkiPathMatchSyntax
ID id-mr-pkiPathMatch }
PkiPathMatchSyntax ::= SEQUENCE {
firstIssuer Name,
lastSubject Name }

Это правило соответствия возвращает значение TRUE, если значение в компоненте firstIssuer соответствует аналогичным элементам поля issuer первого сертификата в SEQUENCE в хранимом значении и представленное значение в компоненте lastSubject соответствует аналогичным элементам поля subject последнего сертификата в SEQUENCE в хранимом значении. Это правило соответствия возвращает значение FALSE, если оба случая не удачны.

10 Сертификаты атрибута

10.1 Структура сертификата атрибута

Структура сертификата атрибута, определенная в данном стандарте, предоставляет основу, на которой могут быть построены PMI.

Эти инфраструктуры могут поддерживать приложения, такие как управление доступом.

Ограничение привилегии для объекта обеспечивается органом посредством структуры цифровых подписанных данных, называемых сертификатом атрибута или посредством сертификата открытого ключа, содержащего расширение, определенное точно для этой цели.

Формат сертификата атрибута, определенный в данном стандарте, включает расширяющийся механизм и набор специфических расширений сертификата.

Если, по какой-либо причине, орган аннулирует прежде выданный сертификат, пользователи должны знать, что произошло аннулирование,

чтобы они не использовали не надежный сертификат. Списки аннулирования это схема, которая может быть использована для уведомления пользователей об аннулировании.

Система, использующая сертификат атрибута, должна утвердить сертификат до использования этого сертификата для приложения. Процедуры выполнения этого утверждения также определенны в данном стандарте и включают проверку целостности самого сертификата, его статус аннулирования и его действительность, касающуюся его использования по назначению. Эта структура включает большое количество необязательных элементов, которые свойственны только в некоторых средах. Также эта структура может быть использована в средах, где используются не все компоненты определенных моделей, хотя модели определяются как полные. Например, существуют среды, где не требуется аннулирование сертификатов атрибута. Делегирование привилегий и использование ролей, которые не являются универсальными, также аспекты данного стандарта.

Директория использует сертификаты атрибута для управления доступом к информации директории на основе правил.

В данном стандарте предоставлены два механизма для ограничения полномочий владельца атрибута.

Сертификаты открытого ключа, используемые в сочетании с услугой аутентификации объекта, могут непосредственно предоставить услугу предоставления прав доступа (авторизации), если привилегии ассоциированы с субъектом посредством инструкций издающего ЦР. Сертификаты открытого ключа могут содержать расширение subjectDirectoryAttirbutes, которое содержит привилегии, ассоциированные с субъектом сертификата открытого ключа. Этот механизм, подходит для ситуаций, где орган, издающий сертификат открытого ключа, также является органом для передачи привилегий, и период действительности привилегий соответствует периоду действительности сертификата открытого ключа. Конечные объекты не могут действовать как орган по передаче привилегий. Если какие-нибудь из расширений, определенных в разделе 12 данного стандарта, включены в сертификат открытого ключа, эти расширения применяются одинаково для всех привилегий, назначенных в расширении subjectDirectoryAttributes данного сертификата открытого ключа.

В общих случаях, привилегии объекта будут иметь жизненный цикл, который не соответствует периоду действительности сертификата открытого ключа. Привилегии обычно имеют меньший жизненный цикл. Орган для назначения привилегий, кроме этого, часто является органом, выдающим сертификат открытого ключа одному и тому же объекту.

Использование сертификатов атрибута изданных органом по присвоению атрибутов предоставляет гибкую инфраструктуру управления привилегиями, которая может быть установлена и управляться независимо от PKI. В то же время, существует взаимосвязь между этими двумя

инфраструктурами, на основе которой PKI используется для установления подлинности издателей и владельцев в сертификате атрибута.

Сертификат атрибута является отдельной структурой, в отличие от сертификата открытого ключа субъекта. Субъект может иметь множество сертификатов атрибута, каждый из которых связан со своим сертификатом открытого ключа. Нет такого требования, чтобы один и тот же орган создавал оба сертификата и сертификат открытого ключа, и сертификат атрибута для пользователя. В средах, где различные органы ответственны за издание сертификата открытого ключа и сертификата атрибута, сертификаты открытого ключа, изданные ЦР, и сертификаты атрибута, изданные АА должны быть подписаны, используя различные подписывающие закрытые ключи. В средах, где один объект является и ЦР, издающим сертификат открытого ключа, и органом по присвоению атрибутов, издающим сертификат атрибута, строго рекомендуется, чтобы использовались различные ключи для подписания сертификата атрибута и для подписания сертификата открытого ключа. Обмен между издающим органом и объектом, получающим сертификат, находится за пределами рассмотрения данного стандарта.

Сертификат атрибута определяется следующим образом:

```
AttributeCertificate ::= SIGNED {AttributeCertificateInfo}
AttributeCertificateInfo ::= SEQUENCE
{
version       AttCertVersion, --версия v2
holder        Holder,
issuer        AttCertIssuer,
signature       AlgorithmIdentifier, O¢z DSt 1108:2006
 58
serialNumber      CertificateSerialNumber,
attrCertValidityPeriod    AttCertValidityPeriod,
attributes        SEQUENCE OF Attribute,
issuerUniqueID      UniqueIdentifier OPTIONAL,
extensions        Extensions OPTIONAL
}
AttCertVersion ::= INTEGER { v2(1) }
Holder   ::= SEQUENCE
{
baseCertificateID     [0] IssuerSerial OPTIONAL,
entityName        [1] GeneralNames OPTIONAL,
objectDigestInfo      [2] ObjectDigestInfo OPTIONAL
}
ObjectDigestInfo   ::= SEQUENCE {
digestedObjectType ENUMERATED {
publicKey      (0),
publicKeyCert    (1),
```

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA
MAHSULOT SIFATINI
NAZORAT QILISH

```
otherObjectTypes    (2) },
otherObjectTypeID      OBJECT IDENTIFIER OPTIONAL,
digestAlgorithm      AlgorithmIdentifier,
objectDigest        BIT STRING }

AttCertIssuer ::=   [0] SEQUENCE {
issuerName          GeneralNames OPTIONAL,
baseCertificateID    [0] IssuerSerial OPTIONAL,
objectDigestInfo    [1] ObjectDigestInfo OPTIONAL }
( WITH COMPONENTS { ..., issuerName PRESENT } |
WITH COMPONENTS { ..., baseCertificateID PRESENT } |
WITH COMPONENTS { ..., objectDigestInfo PRESENT } )
IssuerSerial    ::= SEQUENCE {
issuer    GeneralNames,
serial    CertificateSerialNumber,
issuerUID  UniqueIdentifier OPTIONAL }
AttCertValidityPeriod   ::= SEQUENCE {
notBeforeTime    GeneralizedTime,
notAfterTime    GeneralizedTime }
```

Поле version используется для различия версий сертификатов атрибута. Версия сертификата атрибута, изданного, согласно с синтаксисом, приведенным в ITU-T X.509 [2], будет v.3.

Поле holder сообщает об идентичности владельца сертификата атрибута.

Компонента baseCertificateID, если она представлена, устанавливает специфический сертификат открытого ключа, который необходимо использовать для установления идентичности личности этого владельца при утверждении полномочий этим сертификатом атрибута.

Компонента entityName, если она представлена, то она устанавливает один или несколько имен владельца. Если entityName является единственной компонентой, которая существует в holder, любой сертификат открытого ключа, который имеет один из этих названий в качестве субъекта, может быть использован для установления идентичности личности этого владельца, когда утверждаются привилегии этим сертификатом атрибута.

Если существуют обе компоненты и baseCertificateID и entityName, то может быть использован только тот сертификат, который определен с помощью baseCertificateID. В этом случае, компонента entityName включается только в качестве инструмента, который помогает верификатору привилегий определить местонахождение сертификата открытого ключа.

Существует риск при использовании одного только GeneralNames для идентификации владельца, так как оно указывает только на имя владельца. Это не является достаточным, чтобы идентифицировать

личность владельца, чтобы выдать привилегии для этого владельца. Однако, использование имени издателя и серийного номера специфического сертификата открытого ключа позволяет издателю сертификатов атрибута доверять процессу аутентификации, выполняемому ЦР при издании определенного сертификата открытого ключа. Также некоторые опции в GeneralNames (например, IPAddress) не подходят для использования их при определении названия владельца сертификата атрибута, особенно когда владелец является ролевым, а не индивидуальным объектом.

Компонента objectDigestInfo, если она представлена, то она используется непосредственно для установления подлинности владельца. Владелец аутентифицируется посредством сравнения дайджеста соответствующей информации, созданной верификатором привилегий с помощью алгоритма, установленным в objectDigestInfo с содержимым objectDigest. Если они оба одинаковы, пользователь аутентифицируется для утверждения привилегий этим сертификатом атрибута.

Компонента publicKey будет указываться, когда включается хэш открытого ключа объекта. Хэширование открытого ключа может не уникально идентифицировать один сертификат (т.е. идентичное значение ключа может появляться во многих сертификатах).

Чтобы связать сертификат атрибута с открытым ключом, хэш вычисляется посредством образа этого открытого ключа, который должен быть представлен в сертификате открытого ключа. Точнее, входным параметром хэш алгоритма будет DER кодирование образа SubjectPublicKeyInfo ключа. Следует заметить, что если значение открытого ключа, используемое как входной параметр хэш функции, извлечено из сертификата открытого ключа, тогда возможно в этом случае может быть не достаточным входным параметром для HASH. Правильный входной параметр для хэширования в этой ситуации включает значение унаследованных параметров, и поэтому может отличаться от представленного в сертификате открытого ключа SubjectPublicKeyInfo.

Компонента publicKeyCert указывается, когда хэшируется сертификат открытого ключа, хэширование считается законченным, когда сертификат открытого ключа полностью кодируется с помощью DER, включая биты подписи.

Компонента otherObjectTypes указывается, когда кроме объектов открытых ключей и сертификатов открытых ключей хэшируются и другие объекты (например, объекты программного обеспечения). Идентичность типа объекта может поддерживаться по выбору.

Поле issuer сообщает подлинность органа по присвоению атрибутов, который выдал сертификат и если:

- представлена компонента issuerName, то она указывает на одно или несколько имен издателя;

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

- представлена компонента baseCertificateID, то она указывает издателя согласно ссылке на определенный сертификат открытого ключа, для которого этот издатель является субъектом;

- приведена компонента objectDigestInfo, то она указывает на издателя представлением хэша идентифицируемой информации.

Signature указывает на криптографический алгоритм, используемый для вставки ЭЦП сертификата атрибута.

serialNumber серийный номер, который уникально идентифицирует сертификат атрибута в контексте его издателя.

Поле attrCertValidityPeriod сообщает о промежутке времени, в течение которого сертификат атрибута рассматривается как действительный. Оно выражается в формате GeneralizedTime.

Поле attributes содержит атрибуты, связанные с владельцем который сертифицируется.

В случае дескриптора атрибута сертификатов атрибута эта последовательность атрибута может быть пустой.

issuerUniqueID может использоваться для идентификации издателя сертификата атрибута в случае, когда компонента issuer не достаточна.

Поле extensions позволяет добавлять новые поля в сертификате атрибута.

Структура сертификата атрибута, описанная в этом подразделе, главным образом сосредоточена на модели, в которой привилегии помещаются в сертификаты атрибута.

Однако, как упомянуто ранее, расширения сертификата, определенные в данном подразделе, могут быть также помещены в сертификате открытого ключа, используя расширение subjectDirectoryAttributes.

10.2 Пути сертификата атрибута

Сертификаты открытого ключа могут быть необходимы для сообщения пути сертификата атрибута (например, для утверждения привилегий внутри протокола прикладной программы). Следующий тип данных ASN.1 может использоваться для представления пути сертификата атрибута:

```
AttributeCertificationPath ::= SEQUENCE {
attributeCertificate AttributeCertificate,
acPath SEQUENCE OF ACPathData OPTIONAL }
ACPathData ::= SEQUENCE {
certificate [0] Certificate OPTIONAL,
attributeCertificate [1] AttributeCertificate OPTIONAL }
```

11 Взаимоотношения органа по присвоению атрибутов, источника полномочий и сертифицирующего органа

11.1 Привилегия в сертификатах атрибута

Орган по присвоению атрибутов и сертифицирующий орган логически (во многих случаях физически) совершенно независимы. Создание и поддержание идентичности может быть (часто должно быть) разделено от PMI. Таким образом, полная PKI, включающая ЦР, может существовать и действовать до установления PMI. Хотя ЦР является источником полномочий для идентификации в его домене, он не может автоматически являться источником полномочий для привилегий. Поэтому ЦР не обязательно самому быть органом по присвоению атрибутов, т.е. ему не обязательно быть ответственным за решение того, какие объекты будут иметь возможность функционирования как органы по присвоению атрибутов.

Источник полномочий это объект, который доверен верификатором полномочий, как объект с наибольшей ответственностью для назначения набора привилегий. Источник полномочий, когда издает сертификаты другим объектам, является органом по присвоению атрибутов, в которых устанавливаются привилегии для этих объектов. В некоторых средах нужны ЦР, чтобы обеспечить жесткий контроль над объектами, которые могут действовать как источник полномочий. Эта структура обеспечивает механизм для поддержания этих требований. В других средах этот контроль не нужен и механизм для определения объектов, которые могут действовать как источник полномочий, выходит за пределы рассмотрения данного стандарта.

Эта структура гибкая и может удовлетворять требования многих типов сред:

a) во многих средах привилегии могут быть назначены индивидуальным объектам непосредственно одним органом по присвоению атрибутов, а именно источником полномочий;

b) другие среды могут требовать поддержания необязательных характеристик ролей, при помощи которых личности получают сертификаты, которые назначают различные роли. Привилегии, связанные с ролями косвенно, назначаются для таких личностей;

c) другая необязательная характеристика данной структуры это поддержание делегирования привилегий. Если делегирование выполнено, источник полномочий назначает привилегии объекту, которому разрешается также действовать как орган по присвоению атрибутов и в дальнейшем делегировать привилегии. Делегирование может проходить через промежуточные органы по присвоению атрибутов и продолжается до тех пор, пока назначается конечный объект, который не может делегировать эти привилегии. Промежуточные органы по присвоению

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

атрибутов могут или не могут действовать как назначитель привилегий для привилегий, которые они делегируют;

d) в некоторых средах некоторые физические объекты могут действовать и как орган по присвоению атрибутов и как ЦР. Эта двойная логическая роль для одного и того же физического объекта всегда является случаем, когда привилегия сообщается в расширении subjectDirectoryAttributes сертификата открытого ключа. В некоторых средах индивидуальные физические объекты действуют как ЦР и орган по присвоению атрибутов. В последнем случае привилегия назначается, используя сертификаты атрибута вместо сертификатов открытого ключа.

В то время как сертификаты атрибута указывают на сертификаты открытого ключа для их издателей и владельцев, PKI используется для установления подлинности владельца (назначителя привилегий) и проверки ЭЦП издателя.

Объекты могут получать привилегию в двух случаях:

- орган по присвоению атрибутов может односторонне присвоить привилегию объекту посредством создания сертификата атрибута. Этот сертификат может храниться в открытом хранилище и может последовательно быть обработан одним или несколькими верификаторами привилегий для того, чтобы принять решение об авторизации. Все это может произойти без уведомления об этом объекта или без специального действия;

- в качестве альтернативы, объект сам может запросить привилегии какого-нибудь ОРГАНА ПО ПРИСВОЕНИЮ АТРИБУТОВ. Созданный сертификат может быть возвращен (только) запрашивающему объекту, который точно поддерживает его, когда запрашивается доступ к некоторому защищенному ресурсу.

11.2 Привилегия в сертификатах открытого ключа

В некоторых средах привилегии связываются с субъектом посредством применения ЦР.

Такие привилегии могут быть помещены непосредственно в сертификаты открытого ключа, а не в издаваемые сертификаты атрибута. В таких случаях привилегии включаются в расширение subjectDirectoryAttributes сертификата открытого ключа.

Этот механизм пригоден в тех средах, где одно или несколько из следующих действий является действительным:

- один и тот же физический объект действует и как ЦР и как орган по присвоению атрибутов;

- продолжительность времени действия привилегии приравнивается к продолжительности времени действия открытого ключа, включенного в сертификат;

- делегирование привилегий не разрешено;

- делегирование разрешено, но для одного делегирования, все привилегии в сертификате (в расширении subjectDirectoryAttributes) имеют одинаковые параметры делегирования и все расширения, относящиеся к делегированию, применяются одинаково ко всем привилегиям в сертификате.

12 Модели PMI

12.1 Общая модель

12.1.1 PMI в контексте управления доступом

Общая модель управления привилегиями состоит их трех частей: объект, назначитель привилегий и верификатор привилегий.

Объект может быть защищаемым ресурсом. К ресурсу, который защищается, обращаются как к объекту. Этот тип объекта имеет методы, которые могут быть активизированы (например, объектом может быть межсетевой экран, который имеет объектный метод “Allow Entry”, или объектом может быть файл в файловой системе, который имеет объектные методы чтения, записи и выполнения).

Назначитель привилегий это объект, который владеет определенной привилегией и назначает свои привилегии для применения в определенной ситуации.

Верификатор привилегий это объект, который определяет, являются ли достаточными назначенные привилегии для использования в данной ситуации.

Определение успешного/неудачного установления привилегий, выполненное верификатором привилегий, зависит от следующих четырех аспектов:

- привилегий назначителя привилегий;
- местной политики привилегий;
- текущих переменных сред;
- критичности объектного метода.

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

Привилегия владельца привилегий отражает уровень доверия, установленный для него, определяемый издателем сертификата. Эта привилегия инкапсулирована в сертификатах атрибута владельца привилегий (или в расширении subjectDirectoryAttrubute в его сертификате открытого ключа), которые могут быть представлены верификатору привилегий во время запроса, или могут быть распространены другими способами, например через директорию. Шифрование выполняется с помощью использования конструкции Attribute, содержащей значения AttributeType и SET OF AttributeValue.

Политика привилегий определяет уровень привилегии, который считается достаточным для критичности данного объектного метода или контекста использования. Политика привилегий должна быть защищена в целях целостности и подлинности. Существует несколько вариантов для передачи политики. Сущность идеи в том что, политика в действительности не сообщается вообще, а просто определяется и всегда

хранится локально в среде верификатора привилегий. Другая сущность идеи в том, что некоторые политики являются «универсальными» и должны быть переданы и быть известными каждому объекту в системе. Схема компоненты для хранения информации политики привилегий в директории определена в данном стандарте.

Политика привилегий определяет порог на одобрение для данного набора привилегий.

То есть, она указывает в точности, когда верификатор привилегий должен выводить заключение о том, что представленный набор привилегий является «достаточным» для того, чтобы он мог дать доступ (к запрашиваемому объекту, ресурсу, приложению и т.д.) к назначителю привилегий.

Синтаксис определения политики привилегий не стандартизируется данным стандартом. Для этой цели может быть использован любой синтаксис, включающий чистый текст. Независимо от использования синтаксиса для определения политики привилегий, каждый пример политики привилегий должен быть уникально идентифицирован.

Идентификаторы объектов используются для этих целей.

PrivilegePolicy ::= OBJECT IDENTIFIER

Критичность объектного метода, если это предусмотрено, может отражать атрибуты документа или запроса, которые необходимо обработать, и эта обработка имеет в качестве цели авторизацию или конфиденциальность содержания документа. Критичность объектного метода может быть точно зашифрована в ассоциированной метке безопасности или в сертификате атрибута, хранящегося объектным методом, или может быть косвенно инкапсулирована в структуре и содержании ассоциированного объекта данных. Она может быть зашифрована одним из многочисленных методов. Альтернативно, это может быть выполнено внутри PMI, в сертификате атрибута, ассоциированного с объектным методом.

Нет никакой необходимости в связывающих взаимоотношениях между верификатором привилегий и каким-либо отдельным органом по присвоению атрибутов. В то время, как владельцы привилегий могут иметь сертификаты атрибута, изданные большинством различных органов по присвоению атрибутов, верификаторы привилегий могут признавать сертификаты, изданные многочисленными органами по присвоению атрибутов, которые не должны быть иерархически связанными между собой, для того, чтобы дать доступ к определенному ресурсу.

Структура сертификата атрибута может быть использована для управления привилегиями различных типов и для множества различных целей.

Существует стандартная структура для управления доступом, которая определяет соответствующий набор соглашений, которые характерны для приложения управления доступом.

Назначитель привилегий в данном стандарте предполагается в роли «инициатора» в структуре управления доступом.

Верификатор привилегий в данном стандарте предполагается в роли «исполнителя функции управления доступом».

Объектный метод, для которого утверждена привилегия, в данном стандарте предполагает «цель», определенную в структуре управления доступом.

Переменные среды в данном стандарте предполагают «информацию, вытекающую из контекста» в структуре управления доступом.

Политика привилегий, рассмотренная в данном стандарте, должна включать «политику управления доступом» и «правила политики управления доступом», как определено в структуре управления доступом.

Эта модель позволяет PMI быть достаточно равномерно защищенной в существующей сети ресурсов. Верификатор привилегий экранирует все запросы и только те, которые авторизованы правильно, пропускаются к соответствующему объектному методу.

12.1.2 PMI в контексте безотказности от авторства

Существует стандартная структура для безотказности, которая определяет соответствующий набор соглашений, которые характерны для безотказности от авторства.

Назначитель привилегий в этом стандарте предполагается в роли «признака субъекта» или «автора» в структуре безотказности от авторства.

Верификатор привилегий в этом стандарте предполагается в роли «признака пользователя» или «получателя» в структуре безотказности от авторства.

Объектный метод, для которого в этом стандарте утверждена привилегия, предполагает «цели» в структуре безотказности от авторства.

Переменные среды в этом стандарте предполагают «признак генерированной и проверенной даты или времени» в структуре безотказности от авторства.

Политика привилегий, рассматриваемая в этом стандарте, может включать «политику безопасности безотказности от авторства» в структуре безотказности от авторства.

STANDART AGENTLIGI
TASDIQLANGAN NUSXASI
MUHRLANGAN

12.2 Модель управления PMI

Модель управления иллюстрирует, как управление влияет на доступ к критичному объектному методу. Модель имеет пять компонентов: назначитель привилегий, верификатор привилегий, объектный метод, политика привилегий и переменные среды (рисунок 2).

Назначитель привилегий имеет привилегию; объектный метод имеет критичность.

Методы, описанные в данном стандарте, дают возможность назначителю привилегий управлять доступом к объектному методу, в

соответствии с политикой привилегий. И привилегии и критичность могут быть многозначными параметрами.



Рисунок 2 – Модель управления PMI

Назначитель привилегий может быть объектом, установленным сертификатом открытого ключа.

12.3 Модель делегирования

В некоторых средах может быть необходимо делегирование привилегий, однако это необязательный аспект структуры и не требуется во всех средах. Модель делегирования имеет четыре компоненты: верификатор привилегий, источник полномочий, другие органы по присвоению атрибутов и назначитель привилегий (рисунок 3).

Также как в средах, где делегирование не используется, источник полномочий является исходным издателем сертификатов, который назначает привилегии уполномоченным владельцам. Несмотря на это, в этом случае источник полномочий разрешает владельцу привилегий, чтобы тот действовал как орган по присвоению атрибутов и в дальнейшем делегировал эти привилегии другим объектам посредством выдачи сертификатов, которые содержат такие же привилегии (или их подмножества).

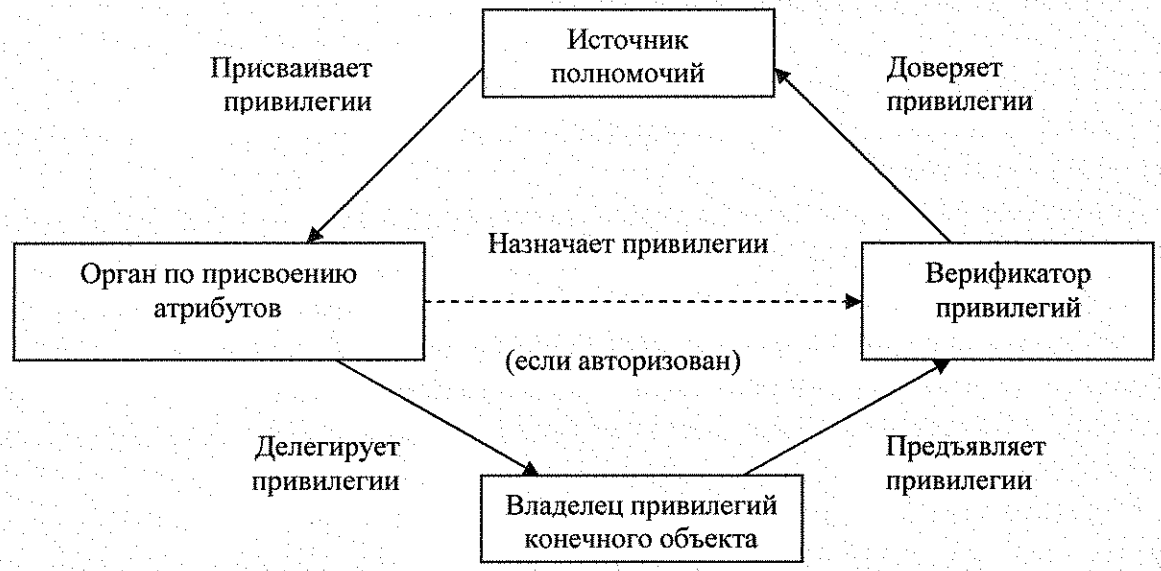


Рисунок 3 – Модель делегирования

Источник полномочий может налагать ограничения на делегирование. Каждый из этих промежуточных органов по присвоению атрибутов может разрешать дальнейшее делегирование в сертификатах, которые он издает для более отдаленных владельцев привилегии, чтобы делегирование дальше выполнялось этими владельцами, и эти владельцы также действовали как орган по присвоению атрибутов. Делегатор также может ограничивать нижестоящие органы по присвоению атрибутов.

