ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

СПРАВОЧНИК

ЧАСТЬ 8

ОСНОВЫ АУТЕНТИФИКАЦИИ

Издание официальное

БЗ 3—98/482

ГОССТАНДАРТ РОССИИ
Москва

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Московским научно-исследовательским центром (МНИЦ) Государственного Комитета Российской Федерации по связи и информатизации

ВНЕСЕН Техническим Комитетом по стандартизации ТК 22 «Информационные технологии»

2 ПРИНЯТ И ВВЕДЕН В ДЕЙСТВИЕ Постановлением Госстандарта России от 19 мая 1998 г. № 215

3 Настоящий стандарт содержит полный аутентичный текст международного стандарта ИСО/МЭК 9594-8—94 «Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 8. Основы аутентификации»

4 ВВЕДЕН ВПЕРВЫЕ

© ИПК Издательство стандартов, 1998

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Госстандарта России

Введение

Настоящий стандарт разработан с целью обеспечения взаимосвязи систем обработки информации, предназначенных для предоставления услуг справочника. Совокупность подобных систем вместе с содержащейся в них информацией справочника может рассматриваться как единое целое, называемое справочником. Информация, хранимая справочником и называемая в целом «информационной базой справочника» (ИБС), используется обычно для обеспечения обмена данными между такими объектами, как логические объекты прикладного уровня, персонал, терминалы и дистрибутивные списки.

Справочник играет существенную роль во взаимосвязи открытых систем (ВОС), цель которой состоит в том, чтобы при минимуме технических согласований вне стандартов по ВОС обеспечить взаимосвязь систем обработки информации:

- поставляемых от различных изготовителей;
- использующих различные методы административного управления;
- имеющих различные уровни сложности;
- использующих различные технологии.

Для многих применений требуются средства защиты от угроз нарушения обмена информацией. Некоторые наиболее известные угрозы вместе с услугами защиты и механизмами, которые могут быть использованы для защиты от них, кратко изложены в приложении В. Фактически все услуги защиты зависят от идентичности взаимодействующих сторон хорошо друг другу известных, то есть от их аутентичности.

Настоящий стандарт определяет основы услуг аутентификации, предоставляемых справочником своим пользователям. К этим пользователям относится сам справочник, а также другие прикладные программы и услуги. Справочник может оказаться полезным при обеспечении других потребностей в аутентификации и других услуг защиты, поскольку он представляет собой естественное место, из которого взаимодействующие стороны могут получить информацию друг о друге, то есть сведения, являющиеся основой аутентификации. Справочник — это естественное место потому, что он хранит и другую информацию, необходимую для обмена и полученную до его осуществления. При таком подходе получение из справочника информации аутентификации потенциального партнера подобно получению адреса. Предполагается, что вследствие широкой доступности справочника для целей обмена эти основы аутентификации будут широко использованы многими прикладными программами.

В обязательном приложении А представлен модуль АСН.1, в котором содержатся все определения, используемые в основных положениях аутентификации.

В приложении В изложены требования защиты.

В приложении С приведены вводные сведения о криптографии на основе ключа общего пользования.

В приложении D изложена криптосистема ключа общего пользования RSA.

В приложении Е описаны хеш-функции.

В приложении F описана защита от угроз методом строгой аутентификации.

В приложении G изложена информация относительно конфиденциальности данных.

В приложении Н приведены идентификаторы объектов, присвоенные в алгоритмах аутентификации и шифрования при отсутствии формального регистра.

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ

СПРАВОЧНИК

Часть 8

Основы аутентификации

Information technology. Open Systems Interconnection. The directory. Part 8. Authentication framework

Дата введения 1999—01—01

ГЛАВА 1. ОБЩЕЕ ОПИСАНИЕ

1 **Область применения**

Настоящий стандарт:

- определяет формат информации аутентификации, хранимой справочником;
- описывает способ получения из справочника информации аутентификации;
- устанавливает предпосылки о способах формирования и размещения в справочнике информации аутентификации;
- определяет три способа, с помощью которых прикладные программы могут использовать такую информацию аутентификации для выполнения аутентификации, и описывает, каким образом с помощью аутентификации могут быть обеспечены другие услуги защиты.

В настоящем стандарте изложены два вида аутентификации: простая, использующая пароль как проверку заявленной идентичности, и строгая, использующая удостоверения личности, созданные с использованием криптографических методов. Если простая аутентификация предлагает некоторые ограниченные возможности защиты от несанкционированного доступа, то в качестве основы услуг, обеспечивающих защиту, должна использоваться только строгая аутентификация. Здесь не ставится задача установить эти методы в качестве общей основы аутентификации, но они могут получить общее применение со стороны тех прикладных программ, которые рассматривают эти методы адекватными.

Аутентификация (и другие услуги защиты) могут быть предложены только в контексте определенной стратегии защиты. Пользователи прикладной программы должны сами определить свою собственную стратегию защиты, которая может быть ограничена услугами, предусмотренными стандартом.

Задача стандартов, определяющих прикладные программы, которые используют основы аутентификации, состоит в том, чтобы установить правила протокольных обменов, которые необходимо осуществить для обеспечения аутентификации, основываясь на информации аутентификации, полученной из справочника. Протоколом, используемым прикладными программами для получения удостоверения личности из справочника, является протокол доступа к справочнику (ПДС), определенный в ИСО/МЭК 9594-5.

Метод строгой аутентификации, определенный в настоящем стандарте, основан на криптосистемах ключа общего пользования. Главное преимущество таких систем состоит в том, что сертификаты пользователей могут храниться в справочнике в виде атрибутов, свободно участвовать в обме-

Издание официальное

не внутри системы справочника и могут быть получены пользователями справочника таким же способом, как и любая другая информация справочника. Предполагается, что сертификаты пользователей должны формироваться «автономными» средствами и вводиться в справочник их создателями. Выработка сертификатов пользователей выполняется некоторыми автономными уполномоченными по сертификации (УС), которые полностью отделены от агентов системы в справочнике. В частности, поставщикам справочника не предъявляется никаких специальных требований по записи или обмену сертификатами пользователей надежным способом.

Краткое описание криптографии ключа общего пользования приведено в приложении С.

В общем случае основы аутентификации не зависят от использования конкретного алгоритма криптографии, при условии, что он обладает свойствами, приведенными в 7.1. Потенциально может быть использовано несколько различных алгоритмов. Однако два пользователя, желающие осуществить аутентификацию, должны придерживаться одного и того же алгоритма криптографии. Таким образом, в контексте набора соответствующих прикладных программ выбор одного алгоритма позволит расширить семейство пользователей, способных к аутентификации и надежной передаче. Пример алгоритма криптографии ключа общего пользования приведен в приложении D.

Точно также два пользователя, желающие осуществить аутентификацию, должны поддерживать одну и ту же хеш-функцию (см. 3.3f), используемую при формировании удостоверений личности и маркеров аутентификации. Опять-таки, в принципе может быть использован ряд альтернативных хеш-функций ценой уменьшения числа пользователей, способных к аутентификации. Краткое описание хеш-функций приведено в приложении Е.

2 Нормативные ссылки

В настоящем стандарте содержатся ссылки на следующие документы:

ГОСТ Р ИСО/МЭК 8824—93 Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии один (АСН.1).

ГОСТ Р ИСО/МЭК 8825—93 Системы обработки информации. Взаимосвязь открытых систем. Спецификация базовых правил кодирования для нотации абстрактного синтаксиса версии один (АСН.1)

ГОСТ Р ИСО/МЭК 9594-1—95 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 1. Общее описание принципов, моделей и услуг

ГОСТ Р ИСО/МЭК 9594-3—97 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 3. Определение абстрактных услуг

ГОСТ Р ИСО/МЭК 9594-5—97 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 5. Спецификации протокола

ГОСТ Р ИСО/МЭК 9594-6—97 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 6. Выбранные типы атрибутов

ГОСТ Р ИСО/МЭК 9594-7—95 Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 7. Выбранные классы объектов

ИСО 7498-2—89* Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 2. Архитектура защиты информации

ИСО/МЭК 9594-2—93* Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 2. Модели

ИСО/МЭК 9594-4—93* Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 4. Процедуры распределенных операций

ИСО/МЭК 9594-9—93* Информационная технология. Взаимосвязь открытых систем. Справочник. Часть 9. Дублирование

---

* Оригиналы стандартов и проектов ИСО/МЭК — во ВНИИКИ Госстандарта России.

ИСО/МЭК 13712-1—94* Информационная технология. Взаимосвязь открытых систем. Удаленные операции. Часть 1. Концепции, модель и нотация

ИСО/МЭК 13712-2—94* Информационная технология. Взаимосвязь открытых систем. Удаленные операции. Часть 2. Определение услуг сервисного элемента удаленных операций

3 Определения

3.1. В настоящем стандарте применяют следующие термины, определенные в ГОСТ Р ИСО 7498-2:

a) **асимметричное (шифрование);**
b) **обмен аутентификацией;**
c) **информация аутентификации;**
d) **конфиденциальность;**
e) **удостоверение личности;**
f) **криптография;**
g) **аутентификация отправителя данных;**
h) **дешифрование;**
i) **шифрование;**
j) **ключ;**
k) **пароль;**
l) **аутентификация равноправного логического объекта;**
m) **симметричное (шифрование).**

3.2 В настоящем стандарте используют следующие термины, определенные в ИСО/МЭК 9594-2:

a) **атрибут;**
b) **информационная база справочника;**
c) **дерево информации справочника;**
d) **агент системы справочника;**
e) **агент пользователя справочника;**
f) **различительное имя;**
g) **запись;**
h) **объект;**
i) **корень.**

3.3 В настоящем стандарте применимы следующие определения:

a) **маркер аутентификации (маркер)** — информация, передаваемая во время обмена строгой аутентификацией, которая может быть использована для аутентификации отправителя;
b) **сертификат пользователя (сертификат)** — ключи общего пользования вместе с некоторой другой информацией, к которой применено шифрование личным ключом УС, выдавшим эту информацию;
c) **уполномоченный по сертификации** — уполномоченный, которому один или несколько пользователей доверяют создавать и присваивать сертификаты. Факультативно УС может создавать ключи пользователя;
d) **путь сертификации** — упорядоченная последовательность сертификатов объектов в дереве информации справочника (ДИС), которая может обрабатываться вместе с ключом общего пользования начального объекта пути для достижения конечного объекта пути;
e) **криптографическая система (криптосистема)** — совокупность преобразований простого текста в шифротекст и обратно, где конкретное(ые) преобразование(я) должно(ы) использо-

---

* Оригиналы стандартов и проектов ИСО/МЭК — во ВНИИКИ Госстандарта России.

ваться выбранными ключами. Преобразования обычно определяются математическим алгоритмом;

f) **хеш-функция** — функция (математическая), которая преобразует значения из большой (возможно очень большой) области в область меньшего масштаба. «Хорошей» считается такая хеш-функция, результаты применения которой к (большому) набору значений в области будут равномерно распределены (и очевидно на случайной основе) по всему диапазону;

g) **однонаправленная функция** — функция (математическая) *f*, которую легко вычислить, но для общего значения *y* из области значений функций трудно вычислительным путем найти такое значение *x* при котором $f(x) = y$. Возможно существует несколько значений *y*, для которых вычислить *x* несложно;

h) **ключ общего пользования** (в криптосистеме ключей общего пользования) — общеизвестный ключ пары ключей пользователя;

i) **личный ключ** (не рекомендуется «секретный ключ») — ключ пары ключей пользователей (в криптосистеме ключей общего пользования), известный только данному пользователю;

j) **простая аутентификация** — аутентификация, осуществляемая путем назначения простого пароля;

k) **стратегия защиты** — набор правил, установленных уполномоченными защиты, которые управляют использованием и предоставлением услуг и средств защиты;

l) **строгая аутентификация** — аутентификация, осуществляемая удостоверениями личности, полученными криптографическим способом;

m) **доверие** — в общем случае один логический объект может сообщить другому логическому объекту о «доверии», если он (первый объект) исходит из предположения о том, что этот другой логический объект будет вести себя в точности так, как ожидает первый логический объект. Такое доверие можно применять только для некоторой конкретной функции. Роль ключа доверия в основах аутентификации состоит в отражении взаимоотношений между аутентифицирующим логическим объектом и УС; аутентифицирующий логический объект должен убедиться в том, что он может доверять уполномоченному по сертификации в создании действительных и надежных сертификатов;

n) **порядковый номер сертификата** — значение целого числа, уникальное для выдающего УС, которое недвусмысленно связано с сертификатом, выданным этим УС.

4 Сокращения

АСС — агент системы справочника
АПС — агент пользователя справочника
ДИС — дерево информации справочника
ИБС — информационная база справочника
ККОП — криптосистема ключа общего пользования
ПДС — протокол доступа к справочнику
УС — уполномоченный по сертификации

5 Соглашения

В настоящем стандарте под понятием «спецификация справочника» следует понимать ГОСТ Р ИСО/МЭК 9594-8, а под понятием «спецификации справочника» — части 1—9 ГОСТ Р ИСО/МЭК 9594.