Когда используется делегирование, верификатор привилегий доверяет источнику полномочий делегирование некоторых или всех привилегий владельцу привилегий, некоторые из которых могут в дальнейшем делегировать некоторые или все из этих привилегий другим владельцам.

Верификатор привилегий доверяет источнику полномочий, как органу для данного набора привилегий, для ресурса. Если сертификат утвердителя привилегий не выдан этим источником полномочий, тогда верификатор привилегий устанавливает путь делегирования сертификатов с этого утвердителя привилегий к тому источнику полномочий, который выдал сертификат. Утверждение пути делегирования включает проверку того, имеет ли орган по присвоению атрибутов достаточно привилегий, и был ли правильно уполномочен для делегирования этих привилегий.

Для случая, в котором привилегии сообщаются посредством сертификатов атрибута, путь делегирования отличается от пути утверждения сертификата и используется для утверждения сертификатов открытого ключа объектов, включенных в процесс делегирования. Однако, качество подлинности, предложенное процессом утверждения

TSTEHSIDA FOYDALIGI
TAQDIMOTI TAQIQLANISH,
HUQUQIY AKTLARGA
VA DAVLAT NAZORATI

сертификата открытого ключа, будет соответствовать критичности объектного метода, т.е. будет защищаться.

12.4 Модель ролей и атрибут роли

Роли предоставляют средства для косвенного назначения привилегий пользователям.

Личности получают сертификаты назначения роли, которые назначают один или более ролей для них посредством атрибута ролей, входящих в состав сертификата. Определенные привилегии назначаются названиям ролей посредством сертификатов определения роли, а не индивидуальным владельцам привилегий посредством сертификата атрибута. Сертификаты назначения роли могут быть сертификатом атрибута или сертификатом открытого ключа.

Сертификаты определения роли могут быть сертификатом атрибута, но не сертификатом открытого ключа. Если сертификаты определения роли не используются, назначение привилегии для роли может быть произведено посредством других способов.

Возможны все нижеследующие случаи:

- любым органом по присвоению атрибутов может быть определено любое количество ролей;
- сами роли и участники роли могут быть определены и управляться раздельно, различными органами по присвоению атрибутов;
- членство ролей может быть делегированы также, как любые другие привилегии;
- ролям могут быть назначены любые подходящие продолжительности жизни.

Если сертификат назначения роли это сертификат атрибута, то атрибут role содержится в компоненте attributes сертификата атрибута. Если сертификат назначения роли это сертификат открытого ключа, то атрибут role содержится в расширении subjectDirectoryAttributes. В последнем случае, любые добавочные привилегии, содержащиеся в сертификате открытого ключа, это привилегии, которые непосредственно назначены субъекту сертификата, а не для роли.

Таким образом, назначитель привилегий может предоставить сертификат назначения роли верификатору привилегий, который наглядно показывает только то, что назначитель привилегий имеет специфическую роль. Верификатор привилегий может знать априори или может обнаружить некоторыми другими методами привилегии, связанные с назначенной ролью, чтобы принять решение об удачной/неудачной авторизации. Сертификат, определяющий роль, может быть использован для этой цели.

Верификатор привилегий должен иметь согласование определенных привилегий для роли. Назначение этих привилегий для роли может быть сделано в пределах PMI в сертификате определения роли или за пределами PMI. Если привилегии роли утверждены в сертификате определения роли, механизмы для сцепления этого сертификата с уместным сертификатом

назначения роли предусмотрено в данном стандарте. Сертификат, определяющий роли, не может быть делегирован другому объекту. Издатель сертификата назначения роли может не зависеть от издателя сертификата, определяющего роль, и они могут управляться (утрачивать силу, аннулироваться и т.д.) раздельно. Тот же самый сертификат (сертификат атрибута или сертификат открытого ключа) может быть сертификатом назначения роли и заодно содержать назначение других привилегий непосредственно одной и той же личности. Однако, сертификат определения роли должен быть отдельным сертификатом.

Определение значений для атрибута роли выходит за пределы данного стандарта.

```
role ATTRIBUTE ::= {
WITH SYNTAX   RoleSyntax
ID      id-at-role }
RoleSyntax ::= SEQUENCE {
roleAuthority   [0]   GeneralNames OPTIONAL,
roleName    [1]   GeneralName }
```

Атрибут привилегий должен быть использован для заполнения поля attributes сертификата назначения роли. Если сертификат назначения роли это сертификат открытого ключа, атрибут должен быть использован для заполнения расширения subjectDirectoryAttributes этого сертификата открытого ключа.

Если представлен roleAuthority, то он указывает признанный орган, который ответственный за издание сертификата, определяющего роль.

Если указан roleAuthority, и если верификатор привилегий использует сертификат определения роли для определения привилегий, назначенных для роли, то по крайней мере одно название в roleAuthority должно представляться в поле issuer в этом сертификате назначения роли. Если верификатор привилегий использовал средство другое, чем сертификат, определяющий роли, для определения привилегий назначенных для роли, тогда механизмы для гарантирования того, что привилегии были назначены органом, указанным в этой компоненте, выходят за пределы данного стандарта.

Если отсутствует roleAuthority, идентификация ответственного органа должна быть определена другими способами.

Компонента roleName идентифицирует роль, которая назначается для владельца сертификата назначения роли, содержащего этот атрибут. Если верификатор привилегий использует сертификат определения роли для определения привилегий, назначенных для этой роли, то это название роли должно также представляться в поле holder сертификата, определяющего роль.

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
DAVLAT MILLIY

12.5 Признание полномочий модели

Рисунок 4 показывает модель контроля для единого домена ITU-T X.509 [2] PMI.

Политика PMI содержит информацию, которая направляет PDP в принятии решений по контролю доступа. Эта информация включает данные о SOA, которому доверяют, правилах делегации, известных и используемых признаках, необходимых привилегиях для получения доступа к ресурсам. Информация политики конфигурируется в PDP статически, либо динамически, пропуская защищенный сертификат атрибута политики привилегий PDP.

Для поддержания связи между организациями, различные PMI должны использоваться вместе, чтобы сертификаты атрибута, выпущенные в одном домене, могли эффективно использоваться в другом домене PMI, чтобы получить доступ к его ресурсам.

Признание привилегий позволяет интегрировать PMI от различных доменов в единый объединенный PMI.

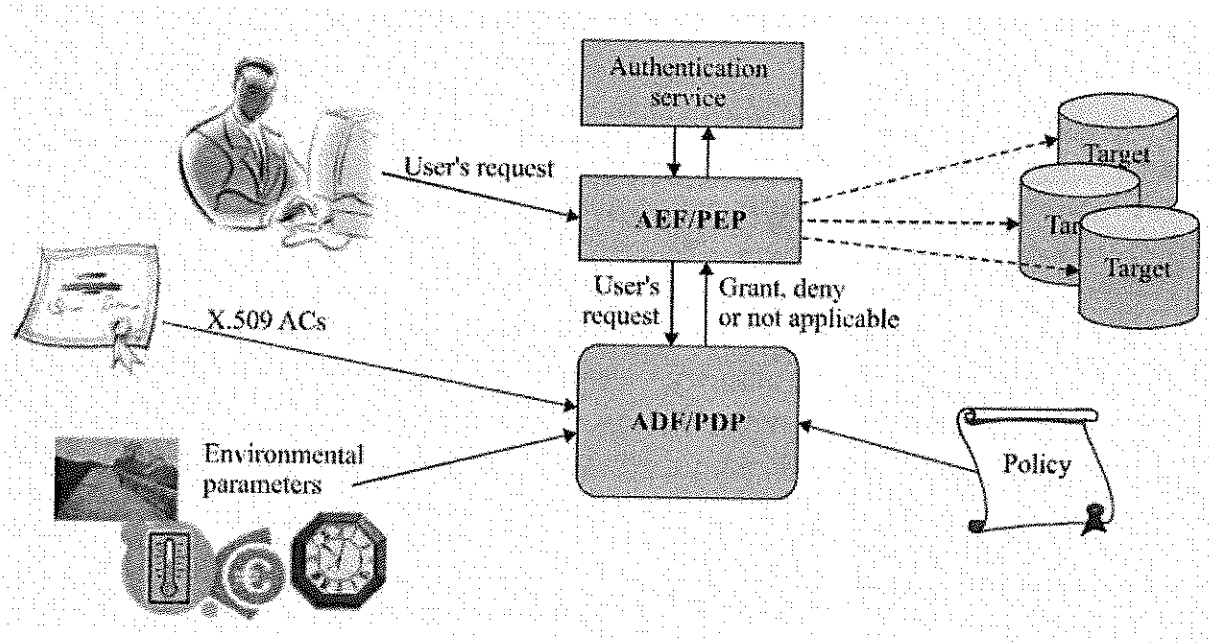


Рисунок 4 – Модель контроля для единого домена PMI

На рисунке 5 показаны два объединенных домена PMI

Пользователь, который является членом домена TopLeft, получает доступ к ресурсам домена BottomRight следующими способами:

1) непосредственная передача запроса пользователем домена TopLeft;

2) PDP в домене BottomRight должен понять CA, выпущенный доменом TopLeft, и политика BottomRight должна с помощью BottomRight PDP определить, достаточны ли они, чтобы предоставить доступ к требуемому ресурсу или нет.

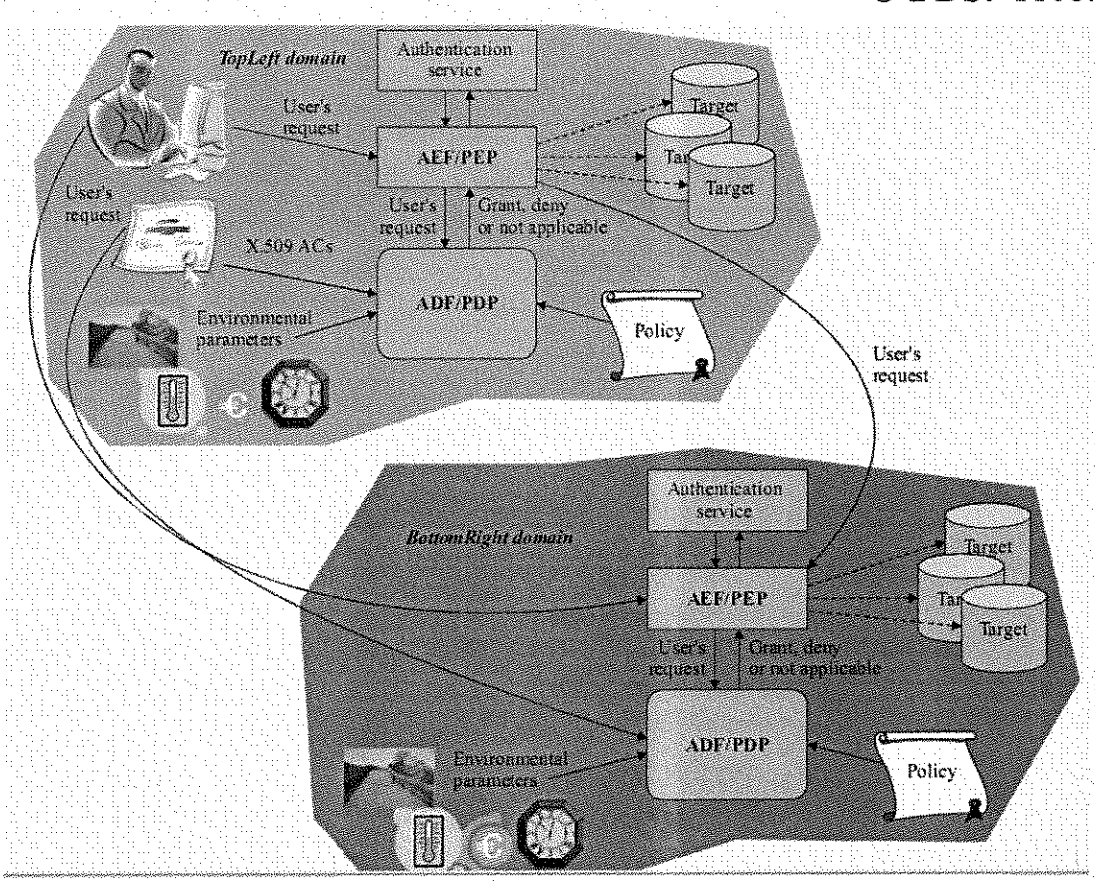


Рисунок 5 – Два объединенных домена PMI

SOA в доверяющем домене должен модернизировать ее политику так, чтобы SOA удаленного домена стал доверенным.

Локальная политика может быть модернизирована следующим образом:

a) статически, с добавлением дополнительной информации в политику, которая загружена в локальный PDP до этого принимающий решения по контролю доступа;

b) динамически, выпуская новую дополнительную политику, которая добавляет дополнительную информацию к текущей политике.

Динамическое дополнение к локальной политике должно определяться локальным SOA, выпускающим переменную политику к удаленному SOA, или локальным SOA, выпускающим административный переменный CA роли к удаленному SOA так, чтобы удаленный SOA мог выпустить собственный переменный CA политики.

Локальные SOA выпускают переменный CA политики к удаленному SOA, содержащий следующие поля:

– поле владельца, идентифицирующее SOA удаленного домена;

– поле издателя, идентифицирующее локальный SOA;

– атрибуты CA - совокупность всех атрибутов привилегии, для которой удаленный SOA является доверенным.

O‘ZSTANDART AGENTLIGI
STANDARTINFORM
MO‘LJALLANGAN
VA DAVLAT HAZORATI

Если любая из этих привилегий атрибута присутствует в вновь определенной роли, то новые CA спецификации роли также должны быть выданы.

Расширение basicAttConstraints входит в полномочия с установленным значением TRUE, чтобы указать, что удаленный SOA является AA.

Ограничение длины пути (pathLenConstraint) устанавливается по мере необходимости, чтобы указать длину – цепь делегации, которая допускается в удаленном домене.

holderNameConstraints служит для ограничения имени формы и пространства имен, в которых удаленный SOA может назначить привилегию атрибутов для пользователей.

allowedAttributeAssignments может ограничить, каким группам удаленных владельцев могут быть назначены какие наборы привилегий атрибута.

attributeMappings может быть установлен для информирования локальных PDP, которые удаленно назначают атрибуты, какие из них следует считать одинаковыми с локально присвоенными атрибутами.

Локальные SOA выпускают административную роль CA для удаленного SOA следующим образом:

1) локальный SOA определяет административную роль для локального домена и возможных разрешений, которые управляются этой административной ролью. Это определяется в CA спецификации роли, в которой владелец является административной ролью, и атрибут - атрибутом разрешения.

Набор разрешений для административной роли называют административными возможностями административной роли. Эти разрешения могут также быть назначены для локальных ролей, так, чтобы пользователи с этими локальными ролями могли использовать эти разрешения. Выпуск административного CA спецификации роли позволяет удаленным администраторам использовать их административные возможности;

2) локальный SOA делегирует эту административную роль удаленному SOA, выпуская CA назначенной роли удаленному SOA, содержащему назначенную административную роль. Удаленный SOA позволяет делегировать административную роль другим администраторам в удаленном домене с помощью pathLenConstraint в basicAttConstraints расширении в CA назначения роли;

3) удаленный SOA (или подчиненный AA), которому была назначена административная роль, признается как юридическое лицо, способное выпустить два типа делегированного CA политики, или делегированный CA спецификации роли, или делегированный атрибут, определяющий CA. В делегированном CA спецификации роли, удаленный SOA (или AA) непосредственно назначает разрешения от административных возможностей до новых атрибутов;

4) для того, чтобы удаленный SOA (или AA) не мог превысить своих делегированных полномочий, система авторизации должна подтверждать, что привилегии CA политики находятся в рамках административных возможностей, установленных для административной роли. В этом случае делегированная CA политика принимается, и правила политики динамично включаются в политику локальных SOA. Иначе делегированный CA политики отклоняется, и его правила политики игнорируются.

Делегированный CA спецификации роли включает следующие поля:

1) Владелец;

2) Поле издателя, которое идентифицирует удаленный SOA (или AA) удаленного домена, которая выпустила этот CA;

3) Атрибуты CA;

4) holderNameConstraints,

5) allowedAttributeAssignments;

6) attributeMappings.

Удаленный SOA будет впоследствии использовать CA привилегии атрибута для конечных пользователей и/или AA в своем домене.

Атрибуты привилегии в CA, выпущенном отдаленным SOA, могут содержать также разрешения, которые понятны PDP в локальном домене или роли, которые могут быть поняты PDP в локальном домене.

Когда CA содержит роли, которые не понятны локальным PDP, последние должны знать, как отобразить их неизвестные роли в локальных разрешениях. Это достигается следующим образом, если:

1) локальный SOA знает необходимые роли до признания удаленного SOA, то вопросы политики CA с удаленным SOA отображает атрибут расширения, который помещается в политике CA и выдается удаленному SOA;

2) удаленные роли не будут известны до признания удаленного SOA, то удаленный SOA должен будет или выпустить атрибут, отображающий CA политики или поместить атрибут, отображающий расширение в CA, который выпускается для пользователей;

3) в удаленном SOA политика содержит атрибут отображения CA, то в нем должно содержаться:

– наименование владельца и издателя удаленного SOA;

– поле атрибутов, которое является пустым;

– набор расширения attributeMappings для описания картографии атрибута.

Политику атрибутов отображения CA необходимо сделать доступной для локальных PDP на время принятия решения.

Это должно быть сделано либо сохранением политики CA в записи каталога на удаленном SOA и предоставлением локальным PDP доступа на чтение к нему, или путем включения политики CA в множество CA, представленных удаленным пользователем при доступе к локальным ресурсам(Push-модели).

O'ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASH
DAVLAT QO'MITASI

12.6 XML привилегии атрибутивной информации

В сертификате атрибута XML-закодированные привилегии могут быть переданы следующим образом:

```
xmlPrivilegeInfo ATTRIBUTE ::= {
WITH SYNTAX UTF8String --contains XML-encoded privilege information
ID id-at-xMLPrivilegeInfo }
```

Схема XML для типа атрибута роли определяется в соответствии с ASN.1 или с XSD.

XML-код, содержащийся в пределах UTF8String всегда должен совпадать.

Следующая схема ASN.1, определяет тип атрибута роли XML. Он сопровождается спецификацией XSD для того же самого типа атрибута, что и XML.

Схема определяет атрибут роли с удостоверением привилегий и названия роли.

```
CERTIFICATE-ATTRIBUTE DEFINITIONS ::=
BEGIN
Role ::= [UNCAPITALIZED] SEQUENCE {
id [ATTRIBUTE] XML-ID,
authorities SEQUENCE (1..MAX) OF
authority UTF8String,
name UTF8String }

XML-ID ::= UTF8String
END
```

Следующая схема XSD является альтернативной предыдущей:

```
<schema xmlns="http://www.w3.org/2000/08/XMLSchema">
 <element name="role">
  <attribute name="id" type="ID"/>
   complexType>
    <sequence>
     <element name="authorities">
      <complexType>
       <sequence>
        <element name="authority" type="string" minOccurs="1" maxOccurs="*"/>
       </sequence>
      </complexType>
```

```
    </element>
    <element name="name" type="string"/>
   </sequence>
  </complexType>
 </element>
</schema>
```

Данная схема, с учетом атрибута xMLPrivilegeInfo, выглядит следующим образом:

```
<role id="123"
xmlns="http://www.example.org/certificates/attribute">
<authorities>
<authority>Fictitious Organization</authority>
</authorities>
<name>manager</name>
</role>
```

12.7 Атрибут разрешения и согласования правил

12.7.1 Атрибут разрешения

Данный атрибут определяет общее разрешение, с помощью которого осуществляется управление на объекте. Он описывается следующим образом:

```
permission ATTRIBUTE ::= {
WITH SYNTAX DualStringSyntax
EQUALITY MATCHING RULE dualStringMatch
ID id-at-permission }

DualStringSyntax ::= SEQUENCE {
operation [0] UnboundedDirectoryString,
object [1] UnboundedDirectoryString }
```

Атрибут разрешения используется для заполнения атрибутов в сертификатах атрибутов и не предназначен для хранения информации, как атрибут каталога.

12.7.2 Правило использования двойных строк

Правило использования двойных строк dualStringMatch определяется следующим образом:

```
dualStringMatch MATCHING-RULE ::= {
SYNTAX DualStringSyntax
ID id-mr-dualStringMatch }
```

DualStringMatch выполняет чувствительное к регистру сравнение на равенство между данной парой строк и значением атрибута типа DualStringSyntax, в котором первая строка является строкой операций, а вторая - строкой объекта.

13 Расширения сертификата для управления привилегиями

13.1 Базовое расширение управления привилегиями

13.1.1 Необходимые условия

Следующие расширения сертификата могут быть включены в сертификат в целях управления привилегиями. Вместе с определением самих расширений, так же предоставлены правила для типов сертификата, в которых могут быть представлены эти расширения.

За исключением расширения идентификатора источника полномочий, любые из расширений, которые могут быть включены в сертификат открытого ключа, могут быть включены только в том случае, если этот сертификат открытого ключа является одним из тех, который назначает привилегии для своего субъекта (т.е. должно быть представлено расширение subjectDirectoryAttributes). Если любое из этих расширений представлено в сертификате открытого ключа, это расширение применяют ко всем привилегиям, представленным в расширении subjectDirectoryAttribute.

Список аннулирований, использованный для опубликования извещений об аннулировании для сертификатов атрибута (ACRL и AARL), может содержать любой CRL или расширение элемента CRL, как это определено для использования в CRL и CARL в разделах с 4 по 11 настоящего стандарта.

Следующие области расширения включают:

a) базовое управление привилегиями: Эти расширения сертификата сообщают информацию, связанную с предъявлением привилегий;

b) аннулирование привилегии: Эти расширения сертификата сообщают информацию, связанную с местонахождением информации о статусе аннулирования;

c) источник полномочий: Эти расширения сертификата связывают с достоверным источником привилегий, назначаемым верификатором для данного ресурса;

d) роли: Эти расширения сертификата сообщают информацию, касающуюся местонахождения, связанных сертификатов определения роли;

e) делегирование: Эти расширения сертификата позволяют ставить ограничения на последующее делегирование назначенных привилегий.

Следующие требования связаны с базовым управлением привилегиями:

a) издателям необходимо иметь возможность устанавливать условные ограничения на время, в течение которого привилегия может быть предъявлена;

b) издателю необходимо иметь возможность направлять сертификаты атрибута на специфический сервер/сервис;

c) возможно, издателям будет необходимо сообщать информацию, предназначенную чтобы показать назначителю привилегий и/или верификатору привилегий, используя сертификат;

d) издателям, возможно, будет необходимо иметь возможность ставить ограничивающие условия на политику привилегии, вместе с которой может быть использована назначенная привилегия.

13.1.2 Поля базового расширения управления привилегиями

13.1.2.1 Расширение спецификации времени

Определены следующие поля расширения:

a) спецификация времени;

b) намечаемая информация;

c) извещение пользователя;

d) допустимые политики привилегий.

Расширение спецификации времени может быть использовано органом по присвоению атрибутов для того, чтобы ограничить определенный период времени, в течение которого привилегия, назначенная в сертификате, содержащем это расширение, может быть назначена владельцем привилегий. Например, орган по присвоению атрибутов может издать сертификат, назначив привилегии, которые могут быть предоставлены только с понедельника до пятницы с 9:00 до 17:00. Это поле определяется следующим образом:

timeSpecification EXTENSION ::= {
SYNTAX TimeSpecification
IDENTIFIED BY id-ce-timeSpecification }

Это расширение может быть представлено в сертификатах атрибута или в сертификатах открытого ключа, изданных органом по присвоению атрибутов, включающий источник полномочий, для объектов, которые могут действовать, как назначители привилегий. Это расширение не должно быть включено в сертификаты, которые содержат расширение идентификатора источника полномочий, или в сертификаты, изданные для органа по присвоению атрибутов, которые не могут действовать так же, как назначители привилегий.

Если это расширение представлено в сертификате, изданном для объекта, который является органом по присвоению атрибутов, то он применяет только те утверждения объекта привилегий, которые содержатся в сертификате.

Из-за того, что данное расширение точно определяет период действительности сертификата, который содержит это расширение, данное расширение должно быть отмечено как критическое (т.е. издатель, посредством включения этого расширения, точно определяет не действительность назначения привилегии за пределами указанного времени).

Если это расширение существует, но не понимается верификатором привилегий, то сертификат будет отвергнут.

13.1.2.2 Соответствие спецификации времени

Правила согласования спецификации времени сравнивает равенство представленного значения со значением атрибута типа AttributeCertificate.

timeSpecificationMatch MATCHING-RULE ::= {
SYNTAX TimeSpecification
ID id-mr-timeSpecMatch }

Это правило согласования возвращает значение TRUE, если хранимое значение содержит расширение timeSpecification и если компоненты, которые даны в представленном значении, совпадают с соответствующими компонентами хранимого значения.

13.1.2.3 Расширение намечаемой информации

Расширение намечаемой информации дает возможность определять сертификат атрибута на специфический набор серверов/сервисов. Сертификат атрибута, содержащий это расширение, должен быть применим только в этих, точно указанных серверах/сервисах. Это поле определяется следующим образом:

targetingInformation EXTENSION ::= {
SYNTAX SEQUENCE SIZE (1..MAX) OF Targets
IDENTIFIED BY id-ce-targetInformation }
Targets ::= SEQUENCE SIZE (1..MAX) OF Target
Target : := CHOICE {
targetName [0] GeneralName,
targetGroup [1] GeneralName,
targetCert [2] TargetCert }
TargetCert ::= SEQUENCE {
targetCertificate IssuerSerial,
targetName GeneralName OPTIONAL,
certDigestInfo ObjectDigestInfo OPTIONAL }

Компонента targetName, если она существует, представляет название целевых серверов/сервисов, для которых предназначен содержащийся сертификат атрибута.