Если элементы списков пронумерованы (в противоположность использованию дефиса «-» или букв), то такие элементы должны рассматриваться как шаги процедуры.

Обозначения, используемые в настоящем стандарте, приведены в таблице 1.

Примечание — Символы X, X_1, X_2 и т. д. в таблице обозначают имена пользователей, а символ I — произвольную информацию.

Таблица 1 — Обозначения

Обозначение	Назначение
X_o	Ключ общего пользования (пользователя X)
$X_л$	Личный ключ (пользователя X)
X_o [I]	Шифрование некоторой информации (I) с использованием ключа общего пользования пользователя X
$X_л$ [I]	Шифрование (I) с использованием личного ключа пользователя X
X [I]	Подпись I пользователем X. Она включает I, к которой присоединены зашифрованные сводные сведения
УС (X)	Уполномоченный по сертификации пользователя X
$УС^n$ (X)	(где n>1): УС (УС (... n раз ... (X)))
$X_1\ll X_2\gg$	Сертификат пользователя X_2, выданный уполномоченным по сертификации пользователя X_1.
$X_1\ll X_2\gg X_2\ll X_3\gg$	Цепочка сертификатов (может иметь произвольную длину), где каждый элемент — это сертификат для уполномоченного по сертификации, который выдает следующий сертификат. Это функциональный эквивалент следующему сертификату $X_1 \ll X_{n+1}\gg$. Например, обладание A ≪B≫ B ≪C≫ обеспечивает те же возможности, что и A ≪C≫, а именно, способность найти C_o данного A_o
$X_{1o} * X_1\ll X_2\gg$	Операция развертывания сертификата (или цепочки сертификатов) для извлечения ключа общего пользования. Это — инфиксная операция, где левый операнд — ключ общего пользования уполномоченного по сертификации, а правый — сертификат, выданный уполномоченным по сертификации. В результате получаем ключ общего пользования того пользователя, сертификат которого — правый операнд. Например: A_o * A ≪B≫ B ≪C≫ означает операцию, использующую ключ общего пользования пользователя A для получения ключа общего пользования B_o пользователя B из его сертификата с последующим использованием B_o для развертывания сертификата C. Результатом операции является ключ общего пользования C_o пользователя C
A → B	Путь сертификации от A к B, сформированный из цепочки сертификатов, начинающийся с УС (A) ≪$УС^2$ (A)≫ и заканчивающийся с УС (B) ≪B≫.

ГЛАВА 2. ПРОСТАЯ АУТЕНТИФИКАЦИЯ

6 Процедура простой аутентификации

Простая аутентификация предназначена для предоставления локальных полномочий на основе различительного имени пользователя, двусторонне согласованного (факультативно) пароля и двустороннего понимания способов использования и обработки этого пароля в пределах одного региона. Простая аутентификация предназначена в основном для локального использования, то есть для аутентификации равноправных логических объектов между одним АПС и одним АСС или между двумя АСС. Простая аутентификация может быть выполнена несколькими способами:

a) передачей различительного имени пользователя и (факультативно) пароля в открытом (незащищенном) тексте получателю для оценки;

b) передачей различительного имени пользователя, пароля и случайного числа и/или отметкой времени и всего того, что защищено применением однонаправленной функции;

c) передачей защищенной информации, описанной в b), вместе со случайным числом и/или отметкой времени и всего того, что защищено применением однонаправленной функции.

Примечания

1 Не предъявляется никаких требований к тому, чтобы применение однонаправленных функций было различным.

2 Сигнализация процедур защищающих паролей может быть поводом для расширения документа.

Если пароли не защищены, то предполагается минимальная степень защиты от несанкционированного доступа. Это не должно рассматриваться как основа услуг защиты. Защита различительного имени пользователя и пароля обеспечивает большую степень защиты. Алгоритмы, которые должны использоваться для механизма защиты, обычно не шифруются однонаправленными функциями, которые очень просты в реализации.

Общая процедура простой аутентификации приведена на рисунке 1.

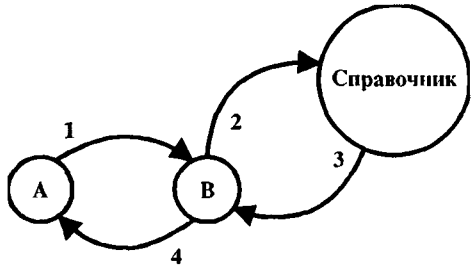


Рисунок 1 — Процедура незащищенной простой аутентификации

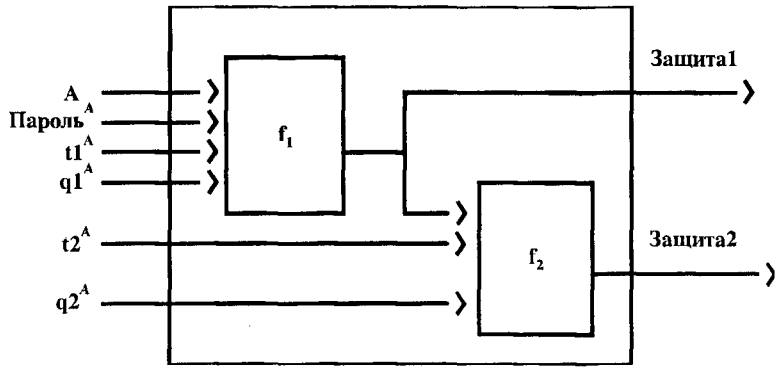
Она состоит из следующих шагов:

1) пользователь — отправитель А посылает свои различительное имя и пароль пользователю — получателю В;
2) пользователь В посылает предполагаемое различительное имя и пароль пользователя А в справочник, где пароль проверяется относительно того пароля, который сохранен в виде атрибута парольПользователя в записи справочника для пользователя А, используя операцию сравнения справочника;
3) справочник подтверждает пользователю В (или отрицает) действительность удостоверения личности;
4) результат положительной (или отрицательной) аутентификации может быть передан пользователю А.

Самая простая форма аутентификации включает только шаг 1, а после проверки пользователем В различительного имени и пароля может выполняться шаг 4.

6.1 Генерация защищенной идентифицирующей информации

На рисунке 2 приведены два подхода генерации защищенной идентифицирующей информации. f_1 и f_2 — это однонаправленные функции (одинаковые либо различные) и отметки времени, а случайные числа являются факультативными и подчиняются двусторонним соглашениям.



Обозначения:

A = Различительное имя пользователя
t^ = Отметка времени
Пароль^ = пароль пользователя А
q^ = Случайные числа, с факультативным счетчиком

Рисунок 2 — Защищенная простая аутентификация

6.2 Процедура защищенной простой аутентификации

На рисунке 3 показана процедура защищенной простой аутентификации.

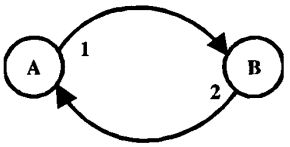


Рисунок 3 — Процедура защищенной простой аутентификации

Процедура защищенной простой аутентификации включает следующие шаги (первоначально использующие только f_1):

1) пользователь — отправитель А посылает свою защищенную идентифицирующую информацию (аутентификатор1) пользователю В. Защита обеспечивается применением однонаправленной функции (f_1), как на рисунке 2, где отметка времени и/или случайное число (при его использовании) используются, чтобы уменьшить ответ и скрыть пароль.

Защита пароля пользователя А имеет вид:

Защита1 = f_1 (t1^A, q1^A, A, парольА)

Информация, переданная пользователю В, имеет вид:

Аутентификатор1 = t1^A, q1^A, A, Защита1

Пользователь В проверяет защищенную идентифицирующую информацию, предлагаемую пользователем А путем создания (используя различительное имя и факультативно отметку времени и/или случайное число, обеспечиваемые пользователем А, вместе с локальной копией пароля пользователя А) локальной защищенной копии пароля пользователя А (в виде Защита1). Пользователь В сравнивает идентифицирующую информацию (Защита1) с локально созданным значением;

2) пользователь В подтверждает пользователю А (или отрицает) наличие защищенной идентифицирующей информации.

С целью предоставления большей степени защиты процедура может быть изменена путем использования функций f_1 и f_2. Основные отличия состоят в следующем:

1) А посылает В дополнительно защищенную идентифицирующую информацию (Аутентификатор2). Дополнительная защита достигается путем применения далее однонаправленной функции f_2, как показано на рисунке 2. Последующая защита имеет вид:

Защита2 = f_2 (t2^A, q2^A, Защита1)

Информация, переданная В, имеет вид:

Аутентификатор2 = t1^A, t2^A, q1^A, q2^A, A, Защита2

При выполнении операции сравнения В генерирует локальное значение дополнительно защищенного пароля и сравнивает его с значением Защита2 (в принципе это аналогично шагу 1 в 6.2).

2) В подтверждает или отрицает для А верификацию защищенной идентифицирующей информации.

Примечание — Процедуры, описываемые в этих пунктах, определены в понятиях пользователей А и В. Применительно к справочнику (определенного в ГОСТ Р ИСО/МЭК 9594-3 и ИСО/МЭК 9594-4). АПС может быть привязана к АСС пользователя В; как вариант, пользователем А может быть АСС, связанный с другим АСС пользователя В.

6.3 Тип атрибута «пароль пользователя»

Этот тип атрибута содержит пароль объекта. Значение атрибута для пароля пользователя — это строка, определенная объектом.

```
userPassword ATTRIBUTE                  ::= {
    WITH SYNTAX                         OCTET STRING (SIZE
                                        (O . . ub-user-password))
    EQUALITY MATCHING RULE              octetStringMatch
        ID                              id-at-userPassword }
```

ГЛАВА 3. СТРОГАЯ АУТЕНТИФИКАЦИЯ

7 Основы строгой аутентификации

Принятый в настоящей спецификации справочника подход к строгой аутентификации основан на использовании свойств семейства криптографических систем, известных под названием «крип-

тосистемы ключа общего пользования» (ККОП). Такие криптосистемы, описываемые также как асимметричные, используют пару ключей, один — секретный, а другой — общего пользования вместо одного ключа в соответствующих криптографических системах.

В приложении С приведено краткое описание таких криптосистем и свойств, которые делают их полезными при выполнении аутентификации. Для того, чтобы ККОП могли использоваться в настоящее время в основах такой аутентификации, они должны обладать таким свойством, чтобы при шифровании использовались оба ключа ключевой пары, то есть, чтобы для дешифрования использовался личный ключ, если ключ общего пользования уже применен, и использовался ключ общего пользования, если личный ключ уже применен. Другими словами, $X_o * X_л = X_o * X_л$, где $X_o / X_л$ — функции шифрования / дешифрования, использующие ключ общего пользования / личный ключ пользователя X.

Примечание — В будущем возможно введение других дополнительных типов ККОП, которые не должны требовать такого свойства перестановки и которые могут быть поддержаны этой спецификацией справочника без особых изменений.

Эти основы аутентификации не предписывают использования конкретной криптосистемы. Задача состоит в том, чтобы эти основы могли быть применимы к любой приемлемой криптосистеме ключа общего пользования и тем самым поддерживали изменения в используемых методах, возникающие в результате дальнейших усовершенствований в криптографии, математических методах или вычислительных возможностях. Однако два пользователя, желающие осуществить аутентификацию, должны для правильного ее выполнения поддерживать один и тот же криптографический алгоритм. Таким образом, в контексте набора соответствующих применений выбор единственного алгоритма должен обеспечить расширение общества пользователей, способных к осуществлению аутентификации и обеспечению защищенного обмена. Пример криптографического алгоритма приведен в приложении D.

Аутентификация рассчитана на любого пользователя, обладающего уникальным различительным именем. За распределение различительных имен несут ответственность уполномоченные по присвоению имен. Поэтому каждый пользователь должен доверить уполномоченным по присвоению имен право не выдавать дубликаты различительных имен.

Каждый пользователь идентифицируется на основе владения личным ключом. Другой пользователь может определить, обладает ли его партнер по обмену личным ключом, и может использовать это для подтверждения того, что партнером по обмену действительно является данный пользователь. Достоверность этого подтверждения зависит от личного ключа, являющегося конфиденциальным для пользователя.

Для того, чтобы пользователь мог определить, что партнер по обмену обладает личным ключом другого пользователя, он должен сам обладать ключом общего пользования этого пользователя. Если процедура получения значения ключа общего пользования непосредственно из записи пользователя в справочнике является достаточно прямолинейной, то проверка его правильности более проблематична. Существует несколько способов решения этой проблемы: в разделе 8 описан процесс, с помощью которого ключ общего пользования может быть проверен путем обращения к справочнику. Такой процесс может действовать только в том случае, если между пользователями, нуждающимися в аутентификации, существует непрерывная цепочка доверительных точек в справочнике. Такая цепочка может быть построена путем идентификации общей точки доверия. Эта общая точка должна быть связана с каждым пользователем непрерывной цепочкой доверительных точек.