Компонента targetGroup, если она существует, то она предоставляет название целевой группы, для которой предназначен содержащийся сертификат атрибута. Определение количества целевых членов в пределах компоненты targetGroup выходит за области рассмотрения данного стандарта.

Компонента targetCert, если существует, то она идентифицирует целевые серверы/сервисы, ссылаясь на их сертификат.

Это расширение может присутствовать в сертификатах атрибута, изданных органом по присвоению атрибутов, включающих источник полномочий, для объектов, которые могут действовать в качестве назначителя привилегий. Это расширение не должно быть включено в сертификаты открытого ключа или сертификаты атрибута, изданных для органа по присвоению атрибутов, которые не могут действовать также как назначитель привилегий.

Если это расширение присутствует в сертификате атрибута, изданном для объекта, который является органом по присвоению атрибутов, то он применяется только на те утверждения объекта привилегий, которые содержатся в этом сертификате. Это не влияет на возможности органа по присвоению атрибутов издавать сертификаты. Это расширение всегда является критическим. Если это расширение присутствует, но верификатор привилегий не является одним из тех, которые определены, то сертификат атрибута должен быть отвергнут. Если это расширение не присутствует, то сертификат атрибута считается не целевым и может быть признан любым сервером.

13.1.2.4 Расширение извещения пользователя

Расширение извещения пользователя позволяет органу по присвоению атрибутов включить извещение, которое должно демонстрироваться владельцам, когда предъявляются их привилегии, и/или верификатору привилегий при использовании сертификата атрибута, содержащего это расширение. Это поле определяется следующим образом:

userNotice EXTENSION ::= {
SYNTAX SEQUENCE SIZE (1..MAX) OF UserNotice
IDENTIFIED BY id-ce-userNotice }

Это расширение может присутствовать в сертификатах атрибута или в сертификатах открытого ключа, изданных органом по присвоению атрибутов, включающих источник полномочий, для объектов, которые могут действовать в качестве назначителя привилегий, включая другие органы по присвоению атрибутов и конечные объекты. Это расширение не должно быть включено в сертификаты, которые содержат расширения идентификатора источника полномочий, или в сертификаты, изданные для органа по присвоению атрибутов, которые не могут действовать также как назначитель привилегий. Если это расширение присутствует в сертификате атрибута, изданном для объекта, который является органом по присвоению атрибутов, то он применяет только те утверждения объекта привилегий, которые содержатся в сертификате. Это не влияет на возможности органа по присвоению атрибутов издавать сертификаты. Это расширение может по выбору издателя сертификата быть либо критическим или

некритическим. Если это расширение отмечено как критическое, извещения о пользователе будут представлены верификатору привилегий каждый раз при предъявлении привилегии. Если назначитель привилегий предоставляет сертификат атрибута верификатору привилегий (т.е. верификатор привилегий не получает его самостоятельно напрямую из хранилища), то извещение о пользователе также должно быть представлено назначителю привилегий.

Если это расширение отмечено как некритическое, то привилегия, утвержденная в сертификате, может быть предоставлена верификатором привилегий, несмотря на то, что были показаны извещения о пользователе назначителю привилегий и/или верификатору привилегий.

13.1.2.5 Расширение допустимых политик привилегий

Поле допустимых политик привилегий используется для ограничения утверждения назначенных привилегий для использования вместе с определенным набором политик привилегий. Это поле определяется следующим образом:

acceptablePrivilegePolicies EXTENSION ::= {
SYNTAX AcceptablePrivilegePoliciesSyntax
IDENTIFIED BY id-ce-acceptablePrivilegePolicies }
AcceptablePrivilegePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF
PrivilegePolicy

Это расширение может быть представлено в сертификатах атрибута или в сертификатах открытого ключа, изданных органом по присвоению атрибутов, включающих источник полномочий, для других органов по присвоению атрибутов или для конечных объектов.

Если, это расширение содержится в сертификате открытого ключа, то оно связано только со способностями субъекта действовать, как назначитель привилегий, согласно привилегиям содержащимся в расширении subjectDirectoryAttributes.

В случае присутствия это расширение должно быть помечено как критическое.

Если это расширение присутствует, и верификатор привилегий признает его, то верификатор должен гарантировать, что политика привилегий, с которой эти привилегии сравниваются, является одной из тех, которые установлены в этом расширении.

Если это расширение присутствует, но оно не понятно верификатору привилегий, то сертификат будет игнорирован.

13.2 Расширения аннулирования привилегий

13.2.1 Необходимые условия

Следующие условия касаются аннулирования сертификатов атрибута:

a) для того чтобы контролировать размеры CRL, необходимо назначить поднабор набора всех сертификатов, изданных одним органом по присвоению атрибутов для различных CRL;

b) издателям сертификата атрибута необходимо иметь возможность указать в сертификате атрибута, что никакая информация об аннулировании непригодна для этого сертификата.

13.2.2 Поля расширения аннулирования привилегий

13.2.2.1 Расширение пункта распределения CRL

Определены следующие поля расширения:

a) пункты распределения CRL;

b) расширение о не существовании информации об аннулировании.

Расширение пункта распределения CRL определено в разделах 5, 6, 7, 8 и 9 данного стандарта для использования в сертификатах открытого ключа. Это поле также может быть включено в сертификат атрибута. Оно может присутствовать в сертификатах, изданных для органа по присвоению атрибутов, включающих источник полномочий, также как и в сертификатах, изданных для конечных объектов.

Если это расширение присутствует в сертификате, верификатор привилегий должен обработать это расширение точно таким же образом, как это описано в разделах 5, 6, 7, 8 и 9 для сертификатов открытого ключа.

13.2.2.2 Расширение о несуществовании информации об аннулировании

В некоторых средах (например, когда сертификаты атрибута изданы с очень маленьким периодом действия), не будет необходимости аннулировать сертификаты атрибута. Орган по присвоению атрибутов может использовать это расширение для того, чтобы указать, что не предоставляется информация о статусе аннулирования этого сертификата атрибута. Это поле определяется следующим образом:

noRevAvail EXTENSION ::= {
SYNTAX NULL
IDENTIFIED BY id-ce-noRevAvail }

Это расширение может присутствовать в сертификатах атрибута, изданных органом по присвоению атрибутов, включающих источник полномочий, для конечных объектов. Это расширение не должно включаться в сертификаты открытого ключа или в сертификаты атрибута, изданные для органа по присвоению атрибутов. Это расширение всегда является некритическим.

Если это расширение присутствует в сертификате атрибута, верификатору привилегии нет необходимости искать информацию о статусе аннулирования.

13.3 Расширения источника полномочий

13.3.1 Необходимые условия

К источнику полномочий предъявляются следующие условия:

a) в некоторых средах существует необходимость для тесного контроля со стороны ЦР объектов, которые могут действовать как источник полномочий;

b) существует необходимость выполнения обоснованного определения синтаксиса и преобладание правил для атрибутов привилегий, которые имеются в распоряжении ответственных источников полномочий.

13.3.2 Поля расширения источника полномочий

13.3.2.1 Расширение идентификатора источника полномочий

Определены следующие поля расширения:

a) идентификатор источника полномочий;

b) дескриптор атрибута.

Расширение идентификатора источника полномочий указывает, что субъект сертификата может действовать как источник полномочий, в целях управления привилегиями. По существу, субъект сертификата может определять атрибуты, которые назначают привилегии, издавать сертификаты дескриптора атрибута для этих атрибутов и использовать закрытый ключ, соответствующий удостоверенному открытому ключу, для того, чтобы издавать сертификаты, которые назначают привилегию владельцу. Эти последующие сертификаты могут быть сертификатами атрибута или сертификатами открытого ключа с расширением subjcetDirectoryAttributes, содержащим эти привилегия.

В некоторых средах это расширение не требуется, а вместо этого могут быть использованы другие механизмы, чтобы определить то, что объекты могут действовать как источник полномочий. Это расширение требуется только в тех обстоятельствах, где требуется тесный централизованный контроль со стороны ЦР, для того чтобы управлять объектами, которые действуют как источник полномочий. Это поле определяется следующем образом:

sOAIdentifier EXTENSION ::= {
SYNTAX NULL
IDENTIFIED BY id-ce-sOAIdentifier }

Если это расширение не присутствует в сертификате, возможность действия субъекта/владельца в качестве источника полномочий определяется другими способами.

Это поле может присутствовать в сертификатах открытого ключа, изданных для источника полномочий. Оно не должно включатся в сертификаты атрибута или в сертификаты открытого ключа, изданных для других органом по присвоению атрибутов или для владельца привилегий конечного объекта. Это расширение всегда является некритическим.

13.3.2.2 Расширение дескриптора атрибута и соответствия дескриптора атрибута

Определение атрибута привилегии и доминирующих правил, контролирующих последующее делегирование этих привилегий, необходимы для верификаторов привилегий для того, чтобы гарантировать, что авторизация выполнена правильно. Эти определения и правила могут быть предоставлены верификатору привилегий различными путями, описание которых лежит за области рассмотрения данного стандарта.

Это расширение предоставляет механизм, который может быть использован со стороны источника полномочий для определения атрибутов привилегий и связанных контролирующих правил, доступных верификатору привилегий. Сертификат атрибута, содержащий такого рода расширения, называется сертификатом дескриптора атрибута, и он является специальным типом сертификата атрибута. Хотя синтаксически этот сертификат является идентичным к AttributeCertificate, но сертификат дескриптора атрибута отличается следующим:

- содержит пустое SEQUENCE в своем поле attributes;
- является сертификатом, изданным самому себе (т.е. издатель и владелец является одним и тем же объектом);
- включает расширение дескриптора атрибута.

Это поле определяется следующим образом:

```
attributeDescriptor EXTENSION ::= {
SYNTAX    AttributeDescriptorSyntax
IDENTIFIED BY   {id-ce-attributeDescriptor } }
AttributeDescriptorSyntax ::= SEQUENCE {
identifier    AttributeIdentifier,
attributeSyntax   OCTET STRING (SIZE(1..MAX)),
name    [0]   AttributeName OPTIONAL,
description   [1]   AttributeDescription OPTIONAL,
dominationRule   PrivilegePolicyIdentifier}
AttributeIdentifier ::= ATTRIBUTE.&id({AttributeIDs})
AttributeIDs ATTRIBUTE ::= {...}
AttributeName ::= UTF8String(SIZE(1..MAX))
AttributeDescription ::= UTF8String(SIZE(1..MAX))
PrivilegePolicyIdentifier ::= SEQUENCE {
privilegePolicy    PrivilegePolicy,
privPolSyntax    InfoSyntax }
```

Компонента identifier со значением, равным расширению attribteDescriptor, является идентификатором объекта, идентифицирующий тип атрибута.

Компонента attributeSyntax содержит ASN.1 описание синтаксиса атрибута.

O'ZSTANDART AGENTLIGI
STANDARTSHUNOSLIK,
METROLOGIYA VA
MUVOFIQLASHTIRISH
DAVLAT HAZORATI

Компонента name возможно (необязательно) будет содержать простое удобное для пользователя имя, по которому будет распознаваться атрибут.

Компонента description возможно (необязательно) будет содержать простое удобное для пользователя описание атрибута.

Компонента dominationRule устанавливает для атрибута, что он означает для делегирования привилегии, будучи «меньше чем» соответствующая привилегия, хранимая делегатором.

Компонента privilegePolicy идентифицирует (указывает) по своему идентификатору объекта экземпляр политики привилегии, которая содержит правила. Компонента privPolSyntax содержит либо саму политику привилегий, либо указатель на местонахождение, где он может быть расположен. Если включен указатель, то не обязательное (опциональное) хэш значение политики привилегий может также быть включено, для того чтобы позволить проверить на целостность по сосланной политике привилегии.

Это расширение может присутствовать только в сертификатах дескриптора атрибута.

Это расширение не будет присутствовать в сертификатах открытого ключа или в сертификатах атрибута, кроме тех, которые являются сертификатами источника полномочий, изданных самому себе. Это расширение всегда будет некритическим.

Правило соответствия дескриптора атрибута сравнивает на равенство представленное значение со значением атрибута типа AttributeCertificate.

attDescriptor MATCHING-RULE ::= {
SYNTAX AttributeDescriptorSyntax
ID id-mr-attDescriptorMatch }

Правило соответствия возвращает значение TRUE, если хранимое значение содержит в себе расширение attributeDescriptor и если присутствующие компоненты в представленном значении совпадают с аналогичными компонентами хранимого значения.

13.4 Расширения роли

13.4.1 Необходимые условия

Если сертификат является сертификатом назначения роли, верификатору привилегий необходимо иметь возможность определить место соответствующего сертификата определяющего роль, которая содержит специфические привилегии, назначенные для этой роли.

13.4.2 Поля расширения роли

13.4.2.1 Расширение идентификатора сертификата определяющего роль и правило соответствия идентификационного номера (ID) сертификата, определяющего роль

Расширение идентификатора сертификата, определяющего роль, может быть использовано органом по присвоению атрибутов в качестве указателя на сертификат, определяющего роль, который содержит назначение привилегии на роль. Оно может присутствовать в сертификате назначения роли (т.е. сертификат, который содержит атрибут role).

Когда верификатор привилегий имеет дело с сертификатом назначения роли, то ему необходимо получить набор привилегий этой роли, для того, чтобы определить результат верификации либо как успешно завершенный, либо неудачный. Если привилегии были назначены для роли в сертификате определения роли, то это поле может быть использовано для того, чтобы определить место этого сертификата. Это поле определяется следующим образом:

```
roleSpecCertIdentifier EXTENSION ::=
{
SYNTAX    RoleSpecCertIdentifierSyntax
IDENTIFIED BY   { id-ce-roleSpecCertIdentifier }
}
RoleSpecCertIdentifierSyntax ::= SEQUENCE SIZE (1..MAX) OF
RoleSpecCertIdentifier
RoleSpecCertIdentifier ::= SEQUENCE {
roleName      [0]   GeneralName,
roleCertIssuer      [1]   GeneralName,
roleCertSerialNumber   [2]   CertificateSerialNumber OPTIONAL,
roleCertLocator     [3]   GeneralNames OPTIONAL}
```

Поле roleName идентифицирует роль. Его имя должно быть таким же, как и в компоненте holder сертификата, определяющего роль.

roleCertIssuer идентифицирует орган по присвоению атрибутов, который издал сосланный сертификат определения роли.

roleCertSerialNumber, если существует, то он содержит серийный номер сертификата определения роли. Нужно заметить, что если изменяются сами привилегии, назначенные на роль, то новый сертификат, определяющий роль, должен быть издан для этой роли. Любые сертификаты, содержащие это расширение, включая компоненту roleCertSerialNumber, должны быть заменены сертификатами, на которые ссылается новый серийный номер.

roleCertLocator, если существует, то содержит информацию, которая может быть использована для определения места сертификата определения роли.

Это расширение, может присутствовать в сертификатах назначения роли, которые являются сертификатами атрибута или сертификатами

MUVOFIQLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

открытого ключа, изданными со стороны органа по присвоению атрибутов, включающих источник полномочий, для других органов по присвоению атрибутов или владельцем привилегий конечного объекта. Это расширение не должно включаться в сертификаты, которые содержат расширение идентификатора источника полномочий.

При присутствии этого расширения, его можно использовать верификатором привилегий, для того, чтобы определить место сертификата определения роли.

Если это расширение не присутствует:

a) либо другое средство будет использовано для того, чтобы определить место сертификата определения роли;

b) либо для того, чтобы назначить привилегии для роли используются другие механизмы, отличающиеся от сертификата определения роли (т.е. привилегии роли могут быть локально сконфигурированы на стороне верификатора привилегий).

Это расширение всегда является некритическим.

Правило соответствия идентификатора сертификата, определяющего роль, сравнивает на равенство представленное значение со значением атрибута типа AttributeCertificate.

roleSpecCertIdMatch MATCHING-RULE ::= {
SYNTAX RoleSpecCertIdentifierSyntax
ID id-mr-roleSpecCertIdMatch }

Это правило соответствия возвращает значение true, если хранимое значение содержит расширение roleSpecCertIdentifier и если компоненты, которые присутствуют в представленном значении, соответствуют аналогичным компонентам хранимых значений.

13.5 Расширение делегирования

13.5.1 Необходимые условия

Следующие условия относятся к делегированию привилегий:

a) сертификаты привилегий конечного объекта необходимы, чтобы отличить от сертификатов органов по присвоению атрибутов, для того, чтобы защититься от того, чтобы конечные объекты не устанавливали себя как орган по присвоению атрибутов без авторизации. Это также необходимо для органа по присвоению атрибутов, чтобы он имел возможность установить лимит на длину пути последующего делегирования;

b) орган по присвоению атрибутов должен иметь возможность указать подходящее пространство имен, внутри которого может произойти делегирование привилегий. Строгое соблюдение этих ограничений требуется, чтобы быть контролепригодным со стороны верификатора привилегий;

c) орган по присвоению атрибутов должен иметь возможность указать допустимые политики сертификата, которые предъявители

привилегий передают дальше «вниз» по пути делегирования и которые будут использоваться для аутентификации их самих при предъявлении делегирования привилегий этим органом по присвоению атрибутов;

d) верификатор привилегий должен иметь возможность определения места, соответствующего сертификата атрибута для издателя, для того чтобы гарантировать, что издатель имел достаточные привилегии для делегирования привилегий в текущем сертификате.

13.5.2 Поля расширения делегирования

13.5.2.1 Расширение ограничения базового атрибута и правило соответствия ограничений базового атрибута

Определены следующие поля расширения:

a) ограничения базового атрибута;

b) ограничения делегированных имен;

c) допустимая политика сертификата;

d) идентификатор атрибута органа.

Расширение ограничения базового атрибута указывает на то, что позволено ли последующее делегирование привилегий, назначенных в сертификате, содержащем это расширение. Если так, то можно указать ограничение длины пути делегирования.

Это поле определяется следующим образом:

```
basicAttConstraints EXTENSION ::=
{
SYNTAX    BasicAttConstraintsSyntax
IDENTIFIED BY   { id-ce-basicAttConstraints }
}
BasicAttConstraintsSyntax ::= SEQUENCE
{
authority    BOOLEAN DEFAULT FALSE,
pathLenConstraint   INTEGER (0..MAX) OPTIONAL}
```

Компонента authority указывает, разрешено ли хранителю дальнейшее делегирование привилегий. Если значение authority равно TRUE, то хранителем является орган по присвоению атрибутов и ему разрешается дальнейшее делегирование привилегий, в зависимости от релевантных ограничений. Если значение authority равняется FALSE, то хранителем является конечный объект и ему не разрешено дальнейшее делегирование привилегии.

Компонента pathLenConstraint имеет значение, только если значение компоненты authority равно TRUE. Она указывает на максимальное количество сертификатов органа по присвоению атрибутов, за которыми может проследовать этот сертификат по пути делегирования. Значение 0 указывает, что субъект этого сертификата может издавать сертификаты только для конечных объектов, но не для органа по присвоению атрибутов.

Если в сертификате делегирования пути не встречается поле pathLenConstraint, то это означает, что нет ограничения на дозволенную длину пути делегирования. Нужно заметить, что ограничение вступает в силу, начиная со следующего сертификата в пути. Это ограничение контролирует количество сертификатов органа по присвоению атрибутов между сертификатом органа по присвоению атрибутов, содержащим это ограничение и сертификатом конечного объекта.

Это расширение может присутствовать в сертификатах атрибута или в сертификате открытого ключа, изданных органом по присвоению атрибутов, включающим источник полномочий, для других органов по присвоению атрибутов или конечных объектов. Это расширение не должно включаться в сертификаты, которые содержат расширение идентификатора источника полномочий.

Если это расширение присутствует в сертификате атрибута, но не в сертификаты открытого ключа, и значение authority равняется TRUE, то владельцу разрешается издавать последующие сертификаты атрибута, делегирующие содержащиеся привилегии для других объектов.

Если это расширение присутствует в сертификате открытого ключа и если расширение basicConstraints указывает на то, что субъект также является ЦР, то субъекту разрешается издавать последующие сертификаты открытого ключа, которые делегируют эти привилегии другим объектам, но не сертификаты атрибута. Если ограничение длины пути включено, то субъект может делегировать в пределах пересечения ограничений, указанных в этом расширении, а также любых ограничений, указанных в расширении basicConstraints. Если это расширение присутствует в сертификате открытого ключа, но в нём отсутствует расширение basicConstraints, или указывает, что субъект является конечным объектом, то субъекту запрещается делегировать привилегии.

Это расширение может быть по выбору издателя сертификата либо критическим, либо некритическим. Рекомендуется, чтобы это расширение было отмечено как критическое, в противном случае владелец, которому не разрешено быть органом по присвоению атрибутов, может издать сертификаты и верификатор привилегий, может непреднамеренно использовать такой сертификат.

Если это расширение присутствует и отмечено как критическое, тогда:

- если значение authority не установлено на TRUE, то делегированный атрибут не будет использоваться для дальнейшего делегирования;

- если значение authority установлено на true и присутствует pathLenConstraint, то верификатор привилегий будет проверять этот обрабатываемый путь делегирования, который является совместимым со значением pathLenConstraint.

Если это расширение присутствует и отмечено как некритическое, и не опознано со стороны верификатора привилегий, то эта система должна использовать другие средства для определения можно ли использовать делегированный атрибут для дальнейшего делегирования.

Если расширение не присутствует, или это расширение присутствует с пустым значением SEQUENCE, то владелец ограничивается только, чтобы быть конечным объектом, а не органом по присвоению атрибутов, и не имеет разрешения делегирования привилегий, содержащихся в сертификате атрибута.

Правило соответствия ограничений базового атрибута сравнивает на равенство представленное значение со значением атрибута типа AttributeCertificate.

basicAttConstraintsMatch MATCHING-RULE ::= {
SYNTAX BasicAttConstraintsSyntax
ID id-mr-basicAttConstraintsMatch }

Это правило соответствия возвращает значение true, если хранимые значения содержат расширение BasicAttConstraints, и если компоненты, которые существуют в представленном значении, соответствуют аналогичным компонентам хранимых значений.

13.5.2.2 Расширение ограничения делегированного имени и правило соответствия ограничения делегированного имени

Поле ограничений делегированного имени указывает на пространство имен, внутри которого должны быть расположены все имена держателей последующих сертификатов, указанных в пути делегирования. Это поле определяется следующим образом:

delegatedNameConstraints EXTENSION ::= {
SYNTAX NameConstraintsSyntax
IDENTIFIED BY id-ce-delegatedNameConstraints }

Это расширение обрабатывается таким же образом, как и расширение nameConstraints для сертификатов открытого ключа. Если присутствует permittedSubtrees, то из всех сертификатов атрибута, изданных со стороны держателя органа по присвоению атрибутов и последующих держателей органа по присвоению атрибутов, находящихся в пути делегирования, допустимы только те сертификаты атрибута в пределах этого поддерева, которые с именем держателя. Если существует excludedSubtrees, то любой сертификат атрибута, изданный со стороны держателя органа по присвоению атрибутов или последующего органа по присвоению атрибутов, находящегося в пути делегирования, который имеет имя держателя внутри этих поддеревьев, считается недопустимым. Если оба и permittedSubtrees и excludedSubtrees присутствуют и пространства имен как-то перекрывают друг друга, то сначала выполняется оператор исключения этих двух пространств (результат оператора исключения этих пространств имеет превосходство).

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

Это расширение может быть представлено в сертификатах атрибута или в сертификатах открытого ключа, изданного со стороны органа по присвоению атрибутов, включая источник полномочий, для других органов по присвоению атрибутов. Это расширение не должно включаться в сертификаты, изданные для конечных объектов или в сертификаты, которые содержат расширение идентификатора источника полномочий.

Если это расширение присутствует в сертификате открытого ключа, и также присутствует расширение nameConstraints, то субъект может делегировать только в пределах пересечения ограничений, указанного в этом расширении и указанного в расширении nameConstraints.

Это расширение может быть по выбору издателя сертификата атрибута либо критическим или некритическим. Рекомендуется, чтобы это расширение отмечалось как критическое, в противном случае издатель сертификата атрибута может не проверить те последующие сертификаты атрибута, находящиеся на пути делегирования, которые помещены в пространстве имен, подразумеваемых издающим органом по присвоению атрибутов.

Правило соответствия ограничений делегированного имени сравнивает на равенство представленное значение со значением атрибута типа AttributeCertificate.

delegatedNameConstraintsMatch MATCHING-RULE ::= {
SYNTAX NameConstraintsSyntax
ID id-mr-delegatedNameConstraintsMatch}

Это правило соответствия возвращает значение TRUE, если хранимое значение содержит расширение attributeNameConstratints и если компоненты, которые присутствуют в представленном значении, соответствуют аналогичным компонентам хранимого значения.

13.5.2.3 Расширения допустимой политики сертификата и правило соответствия допустимой политики сертификата

Поле допустимой политики сертификата используется при делегировании сертификатов атрибута, для того, чтобы контролировать допустимую политику сертификата, согласно которой сертификаты открытого ключа для последующих владельцев, находящихся в пути делегирования, были изданными. Перечисляя набор политики в этом поле, орган по присвоению атрибутов требует, чтобы последующие издатели, находящиеся в пути делегирования, делегировали только содержащиеся привилегии владельцам, которые имеют сертификаты открытого ключа, изданные согласно одному или нескольким перечисленным политикам сертификата. Политики, перечисленные здесь, не являются теми политиками, согласно которым сертификат атрибута был издан, а являются политиками, согласно которым должны быть изданы допустимые

сертификаты открытого ключа для последующих владельцев. Это поле определяется следующим образом:

acceptableCertPolicies EXTENSION ::= {
SYNTAX AcceptableCertPoliciesSyntax
IDENTIFIED BY id-ce-acceptableCertPolicies }
AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId
CertPolicyId ::= OBJECT IDENTIFIER

Это расширение может присутствовать только в сертификатах атрибута, изданных органом по присвоению атрибутов, включающих источник полномочий, для других органов по присвоению атрибутов. Это расширение не должно включаться в сертификаты атрибута конечного объекта или в любой сертификат открытого ключа. В случае делегирования, используя сертификаты открытого ключа, подобную функциональность можно предоставить с помощью cetificatePolicies и другими подобными расширениями.

В случае присутствия, это расширение должно быть помечено как критическое.

Если это расширение присутствует, и верификатор привилегий может понять его, то верификатор гарантирует, что все последующие назначители привилегий, находящиеся в пути делегирования, удостоверены сертификатом открытого ключа согласно одной или нескольким перечисленным политикам сертификата.

Если это расширение присутствует, но оно не понятно верификатору привилегий, то сертификат должен быть отвергнут.

Правило соответствия допустимой политики сертификата сравнивает на равенство представленное значение со значением атрибута типа AttributeCertificate.

acceptableCertPoliciesMatch MATCHING-RULE ::= {
SYNTAX AcceptableCertPoliciesSyntax
ID id-mr-acceptableCertPoliciesMatch }

ALOQA AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA
VA DAVLAT NAZORATI

Это правило соответствия возвращает значение TRUE, если хранимое значение содержит расширение acceptableCertPolicies и если компоненты, которые существуют в представленном значении, соответствуют аналогичным компонентам хранимого значения.

13.5.2.4 Расширение идентификатора органа по присвоению атрибутов и правило соответствия идентификатора органа по присвоению атрибутов

При делегировании привилегий орган по присвоению атрибутов, который делегирует привилегии, должен иметь хотя бы минимальные

привилегии и полномочия для делегирования таких привилегий. Орган по присвоению атрибутов, делегирующий привилегии другому органу по присвоению атрибутов, или для конечного объекта, может поместить это расширение в сертификат органа по присвоению атрибута или конечного объекта, которым он издаёт сертификат. Это расширение является указателем на сертификат, в котором издатель сертификата, содержащий это расширение, назначил его соответствующие привилегии. Это расширение может быть использовано со стороны верификатора привилегий, чтобы гарантировать, что издающий орган по присвоению атрибутов имел достаточную привилегию, для того чтобы иметь возможность делегировать держателю сертификата, который содержит это расширение. Это поле определяется следующим образом:

authorityAttributeIdentifier EXTENSION ::=
{
SYNTAX AuthorityAttributeIdentifierSyntax
IDENTIFIED BY { id-ce-authorityAttributeIdentifier }
}
AuthorityAttributeIdentifierSyntax ::= SEQUENCE SIZE (1..MAX) OF AuthAttId
AuthAttId ::= IssuerSerial

Сертификат, содержащий это расширение, может включать делегирование множества привилегий владельцу сертификата. Если назначение тех привилегий к органу по присвоению атрибутов, который издаёт этот сертификат, были назначены более чем одним сертификатом, то это расширение должно включать более чем один указатель.

Это расширение может присутствовать в сертификатах атрибута или в сертификатах открытого ключа, изданных органом по присвоению атрибутов для других органов по присвоению атрибутов или для владельцев привилегий конечных объектов. Это расширение не должно быть включено в сертификаты, изданные со стороны источника полномочий или в сертификаты открытого ключа, которые содержат расширение идентификатора источника полномочий. Это расширение всегда является некритическим.

Правило соответствия идентификатора органа по распространению атрибутов сравнивает на равенство представленное значение со значением атрибута типа AttributeCertificate.

authAttIdMatch MATCHING-RULE ::= {
SYNTAX AuthorityAttirbuteIdentifierSyntax
ID id-mr-authAttIdMatch }

Это правило соответствия возвращает значение true, если хранимое значение содержит расширение authorityAttributeIdentifier и если

компоненты, которые присутствуют в представленном значении, соответствуют аналогичным компонентам хранимого значения.

13.6 Признание полномочий расширений

13.6.1 Требования

Признание полномочий характеризуется следующими требованиями:

a) локальный SOA должен знать, как атрибуты, назначенные в удаленного домена, отображены в существующей роли в локальном домене;

b) локальный SOA должен ограничить атрибуты привилегии удаленному SOA для назначенных пользователей;

c) локальный SOA должен ограничивать формы названия и места, в пределах которых удаленный SOA может назначить атрибуты привилегии пользователям.

13.6.2 Поля расширения RoA

13.6.2.1 Расширение допустимого назначения атрибута

Расширение допустимого назначения атрибута описывается следующим образом:

```
allowedAttributeAssignments EXTENSION ::= {
SYNTAX AllowedAttributeAssignments
IDENTIFIED BY id-ce-allowedAttAss }

AllowedAttributeAssignments ::= SET OF SEQUENCE {
attributes [0] SET OF CHOICE {
 attributeType [0] AttributeType,
 attributeTypeandValues [1] Attribute{{SupportedAttributes}} },
holderDomain [1] GeneralName }
```

Каждое допустимое назначение атрибута включает в себя множество типов атрибутов вместе с названием расширения, которое определяет владельца домена.

Формы названия, определяемые типом GeneralName должны иметь четкую иерархическую структуру.

Все допустимые атрибуты, определенные в этом расширении должны также быть определены в компоненте атрибутов сертификата атрибута.

Если атрибут определен в этом расширении, но не находится в компоненте атрибутов, то он игнорируется. Если атрибут находится в компоненте атрибутов, но не в этом расширении, то ему доверяют и используют.

13.6.2.2 Расширение картографий атрибута

Расширение картографий атрибута представляется следующим образом:

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
BO‘LIMLARAGA

```
attributeMappings EXTENSION ::= {
 SYNTAX  AttributeMappings
 IDENTIFIED BY  id-ce-attributeMappings }

AttributeMappings ::=  SET OF CHOICE {
 typeMappings  [0] SEQUENCE {
  local   [0] AttributeType,
  remote  [1] AttributeType},
 typeValueMappings [1] SEQUENCE {
  local   [0] AttributeTypeAndValue,
  remote  [1] AttributeTypeAndValue} }
```

13.6.2.3 Ограничения расширения «Наименование владельца»

Это расширение ограничивает формы названия, для которых удаленный SOA и его зависимый AA могут выпустить CA.

В этом расширении указываются формы названия CA, выпускаемые всеми AA в цепочке CA.

Если это расширение присутствует в сертификате CA, то ограничения автоматически помещаются в места для названия каждой формы названия в цепочке CA.

Это расширение определяется следующим образом:

```
holderNameConstraints  EXTENSION ::= {
 SYNTAX  HolderNameConstraintsSyntax
 IDENTIFIED BY  id-ce-holderNameConstraints }

HolderNameConstraintsSyntax ::= SEQUENCE {
 permittedSubtrees [0] GeneralSubtrees,
 excludedSubtrees [1] GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
 base   GeneralName,
 minimum  [0] BaseDistance DEFAULT 0,
 maximum  [1] BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)
```

Компоненты permittedSubtrees и excludedSubtrees определяют одно или более поддеревьев для обозначения одной или нескольких форм названия. Каждое поддерево определяется названием корня поддерева, то есть компонентой основы, и в пределах того поддерева, домен которого ограничен верхними или нижними уровнями.

Минимальный компонент определяет верхнюю границу домена в поддереве.

Значение минимума по умолчанию равно нулю и соответствует основе, то есть главному узлу поддерева.

Максимальный компонент определяет нижние связи домена в пределах поддерева. Вес максимума равный нулю соответствует началу поддерева. Отсутствующий максимальный компонент указывает, что никакой более низкий предел не должен быть наложен на домен в пределах поддерева.

Компонент permittedSubtrees используется, чтобы уменьшить ограничения, установленные для форм названия.

Дополнительный компонент excludedSubtrees используется, чтобы исключить одно или более зависимых поддеревьев из permittedSubtrees.

Все названия владельцев CA в цепочке владельцев сертификата должны быть расположены в разрешенных местах для названия сертификата, чтобы быть приемлемыми.

Если владелец сертификата имеет несколько названий одного и того же типа имени, то все такие имена должны быть расположены в разрешенном пространстве имен для того, чтобы имя формы сертификата было приемлемым.

Когда владелец сертификата имеет несколько названий различных типов имени, каждое название должно быть расположено в разрешенном месте названия формы названия для сертификата.

Формы названия должны быть доступными через тип GeneralName и иметь четкую иерархическую структуру.

CA считается подчиненным базовому, если последовательность RDN, которая формирует полный DN в базе, соответствует исходной последовательности одного и того же числа RDN, которое образует первую часть DN владельца CA.

DN владельца сертификата может иметь дополнительное число RDN в своей последовательности, которые встречаются в базовом DN.

DistinguishedNameMatch используется для сравнения значения базы с начальной последовательностью RDN в DN субъекта сертификата.

Если CA не признает форму названия, используемую в каком-нибудь компоненте основы, то этот компонент должен быть обработан, как компонент с неизвестным расширением.

Если CA, не признает форму названия, которая используется в поле владельца последующего CA в цепи после основного и эта форма названия не относится ни к какому компоненту основы, то этот CA должен быть отклонен.

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
BOSHQARMASI

14 Процедура обработки пути привилегий

14.1 Процедура простой обработки

Обработка пути привилегий завершается верификатором привилегий. Правила обработки пути сертификатов атрибута, в некоторой степени, аналогичны правилам обработки пути сертификатов открытого ключа.

Для путей привилегий, состоящих из одного сертификата, т.е. привилегий, назначенных непосредственно назначителю привилегий источником полномочий, пока привилегия не будет назначена для роли, требуется только простая процедура, описанная в данном подразделе. В этом случае, если верификатор привилегии не сконфигурирован со специфическими привилегиями роли, может быть необходимым получение сертификата определения роли, который назначает специфические привилегии ролям, как описано в 14.2.

Если эти привилегии даны назначителю привилегий промежуточным органом по присвоению атрибутов, тогда требуется процедура пути делегирования, описанная в 14.3.

Эта процедура не выполняется последовательно. Процедура обработки пути и процедура обработки делегирования, прежде всего, нужна для того, чтобы узнать предъявлены ли достаточные привилегии или нет для контекста использования в простой процедуре.

Подпись на всех сертификатах в пути должна быть проверена. Верификатор привилегий проверяет идентичность каждого объекта в пути, используя процедуры, приведенные в разделе 7. Заметим, что проверка подписи на сертификате атрибута обязательно включает проверку сосланного сертификата открытого ключа на его действительность. Там где привилегии назначаются, используя сертификат атрибута, средства обработки пути необходимы, для того чтобы принимать во внимание элементы и PMI и PKI в ходе определения окончательного утверждения утвердителя привилегии сертификата атрибута. Однажды подтвержденная действительность привилегии, содержащейся в сертификате, может быть использована в зависимости от сравнения релевантной политики привилегий и другой информации, связанной с контекстом, в котором сертификат используется.

Контекст использования устанавливается, если владелец привилегии действительно предназначен для назначения содержащихся привилегий для использования в этом контексте. Цепь сертификатов в доверенном источнике полномочий не достаточна для того, чтобы сделать это определение. Готовность хозяина привилегий чтобы использовать такой сертификат должна быть точно указана и проверена. Однако, механизмы гарантирования того, что такое назначение привилегии адекватно продемонстрировано владельцем привилегий выходит за пределы рассмотрения данного стандарта. В качестве примера такого утверждения

привилегии может быть верифицируемость, если владелец привилегии подписал привилегии для данного сертификата, таким образом, указывается их готовность для использования этого сертификата в этом контексте.

Для каждого сертификата атрибута в пути, который не содержит расширение noRevAvail, верификатор привилегий должен гарантировать, что сертификат атрибута неаннулирован.

Верификатор привилегий должен гарантировать, что выданные привилегии действительны в течении времени называемого «время оценки». В контексте услуги управления доступом проверка всегда делается для настоящего времени. Когда сертификат утвержден, верификатор привилегий должен гарантировать, что время оценки истекло не позднее всего периода действительности всех сертификатов, используемых в пути. Также, если любые из сертификатов в пути содержат расширение timeSpecification, ограничения, расположенные на протяжении времени привилегии, могут быть назначены также, чтобы позволить утверждению привилегии быть действительным ко времени оценки.

Если расширение targetingInformation представлено в сертификате и используется для утверждения привилегии, то верификатор привилегий должен проверить, что сервер/сервис, для которого он верифицируется, включен в список целей.

Если сертификат является сертификатом назначения роли, то процедура обработки роли, описанная в 14.2, необходима для гарантирования того, что соответствующие привилегии идентифицируются. Если привилегия делегирована объекту, а не назначена непосредственно источником полномочий, утвержденным верификатором привилегий, то процедура обработки делегирования, описанная в 14.3, необходима для того, чтобы гарантировать, что делегирование выполнено правильно.

Верификатор привилегий также нужен для определения того, что достаточны или нет привилегии, утвержденные для контекста использования. Политика привилегий устанавливает правила для создания этого определения и включения спецификации какой-либо переменной среды, которую необходимо учитывать. Утвержденные привилегии, включающие результаты из процедуры роли в 14.2 и процедуры делегирования 14.3 и любые переменные среды (например, время дня или баланс текущего счета) сравниваются с политикой привилегий для определения достаточны они или нет для контекста использования. Если представлено расширение acceptablePrivilegePolicies, утверждение привилегии может удастся, только если политика привилегий верификатора привилегий сравнивается с одной из тех политик привилегий, которая включена в это расширение.

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA HAMDA
DAVLAT NAZORATI

14.2 Процедура обработки роли

Если утвержденный сертификат это сертификат утверждения роли, то верификатор привилегий должен получить специфические привилегии, назначенные для этой роли.

Название роли, для которой предназначен назначитель привилегий, содержится в атрибуте сертификата role. Верификатор привилегий, если не сконфигурирован с привилегиями назначенных ролей, может быть необходим для установления сертификата определения роли, который назначает привилегии для этой роли. Информация в атрибуте role в расширении roleSpecCertIdentifier может быть использована для установления этого сертификата.

Привилегии, назначенные для роли косвенным образом, предназначены для назначителя привилегий и поэтому включены между утвержденными привилегиями, которые сравниваются с политикой привилегий в базовой процедуре описанной в 14.1 для того, чтобы определить достаточны или нет назначенные привилегии для контекста использования.

14.3 Процедура обработки делегирования

14.3.1 Проверка целостности доминирующего правила

Если назначенные привилегии делегированы назначителю привилегий промежуточным органом по присвоению атрибутов, то верификатор привилегий должен гарантировать, что путь это действительный путь делегирования, гарантирующий что:

- каждый орган по присвоению атрибутов, который издал сертификат в пути делегирования, был авторизован, чтобы делать это;
- каждый сертификат в пути делегирования действителен по отношению к пути и ограничений названия, наложенных на него;
- каждый объект в пути делегирования удостоверяется сертификатом открытого ключа, который действителен согласно с любым наложенным ограничением политики;
- нет более значительной привилегии делегирования, чем привилегии, имеющиеся у органа по присвоению атрибутов.

До начала утверждения, пути делегирования верификатор привилегий получает следующим образом. Любой из нижеприведенных может быть предоставлен назначителю привилегий или получен верификатором привилегий из какого-нибудь другого источника, такого как директория. Атрибуты сервиса могут быть предоставлены верификатору привилегий в виде структурированного документа или посредством какого-нибудь другого средства:

- установленное доверие в проверяющем открытом ключе используется для утверждения подписи доверенного источника полномочий. Это доверие может быть установлено посредством объединенных средств или посредством сертификата открытого ключа,

изданного источником полномочий для ЦР, в котором верификатор привилегий уже установил доверие. Такой сертификат должен содержать расширение sOAIdentifier;

- привилегия назначителя привилегий закодирована в его сертификате атрибута или в расширении атрибутов субъекта директории их сертификата открытого ключа;

- путь делегирования сертификатов от утвердителя привилегий до доверенного источника полномочий;

- будет установлено доминирующее правило для привилегии; это может быть получено из дескриптора атрибута выданного ответственным за рассматриваемый атрибут источником полномочий, или он может быть получен посредством объединенных средств;

- политика привилегий; она может быть получена из директории или посредством некоторых объединенных средств;

- переменные среды, включающие, например, текущую дату/время, баланс текущего счета и т.д.

Реализация должна быть функционально эквивалентна внешнему режиму работы, получающемуся из этой процедуры. Однако, алгоритм, используемый исключительной реализацией для получения правильных выходных параметров из данных входных параметров, не стандартизируется.

Доминирующее правило связано с делегируемыми привилегиями. Синтаксис и метод для получения доминирующего правила не стандартизируется. Однако может быть проверена целостность восстановленных доминирующих правил. Сертификат дескриптора атрибута, выданный источником полномочий ответственного за делегирование атрибута, может содержать HASH доминирующего правила. Верификатор привилегий может воспроизвести HASH функцию восстановленной копии доминирующего правила и сравнить два хэша. Если они идентичны, то верификатор привилегий принимает, что доминирующее правило правильно.

14.3.2 Установление действительности пути делегирования

Верификатор привилегий находит путь делегирования и получает сертификаты для каждого объекта в пути. Путь делегирования расстилается от непосредственного назначителя привилегий до источника полномочий. Каждый промежуточный сертификат в пути делегирования должен содержать расширение BasicAttConstraints с набором компонент авторизации, установленным на true. Издатель каждого сертификата должен быть одинаковым с владельцем/субъектом сертификата, который является смежным к нему в пути делегирования. Расширение authorityAttributeIdentifier используется для установления соответствующего сертификата смежного объекта в пути делегирования. Количество сертификатов в пути от каждого объекта к непосредственному назначителю привилегий (включительно) не должен

O'ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

превышать 2 и границ значения pathLenConstraint в расширении объекта basicAttConstraints. Это потому, что pathLenConstraint ограничивает количество промежуточных сертификатов между двумя конечными точками (т.е. сертификат, содержащий ограничения и сертификат конечного объекта) так, что максимальная длина это значение этого ограничения плюс сертификаты, являющиеся конечными точками.

Если расширение delegatedNameConstraints представляется в любых сертификатах в пути делегирования, ограничения обрабатываются таким же образом, как расширение nameConstraints, которое обрабатывается процедурой обработки пути сертификации, приведенным в разделе 7.

Если расширение представляется в любых сертификатах в пути делегирования, верификатор привилегий гарантирует, что аутентификация каждого последующего объекта в пути делегирования выполняется сертификатом открытого ключа, который содержит, по крайней мере, одну допустимую политику.

14.3.3 Проверка делегирования привилегии

Не один делегатор не может делегировать привилегии, которые больше чем привилегии, которые он имеет. Доминирующее правило в атрибуте дескриптора атрибута обеспечивает правила для случая, когда данное значение «меньше чем» другое значение для атрибута, который делегируется.

Для каждого сертификата в пути делегирования, включающего сертификат непосредственного назначителя привилегий, верификатор привилегий гарантирует, что делегатор был авторизован для делегирования привилегий, которыми он обладает, и что делегированные привилегии не больше чем те привилегии, которыми они обладают.

Для каждого из этих сертификатов верификатор привилегий сравнивает делегированные привилегии с привилегиями, которые принадлежат делегатору, согласно с этим делегатором в соответствии с доминирующим правилом для привилегий. Привилегии, обладаемые делегатором, получаются из смежных сертификатов в пути делегирования как описано в 14.3.1.

14.3.4 Определение успешности/неудачи проверки

При условии, что действительный путь делегирования установлен, привилегии непосредственного назначителя привилегий представляются, как входные параметры, для сравнения с политикой привилегий, как описано в 14.1, чтобы определить достаточно ли привилегий имеет или не имеет непосредственный назначитель привилегий для контекста использования.

15 Схема директории PMI

15.1 Класс объекта директории PMI

15.1.1 Классы объекта пользователя PMI и органа по присвоению атрибутов PMI

Здесь определены элементы схемы директории, использованные для представления информации PMI в директории. Класс объекта директории PMI включает релевантный класс объекта, атрибуты и правила согласования значений атрибута.

Класс объекта пользователя PMI используется для определения элементов для объектов, которые могут быть владельцем сертификата атрибута.

pmiUser OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {attributeCertificateAttribute}
ID id-oc-pmiUser }

Класс объекта орган по присвоению атрибутов PMI использует в определении элементов для объектов, которые действуют как орган по присвоению атрибутов.

pmiAA OBJECT-CLASS ::= {

SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {aACertificate |
attributeCertificateRevocationList |
attributeAuthorityRevocationList}
ID id-oc-pmiAA }

15.1.2 Класс объекта источник полномочий PMI

Класс объекта источник полномочий PMI используется в определении элементов для объектов, которые действуют как источник полномочий. Заметим, что если объект был авторизован, чтобы действовать, как источник полномочий посредством выдачи сертификата открытого ключа, содержащего расширение sOAIdentifier, содержимое директории показывает, что объект также должен содержать класс объекта pkiCA.

pmiSOA OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {attributeCertificateRevocationList |
attributeAuthorityRevocationList |
attributeDescriptorCertificate}

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

ID id-oc-pmiSOA }

15.1.3 Класс объекта пункт распределения сертификата атрибута CRL

Класс объекта пункт распределения сертификата атрибута CRL используется в определении элементов для объектов, которые содержат сертификат атрибута и сегменты списка аннулирования органа по присвоению атрибутов. Этот вспомогательный класс предназначен для объединения со структурным классом объекта crlDistributionPoint, когда содержимое подвергается обработке. Поскольку атрибуты в certificateRevocationList и authorityRevocationList не обязательны в этом классе, можно создать элементы, которые содержат, например, только список аннулированных органов по присвоению атрибутов или элементы, которые содержат списки аннулирования множественных типов, зависящих от требований.

attCertCRLDistributionPt OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN { attributeCertificateRevocationList |
attributeAuthorityRevocationList }
ID id-oc-attCertCRLDistributionPts }

15.1.4 Делегирование пути PMI

Класс объекта пути делегирования PMI используется для определения элемента для объектов, которые могут делегировать пути. Он обычно используется в сочетании с элементом структурного класса объекта pmiAA.

pmiDelegationPath OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN { delegationPath }
ID id-oc-pmiDelegationPath }

15.1.5 Класс объекта политики привилегий

Класс объекта политики привилегий используется в определении элемента для объекта, который содержит информацию о политике привилегии.

privilegePolicy OBJECT-CLASS ::= {
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {privPolicy }
ID id-oc-privilegePolicy

15.2 Атрибуты директории PMI

15.2.1 Атрибут сертификата атрибута

Этот подраздел определяет атрибуты директории, использованные для хранения данных PMI в элементах директории.

Следующий атрибут содержит сертификаты атрибута, выданные для специфических владельцев и хранится в элементе директории этого держателя.

attributeCertificateAttribute ATTRIBUTE ::= {
WITH SYNTAX AttributeCertificate
EQUALITY MATCHING RULE attributeCertificateExactMatch
ID id-at-attributeCertificate

15.2.2 Сертификат атрибута органа по присвоению атрибута

Следующий атрибут содержит сертификаты атрибута, выданные для органа по присвоению атрибутов и хранится в элементе директории владельца органа по присвоению атрибутов.

aACertificate ATTRIBUTE ::= {
WITH SYNTAX AttributeCertificate
EQUALITY MATCHING RULE attributeCertificateExactMatch
ID id-at-aACertificate }

O'ZBEKISTON RESPUBLIKASI DAVLAT XAVFSIZLIGI XIZMATI HUZURIDAGI AXBOROT-TAHLIL MARKAZI

15.2.3 Сертификат атрибута дескриптора атрибута

Следующий атрибут содержит сертификаты атрибута, выданные источником полномочий, которые содержат расширения attributeDescriptor. Эти сертификаты атрибута содержат действительный синтаксис и спецификацию доминирующего правила атрибута привилегий и хранится в элементе директории издающего источника полномочий.

attributeDescriptorCertificate ATTRIBUTE ::= {
WITH SYNTAX AttributeCertificate
EQUALITY MATCHING RULE ttributeCertificateExactMatch
ID id-at-attributeDescriptorCertificate

15.2.4 Атрибут списка аннулированных сертификата атрибутов

Следующий атрибут содержит список аннулированных сертификатов атрибута. Эти списки могут храниться в элементе директории издающего органа, или в других элементах директории (например, пункт распределения).

attributeCertificateRevocationList ATTRIBUTE ::= {
WITH SYNTAX CertificateList
EQUALITY MATCHING RULE certificateListExactMatch

ID id-at-attributeCertificateRevocationList}

15.2.5 Атрибут списка аннулированных сертификатов органа по присвоению атрибутов

Следующий атрибут содержит список аннулированных сертификатов атрибута, выданных для органа по присвоению атрибутов. Эти списки могут храниться в элементе директории издающего органа или в других элементах директории (например, пункт распределения).

attributeAuthorityRevocationList ATTRIBUTE ::= {
WITH SYNTAX CertificateList
EQUALITY MATCHING RULE certificateListExactMatch
ID id-at-attributeAuthorityRevocationList }

15.2.6 Атрибут пути делегирования

Атрибут пути делегирования содержит пути делегирования, каждый из которых состоит из последовательности сертификатов атрибута.

delegationPath ATTRIBUTE ::= {
WITH SYNTAX AttCertPath
ID id-at-delegationPath }
AttCertPath ::= SEQUENCE OF AttributeCertificate

Этот атрибут может храниться в элементе директории органа по присвоению атрибутов и должен содержать какие-либо пути делегирования от этого органа по присвоению атрибутов к другим органам по присвоению атрибутов. Этот атрибут, если он используется, позволяет более эффективно восстанавливать делегированные сертификаты атрибута, форма которых часто используется путями делегирования.