8 Получение ключа общего пользования

Для того, чтобы пользователь доверял процедуре аутентификации, он должен получить ключ общего пользования других пользователей от источника, которому он доверяет. Такой источник, называемый уполномоченным по сертификации (УС), для сертификации ключа общего пользования использует алгоритм ключа общего пользования. Сертификат, форма которого будет определена ниже в этом разделе, обладает следующими свойствами:

- любой пользователь, имеющий доступ к ключу общего пользования уполномоченного по сертификации, может восстановить ключ общего пользования, который был подтвержден;

- никто другой, кроме уполномоченного по сертификации, не может изменить сертификат без того, чтобы это не было обнаружено (сертификаты неподдельны).

Поскольку сертификаты неподдельны, их можно сделать общедоступными, поместив в справочнике без необходимости принятия специальных мер по их защите.

Примечание — Хотя уполномоченные по сертификации недвусмысленно определены по их различительным именам в ДИС, это еще не означает наличие какого-либо взаимоотношения между организацией СУ и ДИС.

Уполномоченный по сертификации создает сертификат пользователя, подписывая (см. раздел 9) собранную информацию, которая включает различительное имя пользователя и его ключ общего пользования, а также факультативный уникальный идентификатор, содержащий дополнительную информацию о пользователе. Точная форма содержимого уникального идентификатора здесь не определяется и оставлена на усмотрение уполномоченного по сертификации; она может иметь вид, например, идентификатора объекта, сертификата, даты или некоторый другой вид сертификата достоверности различительного имени. В частности, сертификат пользователя с различительным именем А и уникальным идентификатором УА, созданный уполномоченным по сертификации с именем УС и уникальным идентификатором УУС, имеет вид:

$$УС \ll A \gg = УС \{В, ПН, ИА, УС, УУС, УА, А, А_{о}, Т^{А}\},$$

где В — версия сертификата, ПН — порядковый номер сертификата, ИА — идентификатор алгоритма, используемый для подписания сертификата, УУС — факультативный уникальный идентификатор УС, УА — факультативный уникальный идентификатор пользователя А, $Т^{А}$ указывает время действия сертификата и состоит из двух дат: начала и конца действия сертификата. Поскольку $Т^{А}$ может изменяться не чаще, чем через 24 ч, предполагается, что системы должны использовать в качестве базового эталонного времени Всемирное координированное время. Действительность подписи на сертификате может быть проверена любым пользователем, знающим $СУ_{о}$. Для представления сертификатов может быть использован следующий тип данных АСН.1:

```
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,
                  - - при его наличии, должна быть версия v2
subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL
                  - - при его наличии, должна быть версия v2 - - }}
Version                    : : = INTEGER {v1(0), v2(1)}
CertificateSerialNumber    : : = INTEGER
AlgorithmIdentifier        : : = SEQUENCE {
   algorithm                   ALGORITHM. &id ({SupportedAlgorithms}),
   parameters                  ALGORITHM. &Type ({SupportedAlgorithms}
                                        {Algorithm}) OPTIONAL }
- Определение следующего набора информационных объектов отложено, возможно, до раз-
- работки стандартизованных профилей или заявок о соответствии реализации протоколу.
- Такой набор необходим для определения таблицы ограничений на компоненту «параметр»
- атрибута AlgorithmIdentifier.
- SupportedAlgorithms ALGORITHM : : = {. . . I . . . }
Validity                   : : =     SEQUENCE {
   notBefore                   UTCTime,
   notAfter                    UTCTime }
SubjectPublicKeyInfo       : : = SEQUENCE {
```

```
algorithm           AlgorithmIdentifier,
subjectPublicKey    BIT STRING  }
```

Примечание — В случаях, когда различительное имя может быть повторно назначено другому пользователю уполномоченным по присвоению имен, УС могут использовать уникальный идентификатор различения имен при многократном использовании. Однако, если одного и того же пользователя сертификатами обеспечивают несколько УС, рекомендуется, чтобы УС координировали присвоение уникальных идентификаторов в процессе выполнения процедур по регистрации пользователей.

Запись справочника каждого пользователя А, участвующего в строгой аутентификации, содержит сертификат(ы) А. Такой сертификат создается уполномоченным по сертификации пользователя А, являющимся логическим объектом в ДИС. Уполномоченный по сертификации пользователя А, который может быть не уникальным, обозначается УС(А), или просто УС, если А подразумевается. Поэтому ключ общего пользования пользователя А может быть открыт любым пользователем, который знает ключ общего пользования УС. Таким образом, процесс раскрытия ключей общего пользования является рекурсивным.

Если пользователь А, пытаясь получить ключ общего пользования пользователя В, уже получил ключ общего пользования УС(В), то процесс заканчивается. Для того, чтобы А мог получить ключ общего пользования УС(В), запись справочника каждого уполномоченного по сертификации X содержит несколько сертификатов. Существует два типа таких сертификатов. К первому типу относятся срочные сертификаты X, созданные другими уполномоченными по сертификации. Ко второму — реверсивные сертификаты, созданные самим X, которые являются заверенными общими ключами других уполномоченных по сертификации. Наличие таких сертификатов позволяет пользователям строить путь сертификации от одной точки к другой.

Список сертификатов, необходимый для того, чтобы конкретный пользователь мог получить общий ключ другого пользователя, называется «путь сертификации». Каждый элемент такого списка является сертификатом уполномоченного по сертификации следующего элемента в списке. Путь сертификации от А к В (обозначаемый А —> В)

- начинается от сертификата, созданного УС (А), а именно, УС (А)$\ll X^1 \gg$ для некоторого логического объекта X^1;
- продолжается последующими сертификатами $X^i \ll X^{i+1} \gg$;
- заканчивается сертификатом В.

Логически путь сертификации формирует непрерывную цепочку доверительных точек в дереве информации справочника между двумя пользователями, желающими выполнить аутентификацию. Точный метод, применяемый пользователями А и В для получения пути сертификации А — $\rightarrow$В и В — $\rightarrow$А, может изменяться. Один из способов, ведущих к упрощению, состоит в том, чтобы построить иерархию УС, совпадающую с иерархией ДИС полностью или частично. Преимущество такого подхода состоит в том, что пользователи, имеющие УС в иерархии, могут, используя справочник, установить между ними путь сертификации без какой-либо предварительной информации. Для того, чтобы достичь этого, каждый такой УС может хранить один сертификат и один реверсивный сертификат, предназначенный для передачи своему старшему УС.

Сертификаты хранятся в записях справочника в виде атрибутов типа UserCertificate, CACertificate и CrossCertificatePair. Эти типы атрибутов известны справочнику. Такие атрибуты могут действовать при использовании тех же операций протокола, которые используют другие атрибуты. Определения этих типов приведены в 3.3; спецификация этих типов атрибутов имеет следующий вид:

```
userCertificate                 ATTRIBUTE : : = {
        WITH SYNTAX             Certificate
        ID                      id-at-userCertificate}
cACertificate                   ATTRIBUTE : : = {
        WITH SYNTAX             Certificate
        ID                      id-at-cAcertificate}
crossCertificatePair            ATTRIBUTE : : = {
        WITH SYNTAX             CertificatePair
        ID                      id-at-crossCertificatePair}
CertificatePair     : =         SEQUENCE {
        forward     [0]         Certificate OPTIONAL,
        reverse     [1]         Certificate OPTIONAL
```

- - Должна иметь место, по меньшей мере, одна из пар - - }

Пользователь может получить один или несколько сертификатов от одного или нескольких уполномоченных по сертификации. Каждый сертификат носит имя того уполномоченного по сертификации, который его выдал. Для представления сертификатов и пути сертификации могут быть использованы следующие типы данных АСН.1:

```
Certificates              : : =   SEQUENCE {
    userCertificate               Certificate,
    certif icationPath            ForwardCertificationPath OPTIONAL }
CertificationPath         : : =   SEQUENCE {
    userCertificate               Certificate,
    theCACertificates              SEQUENCE OF CertificatePair OPTIONAL }
```

Кроме того, для представления срочного пути сертификации может быть использован следующий тип данных АСН.1. Эта компонента содержит путь сертификации, который может привести обратно к отправителю.

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

8.1 Оптимизация количества информации, полученной из справочника

В общем случае, прежде чем пользователи выполнят взаимную аутентификацию, справочник должен выполнить полную сертификацию и выдать путь аутентификации. Однако практически объем информации, которая должна быть получена из справочника, может быть сведена к конкретному сеансу аутентификации следующим образом:

a) если два пользователя, желающие выполнить аутентификацию, обслуживаются одним и тем же уполномоченным по сертификации, то путь сертификации становится тривиальным, и пользователи сами непосредственно сертифицируют друг друга;

b) если УС пользователей расположены иерархически, то пользователь должен хранить ключи общего пользования, сертификаты и повторные сертификаты всех уполномоченных между пользователем и корнем ДИС. Обычно это требует от пользователя знания ключей общего пользования и сертификатов только трех или четырех уполномоченных по сертификации. При этом пользователю потребуется только получить путь сертификации от общей точки доверия;

c) если пользователь часто связывается с другими пользователями, получившими сертификаты от другого конкретного УС, то для того чтобы такой пользователь мог узнать путь сертификации к этому УС и обратный путь от него, ему необходимо только получить сертификат другого пользователя из справочника;

d) уполномоченные по сертификации могут пересекаться друг с другом при выдаче сертификатов на основе двустороннего соглашения. Результатом является более короткий путь сертификации;

e) если два пользователя взаимодействовали до этого и знают сертификаты друг друга, они могут выполнить аутентификацию без какого-либо обращения к справочнику.

В любом случае, определив сертификаты других из пути сертификации, пользователи должны проверить действительность полученных сертификатов.

8.2 Пример

На рисунке 4 приведен гипотетический пример фрагмента ДИС, где УС образуют иерархию. Считается, что кроме информации, приведенной в УС, каждый пользователь знает ключ общего пользования своего уполномоченного по сертификации, свои собственные ключ общего пользования и личный ключ.

Если УС пользователей расположены иерархически, пользователь А для установления пути сертификации к пользователю В может получать из справочника следующие сертификаты:

$$X\langle\langle W\rangle\rangle,\ W\langle\langle V\rangle\rangle,\ V\langle\langle Y\rangle\rangle,\ Y\langle\langle Z\rangle\rangle,\ Z\langle\langle B\rangle\rangle$$

Пользователь А, получив такие сертификаты, может развернуть путь сертификации в такой последовательности, которая представляет содержимое сертификата пользователя В, включая B_o:

$$B_o = X_p * X\langle\langle W\rangle\rangle\ W\langle\langle V\rangle\rangle\ V\ \langle Y\rangle\rangle\ Y\langle\langle Z\rangle\rangle\ Z\langle\langle B\rangle\rangle$$

В общем случае, пользователь А должен получить следующие сертификаты из справочника, чтобы установить обратный путь сертификации от В к А:

Z≪Y≫, Y≪V≫, V≪W≫,
W≪X≫, X≪A≫

При получении таких сертификатов от А пользователь В может развернуть обратный путь сертификации в такой последовательности. которая вырабатывает содержимое сертификата пользователя А, включая A_o:

$A_o = Z_o$* Z≪Y≫ Y≪V≫ V≪W≫
W≪X≫ X≪A≫.

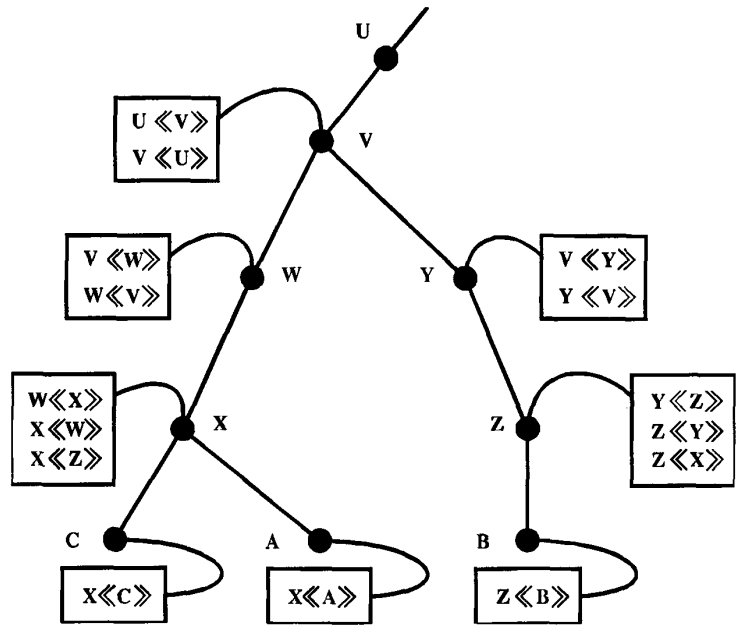


Рисунок 4 — Иерархия. Гипотетический пример

Применяя оптимизацию в соответствии с 8.1:

a) возьмем, например, А и С: оба знают X_o, так что А должен прямо получить сертификат С. Развернутый путь сертификации сводится к

$C_o = X_o$* X≪C≫,

а развернутый обратный путь сертификации сводится к

$A_o = X_o$* X≪A≫;

b) предполагая, что А может таким образом знать W≪X≫, W_o, V≪W≫, V_o, U≪V≫ и т. д., тогда информация, которую А должен получить из справочника для формирования пути сертификации, сводится к