15.2.7 Атрибут политики привилегий

Атрибут политики привилегий содержит информацию о политике привилегий.

privPolicy ATTRIBUTE ::= {
WITH SYNTAX PolicySyntax
ID id-at-privPolicy

Компонента policyIdentifier включает идентификатор объекта, зарегистрированный для отдельного объекта.

Если представлен content, то включается полное содержание политики привилегий.

Если представлен pointer, то компонента name ссылается на одно или более мест, где могут быть расположены копии политики привилегий. Если представлена компонента hash, она содержит HASH содержание политики

привилегий, которая должна быть найдена на сосланном месте. Этот хэш может быть использован для выполнения проверки на целостность сосланного документа.

15.3 Общие правила соответствия директории PMI

15.3.1 Точное соответствие сертификата атрибута

Этот подраздел определяет правила соответствия для атрибутов директории PMI.

Точное правило соответствия сертификата атрибута сравнивает равенство представленного значения со значением атрибута типа AttributeCertificate.

attributeCertificateExactMatch MATCHING-RULE ::= {
SYNTAX AttributeCertificateExactAssertion
ID id-mr-attributeCertificateExactMatch }
AttributeCertificateExactAssertion ::= SEQUENCE {
serialNumber CertificateSerialNumber OPTIONAL,
issuer IssuerSerial

Это правило соответствия возвращает значение TRUE, если компоненты в значении атрибута эквивалентны тем, которые приведены в представленном значении.

15.3.2 Соответствие сертификата атрибута

Правило соответствия сертификата атрибута сравнивает представленное значение с значением типа AttributeCertificate. Эти правила соответствия обеспечивают более комплексное соответствие, чем certificateExactMatch.

attributeCertificateMatch MATCHING-RULE ::= {
SYNTAX AttributeCertificateAssertion
ID id-mr-attributeCertificateMatch }
AttributeCertificateAssertion ::= SEQUENCE {
holder [0] CHOICE {
baseCertificateID [0] IssuerSerial,
holdertName [1] GeneralNames} PTIONAL,
issuer [1] GeneralNames OPTIONAL,
attCertValidity [2] GeneralizedTime OPTIONAL,
attType [3] SET OF AttributeType OPTIONAL}
-- как минимум одна компонента последовательности должна быть представлена

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

Правило соответствия возвращает значение TRUE, если все из компонент, которые представлены в представленном значении соответствуют аналогичным компонентам значения атрибута следующим образом:

- baseCertificateID совпадает, если он эквивалентен компоненте IssuerSerial хранимого значения атрибута;

- holderName совпадает, если хранимое значение атрибута содержит расширение имени с одинаковым типом имени как указано в представленном значении;

- ,issuer совпадает, хранимое значение атрибута содержит компоненту имени вышеупомянутого типа имени как указано в представленном значении;

- attCertValidity совпадает, если он находится в пределах определенного периода действительности хранимого значения атрибута;

- для каждого attType в представленном значении в компоненте attributes хранимого значения представляется атрибут этого типа.

15.3.3 Соответствие издателя и владельца сертификата

Правило соответствия атрибута сертификата держателя и издателя сравнивает равенство представленного значения держателя и/или компонентов издателя представленного значения со значением атрибута типа AttributeCertificate.

```
holderIssuerMatch MATCHING-RULE ::= {
SYNTAX    HolderIssuerAssertion
ID      id-mr-holderIssuerMatch }
HolderIssuerAssertion ::= SEQUENCE {
holder    [0]   Holder OPTIONAL,
issuer    [1]   AttCertIssuer OPTIONAL }
```

Правило согласования возвращает значение TRUE, если все из компонент, которые представлены в данном значении соответствуют аналогичным компонентам значения атрибута.

15.3.4 Соответствие пути делегирования

Правило согласования delegationPathMatch сравнивает равенство представленного значения со значением атрибута типа delegationPath. Верификатор привилегий может использовать это правило согласования, чтобы выбрать путь, начинающийся с сертификата, изданного его источником полномочий и заканчивающийся сертификатом, выданным органом по присвоению атрибутов, который выдан утвержденному конечному держателю сертификата.

```
delegationPathMatch MATCHING-RULE ::= {
SYNTAX    DelMatchSyntax
ID      id-mr-delegationPathMatch }

DelMatchSyntax   ::= SEQUENCE {
firstIssuer   AttCertIssuer,
```

lastHolder Holder }

Это правило согласования возвращает значение TRUE, если представленное значение в компоненте firstIssuer соответствует аналогичному элементу поля издателя первого сертификата в SEQUENCE в хранимом значении и представленное значение в компоненте совпадает (соответствует) элементам поля держателя последнего сертификата в SEQUENCE в хранимом значении. Это правило согласования возвращает значение FALSE, если оба сравнения неудачны.

16 Аутентификация директории

16.1 Процедура простой аутентификации

16.1.1 Генерация защищенной идентифицирующей информации

Директория поддерживает аутентификацию пользователей, обращающихся к директории через DUA и DSA для доступа к пользователем и к другим DSA. В зависимости от обстановки, может быть использована либо простая, либо жесткая аутентификация. Процедуры, используемые для простой и жесткой аутентификации в директории описаны в следующих разделах.

Простая аутентификация предназначена для предоставления локальной авторизации, основанной на различительном имени пользователя, двухсторонне соглашенного (не обязательного) пароля, и двухсторонне понимаемых средств использования и обращения этого пароля внутри одного домена. Использование простой аутентификации главным образом предназначено только для локального использования, т.е. аутентификация равноправных объектов между одним DUA и одним DSA или между двумя DSA. Простая аутентификация может быть реализована несколькими способами:

a) передачей различительного имени пользователя и (необязательного) пароля в незащищенном виде получателю для анализа;

b) передачей различительного имени пользователя, пароля и случайного числа и/или метки времени, каждая из них защищена с помощью применения односторонней функции;

c) передачей защищенной информации, описанной в перечислении b), вместе со случайным числом и/или параметром времени, каждое из которых защищено с помощью применения односторонней функции.

В местах, где пароли не защищены, обеспечивается минимальный уровень безопасности для предотвращения несанкционированного доступа. Это не должно рассматриваться, как основа для служб безопасности. Защита различительного имени пользователя и пароля обеспечивает значительный уровень безопасности. Алгоритмы, используемые для механизма защиты, являются не шифрующими односторонними функциями, которые очень просто реализовать. Общая процедура для реализации простой аутентификации показана на рисунке 6.

O'ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA
VA DAVLAT NAZORATI
DEPARTAMENTI

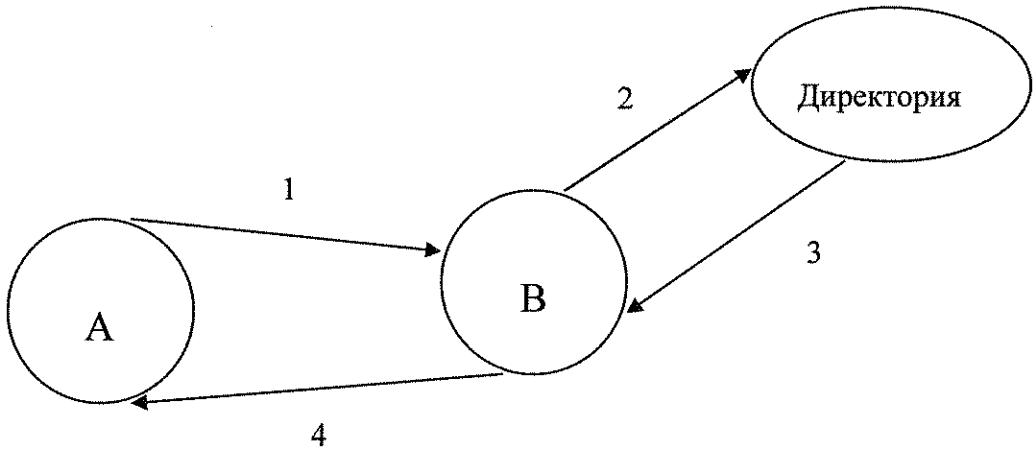


Рисунок 6 – Незащищенная процедура простой аутентификации

Сюда включены следующие шаги:

1) исходный (отправитель) пользователь А отправляет свое различительное имя и пароль пользователю (получателю) В;

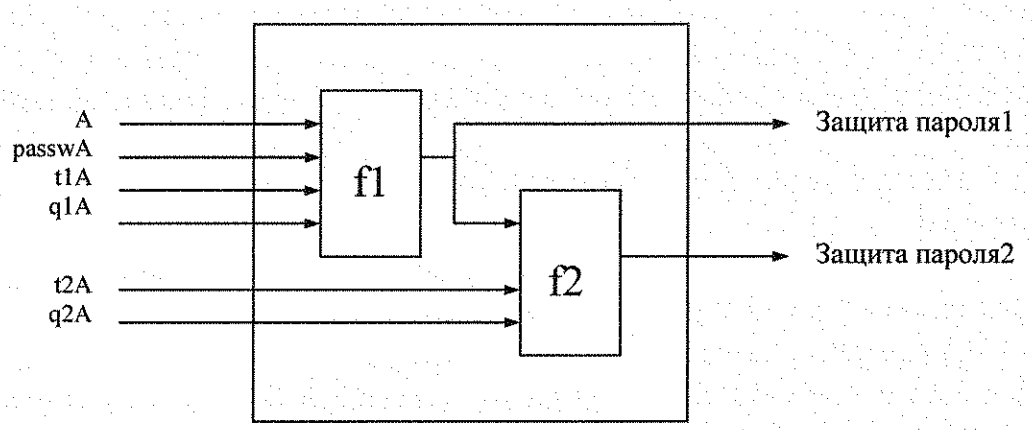
2) В отправляет предполагаемое различительное имя и пароль пользователя А в директорию, где пароль проверяется с тем, который храниться как атрибут UserPassword в элементе директории для пользователя А (используя операцию сравнения директории);

3) директория подтверждает (или отвергает) для В, что переданные данные являются действительными;

4) результат успешности (или неудачи) аутентификации может быть сообщен пользователю А.

Самая базовая форма простой аутентификации включает в себя только шаг 1) и после того как пользователь В проверил различительное имя и пароль, можно включить шаг 4).

Рисунок 7 иллюстрирует два подхода, согласно которым можно сгенерировать защищенную идентифицирующую информацию. f1 и f2 являются односторонними функциями (либо идентичные, либо различные), а также временные параметры (метки) и случайные числа являются необязательными и являются предметом двухстороннего соглашения.



A – различительное имя пользователя
t1A – временной штамп
passwA – пароль пользователя A
qA – случайные числа

Рисунок 7 – Защищенная простая аутентификация

16.1.2 Процедура для простой защищенной аутентификации

Рисунок 8 иллюстрирует процедуру для простой защищенной аутентификации.

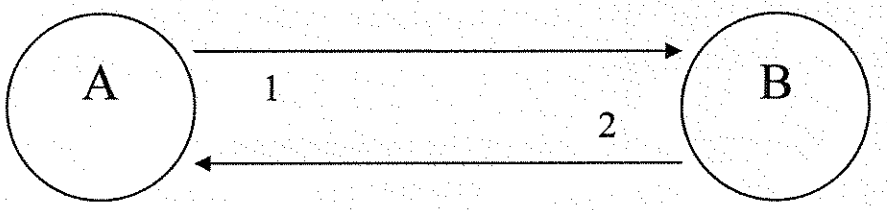


Рисунок 8 – Процедура простой защищенной аутентификации

В эту процедуру включены следующие шаги (первоначально использованием только f1):

1) первичный (отправитель) пользователь A отправляет свою защищенную идентифицирующую информацию (Аутентификатор1) пользователю B. Защита достигается применением односторонней функцию (f1) из рисунка 5, где временной параметр (метка) и/или случайное число (при использовании) используются для минимизации воспроизведения и маскировки (скрытия) пароля.

Защита пароля пользователя A выполняется в форме:

Защищенный пароль1 = f1 (t1A, q1A, A, passwA)

Информация, пользователю B передается в форме:

Аутентификатор1 = t1A, q1A, A, Защищенный пароль1;

2) B проверяет подлинность защищенной идентифицируемой информации предоставленной со стороны A, с помощью генерации (используя отличительное имя и необязательный временной параметр

(метка) и/или случайное число, предоставленное со стороны пользователя А, вместе с локальной копией пароля этого пользователя А) локальной защищенной копии пароля пользователя А (в форме Защищенный пароль1). Пользователь В сравнивает на равенство предъявленную идентифицирующую информацию (Защищенный пароль1) с локально сгенерированным значением;

3) пользователь В подтверждает или отклоняет верификацию защищенной идентифицирующей информации пользователя А.

Процедура может быть модифицирована для того, чтобы предоставить больше защищенности, используя f1 и f2. Основные различия являются следующими:

1) пользователь А отправляет свою дополнительно-защищенную идентифицирующую информацию (Аутентификатор2) пользователю В. Дополнительная защита осуществляется применением дополнительного использования односторонней функции f2, как это показано на рисунке 5. Дополнительная защита будет в следующей форме:

Защищенный пароль 2 = f2(t2A, q2A, Защищенный пароль1)

Информация, передаваемая пользователю В, определяется формулой:

Аутентификатор2 = t1A, t2A, q1A, q2A, A, Защищенный пароль2

Для сравнения, пользователь В генерирует локальное значение дополнительно защищенного пароля пользователя А и сравнивает его на равенство с Защищенный пароль 2;

2) пользователь В подтверждает или игнорирует верификацию защищенной идентифицирующей информации пользователя А.

16.1.3 Тип атрибута пароль пользователя

Тип атрибута пароль пользователя содержит пароль объекта. Значение атрибута для пароля пользователя является строкой, которая определяется объектом.

```
userPassword ATTRIBUTE ::= {
WITH SYNTAX      OCTET STRING (SIZE (0..ub-user-password))
EQUALITY MATCHING RULE   octetStringMatch
ID          id-at-userPassword }
```

16.2 Жесткая аутентификация

16.2.1 Получение из директории сертификата открытого ключа

Процедура, описанная в этом подразделе, используется в аутентификации между DUA и DSA, так же как между парными DSA. Процедуры пользуются структурой сертификата, определенной в настоящем стандарте. Кроме того, процедуры используют саму директорию, как хранилище для информации об открытом ключе требующуюся для аутентификации.

Процедура, использованная здесь для жесткой аутентификации, может также быть использована приложениями, отличающимися от

директории, которые также пользуются такими хранилищами. Для директории, использующей эти процедуры, термин «пользователь» в этих процедурах может означать оба и DUA, и DSA.

Подход к жесткой аутентификации, принятый в спецификации настоящего стандарта, использует свойства семейства криптографических систем, известных, как PKCS. Эти криптосистемы определены как ассиметричные, включающие пару ключей: один частный (закрытый) и один открытый, а не один ключ, как в обычных криптографических системах.

Эта структура не поручает для использования отдельную криптосистему. Она подразумевает, что структура должна быть применима к любой криптосистеме с открытым ключом и должна поддерживать изменения, как результат будущего прогресса в криптографии.

Аутентификация предполагает, что каждый пользователь имеет различительное имя.

Распределение различительных имен это ответственность органа по распределению имен.

Поэтому каждый пользователь доверяет тому, что орган по присвоению имен не издаст дубликат различительных имен.

Каждый пользователь идентифицируется его собственным закрытым ключом. Если партнер имеет закрытый ключ, второй пользователь имеет возможность определить это и может использовать закрытый ключ, чтобы подтвердить, что связанный партнер действительно пользователь. Действительность этого подтверждения зависит от закрытого ключа, остающегося секретным для пользователя.

Пользователь, чтобы определить связанного партнера, получает открытый ключ другого пользователя. Несмотря на то, что получение значения этого открытого ключа из элемента директории легко, проверка правильности более проблематична. Существует много возможных путей для выполнения проверки правильности открытого ключа пользователя: данный подраздел описывает процесс, посредством которого открытый ключ пользователя может быть проверен ссылкой на директорию. Этот процесс может работать, если только существует целая (неразорванная) цепь доверенных пунктов в директории между пользователями, требующими аутентификацию. Такая цепь может быть построена посредством идентификации общего пункта доверия. Этот общий пункт доверия должен быть присоединен к каждому пользователю посредством, неразорванной цепи доверенных пунктов.

Сертификаты содержатся в элементах директории как атрибуты типов UserCertificate, CACertificate и CrossCertificatePair. Эти типы атрибутов известны для директории. Эти атрибуты могут управляться одним и тем же протоколом, работающим как другой атрибут.

Спецификация этого типа атрибута определяется в 9.2.

В общем случае, прежде чем пользователи могут взаимно аутентифицироваться, директория доставляет полный сертификат и обратные пути сертификации. Однако, на практике, количество

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

информации, которая будет получена из директории, может быть уменьшеноы для отдельных случаев аутентификации следующим образом:

a) если два пользователя, которые хотят аутентифицироваться, обслуживаются одним сертифицирующим органом, тогда путь сертификации становится ограниченным и пользователи непосредственно разворачивают сертификаты друг друга;

b) если ЦР пользователей расположены в иерархии, пользователь должен хранить открытые ключи, сертификаты и наоборот сертификаты всех ЦР между пользователем и корнем DIT;

c) если пользователь часто связывается с пользователями сертифицированными другим, отдельным (индивидуальным) ЦР, этот пользователь изучит путь сертификации к этому ЦР и обратный путь от этого ЦР, это необходимо только для получения сертификата другого пользователя из директории;

d) ЦР могут кросс - сертифицировать посредством двусторонних соглашений. В результате происходит сокращение пути сертификации;

e) два пользователя, прежде чем связаться и изучить сертификаты друг друга, они могут аутентифицироваться без какой либо помощи директории.

В любом случае, при знании любых сертификатов из пути сертификации, пользователи должны проверять действительность полученных сертификатов.

16.2.2 Процедуры жесткой аутентификации

16.2.2.1 Односторонняя аутентификация

Простой подход для аутентификации в общих чертах описан выше, т.е. подтверждение подлинности осуществляется посредством обладания закрытым ключом. Тем не менее, многие процедуры аутентификации по возможности используют этот подход. В общем, определение соответствующей процедуры в качестве подходящей политики безопасности приложения это задача специфического приложения. Этот подраздел описывает три отдельные процедуры аутентификации, которые могут быть полезны в ряде приложений.

Эти три процедуры включают различное количество обмена аутентифицирующей информацией и следовательно обеспечивают различные типы гарантий их участникам. А именно:

a) односторонняя аутентификация включает одиночную передачу информации от пользователя А, предназначенную пользователю В, и устанавливает следующее:

- подлинность А и что идентификационный признак действительно был генерирован А;
- подлинность В, и что идентификационный признак действительно был предназначен для передачи В;
- идентификационный признак был передан в целостности и в «оригинальном» виде.

b) двухсторонняя аутентификация, описанная в 16.2.2.2, к тому же включает ответ B к A. Она устанавливается следующим образом:

- идентификационным признаком, генерируемым из ответа, который генерируется B и предназначен для передачи A;

- целостностью и оригинальностью идентификационного признака, переданного в ответе;

c) трехсторонняя аутентификация, описанная в 16.2.2.3, к тому же включает дальнейшую передачу от A к B. Она устанавливает такие же свойства как и двухсторонняя аутентификация, но делает это без потребности сравнения связанной временной метки.

В каждом случае, где жесткая аутентификация имеет место, A должен получить открытый ключ B и обратный путь сертификации от B к A, предшествующий любому обмену информацией. Это может включать доступ к директории, как описано в 16.2.

При использовании временной метки рекомендуется использовать универсальное установленное время.

Для каждого из трех процедур аутентификации, описанных выше, предполагается что сторона A проверяет действительность всех сертификатов в пути сертификации.

Рисунок 9 включает следующие шаги:

1) A генерирует одноразовое число rA, которое используется для обнаружения произведенных атак и предотвращения подделки;

2) A передает следующее сообщение B:

B>A, A{tA, rA, B},

где: tA это временная метка. tA содержит две или более дат: время генерации признака (которое не обязательно) и дату истечения срока. Если аутентификация источника данных это "sgnData" она обеспечивается цифровой подписью:

BA, A{ tA, rA, B, sgnData}

В случае, где информация была сообщена и была использована как закрытый ключ (эта информация обозначается как "encData"):

BA, A{ tA, rA, B, sgnData, Bp[encData]}

Использование "encData" как закрытый ключ подразумевает, что он будет выбран осторожно, например, чтобы быть сильным ключом какой-либо криптосистемы используемой как указано в поле "encData" признака;

3) B выполняет следующие действия:

a) получает Ap от BA, проверяет не истек ли срок сертификата A;

b) проверяет подписи, и таким образом целостность подписанной информации;

c) проверяет, что сам B является получателем (действительно ли предназначено ему);

d) проверяет что временная метка «текущая»;

e) факультативно, проверяет что rA не был воспроизведен. Это, например, может быть получено обладанием rA включая последовательные

O'ZBEKISTON STANDARTLARI MUVOFIQLASHTIRISH VA DAVLAT NAZORATI BOSHQARMASI

части, которые проверены локальным выполнением для уникальности его значения. rA действителен до тех пор, пока дата указанная в tA, не кончилась. rA всегда сопровождается последовательной частью, которая указывает, что A не повторит знака в течении времени tA и поэтому сопоставление своего значения rA не требуется.

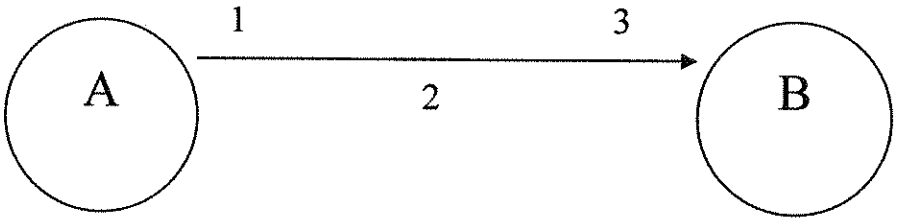


Рисунок 9 – Односторонняя аутентификация

16.2.2.2 Двухсторонняя аутентификация

Рисунок 10 включает следующие шаги:

1) как в 16.2.2.1;

2) как в 16.2.2.1;

3) как в 16.2.2.1;

4) B генерирует одноразовое число rB, использованное для похожих целей как и rA;

5) B посылает следующий идентификационный признак A:

B{tB, rB, A, rA },

где: tB временная метка, определенная также как tA.

Если аутентификация источника данных это "sgnData" она обеспечивается цифровой подписью:

B{tB, rB, A, rA, sgnData}

В случае, где информация была сообщена и была использована как закрытый ключ (эта информация обозначается как "encData"):

B{tB, rB, A, rA, sgnData, Ap[encData]}.

Использование "encData" как закрытый ключ подразумевает, что он будет выбран осторожно, например чтобы быть сильным ключом какой-либо криптосистемы, используемой как указано в поле "encData" признака;

6) A выполняет следующие действия:

a) проверяет подпись и таким образом целостность подписанной информации;

b) проверяет, что A это предназначенный получатель;

c) проверяет что rB не был воспроизведен (см. 16.2.2.1, шаг 3), d)).

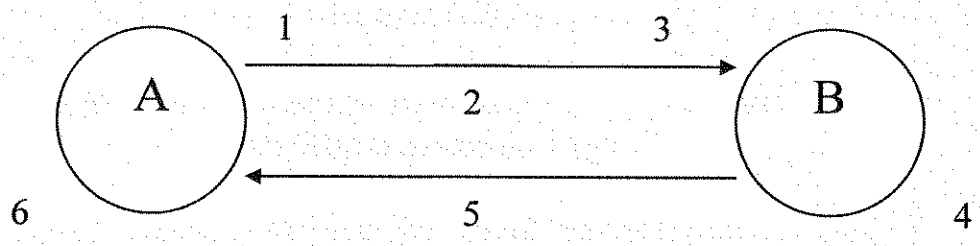


Рисунок 10 – Двухсторонняя аутентификация

16.2.2.3 Трехсторонняя аутентификация

Рисунок 11 включает следующие шаги:

1) как в 16.2.2.2;

2) как для 16.2.2.2 временная метка tA может быть обнулена;

3) как для 16.2.2.2 исключает, что временная метка не должна быть проверена;

4) как в 16.2.2.2;

5) как для 16.2.2.2 временная метка tA может быть обнулена;

6) как для 16.2.2.2 исключает, что временная метка не должна быть проверена;

7) A проверяет, что принятый rA идентичен rA , который был проверен;

8) A посылает следующий идентификационный признак B:

A{rB,B};

9) B выполняет следующие действия:

a) проверяет подпись и таким образом целостность подписанной информации;

b) проверяет, что полученный rB идентичен rB, который был принят B.

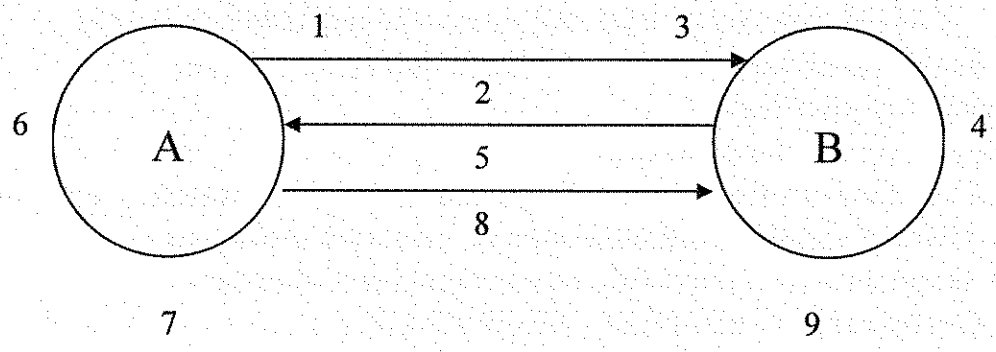


Рисунок 11 – Трехсторонняя аутентификация

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

Приложение А
(обязательное)

Структура сертификата открытого ключа и сертификата атрибута

А.1 Данное приложение включает все типы языка ASN.1, значение и определения классов информационных объектов в форме трех модулей языка ASN.1: AuthenticationFramework, CertificateExtensions и AttributeCertificate Derinitions.