V≪Y≫, Y≪Z≫, Z≪B≫,

а информация, которую А должен получить из справочника для формирования обратного пути сертификации, сводится к Z≪Y≫, Y≪V≫;

c) предполагая, что А часто взаимодействует с пользователями, получившими сертификаты от Z, он может узнать [дополнительно к ключам общего пользования, известных из b)] V≪Y≫, Y≪V≫, Y≪Z≫, и Z≪Y≫. Для того, чтобы связаться с В, ему, следовательно, необходимо только получить из справочника Z≪B≫;

d) предполагая, что пользователи, получившие сертификаты от X и Z, часто взаимодействуют между собой, X≪Z≫ должен храниться в записи справочника для X, и наоборот (как приведено на рисунке 4). Если А желает выполнить аутентификацию с В, ему необходимо только получить:

X≪Z≫, Z≪B≫

для формирования пути аутентификации и

Z≪X≫

для формирования обратного пути аутентификации;

е) предполагая, что пользователи А и С уже взаимодействовали ранее и знают сертификаты друг друга, они могут использовать непосредственно ключ общего пользования друг друга, то есть,

$$C_o = X_o * X\ll C\gg$$

и

$$A_o = X_o * X\ll A\gg.$$

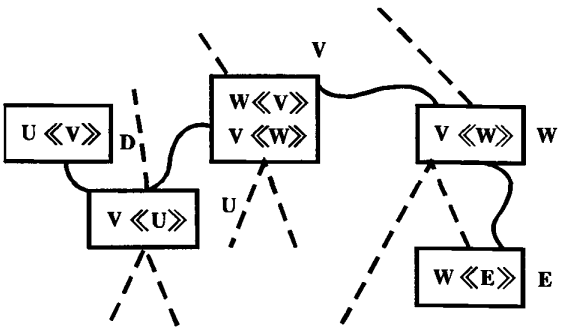


Рисунок 5 — Пример неиерархического пути сертификации

В более общем случае уполномоченные по сертификации иерархически не взаимосвязаны. Обращаясь к гипотетическому примеру на рисунке 5, предположим, что пользователь D, получивший сертификат от U, желает выполнить аутентификацию с пользователем Е, получившим сертификат от W. Запись пользователя D в справочнике должна содержать сертификат U≪D≫, а запись пользователя Е — сертификат W≪E≫.

Пусть V — это УС, с которым уполномоченные по сертификации U и W в предыдущий раз обменялись ключами общего пользования доверительным способом. В результате были созданы сертификаты U≪V≫, V≪U≫, W≪V≫и V≪W≫ и занесены в справочник. Предположим, что U≪V≫ и W≪V≫ содержатся в записи V, V≪U≫ — в записи U, а V≪W≫ — в записи W.

Пользователь D должен найти путь сертификации к Е. Могут быть использованы различные стратегии. Одна из таких стратегий — рассматривать пользователей и УС как узлы, а сертификаты как дуги в направленном графе. В таких понятиях D должен выполнить поиск в графе для нахождения пути от U к Е, которым в данном случае является путь U≪V≫, V≪W≫, W≪E≫. После того, как этот путь будет найден, может быть построен и обратный путь W≪V≫, V≪U≫, U≪D≫.

9 Цифровые подписи

В этом разделе не ставится задача создать общий стандарт по цифровым подписям, а только определить средства, с помощью которых маркеры подписываются в справочнике.

Информация (инф) подписывается путем добавления к ней зашифрованной сводки информации. Эта сводка вырабатывается с использованием однонаправленной хеш-функции, а шифрование выполняется с использованием личного ключа подписывающего лица (см. рисунок 6).

Таким образом

$$X\{\text{Инф}\} = \text{Инф}, X\text{л}[h(\text{Инф})]$$

Примечание — Шифрование, использующее личный ключ, гарантирует, что подпись не может быть подделана. Свойство однонаправленности хеш-функции гарантирует, что ложная информация, которая вырабатывается так, чтобы иметь тот же результат хеширования (и, тем самым, подпись), не может послужить заменой.

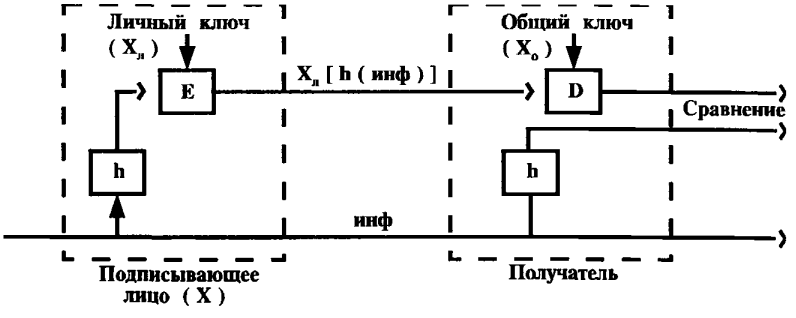


Рисунок 6 — Цифровые подписи

Получатель подписанной информации проверяет подпись путем:
- применения к информации однонаправленной хеш-функции,
- сравнением результата с полученной после дешифрования подписью, используя общий ключ подписывающего лица.

Настоящий стандарт не предписывает использовать для подписания единственную однонаправленную хеш-функцию. Задача состоит в том, чтобы стандартизуемые здесь основы аутентификации могли быть применимы к любой подходящей хеш-функции и могли таким образом поддерживать изменения в используемых методах, которые могут появиться в результате будущих усовершенствований в криптографии, математических методах или вычислительных возможностях. Однако два пользователя, желающие выполнить правильно аутентификацию, должны обеспечивать одну и ту же хеш-функцию. Таким образом, в контексте набора соответствующих применений выбор единственной функции должен послужить максимизации количества пользователей, способных выполнить аутентификации и обеспечить надежный обмен данными.

Подписанная информация включает указатели, которые идентифицируют алгоритмы хеширования и шифрования, используемые для вычисления цифровой подписи.

Шифрование некоторого элемента данных может быть описано в АСН.1 следующим образом:

ENCRYPTED {ToBeEnciphered}: : = BIT STRING (CONSTRAINED BY {
- - применение процедуры шифрования должно приводить - -
- - к закодированным по базовым правилам кодирования
- - октетам значения - - ToBeEnciphered })

Это значение строки битов образуется из октетов, которые создают полное кодирование (используя базовые правила кодирования АСН.1 по ГОСТ Р ИСО/МЭК 8825) значения типа ToBeEnciphered, путем применения процедуры шифрования к этим октетам.