-- А.2 Модуль структуры Аутентификации

```
AuthenticationFramework {joint-iso-itu-t ds(5) module(1)
authenticationFramework(7)
4}
DEFINITIONS ::=
BEGIN
-- EXPORTS All –
IMPORTS
  id-at, id-nf, id-oc, informationFramework, upperBounds,
selectedAttributeTypes,
basicAccessControl,
  certificateExtensions
    FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
usefulDefinitions(0) 4}
  Name, ATTRIBUTE, OBJECT-CLASS, NAME-FORM, top
    FROM InformationFramework informationFramework
  ub-user-password, ub-content
    FROM UpperBounds upperBounds
  UniqueIdentifier, octetStringMatch, DirectoryString, commonName
    FROM SelectedAttributeTypes selectedAttributeTypes
  certificateExactMatch, certificatePairExactMatch,
certificateListExactMatch,
KeyUsage, GeneralNames,
  CertificatePoliciesSyntax, algorithmIdentifierMatch, CertPolicyId
   FROM CertificateExtensions certificateExtensions ;
-- Определение сертификата открытого ключа --
  Certificate ::= SIGNED { SEQUENCE {
    version [0] Version DEFAULT v1,
    serialNumber CertificateSerialNumber,
    signature AlgorithmIdentifier,
    issuer Name,
    validity Validity,
    subject Name,
```

```
        subjectPublicKeyInfo SubjectPublicKeyInfo,
        issuerUniqueIdentifier  [1] IMPLICIT UniqueIdentifier
OPTIONAL,
        subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier
OPTIONAL,
        extensions          [3] Extensions OPTIONAL} }
      Version ::= INTEGER { v1(0), v2(1), v3(2) }
      CertificateSerialNumber ::= INTEGER
      AlgorithmIdentifier ::= SEQUENCE {

        algorithm ALGORITHM.&id ({SupportedAlgorithms}),
        parameters ALGORITHM.&Type ({SupportedAlgorithms}{
@algorithm})
      OPTIONAL }
        SupportedAlgorithms ALGORITHM ::= { ... }

        Validity ::= SEQUENCE {
          notBefore Time,
          notAfter Time }
        SubjectPublicKeyInfo ::= SEQUENCE {
          algorithm AlgorithmIdentifier,
          subjectPublicKey BIT STRING }
        Time ::= CHOICE {
          utcTime UTCTime,
          generalizedTime GeneralizedTime }
        Extensions ::= SEQUENCE OF Extension
        Extension ::= SEQUENCE {
          extnId EXTENSION.&id ({ExtensionSet}),
          critical BOOLEAN DEFAULT FALSE,
          extnValue OCTET STRING}
        ExtensionSet EXTENSION ::= { ... }
        EXTENSION ::= CLASS {
          &id OBJECT IDENTIFIER UNIQUE,
          &ExtnType }
        WITH SYNTAX {
          SYNTAX &ExtnType
          IDENTIFIED BY &id }
        -- другие конструкции сертификата PKI
       Certificates ::= SEQUENCE {
          userCertificate Certificate,
          certificationPath ForwardCertificationPath OPTIONAL}
        ForwardCertificationPath ::= SEQUENCE OF CrossCertificates
        CrossCertificates ::= SET OF Certificate
        CertificationPath ::= SEQUENCE {
          userCertificate Certificate,
```

O‘ZBEKISTON RESPUBLIKASI STANDARTLASHTIRISH, METROLOGIYA VA SERTIFIKATLASHTIRISH DAVLAT QO‘MITASI

```
        theCACertificates SEQUENCE OF CertificatePair OPTIONAL}
      CertificatePair ::= SEQUENCE {
        forward [0] Certificate OPTIONAL,
        reverse [1] Certificate OPTIONAL}
      (WITH COMPONENTS {..., forward PRESENT} |
       WITH COMPONENTS {..., reverse PRESENT})
      -- Список отозванных сертификатов (CRL)
      CertificateList ::= SIGNED { SEQUENCE {
        version Version OPTIONAL,
        signature AlgorithmIdentifier,
        issuer Name,
        thisUpdate Time,
        nextUpdate Time OPTIONAL,
        revokedCertificates SEQUENCE OF SEQUENCE {
          serialNumber CertificateSerialNumber,
          revocationDate Time,
          crlEntryExtensions Extensions OPTIONAL } OPTIONAL,
        crlExtensions [0] Extensions OPTIONAL }}
      -- классы информационных объектов --
      ALGORITHM ::= TYPE-IDENTIFIER
      -- параметризованные типы --
      HASH {ToBeHashed} ::= SEQUENCE {
        algorithmIdentifier AlgorithmIdentifier,
        hashValue BIT STRING ( CONSTRAINED BY {
      } ) }
      ENCRYPTED-HASH { ToBeSigned } ::= BIT STRING (
CONSTRAINED BY {
    })
      ENCRYPTED { ToBeEnciphered } ::= BIT STRING (
CONSTRAINED BY {
    })
      SIGNATURE { ToBeSigned } ::= SEQUENCE {
        algorithmIdentifier AlgorithmIdentifier,
        encrypted ENCRYPTED-HASH { ToBeSigned }}
      SIGNED { ToBeSigned } ::= SEQUENCE {
        toBeSigned ToBeSigned,
        COMPONENTS OF SIGNATURE { ToBeSigned }}
      -- классы объекта PKI --
      pkiUser OBJECT-CLASS ::= {
        SUBCLASS OF {top}
        KIND auxiliary
        MAY CONTAIN {userCertificate}
        ID id-oc-pkiUser }
      pkiCA OBJECT-CLASS ::= {
        SUBCLASS OF {top}
```

```
  KIND auxiliary
  MAY CONTAIN {cACertificate |
          certificateRevocationList |
          authorityRevocationList |
          crossCertificatePair }
 ID          id-oc-pkiCA }
cRLDistributionPoint OBJECT-CLASS ::= {
  SUBCLASS OF { top }
  KIND structural
  MUST CONTAIN { commonName }
  MAY CONTAIN { certificateRevocationList |
          authorityRevocationList |
          deltaRevocationList }
  ID          id-oc-cRLDistributionPoint }
cRLDistPtNameForm NAME-FORM ::= {
  NAMES cRLDistributionPoint
  WITH ATTRIBUTES { commonName}
  ID id-nf-cRLDistPtNameForm }
deltaCRL OBJECT-CLASS ::= {
  SUBCLASS OF {top}
  KIND auxiliary
  MAY CONTAIN {deltaRevocationList}
  ID id-oc-deltaCRL }
cpCps OBJECT-CLASS ::= {
  SUBCLASS OF {top}
  KIND auxiliary
  MAY CONTAIN {certificatePolicy |
          certificationPracticeStmt}
  ID          id-oc-cpCps }
pkiCertPath OBJECT-CLASS ::= {
  SUBCLASS OF {top}
  KIND auxiliary
  MAY CONTAIN { pkiPath }
  ID id-oc-pkiCertPath }
-- Атрибуты директории PKI --
userCertificate ATTRIBUTE ::= {
  WITH SYNTAX Certificate
  EQUALITY MATCHING RULE certificateExactMatch
  ID id-at-userCertificate}
cACertificate ATTRIBUTE ::= {
  WITH SYNTAX Certificate
  EQUALITY MATCHING RULE certificateExactMatch
  ID id-at-cAcertificate }
crossCertificatePair ATTRIBUTE ::= {
  WITH SYNTAX CertificatePair
```

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

```
    EQUALITY MATCHING RULE certificatePairExactMatch
    ID id-at-crossCertificatePair }
  certificateRevocationList ATTRIBUTE ::= {
    WITH SYNTAX CertificateList
    EQUALITY MATCHING RULE certificateListExactMatch
    ID id-at-certificateRevocationList }
  authorityRevocationList ATTRIBUTE ::= {
    WITH SYNTAX CertificateList
    EQUALITY MATCHING RULE certificateListExactMatch
    ID id-at-authorityRevocationList }
  deltaRevocationList ATTRIBUTE ::= {
    WITH SYNTAX CertificateList
    EQUALITY MATCHING RULE certificateListExactMatch
    ID id-at-deltaRevocationList }
  supportedAlgorithms ATTRIBUTE ::= {
    WITH SYNTAX SupportedAlgorithm
    EQUALITY MATCHING RULE algorithmIdentifierMatch
    ID id-at-supportedAlgorithms }
  SupportedAlgorithm ::= SEQUENCE {
    algorithmIdentifier AlgorithmIdentifier,
    intendedUsage [0] KeyUsage OPTIONAL,
    intendedCertificatePolicies [1] CertificatePoliciesSyntax
OPTIONAL }
  certificationPracticeStmt ATTRIBUTE ::= {
    WITH SYNTAX InfoSyntax
    ID id-at-certificationPracticeStmt }
  InfoSyntax ::= CHOICE {
    content DirectoryString {ub-content},
    pointer SEQUENCE {
      name GeneralNames,
      hash HASH { HashedPolicyInfo } OPTIONAL } }
  POLICY ::= TYPE-IDENTIFIER
  HashedPolicyInfo ::= POLICY.&Type( {Policies} )
  Policies POLICY ::= {...} -- Определенная со стороны реализатора -
-
  certificatePolicy ATTRIBUTE ::= {
    WITH SYNTAX PolicySyntax
    ID id-at-certificatePolicy }
  PolicySyntax ::= SEQUENCE {
    policyIdentifier PolicyID,
    policySyntax InfoSyntax
    }

  PolicyID ::= CertPolicyId
  pkiPath ATTRIBUTE ::= {
```

```
        WITH SYNTAX PkiPath
        ID id-at-pkiPath }
      PkiPath ::= SEQUENCE OF CrossCertificates
      userPassword ATTRIBUTE ::= {
        WITH SYNTAX OCTET STRING (SIZE (0..ub-user-password))
        EQUALITY MATCHING RULE octetStringMatch
        ID id-at-userPassword }
      id-oc-cRLDistributionPoint      OBJECT IDENTIFIER ::= {id-oc 19}
      id-oc-pkiUser                   OBJECT IDENTIFIER ::= {id-oc 21}
      id-oc-pkiCA                     OBJECT IDENTIFIER ::= {id-oc 22}
      id-oc-deltaCRL                  OBJECT IDENTIFIER ::= {id-oc 23}
      id-oc-cpCps                     OBJECT IDENTIFIER ::= {id-oc 30}
      id-oc-pkiCertPath               OBJECT IDENTIFIER ::= {id-oc 31}
      --формы имен--
      id-nf-cRLDistPtNameForm     OBJECT IDENTIFIER ::= {id-nf 14}
      --атрибуты директории --
      id-at-userPassword              OBJECT IDENTIFIER ::= {id-at 35}
      id-at-userCertificate           OBJECT IDENTIFIER ::= {id-at 36}
      id-at-cAcertificate             OBJECT IDENTIFIER ::= {id-at 37}
      id-at-authorityRevocationList      OBJECT IDENTIFIER ::= {id-at
38}
      id-at-certificateRevocationList    OBJECT IDENTIFIER ::= {id-at
39}
      id-at-crossCertificatePair      OBJECT IDENTIFIER ::= {id-at 40}
      id-at-supportedAlgorithms       OBJECT IDENTIFIER ::= {id-at
52}
      id-at-deltaRevocationList       OBJECT IDENTIFIER ::= {id-at 53}
      id-at-certificationPracticeStmt    OBJECT IDENTIFIER ::= {id-at
68}
      id-at-certificatePolicy         OBJECT IDENTIFIER ::= {id-at 69}
      id-at-pkiPath                   OBJECT IDENTIFIER ::= {id-at 70}
      END
      -- A.3 Модуль расширений сертификата
      CertificateExtensions {joint-iso-itu-t ds(5) module(1)
certificateExtensions(26) 4}
      DEFINITIONS IMPLICIT TAGS ::=
      BEGIN
      IMPORTS
        id-at, id-ce, id-mr, informationFramework,
authenticationFramework,
          selectedAttributeTypes, upperBounds
          FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
          usefulDefinitions(0) 4}
        Name, RelativeDistinguishedName, ATTRIBUTE, Attribute,
MATCHING-RULE
```

```
        FROM InformationFramework informationFramework
      CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, Time, PolicyID
        FROM AuthenticationFramework authenticationFramework
      DirectoryString
        FROM SelectedAttributeTypes selectedAttributeTypes
      ub-name
        FROM UpperBounds upperBounds
      ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;

    -- Сертификат открытого ключа и расширения CRL --
    authorityKeyIdentifier EXTENSION ::= {
      SYNTAX AuthorityKeyIdentifier
      IDENTIFIED BY id-ce-authorityKeyIdentifier }
    AuthorityKeyIdentifier ::= SEQUENCE {
      keyIdentifier [0] KeyIdentifier OPTIONAL,
      authorityCertIssuer [1] GeneralNames OPTIONAL,
      authorityCertSerialNumber [2] CertificateSerialNumber
OPTIONAL }
      ( WITH COMPONENTS {..., authorityCertIssuer PRESENT,
        authorityCertSerialNumber PRESENT} |
      WITH COMPONENTS {..., authorityCertIssuer ABSENT,
        authorityCertSerialNumber ABSENT} )
    KeyIdentifier ::= OCTET STRING
    subjectKeyIdentifier EXTENSION ::= {
      SYNTAX SubjectKeyIdentifier
      IDENTIFIED BY id-ce-subjectKeyIdentifier }
    SubjectKeyIdentifier ::= KeyIdentifier
    keyUsage EXTENSION ::= {
      SYNTAX KeyUsage
      IDENTIFIED BY id-ce-keyUsage }
    KeyUsage ::= BIT STRING {
      digitalSignature    (0),
      nonRepudiation      (1),
      keyEncipherment     (2),
      dataEncipherment    (3),
      keyAgreement        (4),
      keyCertSign         (5),
      cRLSign             (6),
      encipherOnly        (7),
      decipherOnly        (8) }
    extKeyUsage EXTENSION ::= {
      SYNTAX SEQUENCE SIZE (1..MAX) OF KeyPurposeId
```

```
    IDENTIFIED BY id-ce-extKeyUsage }
  KeyPurposeId ::= OBJECT IDENTIFIER
  privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX PrivateKeyUsagePeriod
    IDENTIFIED BY id-ce-privateKeyUsagePeriod }
  PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore [0] GeneralizedTime OPTIONAL,
    notAfter [1] GeneralizedTime OPTIONAL }
    ( WITH COMPONENTS {..., notBefore PRESENT} |
    WITH COMPONENTS {..., notAfter PRESENT} )
  certificatePolicies EXTENSION ::= {
    SYNTAX CertificatePoliciesSyntax
    IDENTIFIED BY id-ce-certificatePolicies }
  CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF
PolicyInformation
  PolicyInformation ::= SEQUENCE {
    policyIdentifier CertPolicyId,
    policyQualifiers SEQUENCE SIZE (1..MAX) OF
            PolicyQualifierInfo OPTIONAL }
  CertPolicyId ::= OBJECT IDENTIFIER
  PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId CERT-POLICY-QUALIFIER.&id
            ({SupportedPolicyQualifiers}),
    qualifier CERT-POLICY-QUALIFIER.&Qualifier
          ({SupportedPolicyQualifiers}{@policyQualifierId})
          OPTIONAL }
  SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }
  anyPolicy OBJECT IDENTIFIER ::= { 2 5 29 32 0 }
  CERT-POLICY-QUALIFIER ::= CLASS {
    &id OBJECT IDENTIFIER UNIQUE,
    &Qualifier OPTIONAL }
  WITH SYNTAX {
    POLICY-QUALIFIER-ID &id
    [QUALIFIER-TYPE &Qualifier] }
  policyMappings EXTENSION ::= {
    SYNTAX PolicyMappingsSyntax
    IDENTIFIED BY id-ce-policyMappings }
  PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF
SEQUENCE {
    issuerDomainPolicy CertPolicyId,
    subjectDomainPolicy CertPolicyId }
  subjectAltName EXTENSION ::= {
    SYNTAX GeneralNames
    IDENTIFIED BY id-ce-subjectAltName }
  GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName
```

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLIKNI
BA DAVLAT NAZORATI

```
GeneralName ::= CHOICE {
  otherName                  [0]  INSTANCE OF OTHER-NAME,
  rfc822Name                 [1]  IA5String,
  dNSName                    [2]  IA5String,
  x400Address                [3]  ORAddress,
  directoryName              [4]  Name,
  ediPartyName               [5]  EDIPartyName,
  uniformResourceIdentifier  [6]  IA5String,
  iPAddress                  [7]  OCTET STRING,
  registeredID               [8]  OBJECT IDENTIFIER }
OTHER-NAME ::= TYPE-IDENTIFIER
EDIPartyName ::= SEQUENCE {
  nameAssigner  [0]  DirectoryString {ub-name} OPTIONAL,
  partyName     [1]  DirectoryString {ub-name} }
issuerAltName EXTENSION ::= {
  SYNTAX     GeneralNames
  IDENTIFIED BY  id-ce-issuerAltName }
subjectDirectoryAttributes EXTENSION ::= {
  SYNTAX     AttributesSyntax
  IDENTIFIED BY  id-ce-subjectDirectoryAttributes }
AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute
basicConstraints EXTENSION ::= {
  SYNTAX     BasicConstraintsSyntax
  IDENTIFIED BY  id-ce-basicConstraints }
BasicConstraintsSyntax ::= SEQUENCE {
  cA BOOLEAN DEFAULT FALSE,
  pathLenConstraint INTEGER (0..MAX) OPTIONAL }
nameConstraints EXTENSION ::= {
  SYNTAX     NameConstraintsSyntax
  IDENTIFIED BY  id-ce-nameConstraints }

NameConstraintsSyntax ::= SEQUENCE {
  permittedSubtrees  [0]  GeneralSubtrees OPTIONAL,
  excludedSubtrees   [1]  GeneralSubtrees OPTIONAL }
GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree
GeneralSubtree ::= SEQUENCE {
  base      GeneralName,
  minimum   [0]  BaseDistance DEFAULT 0,
  maximum   [1]  BaseDistance OPTIONAL }
BaseDistance ::= INTEGER (0..MAX)
policyConstraints EXTENSION ::= {
  SYNTAX     PolicyConstraintsSyntax
  IDENTIFIED BY  id-ce-policyConstraints }
PolicyConstraintsSyntax ::= SEQUENCE {
  requireExplicitPolicy  [0]  SkipCerts OPTIONAL,
```

```
    inhibitPolicyMapping  [1]  SkipCerts OPTIONAL }
  SkipCerts ::= INTEGER (0..MAX)
  cRLNumber EXTENSION ::= {
    SYNTAX    CRLNumber
    IDENTIFIED BY  id-ce-cRLNumber }
  CRLNumber ::= INTEGER (0..MAX)
  reasonCode EXTENSION ::= {
    SYNTAX    CRLReason
    IDENTIFIED BY  id-ce-reasonCode }
  CRLReason ::= ENUMERATED {
    unspecified          (0),
    keyCompromise        (1),
    cACompromise         (2),
    affiliationChanged   (3),
    superseded           (4),
    cessationOfOperation (5),
    certificateHold      (6),
    removeFromCRL        (8),
    privilegeWithdrawn   (9),
    aaCompromise         (7) }
  holdInstructionCode EXTENSION ::= {
    SYNTAX    HoldInstruction
    IDENTIFIED BY  id-ce-instructionCode }
  HoldInstruction ::= OBJECT IDENTIFIER
  invalidityDate EXTENSION ::= {
    SYNTAX    GeneralizedTime
    IDENTIFIED BY  id-ce-invalidityDate }
  crlScope EXTENSION ::= {
    SYNTAX    CRLScopeSyntax
    IDENTIFIED BY  id-ce-cRLScope }
  CRLScopeSyntax ::= SEQUENCE SIZE (1..MAX) OF
PerAuthorityScope
  PerAuthorityScope ::= SEQUENCE {
    authorityName       [0]  GeneralName OPTIONAL,
    distributionPoint   [1]  DistributionPointName OPTIONAL,
    onlyContains        [2]  OnlyCertificateTypes OPTIONAL,
    onlySomeReasons     [4]  ReasonFlags OPTIONAL,
    serialNumberRange   [5]  NumberRange OPTIONAL,
    subjectKeyIdRange   [6]  NumberRange OPTIONAL,
    nameSubtrees        [7]  GeneralNames OPTIONAL,
    baseRevocationInfo  [9]  BaseRevocationInfo OPTIONAL
    }
  OnlyCertificateTypes ::= BIT STRING {
    user       (0),
    authority  (1),
```

AXBOROT AGENTLIGI
ELEKTRON HUKUMAT
STRATEGIYASI VA
RIVOJLANISH
VAZIRLIGI HUZURIDAGI

```
    attribute   (2) }
  NumberRange ::= SEQUENCE {
    startingNumber   [0]   INTEGER OPTIONAL,
    endingNumber   [1]   INTEGER OPTIONAL,
    modulus      INTEGER OPTIONAL }
  BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier   [0] CRLStreamIdentifier OPTIONAL,
    cRLNumber     [1] CRLNumber,
    baseThisUpdate     [2] GeneralizedTime }
  statusReferrals EXTENSION ::= {
    SYNTAX      StatusReferrals
    IDENTIFIED BY     id-ce-statusReferrals }
  StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral
  StatusReferral ::= CHOICE {
    cRLReferral   [0]   CRLReferral,
    otherReferral   [1]   INSTANCE OF OTHER-REFERRAL}
  CRLReferral ::= SEQUENCE {
    issuer            [0]   GeneralName OPTIONAL,
    location          [1]   GeneralName OPTIONAL,
    deltaRefInfo       [2]   DeltaRefInfo OPTIONAL,
    cRLScope               CRLScopeSyntax,
    lastUpdate        [3]   GeneralizedTime OPTIONAL,
    lastChangedCRL     [4]   GeneralizedTime OPTIONAL}
  DeltaRefInfo ::= SEQUENCE {
    deltaLocation   GeneralName,
    lastDelta     GeneralizedTime OPTIONAL }
  OTHER-REFERRAL ::= TYPE-IDENTIFIER
  cRLStreamIdentifier EXTENSION ::= {
    SYNTAX     CRLStreamIdentifier
    IDENTIFIED BY   id-ce-cRLStreamIdentifier }
  CRLStreamIdentifier ::= INTEGER (0..MAX)
  orderedList EXTENSION ::= {
    SYNTAX     OrderedListSyntax
    IDENTIFIED BY   id-ce-orderedList }
  OrderedListSyntax ::= ENUMERATED {
    ascSerialNum   (0),
    ascRevDate     (1) }
  deltaInfo EXTENSION ::= {
    SYNTAX     DeltaInformation
    IDENTIFIED BY   id-ce-deltaInfo }
  DeltaInformation ::= SEQUENCE {
    deltaLocation   GeneralName,
    nextDelta     GeneralizedTime OPTIONAL }
  cRLDistributionPoints EXTENSION ::= {
    SYNTAX     CRLDistPointsSyntax
```

```
    IDENTIFIED BY   id-ce-cRLDistributionPoints }
  CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF
DistributionPoint

  DistributionPoint ::= SEQUENCE {
    distributionPoint   [0]   DistributionPointName OPTIONAL,
    reasons     [1]   ReasonFlags OPTIONAL,
    cRLIssuer     [2]   GeneralNames OPTIONAL }
  DistributionPointName ::= CHOICE {
    fullName        [0]   GeneralNames,
    nameRelativeToCRLIssuer   [1]   RelativeDistinguishedName }
  ReasonFlags ::= BIT STRING {
    unused                  (0),
    keyCompromise           (1),
    cACompromise            (2),
    affiliationChanged      (3),
    superseded              (4),
    cessationOfOperation    (5),
    certificateHold         (6),
    privilegeWithdrawn      (7),
    aACompromise            (8)}
  issuingDistributionPoint EXTENSION ::= {
    SYNTAX     IssuingDistPointSyntax
    IDENTIFIED BY   id-ce-issuingDistributionPoint }
  IssuingDistPointSyntax ::= SEQUENCE {
    distributionPoint          [0]   DistributionPointName OPTIONAL,
    onlyContainsUserCerts      [1]   BOOLEAN DEFAULT FALSE,
    onlyContainsAuthorityCerts     [2]   BOOLEAN DEFAULT FALSE,
    onlySomeReasons            [3]   ReasonFlags OPTIONAL,
    indirectCRL                [4]   BOOLEAN DEFAULT FALSE,
    onlyContainsAttributeCerts     [5]   BOOLEAN DEFAULT FALSE
}
  certificateIssuer EXTENSION ::= {
    SYNTAX     GeneralNames
    IDENTIFIED BY   id-ce-certificateIssuer }
  deltaCRLIndicator EXTENSION ::= {
    SYNTAX     BaseCRLNumber
    IDENTIFIED BY   id-ce-deltaCRLIndicator }
  BaseCRLNumber ::= CRLNumber
  baseUpdateTime EXTENSION ::= {
    SYNTAX     GeneralizedTime
    IDENTIFIED BY   id-ce-baseUpdateTime }
  freshestCRL EXTENSION ::= {
    SYNTAX     CRLDistPointsSyntax
    IDENTIFIED BY   id-ce-freshestCRL }
```

O‘ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
DAVLAT QO‘MITASI

```
inhibitAnyPolicy EXTENSION ::= {
  SYNTAX    SkipCerts
 IDENTIFIED BY  id-ce-inhibitAnyPolicy }
 -- Правила соответствия PKI --
 certificateExactMatch MATCHING-RULE ::= {
  SYNTAX    CertificateExactAssertion
  ID      id-mr-certificateExactMatch }
 CertificateExactAssertion ::= SEQUENCE {
  serialNumber CertificateSerialNumber,
  issuer Name }
 certificateMatch MATCHING-RULE ::= {
  SYNTAX  CertificateAssertion
  ID    id-mr-certificateMatch }

 CertificateAssertion ::= SEQUENCE {
  serialNumber          [0]  CertificateSerialNumber OPTIONAL,
  issuer                [1]  Name OPTIONAL,
  subjectKeyIdentifier     [2]  SubjectKeyIdentifier OPTIONAL,
  authorityKeyIdentifier   [3]  AuthorityKeyIdentifier OPTIONAL,
  certificateValid       [4]  Time OPTIONAL,
  privateKeyValid         [5]  GeneralizedTime OPTIONAL,
  subjectPublicKeyAlgID    [6]  OBJECT IDENTIFIER OPTIONAL,
  keyUsage              [7]  KeyUsage OPTIONAL,
  subjectAltName          [8]  AltNameType OPTIONAL,
  policy                [9]  CertPolicySet OPTIONAL,
  pathToName              [7]  Name OPTIONAL,
  subject               [8]  Name OPTIONAL,
  nameConstraints         [12]  NameConstraintsSyntax OPTIONAL }
AltNameType ::= CHOICE {
  builtinNameForm ENUMERATED {
    rfc822Name              (1),
    dNSName                 (2),
    x400Address             (3),
    directoryName           (4),
    ediPartyName            (5),
    uniformResourceIdentifier   (6),
    iPAddress               (7),
    registeredId            (8)},
  otherNameForm OBJECT IDENTIFIER }
CertPolicySet ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId
certificatePairExactMatch MATCHING-RULE ::= {
  SYNTAX    CertificatePairExactAssertion
  ID      id-mr-certificatePairExactMatch }
CertificatePairExactAssertion ::= SEQUENCE {
```

```
        issuedToThisCAAssertion [0] CertificateExactAssertion
OPTIONAL,
        issuedByThisCAAssertion [1] CertificateExactAssertion
OPTIONAL }
        ( WITH COMPONENTS {..., issuedToThisCAAssertion PRESENT}
|
        WITH COMPONENTS {..., issuedByThisCAAssertion PRESENT} )
      certificatePairMatch MATCHING-RULE ::= {
        SYNTAX    CertificatePairAssertion
        ID      id-mr-certificatePairMatch }
      CertificatePairAssertion ::= SEQUENCE {
        issuedToThisCAAssertion [0] CertificateAssertion OPTIONAL,
        issuedByThisCAAssertion [1] CertificateAssertion OPTIONAL }
        ( WITH COMPONENTS {..., issuedToThisCAAssertion PRESENT}
|
        WITH COMPONENTS {..., issuedByThisCAAssertion PRESENT} )
      certificateListExactMatch MATCHING-RULE ::= {
        SYNTAX    CertificateListExactAssertion
        ID      id-mr-certificateListExactMatch }
      CertificateListExactAssertion ::= SEQUENCE {
        issuer Name,
        thisUpdate Time,
        distributionPoint DistributionPointName OPTIONAL OPTIONAL
}
      certificateListMatch MATCHING-RULE ::= {
        SYNTAX    CertificateListAssertion
        ID      id-mr-certificateListMatch }

      CertificateListAssertion ::= SEQUENCE {
        issuer                    Name OPTIONAL,
        minCRLNumber              [0]  CRLNumber OPTIONAL,
        maxCRLNumber              [1]  CRLNumber OPTIONAL,
        reasonFlags                 ReasonFlags OPTIONAL,
        dateAndTime                   Time OPTIONAL,
        distributionPoint         [2]  DistributionPointName OPTIONAL,
        authorityKeyIdentifier    [3]  AuthorityKeyIdentifier OPTIONAL}
      algorithmIdentifierMatch MATCHING-RULE ::= {
        SYNTAX    AlgorithmIdentifier
        ID      id-mr-algorithmIdentifierMatch }
      policyMatch MATCHING-RULE ::= {
        SYNTAX    PolicyID
        ID    id-mr-policyMatch }
      pkiPathMatch MATCHING-RULE ::= {
      SYNTAX    PkiPathMatchSyntax
      ID      id-mr-pkiPathMatch }
```

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA
VA DAVLAT NAZORATI

```
PkiPathMatchSyntax ::= SEQUENCE {
firstIssuer Name,
lastSubject Name }
-- Присвоение (назначение) идентификатора Объекта --
id-ce-subjectDirectoryAttributes     OBJECT IDENTIFIER ::= {id-ce
9}
id-ce-subjectKeyIdentifier           OBJECT IDENTIFIER ::= {id-ce
14}
id-ce-keyUsage                    OBJECT IDENTIFIER ::= {id-ce 15}
id-ce-privateKeyUsagePeriod          OBJECT IDENTIFIER ::= {id-ce
16}
id-ce-subjectAltName                 OBJECT IDENTIFIER ::= {id-ce
17}
id-ce-issuerAltName                OBJECT IDENTIFIER ::= {id-ce 18}
id-ce-basicConstraints             OBJECT IDENTIFIER ::= {id-ce 19}
id-ce-cRLNumber                    OBJECT IDENTIFIER ::= {id-ce 20}
id-ce-reasonCode                  OBJECT IDENTIFIER ::= {id-ce 21}
id-ce-instructionCode              OBJECT IDENTIFIER ::= {id-ce 23}
id-ce-invalidityDate              OBJECT IDENTIFIER ::= {id-ce 24}
id-ce-deltaCRLIndicator              OBJECT IDENTIFIER ::= {id-ce
27}
id-ce-issuingDistributionPoint       OBJECT IDENTIFIER ::= {id-ce
28}
id-ce-certificateIssuer           OBJECT IDENTIFIER ::= {id-ce 29}
id-ce-nameConstraints                OBJECT IDENTIFIER ::= {id-ce
30}
id-ce-cRLDistributionPoints          OBJECT IDENTIFIER ::= {id-ce
31}
id-ce-certificatePolicies         OBJECT IDENTIFIER ::= {id-ce 32}
id-ce-policyMappings                 OBJECT IDENTIFIER ::= {id-ce
33}
        OBJECT IDENTIFIER ::= {id-ce 34}
id-ce-authorityKeyIdentifier         OBJECT IDENTIFIER ::= {id-ce
35}
id-ce-policyConstraints              OBJECT IDENTIFIER ::= {id-ce
36}
id-ce-extKeyUsage                 OBJECT IDENTIFIER ::= {id-ce 37}
id-ce-cRLStreamIdentifier            OBJECT IDENTIFIER ::= {id-ce
40}
id-ce-cRLScope                    OBJECT IDENTIFIER ::= {id-ce 44}
id-ce-statusReferrals              OBJECT IDENTIFIER ::= {id-ce 45}
id-ce-freshestCRL                  OBJECT IDENTIFIER ::= {id-ce 46}
id-ce-orderedList                 OBJECT IDENTIFIER ::= {id-ce 47}
id-ce-baseUpdateTime                 OBJECT IDENTIFIER ::= {id-ce
51}
```

```
        id-ce-deltaInfo                 OBJECT IDENTIFIER ::= {id-ce 53}
        id-ce-inhibitAnyPolicy          OBJECT IDENTIFIER ::= {id-ce
54}

        -- правило соответствия OID --
        id-mr-certificateExactMatch         OBJECT IDENTIFIER ::= {id-
mr 34}
        id-mr-certificateMatch              OBJECT IDENTIFIER ::= {id-mr
35}
        id-mr-certificatePairExactMatch     OBJECT IDENTIFIER ::= {id-
mr 36}
        id-mr-certificatePairMatch          OBJECT IDENTIFIER ::= {id-mr
37}
        id-mr-certificateListExactMatch     OBJECT IDENTIFIER ::= {id-
mr 38}
        id-mr-certificateListMatch          OBJECT IDENTIFIER ::= {id-mr
39}
        id-mr-algorithmIdentifierMatch      OBJECT IDENTIFIER ::= {id-
mr 40}
        id-mr-policyMatch                   OBJECT IDENTIFIER ::= {id-mr
60}
        id-mr-pkiPathMatch                  OBJECT IDENTIFIER ::= {id-mr
62}
        -- Следующие Идентификаторы Объекта (OBJECT
IDENTIFIERS) не используются в
      этом стандарте:
       -- {id-ce 2}, {id-ce 3}, {id-ce 4}, {id-ce 5}, {id-ce 6}, {id-ce 7},
       -- {id-ce 8}, {id-ce 7}, {id-ce 8}, {id-ce 12}, {id-ce 13},
       -- {id-ce 22}, {id-ce 25}, {id-ce 26}
       END
       -- A.4 Модуль Структуры Сертификата Атрибута
       AttributeCertificateDefinitions {joint-iso-itu-t ds(5) module(1)
      attributeCertificateDefinitions(32) 4}
       DEFINITIONS IMPLICIT TAGS ::=
       BEGIN
       IMPORTS
         id-at, id-ce, id-mr, informationFramework,
authenticationFramework,
           selectedAttributeTypes, upperBounds, id-oc, certificateExtensions
           FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
           usefulDefinitions(0) 4}
         Name, RelativeDistinguishedName, ATTRIBUTE, Attribute,
           MATCHING-RULE, AttributeType, OBJECT-CLASS, top
           FROM InformationFramework informationFramework
         CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
```

STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

```
            EXTENSION, SIGNED, InfoSyntax, PolicySyntax, Extensions,
Certificate
            FROM AuthenticationFramework authenticationFramework
          DirectoryString, TimeSpecification, UniqueIdentifier
            FROM SelectedAttributeTypes selectedAttributeTypes
          GeneralName, GeneralNames, NameConstraintsSyntax,
certificateListExactMatch
            FROM CertificateExtensions certificateExtensions
          ub-name
            FROM UpperBounds upperBounds
          UserNotice
            FROM PKIX1Implicit93 {iso(1) identified-organization(3) dod(6)
internet(1)
      security(5) mechanisms(5)
        pkix(7) id-mod(0) id-pkix1-implicit-93(4)}
        ORAddress
          FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
          modules(0) mts-abstract-service(1) version-1999 (1) } ;
      AttributeCertificate ::= SIGNED {AttributeCertificateInfo}
      AttributeCertificateInfo ::= SEQUENCE
        {
        version AttCertVersion, -- version is v2
        holder       Holder,
        issuer       AttCertIssuer,
        signature       AlgorithmIdentifier,
        serialNumber      CertificateSerialNumber,
        attrCertValidityPeriod    AttCertValidityPeriod,
        attributes       SEQUENCE OFAttribute,
        issuerUniqueID      UniqueIdentifier OPTIONAL,
        extensions Extensions    OPTIONAL
        }
      AttCertVersion ::= INTEGER {v1(0), v2(1) }
      Holder ::= SEQUENCE
        {
        baseCertificateID [0] IssuerSerial OPTIONAL,
          -- издатель и серийный номер сертификата открытого ключа
хранителя
        entityName    [1] GeneralNames OPTIONAL,
          -- имя объекта или роли
        objectDigestInfo    [2] ObjectDigestInfo OPTIONAL}
      ObjectDigestInfo ::= SEQUENCE {
        digestedObjectType ENUMERATED {
          publicKey   (0),
          publicKeyCert    (1),
          otherObjectTypes   (2) },
```

```
  otherObjectTypeID OBJECT IDENTIFIER OPTIONAL,
  digestAlgorithm AlgorithmIdentifier,
  objectDigest BIT STRING }
AttCertIssuer ::= [0] SEQUENCE {
  issuerName GeneralNames   OPTIONAL,
  baseCertificateID    [0] IssuerSerial OPTIONAL,
  objectDigestInfo      [1] ObjectDigestInfo OPTIONAL }
-- По крайне мере одна компонента должна быть представлена
( WITH COMPONENTS { ..., issuerName PRESENT } |
WITH COMPONENTS { ..., baseCertificateID PRESENT } |
WITH COMPONENTS { ..., objectDigestInfo PRESENT } )
IssuerSerial ::= SEQUENCE {
  issuer     GeneralNames,
  serial     CertificateSerialNumber,
  issuerUID     UniqueIdentifier OPTIONAL }
AttCertValidityPeriod ::= SEQUENCE {
  notBeforeTime     GeneralizedTime,
  notAfterTime     GeneralizedTime }
AttributeCertificationPath ::= SEQUENCE {
  attributeCertificate   AttributeCertificate,
  acPath SEQUENCE OF ACPathData OPTIONAL }
ACPathData ::= SEQUENCE {
  certificate      [0]   Certificate OPTIONAL,
  attributeCertificate   [1]   AttributeCertificate OPTIONAL }
PrivilegePolicy ::= OBJECT IDENTIFIER
-- атрибуты привилегий --
role ATTRIBUTE ::= {
     WITH SYNTAX RoleSyntax
     ID id-at-role }
RoleSyntax ::= SEQUENCE {
  roleAuthority [0] GeneralNames OPTIONAL,
  roleName [1] GeneralName }
-- Классы объекта PMI --
pmiUser OBJECT-CLASS ::= {
  SUBCLASS OF {top}
  KIND auxiliary
  MAY CONTAIN {attributeCertificateAttribute}
  ID id-oc-pmiUser
  }
pmiAA OBJECT-CLASS ::= {
-- a PMI AA
SUBCLASS OF {top}
KIND auxiliary
MAY CONTAIN {aACertificate |
attributeCertificateRevocationList |
```

ZSTANDARD ASSENILIG'I
STANDARDLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
AGENTLIGI

```
    attributeAuthorityRevocationList}
    ID id-oc-pmiAA
    }
    pmiSOA OBJECT-CLASS ::= { -- источник полномочий PMI
      SUBCLASS OF {top}
      KIND auxiliary
      MAY CONTAIN {attributeCertificateRevocationList |
      attributeAuthorityRevocationList |
      attributeDescriptorCertificate}
      ID id-oc-pmiSOA
      }
    attCertCRLDistributionPt OBJECT-CLASS ::= {
      SUBCLASS OF {top}
      KIND auxiliary
      MAY CONTAIN { attributeCertificateRevocationList |
      attributeAuthorityRevocationList }
      ID id-oc-attCertCRLDistributionPts
      }
    pmiDelegationPath OBJECT-CLASS ::= {
      SUBCLASS OF {top}
      KIND auxiliary
      MAY CONTAIN { delegationPath }
      ID id-oc-pmiDelegationPath }
    privilegePolicy OBJECT-CLASS ::= {
      SUBCLASS OF {top}
      KIND auxiliary
      MAY CONTAIN {privPolicy }
      ID id-oc-privilegePolicy }
    -- Атрибуты директории  PMI --
    attributeCertificateAttribute ATTRIBUTE ::= {
      WITH SYNTAX AttributeCertificate
      EQUALITY MATCHING RULE attributeCertificateExactMatch
      ID id-at-attributeCertificate }
    aACertificate ATTRIBUTE ::= {
      WITH SYNTAX AttributeCertificate
      EQUALITY MATCHING RULE attributeCertificateExactMatch
      ID id-at-aACertificate }
    attributeDescriptorCertificate ATTRIBUTE ::= {
      WITH SYNTAX AttributeCertificate
      EQUALITY MATCHING RULE attributeCertificateExactMatch
      ID id-at-attributeDescriptorCertificate }
    attributeCertificateRevocationList ATTRIBUTE ::= {
      WITH SYNTAX CertificateList
      EQUALITY MATCHING RULE certificateListExactMatch
      ID id-at-attributeCertificateRevocationList}
```

```
        attributeAuthorityRevocationList ATTRIBUTE ::= {
          WITH SYNTAX CertificateList
          EQUALITY MATCHING RULE certificateListExactMatch
          ID id-at-attributeAuthorityRevocationList }
        delegationPath ATTRIBUTE ::= {
          WITH SYNTAX AttCertPath
          ID id-at-delegationPath }
        AttCertPath ::= SEQUENCE OF AttributeCertificate
        privPolicy ATTRIBUTE ::= {
          WITH SYNTAX PolicySyntax
          ID id-at-privPolicy }
        --Расширения сертификата атрибута и правила соответствия --
        attributeCertificateExactMatch MATCHING-RULE ::= {
          SYNTAX AttributeCertificateExactAssertion
          ID id-mr-attributeCertificateExactMatch }
        AttributeCertificateExactAssertion ::= SEQUENCE {
          serialNumber CertificateSerialNumber OPTIONAL,
          issuer IssuerSerial
          }
        attributeCertificateMatch MATCHING-RULE ::= {
          SYNTAX AttributeCertificateAssertion
          ID id-mr-attributeCertificateMatch }
        AttributeCertificateAssertion ::= SEQUENCE {
          holder      [0]  CHOICE {
            baseCertificateID  [0]  IssuerSerial,
            holderName    [1]  GeneralNames} OPTIONAL,
          issuer      [1]  GeneralNames OPTIONAL,
          attCertValidity    [2]  GeneralizedTime OPTIONAL,
          attType      [3]  SET OF AttributeType OPTIONAL}
        -- По крайне мере одна компонента последовательности должна
присутствовать
        holderIssuerMatch MATCHING-RULE ::= {
          SYNTAX    HolderIssuerAssertion
          ID      id-mr-holderIssuerMatch }
        HolderIssuerAssertion ::= SEQUENCE {
          holder  [0]  Holder OPTIONAL,
          issuer  [1]  AttCertIssuer OPTIONAL
          }
        delegationPathMatch MATCHING-RULE ::= {
          SYNTAX    DelMatchSyntax
          ID      id-mr-delegationPathMatch }
        DelMatchSyntax ::= SEQUENCE {
          firstIssuer AttCertIssuer,
          lastHolder Holder }
        sOAIdentifier EXTENSION ::= {
```

```
        SYNTAX    NULL
        IDENTIFIED BY   id-ce-sOAIdentifier }
      authorityAttributeIdentifier EXTENSION ::=
        {
        SYNTAX    AuthorityAttributeIdentifierSyntax
        IDENTIFIED BY  { id-ce-authorityAttributeIdentifier } }
      AuthorityAttributeIdentifierSyntax ::= SEQUENCE SIZE (1..MAX)
OF AuthAttId
      AuthAttId ::= IssuerSerial
      authAttIdMatch MATCHING-RULE ::= {
        SYNTAX    AuthorityAttributeIdentifierSyntax
        ID      id-mr-authAttIdMatch }
      roleSpecCertIdentifier EXTENSION ::=
        {
        SYNTAX RoleSpecCertIdentifierSyntax
        IDENTIFIED BY { id-ce-roleSpecCertIdentifier } }
      RoleSpecCertIdentifierSyntax ::= SEQUENCE SIZE (1..MAX) OF
    RoleSpecCertIdentifier
      RoleSpecCertIdentifier ::= SEQUENCE {
        roleName      [0]  GeneralName,
        roleCertIssuer    [1]  GeneralName,
        roleCertSerialNumber  [2]  CertificateSerialNumber OPTIONAL,
        roleCertLocator      [3]  GeneralNames OPTIONAL }
      roleSpecCertIdMatch MATCHING-RULE ::= {
        SYNTAX    RoleSpecCertIdentifierSyntax
        ID      id-mr-roleSpecCertIdMatch }
      basicAttConstraints EXTENSION ::=
        {
          SYNTAX    BasicAttConstraintsSyntax
          IDENTIFIED BY   { id-ce-basicAttConstraints }
        }
      BasicAttConstraintsSyntax ::= SEQUENCE
        {
          authority      BOOLEAN DEFAULT FALSE,
          pathLenConstraint   INTEGER (0..MAX) OPTIONAL
        }
      basicAttConstraintsMatch MATCHING-RULE ::= {
        SYNTAX   BasicAttConstraintsSyntax
        ID    id-mr-basicAttConstraintsMatch }
      delegatedNameConstraints EXTENSION ::= {
        SYNTAX    NameConstraintsSyntax
        IDENTIFIED BY   id-ce-delegatedNameConstraints }
      delegatedNameConstraintsMatch MATCHING-RULE ::= {
        SYNTAX   NameConstraintsSyntax
        ID    id-mr-delegatedNameConstraintsMatch}
```

```
        timeSpecification EXTENSION ::= {
          SYNTAX    TimeSpecification
          IDENTIFIED BY   id-ce-timeSpecification }
        timeSpecificationMatch MATCHING-RULE ::= {
          SYNTAX    TimeSpecification
          ID      id-mr-timeSpecMatch }
        acceptableCertPolicies EXTENSION ::= {
          SYNTAX    AcceptableCertPoliciesSyntax
          IDENTIFIED BY   id-ce-acceptableCertPolicies }

        AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF
CertPolicyId
        CertPolicyId ::= OBJECT IDENTIFIER
        acceptableCertPoliciesMatch MATCHING-RULE ::= {
          SYNTAX   AcceptableCertPoliciesSyntax
          ID    id-mr-acceptableCertPoliciesMatch }
        attributeDescriptor EXTENSION ::= {
          SYNTAX    AttributeDescriptorSyntax
          IDENTIFIED BY  {id-ce-attributeDescriptor } }
        AttributeDescriptorSyntax ::= SEQUENCE {
          identifier      AttributeIdentifier,
          attributeSyntax    OCTET STRING (SIZE(1..MAX)),
          name    [0]   AttributeName OPTIONAL,
          description    [1]   AttributeDescription OPTIONAL,
          dominationRule    PrivilegePolicyIdentifier}
        AttributeIdentifier ::= ATTRIBUTE.&id({AttributeIDs})
        AttributeIDs ATTRIBUTE ::= {...}
        AttributeName ::= UTF8String(SIZE(1..MAX))
        AttributeDescription ::= UTF8String(SIZE(1..MAX))
        PrivilegePolicyIdentifier ::= SEQUENCE {
          privilegePolicy   PrivilegePolicy,
          privPolSyntax   InfoSyntax }
        attDescriptor MATCHING-RULE ::= {
          SYNTAX    AttributeDescriptorSyntax
          ID      id-mr-attDescriptorMatch }
        userNotice EXTENSION ::= {
          SYNTAX SEQUENCE SIZE (1..MAX) OF UserNotice
          IDENTIFIED BY id-ce-userNotice }
          targetingInformation EXTENSION ::= {
          SYNTAX SEQUENCE SIZE (1..MAX) OF Targets
          IDENTIFIED BY id-ce-targetInformation }
        Targets ::= SEQUENCE SIZE (1..MAX) OF Target
        Target ::= CHOICE {
          targetName [0] GeneralName,
          targetGroup [1] GeneralName,
```

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT METROLOGIYASI

```
        targetCert [2] TargetCert }
      TargetCert ::= SEQUENCE {
        targetCertificate IssuerSerial,
        targetName GeneralName OPTIONAL,
        certDigestInfo ObjectDigestInfo OPTIONAL }
      noRevAvail EXTENSION ::= {
        SYNTAX NULL
        IDENTIFIED BY id-ce-noRevAvail }
      acceptablePrivilegePolicies EXTENSION ::= {
        SYNTAX AcceptablePrivilegePoliciesSyntax
        IDENTIFIED BY id-ce-acceptablePrivilegePolicies }
      AcceptablePrivilegePoliciesSyntax ::= SEQUENCE SIZE (1..MAX)
OF PrivilegePolicy
    -- присвоение (назначение) идентификатора объекта
    -- классы объекта --
      id-oc-pmiUser OBJECT IDENTIFIER ::= {id-oc 24}
      id-oc-pmiAA OBJECT IDENTIFIER ::= {id-oc 25}
      id-oc-pmiSOA OBJECT IDENTIFIER ::= {id-oc 26}
      id-oc-attCertCRLDistributionPts OBJECT IDENTIFIER ::= {id-oc
27}
      id-oc-privilegePolicy OBJECT IDENTIFIER ::= {id-oc 32}
      id-oc-pmiDelegationPath OBJECT IDENTIFIER ::= {id-oc 33}
      -- атрибуты директории --
      id-at-attributeCertificate OBJECT IDENTIFIER ::= {id-at 58}
      id-at-attributeCertificateRevocationList OBJECT IDENTIFIER ::=
{id-at 59}
      id-at-aACertificate OBJECT IDENTIFIER ::= {id-at 61}
      id-at-attributeDescriptorCertificate OBJECT IDENTIFIER ::= {id-at
62}
      id-at-attributeAuthorityRevocationList OBJECT IDENTIFIER ::=
{id-at 63}
      id-at-privPolicy OBJECT IDENTIFIER ::= {id-at 71}
      id-at-role OBJECT IDENTIFIER ::= {id-at 72}
      id-at-delegationPath OBJECT IDENTIFIER ::= {id-at 73}
      --расширение сертификата атрибута --
        id-ce-authorityAttributeIdentifier OBJECT IDENTIFIER ::= {id-ce
38}
      id-ce-roleSpecCertIdentifier OBJECT IDENTIFIER ::= {id-ce 39}
      id-ce-basicAttConstraints OBJECT IDENTIFIER ::= {id-ce 41}
      id-ce-delegatedNameConstraints OBJECT IDENTIFIER ::= {id-ce
42}
      id-ce-timeSpecification OBJECT IDENTIFIER ::= {id-ce 43}
      id-ce-attributeDescriptor OBJECT IDENTIFIER ::= {id-ce 48}
      id-ce-userNotice OBJECT IDENTIFIER ::= {id-ce 49}
      id-ce-sOAIdentifier OBJECT IDENTIFIER ::= {id-ce 50}
```

```
        id-ce-acceptableCertPolicies OBJECT IDENTIFIER ::= {id-ce 52}
        id-ce-targetInformation OBJECT IDENTIFIER ::= {id-ce 55}
        id-ce-noRevAvail OBJECT IDENTIFIER ::= {id-ce 56}
        id-ce-acceptablePrivilegePolicies OBJECT IDENTIFIER ::= {id-ce
57}
        -- правила соответствия PMI --
        id-mr-attributeCertificateMatch OBJECT IDENTIFIER ::= {id-mr
42}
        id-mr-attributeCertificateExactMatch OBJECT IDENTIFIER ::= {id-
mr 45}
        id-mr-holderIssuerMatch OBJECT IDENTIFIER ::= {id-mr 46}
        id-mr-authAttIdMatch OBJECT IDENTIFIER ::= {id-mr 53}
        id-mr-roleSpecCertIdMatch OBJECT IDENTIFIER ::= {id-mr 54}
        id-mr-basicAttConstraintsMatch OBJECT IDENTIFIER ::= {id-mr
55}
        id-mr-delegatedNameConstraintsMatch OBJECT IDENTIFIER ::=
{id-mr 56}
        id-mr-timeSpecMatch OBJECT IDENTIFIER ::= {id-mr 57}
        id-mr-attDescriptorMatch OBJECT IDENTIFIER ::= {id-mr 58}
        id-mr-acceptableCertPoliciesMatch OBJECT IDENTIFIER ::= {id-
mr 59}
        id-mr-delegationPathMatch OBJECT IDENTIFIER ::= {id-mr 61}
        END
```

Приложение В
(обязательное)

Правила генерации и обработки CRL

В.1 Введение

В.1.1 Типы CRL

Доверяющей стороне (пользователь сертификата) необходимо иметь возможность проверять статус сертификата для того, чтобы облегчить принятие решения о достоверности этого сертификата. CRL являются одним из механизмов, используемых доверяющими сторонами, чтобы получить эту информацию.

Также могут быть использованы и другие механизмы, обсуждение которых находится за областью рассмотрения данного стандарта.

Данное приложение описывает использование доверяющей стороной списков аннулированных сертификатов для проверки статуса аннулирования проверяемого сертификата. Различные органы могут иметь различную политику по отношению издания списков аннулирования. Некоторые органы могут объединить аннулирование сертификата конечного объекта и сертификатов ЦР в один список, в то время как другие органы могут разделить их на различные списки. Некоторые органы могут разделить их совокупность (семейство) сертификата на фрагменты CRL, а некоторые органы могут издать дельта обновления на список аннулирования между регулярными интервалами CRL. В результате доверяющим сторонам необходимо иметь возможность определения границ CRL, которые они восстанавливают, чтобы дать им возможность гарантировать, что они имеют полный набор информации об аннулировании. Расширение clrScope может быть использовано как один из механизмов для определения области. Данное приложение предоставляет механизм, который может быть использован в случае отсутствия расширения crlScope в CRL.

Данное приложение написано для проверки статуса аннулирования сертификатов открытого ключа, использующих CRL, CRL конечного объекта и CARL. Однако, это описание также может быть применено для проверки статуса аннулирования сертификатов атрибута, использующих ACRL и AARL. В данном приложении ACRL может рассматриваться вместо CRL и AARL вместо CARL.

Доверяющей стороне могут быть доступны один или более одного CRL следующих типов основанных на аспектах аннулирования:

- полный и завершенный CRL;
- полный и завершенный EARL;
- полный и завершенный CARL;
- пункт распределения CRL, ACRL конечного объекта или CARL;
- косвенный CRL, ACRL конечного объекта или CARL;
- dCRL, ACRL конечного объекта или CARL;

- косвенный dCRL, ACRL конечного объекта или CARL.

Полный и завершенный CRL - это список всех аннулированных сертификатов конечных объектов и сертификатов ЦР, которые были изданы по различным причинам со стороны органа.

Полный и завершенный ACRL конечного объекта - это список всех аннулированных сертификатов конечных объектов, которые были изданы со стороны органа по различным причинам.

Полный и завершенный CARL - это список аннулированных сертификатов ЦР, которые были изданы со стороны органа по различным причинам.

Пункт распределения CRL, ACRL конечного объекта или CARL это пункт, который охватывает все или подмножество сертификатов, изданных со стороны органа.

Подмножество может основывается на различных критериях.

Косвенный CRL, ACRL конечного объекта или CARL (iCRL) - это CRL, который содержит список аннулированных сертификатов, в котором некоторые или все из сертификатов не были изданы со стороны органа, подписывающего и издающего CRL.

dCRL, ACRL конечного объекта или CARL - это CRL, который содержит только изменения CRL и который является завершенным для данной области.

Необходимо заметить, что сосланный CRL может быть единственным CRL, который является завершенным для данной области, или может являться dCRL, который используется для локального построения CRL, который является завершенным для данной области. Все перечисленные выше типы CRL (за исключением dCRL) это такие CRL, которые являются завершенными, для своих областей. dCRL будет использован совместно с ассоциированным CRL, который является завершенным для той же области, для того, чтобы формировать полное представление о статусе аннулирования сертификатов.

Косвенный dCRL, EARL или CARL является CRL, который содержит только изменения к одному или к нескольким наборам CRL, которые являются завершенными для своих областей и в которых несколько или все из этих сертификатов не могут быть изданы со стороны органа, который и подписывает и издает этот CRL.

Внутри этого приложения «Область CRL» определена двумя независимыми измерениями. Первое измерение это набор сертификатов, охватываемых CRL. Второе измерение это набор кодов причин, охватываемых CRL. Область CRL может быть определена одним или несколькими из следующих путей:

- из расширения IDP в CRL;
- из расширения области применения CRL в CRL;
- другими способами, выходящими за пределы рассмотрения данного стандарта.

O‘ZBEKISTON RESPUBLIKASI
STANDARTLASHTIRISH,
METROLOGIYA VA SERTIFIKATLASHTIRISH
DAVLAT QO‘MITASI

B.1.2 Обработка CRL

Если доверяющие стороны используют CRL как механизм определения аннулирования сертификата, то они должны быть уверены, что используют соответствующий CRL для этого сертификата. Это приложение описывает процедуру получения и обработки соответствующих CRL посредством нескольких специфических шагов. Реализация будет функционально эквивалентной внешнему режиму работы, получающемуся в результате этой процедуры. Алгоритм, используемый специфической реализацией для получения правильных выходных параметров (т.е. статуса аннулирования сертификата) из данных входных параметров (сам сертификат и входные параметры из локальной политики), не стандартизируется.

Данное приложение не включает процедуры для следующих указателей конструкции CRL, содержащей расширение statusReferrals. Любой CRL, содержащий это расширение, не будет использован в качестве источника для проверки статуса аннулирования любого сертификата доверяющей стороной. Скорее всего, CRL, содержащий это расширение может быть использован со доверяющей стороной, как дополнительный инструмент для нахождения соответствующих CRL для проверки статуса аннулирования.

B.2 Определение параметров для CRL

Информация, расположенная в самом сертификате, так же как и информация из политики, согласно которой действует доверяющая сторона, предоставляют параметры для определения соответствия кандидатов CRL. Следующая информация необходима для определения типов CRL, являющихся подходящими:

- тип сертификата (т.е. сертификат конечного объекта или ЦР);
- пункт распределения критического CRL;
- критический самый новый CRL;
- код причины важности.

Тип сертификата может быть определен из расширений базового ограничения сертификата. Если расширение существует, то оно определяет то, что является ли сертификат сертификатом ЦР или сертификатом конечного объекта. Если расширение отсутствует, то тип сертификата рассматривается, как сертификат конечного объекта. Эта информация необходима для определения того, что можно ли использовать CRL, EARL или CARL для проверки сертификата на аннулирование.

Если сертификат содержит критическое расширение пункта распределения CRL, то система обработки сертификата доверяемой стороны должна понимать это расширение для того, чтобы возможно было доверять сертификату. Доверие на полный CRL, может быть не достаточным.

Если сертификат содержит расширение критического самого нового CRL, то доверяющая сторона не может использовать сертификат без предварительного получения и проверки самого нового CRL.

Коды причины важности определяются политикой и обычно поддерживаются приложениями. Они должны включать в себя все коды причины. Эта информация необходима для определения, какие CRL являются достаточными относительно кода причины.

Нужно заметить, что политика также может определять полагается ли проверять доверяемой стороне dCRL на статус аннулирования, даже когда расширение freshnestCRL установлено как некритическое или отсутствует в сертификате. Несмотря на то, что обработка этих необязательных dCRL исключена из этого шага, они описаны в шаге 4).

B.3 Определение необходимых CRL

B.3.1 Конечный объект с критическим CRL DP

Значения параметров, описанных в B.2 определяют критерии, в которых требуются типы CRL для определения статуса аннулирования данного сертификата. Определение типов CRL может быть осуществлено на основе следующего набора критериев, как это описано ниже в B.3.1 - B.3.4:

- сертификат конечного объекта с предъявленным критическим CRL DP;
- сертификат конечного объекта без предъявленного критического CRL DP;
- сертификат ЦР с предъявленным критическим CRL DP;
- сертификат ЦР без предъявленного критического CRL DP.

Обработка остальных параметров приводиться внутри каждого раздела.

Если сертификат является сертификатом конечного объекта и в нём присутствует расширение cRLDistributionPoints, а также если это расширение отмечено как критическое, то будут получены следующие CRL:

a) CRL из одного из назначенных пунктов распределения CRL, который охватывает один или несколько кодов причин важности;

b) если все коды причин важности не охвачены этим CRL, статус аннулирования для остальных кодов важности может быть удовлетворен комбинацией любой, из следующих CRL:

- дополнительный пункт распределения CRL;
- дополнительный полный CRL;
- дополнительные полный EARL.

Если расширение самого нового CRL также присутствует в сертификате и отмечено как критическое, то один или несколько CRL будут также получены из одного или нескольких пунктов распределения, назначенных в этом расширении.

B.3.2 Конечный объект с некритическим CRL DP

Если сертификат является сертификатом конечного объекта, и расширение cRLDistributionPoints отсутствует в сертификате, или же присутствует, но не отмечено как критическое, то статус аннулирования для

кодов причин важности может быть удовлетворен любой комбинацией следующих CRL:

- CRL пунктов распределения (если присутствует);
- полный CRL;
- полный EPRL.

Если расширение самого нового CRL также присутствует в сертификате и если отмечено как критическое, один или несколько CRL будут также получены из одного или нескольких назначенных в этом расширении пунктов распределения.

B.3.3 ЦР с критическим CRL DP

Если сертификат является сертификатом DP и расширение cRLDistributionPoints присутствует в сертификате и отмечено как критическое, то будут получены следующие CRL/CARL:

a) CRL или CARL из одного назначенного пункта распределения, который покрывает один или несколько кодов причин важности;

b) если все коды причин важности не охвачены этим CRL/CARL, статус аннулирования для остальных кодов важности может быть удовлетворен любой комбинацией следующих CRL/CARL:

- дополнительный пункт распределения CRL/CARL;
- дополнительные полный CRL;
- дополнительный полный CARL.

Если расширение самого нового CRL также присутствует в сертификате и отмечено как критическое, то один или более CRL/CARL также будут получены из одного или более назначенных в этом расширении пунктов распределения.

B.3.4 ЦР без критического CRL DP

Если сертификат является сертификатом ЦР и расширение cRLDistributionPoints отсутствует в сертификате или же присутствует, но отмечено как некритическое, то статус аннулирования для кодов причин важности могут быть удовлетворены любой комбинацией следующих CRL:

- пункт распределения CRL/CARL (если присутствует);
- полный CRL;
- полный CARL.

Если расширение самого нового CRL также присутствует в сертификате и отмечено как критическое, то один или более CRL/CARL будут также получены из одного или нескольких назначенных в этом расширении пунктов распределения.

B.4 Получение CRL

Если доверяющая сторона получает подходящие CRL из директории, то эти CRL на самом деле получаются (извлекаются) из

CRL DP или из элемента директории издателя сертификата, извлекая соответствующие атрибуты, т.е. из следующих атрибутов:

- список аннулированных сертификатов;
- список аннулированных органов;
- дельта список аннулирования.

B.5 Обработка CRL

B.5.1 Утверждение базовой области CRL

B.5.1.1 Полный CRL

После рассмотрения параметров, приведенных в B.2, определение подходящих типов CRL, как это описано в B.3 и получение подходящего набора CRL как это описано в B.4, доверяющая сторона готова обработать CRL. Набор CRL будет содержать по крайне мере один базовый CRL и может также содержать один или более dCRL. При каждой обработке CRL, доверяющая сторона должна гарантировать, что CRL является правильным по отношению к своей области. Доверяющая сторона к этому времени уже определит, что CRL является подходящим для области интересующего сертификата по результатам обработки, приведенным в B.2 и B.3. Кроме того, проверки на действительность будут вестись над CRL и они будут проверяться для определения, является ли сертификат аннулированным. Эти проверки описаны в B.5.1 - B.5.4.

Как описано в B.3, возможно будет существовать более чем один тип CRL, который можно будет использовать как базовый CRL для проверки статуса аннулирования сертификата. В зависимости от политики издающего органа по отношению издания CRL, доверяющая сторона может иметь один или несколько следующих базовых типов CRL доступных для него:

- завершенный CRL для всех объектов;
- завершенный EARL;
- завершенный CARL;
- пункт распределения основанный на CRL/EPRL/CARL.

Для того, чтобы определить, что CRL является полным для сертификата конечного объекта и сертификата ЦР для всех кодов причин важности, должны удовлетворятся следующие условия:

- расширение индикатора дельта CRL должно отсутствовать;
- расширение издающего пункта распределения может отсутствовать;
- расширение издающего пункта распределения не должно содержать поле пункта распределения;
- расширение издающего пункта распределения не должно содержать поле onlyContainsUserCerts с установленным значением TRUE;
- расширение издающего пункта распределения не должно содержать поле onlyContainsAuthorityCerts с установленным значением TRUE;
- расширение издающего пункта распределения не должно содержать поле onlyContainsAttributCerts с установленным значением TRUE;

O'ZBEKISTON AGENTLIGI
STANDARTLASHTIRISH,
METROLOGIYA VA SIFAT NAZORATI

- если поле reasonCodes присутствует в расширении издающего пункта распределения, то поле кодов причин должно включать все причины важности для приложения;

- расширение издающего пункта распределения должно или не должно содержать поле indirectCRL (следовательно, это поле необязательно проверять).

B.5.1.2 Завершенный EARL

Для того чтобы определить, что CRL является завершенным EARL для кодов причин важности, должны удовлетворятся все следующие условия:

- расширение индикатора дельта CRL должно отсутствовать;

- расширение издающего пункта распределения должно присутствовать;

- расширение издающего пункта распределения не должно содержать поле пункта распределения;

- расширение издающего пункта распределения должно содержать поле onlyContainsUserCerts. Значение этого поля должно быть установлено на TRUE;

- расширение издающего пункта распределения не должно содержать поле onlyContainsAuthorityCerts с установленным значением TRUE;

- расширение издающего пункта распределения не должно содержать поле onlyContainsAttributeCerts с установленным значением TRUE;

- если поле reasonCodes присутствует в расширении издающего пункта распределения, то поле кодов причин должно включать все причины важности для приложения;

- расширение издающего пункта распределения может или не может содержать поле indirectCRL (следовательно, это поле необязательно проверять).

Этот CRL может быть использован, только если доверяющая сторона определила, что сертификат субъекта является сертификатом конечного объекта. Таким образом, для сертификатов версии 3, если сертификат субъекта содержит расширение basicConstraints, то его значение должно быть cA=FALSE.

B.5.1.3 Завершенный CARL

Для того чтобы определить, что CRL является завершенным CARL для кодов причин важности, должны удовлетворятся все следующие условия:

- расширение индикатора дельта CRL должно отсутствовать;

- должно присутствовать расширение Издающего пункта распределения;

- расширение издающего пункта распределения не должно содержать поле пункта распределения;

- расширение издающего пункта распределения не должно содержать поле onlyContainsUserCerts с установленным значением TRUE;

- расширение издающего пункта распределения не должно содержать поле onlyContainsAttributeCerts с установленным значением TRUE;

- расширение издающего пункта распределения должно содержать в себе поле onlyContainsAuthorityCerts. Значение этого поля должно быть установлено на TRUE;

- если поле reasonCodes присутствует в расширении издающего пункта распределения, то поле кодов причин должно включать все причины важности для приложения;

- расширение издающего пункта распределения может или не может содержать в себе поле inderectCRL (следовательно, это поле необязательно проверять).

Этот CARL может быть использован, только, если сертификат субъекта является сертификатом ЦР. Таким образом, для сертификата версии 3 сертификат субъекта должен содержать расширение basicConstraints со значением cA=TRUE.

B.5.1.4 Пункт распределения основанный на CRL/EPRL/CARL

Для того, чтобы определить, что CRL является одним из CRL, указанных расширением пункта распределения CRL в сертификате, должны удовлетворяться все следующие условия, если:

- поле пункта распределения в расширении издающего пункта распределения CRL должно отсутствовать (только в тех случаях, когда не ожидается критический CRL DP), или одно из имен в поле пункта распределения расширения пункта распределения CRL сертификата соответствует одному из имен в поле пункта распределения расширения издающего пункта распределения CRL;

- сертификат является сертификатом конечного объекта, то CRL не должен содержать поле onlyContainsAuthorityCerts с установленным значением TRUE в расширении CRL издающего пункта распределения;

- в расширении CRL издающего пункта распределения значение onlyContstraintsAuthorityCerts установлено на TRUE, то проверяемый сертификат должен содержать расширение basicConstraints с компонентой cA и установленным значением TRUE;

- поле кодов причин присутствует в расширении пункта распределения CRL сертификата, то это поле должно либо отсутствовать в расширении CRL издающего пункта распределения или содержать по крайне мере один из кодов причин, предъявленных в расширении пункта распределения CRL сертификата;

- поле cRLIssuer отсутствует в расширении пункта распределения CRL сертификата, то CRL должен быть подписан тем ЦР, который подписал сертификат;

- поле cRLIssuer присутствует в расширении пункта распределения CRL сертификата, то CRL должен быть подписан со стороны издателя CRL, установленного в расширении пункта распределения CRL сертификата и

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

CRL должен содержать поле indirectCRL в расширении издающего пункта распределения.

B.5.2 Утверждение области дельта CRL

Доверяющая сторона может также быть проверяющим (контролирующим) dCRL, либо потому, что это затребовано посредством критического расширения freshestCRL в сертификате или потому что политика, согласно которой работает доверяющая сторона, поддерживает проверку dCRL.

Доверяющая сторона может быть всегда уверенной, что она имеет соответствующую информацию CRL для сертификата, если будут выполняться все следующие условия:

- базовый CRL, используемый доверяющей стороной, соответствует сертификату (в рамках области);
- dCRL, используемый доверяющей стороной, соответствует сертификату (в рамках области);
- базовый CRL был издан в одно и тоже время или позже чем базовый CRL, сосланный со стороны dCRL.

Для того чтобы определить, что dCRL является соответствующим для сертификата, должны выполнятся все следующие условия:

- расширение индикатора дельта CRL должно присутствовать;
- dCRL должен быть изданным после CRL. Один из путей гарантирования этого, это проверка того, что номер CRL в расширении crlNumber dCRL больше, чем номер CRL в расширении crlNumber базового CRL, который используется доверяющей стороной, и поля cRLStreamIdentifier базового CRL и dCRL соответствуют. Другой путь - это сравнение полей thisUpdate базового CRL и dCRL, имеющийся у доверяющей стороны;
- базовый CRL, используемый доверяющей стороной, должен быть dCRL, который издан до базового CRL или позже него. Один из путей гарантирования этого - это проверка того, что номер CRL в расширении deltaCRLIndicator dCRL меньше чем или равно номеру CRL в расширении crlNumber базового CRL, который используется доверяющей стороной и поля cRLStreamIdintifier в базовом CRL и dCRL соответствуют. Другим путем является сравнение полей thisUpdate базового CRL, который имеется у доверяющей стороны, и базового CRL, указанного со стороны dCRL;
- если dCRL содержит расширение издающего пункта распределения, то область издающего пункта распределения будет совместимой с сертификатом, как это описано в B.5.1.4;
- если dCRL содержит расширение область CRL, то сертификат будет в области этого CRL;
- если dCRL не содержит какую-нибудь из таких расширений как streamIdentifier, crlScope и issuingDistributionPoint, то он будет использован только в сочетании с полным и завершенным базовым CRL.

B.5.3 Проверка действительности и употребительности базового CRL

Для того чтобы проверить, что базовый CRL является точным и не изменен с тех пор, как его издали, все следующие условия должны выполнятся:

- доверяющая сторона должна иметь возможность получить открытый ключ издателя, установленного в CRL, используя удостоверенные способы;
- подпись на базовом CRL должна быть проверена, используя удостоверенный открытый ключ;
- если поле nextUpdate присутствует, то текущее время должно быть более ранним, чем поле nextUpdate;
- имя издателя в CRL должно совпадать с именем издателя в сертификате, который проверяется на аннулирование, до тех пор, пока CRL не извлечен из CRL DP в сертификате и расширение CRL DP содержит компоненту издателя CRL. В этом случае, одно из имен в компоненте издателя CRL в расширении CRL DP должно соответствовать имени издателя в CRL.

B.5.4 Действительность и контроль над dCRL

Для того чтобы проверить, что dCRL является точным и не был изменен с тех пор как был издан, должны выполнятся все следующие условия:

- доверяющая сторона должна иметь возможность получить открытый ключ издателя, установленного в CRL, используя удостоверенные способы;
- подпись на дельта CRL должна быть проверена, используя удостоверенный открытый ключ;
- если поле nextUpdate присутствует, то текущее время должно быть более ранним, чем поле nextUpdate;
- имя издателя в CRL должно совпадать с именем издателя в сертификате, который проверяется на аннулирование, до тех пор, пока выполняются следующие условия:

a) дельта CRL извлечен из CRL DP в сертификате и расширение CRL DP содержит компоненту издателя CRL. В этом случае, одно из имен в компоненте издателя CRL в расширении CRL DP должно соответствовать имени издателя в CRL;

b) дельта CRL извлечен из расширения freshestCRL в сертификате и расширение CRLScope в dCRL содержит компоненту PerAuthorityScope с authorityName, который соответствует имени издателя сертификата.

O‘ZSTANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI

Приложение С
(обязательное)

Эталонное определение алгоритма идентификаторов объекта

Данное приложение определяет идентификаторы объекта, присвоенные алгоритмам аутентификации и шифрования, при отсутствии формального регистра. Оно предназначено для того, чтобы использовать такой регистр, как только он будет доступным. Определение имеет форму модуля языка ASN.1, AlgorithmObjectIdentifier.

```
AlgorithmObjectIdentifiers {joint-iso-itu-t ds(5) module(1)
algorithmObjectIdentifiers(8) 4}
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
IMPORTS
algorithm, authenticationFramework
FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
usefulDefinitions(0) 4}
ALGORITHM
FROM AuthenticationFramework authenticationFramework ;
-- категории идентификатора объекта --
encryptionAlgorithm OBJECT IDENTIFIER ::= {algorithm 1}
hashAlgorithm OBJECT IDENTIFIER ::= {algorithm 2}
signatureAlgorithm OBJECT IDENTIFIER ::= {algorithm 3}
-- синонимы --
id-ea OBJECT IDENTIFIER ::= encryptionAlgorithm
id-ha OBJECT IDENTIFIER ::= hashAlgorithm
id-sa OBJECT IDENTIFIER ::= signatureAlgorithm
-- алгоритмы --
rsa ALGORITHM ::= {
KeySize
IDENTIFIED BY id-ea-rsa }
KeySize ::= INTEGER
-- присвоение (назначение) идентификатора объекта --
id-ea-rsa OBJECT IDENTIFIER ::= {id-ea 1}
id-ha-sqMod-n OBJECT IDENTIFIER ::= {id-ha 1}
id-sa-sqMod-nWithRSA OBJECT IDENTIFIER ::= {id-sa 1}
END
```

Библиография

[1]	ITU-T X.500:2008	Информационные технологии. Взаимосвязь открытых систем. Директория. Обзор концепций, моделей и услуг
[2]	ITU-T X.509 v.3:2008	Информационные технологии. Взаимосвязь открытых систем. Директория. Структура сертификата открытого ключа и сертификата атрибута
[3]	RFC 2459 :2008	Интернет X.509. Форматы сертификатов и списков отозванных сертификатов инфраструктуры открытых ключей
[4]	ITU-T X.660:2008 | ISO/IEC 9834-1:2009	Информационные технологии. Взаимосвязь открытых систем. Процедуры для работы регистрационных органов в системе OSI. Общие процедуры и высшие разряды дерева идентификаторов объекта ASN.1
[5]	ITU-T X.681:2008	Информационные технологии. Нотация абстрактного синтаксиса один (ASN.1). Спецификация информационных объектов
[6]	ITU-T X.813:1996 | ISO/IEC 9834-1:1997	Информационные технологии. Взаимосвязь открытых систем. Основы безопасности для открытых систем. Основы неотказуемости
[7]	ITU-T X.400:1999	Обзор систем обработки сообщений и служб

STANDART AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA SIFAT NAZORATI

УДК: 621.321 ОКС 35.240

Ключевые слова: сертификат открытого ключа, сертификат атрибута, идентификатор, директория, промежуточный сертификат, владелец, обработка, инициализация, аутентификация, расширение, процедура, схема, ограничение, дескриптор, распределение, структура, привилегия, делегирование.

Заместитель директора
ГУП «UNICON.UZ» _____ Х. Хасанов

Начальник Центра регистрации ключей электронной цифровой подписи _____ Э. Исаков

Инженер 2 категории Центра регистрации ключей электронной цифровой подписи _____ Р. Миндуллина

Нормоконтроль _____ Л. Шаймарданова
24.02.2011

СОГЛАСОВАНО

Начальник отдела производства и рынка программных продуктов Узбекского агентства связи и информатизации

З. Рахматуллаев
Письмо от 01.02.2011г.
№ 17-8/412

СОГЛАСОВАНО

Начальник государственной инспекции связи

Ш. Атамухамедов
Письмо от 14.01.2011г.
№ 32-12/54

O'ZBEKISTON ALOQA AGENTLIGI
STANDARTLASHTIRISH,
MUVOFIQLASHTIRISH
VA DAVLAT NAZORATI
BOSHQARMASI