Примечания

1 Процедура шифрования требует наличия соглашения по применяемому алгоритму, включая все параметры алгоритма, такие как необходимые ключи, значения инициализации и инструкции заполнения. Процедуры шифрования несут ответственность за определение средств, с помощью которых достигается синхронизация отправителя и получателя данных, которые могут включать в передаваемые биты свою информацию.

2 Требуется, чтобы процедура шифрования воспринимала на входе строки октетов и вырабатывала результат в виде одной строки битов.

3 Механизмы согласования защиты по алгоритму шифрования и их параметрам отправителем и получателем данных не входит в предмет рассмотрения настоящего стандарта.

В случае, когда подпись может быть присоединена к типу данных, то для определения типа данных, полученного в результате применения подписи к данному типу данных, может быть использовано следующее описание АСН.1

```
SIGNED {ToBeSigned}          : : =     SEQUENCE {
  toBeSigned                 ToBeSigned,
  COMPONENTS OF              SIGNATURE {ToBeSigned}}
```

В случае, когда требуется только подпись, то для определения типа данных, полученного в результате применения подписи к данному типу данных, может быть использована следующая макрокоманда АСН.1

```
SIGNATURE {OfSignature}      : : =     SEQUENCE {
  algorithmIdentifier        AlgorithmIdentifier,
  encrypted                  ENCRYPTED {HASHED {OfSignature}}}
```

Чтобы иметь возможность проверить правильность типов SIGNED и SIGNATURE в распределенной среде, требуется различительное кодирование. Различительное кодирование значения данных SIGNED или SIGNATURE может быть получено путем применения базовых правил кодирования, определенных в ГОСТ Р ИСО/МЭК 8825, с учетом следующих ограничений:

a) должна быть использована определительная форма кодирования длины, закодированная минимальным числом октетов;
b) созданная форма кодирования не должна использоваться для последовательных типов;
c) если значение типа — это значение по умолчанию, оно должно отсутствовать;
d) компоненты типа Set должны кодироваться в порядке возрастания значений их тегов;
e) компоненты типа Set должны кодироваться в порядке возрастания значений их октетов;
f) если булев тип имеет значение «истинно», то в результате кодирования содержимое октета должно быть установлено в «FF»$_{16}$;

g) при наличии неиспользуемых битов в последнем октете закодированных строк битов они должны быть установлены в 0;

h) при кодировании типа Real не должны использоваться основания 8, 10, и 16, а коэффициент двоичного масштабирования должен быть нулевым.

10 Процедуры строгой аутентификации

10.1 Общее описание

Выше изложен основной подход к аутентификации, а именно подтверждение идентичности путем демонстрации обладания личным ключом. Возможно, однако, большое количество процедур аутентификации, использующих этот подход. В общем случае дело каждого конкретного применения — определить соответствующие процедуры, которые удовлетворяли бы стратегии защиты данного применения. В этом разделе описываются три конкретные процедуры аутентификации, которые могут быть полезны во всем диапазоне применений.

Примечание — Настоящий стандарт не определяет подробно процедур, требуемых для реализации. Однако могут быть предусмотрены дополнительные стандарты, которые могли бы выполнять это специфичным для применения либо универсальным способом.

Три указанные процедуры используют различное число сеансов обмена информацией аутентификации и, следовательно, обеспечивают различные степени гарантий своим участникам. В частности:

a) однонаправленная аутентификация, описанная в 10.2, использует одну передачу информации от одного пользователя (A), предназначенную для другого (B), и устанавливает следующее:
   - подлинность пользователя A и подтверждение того, что маркер аутентификации был действительно сгенерирован A;
   - подлинность пользователя B и подтверждение того, что маркер аутентификации был действительно предназначен для передачи к B;
   - целостность и новизну (передан не более одного раза) передаваемого маркера аутентификации.

   Последние свойства могут быть также установлены для произвольных дополнительных данных, сопровождающих передачу;

b) двунаправленная аутентификация, описанная в 10.3, дополнительно использует ответ от B к A и кроме этого констатирует:
   - что маркер аутентификации в ответе действительно был сгенерирован B и предназначен для передачи к A;
   - целостность и новизну передаваемого маркера аутентификации в ответе;
   - (факультативно) взаимную секретность маркеров;

c) трехнаправленная аутентификация, описанная в 10.4, дополнительно использует последующую передачу от A до B и устанавливает те же свойства, что и двунаправленная аутентификация, но делает это без необходимости соответствующей проверки отметок времени.

Во всех случаях использования строгой аутентификации A должен получить ключ общего пользования пользователя B и обратный путь сертификации от B до A. Это может обусловить обращение к справочнику, как описано в разделе 7. Любое такое обращение не упоминается ниже каждый раз при описании процедур.

Проверка отметок времени, упоминаемая в последующих разделах, используется только в случаях, когда в локальной среде используются синхронизированные часы, или если часы логически синхронизированы в соответствии с двусторонними соглашениями. В любом случае рекомендуется использовать всемирное координированное время.

Для каждой из описываемых ниже трех процедур аутентификации предполагается, что сторона A проверила действительность всех сертификатов в пути аутентификации.

10.2 Однонаправленная аутентификация

При однонаправленной аутентификации (рисунок 7) выполняются следующие шаги:

1) A создает r^A (неповторяющийся номер), который используется для обнаружения повторных угроз и предотвращения подделок;

2) A посылает следующее сообщение к B:

B — ¬A, A {t^A, r^A, B},

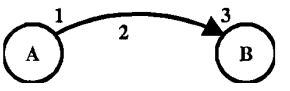


Рисунок 7 — Однонаправленная аутентификация

где t^A — отметка времени, которая состоит из одной или двух дат: даты создания маркера (который является факультативным) и даты истечения срока действия. Как вариант, если аутентификация отправителя данных «sgnData» должна обеспечиваться цифровой подписью, сообщение будет иметь вид:

B — ¬A, A {t^A, r^A, B, sgnData }.

В тех случаях, когда передаваемая информация должна впоследствии использоваться в виде личного ключа (эта информация обозначается «encData»), сообщение будет иметь вид:

B — ¬A, A {t^A, r^A, B, sgnData, B_o [encData]}.

Использование «encData» в виде личного ключа предполагает, что он должен очень внимательно выбираться, например, быть строгим ключом для любой используемой криптосистемы, как указано в поле «sgnData» маркера.

3) Пользователь B выполняет следующие действия:

a) получает A_o от B — ¬A, проверяя, что срок сертификата пользователя A не истек;

b) проверяет подпись, и тем самым целостность подписанной информации;

c) проверяет, что он сам является назначенным получателем;

d) проверяет, что отметка времени имеет значение «текущее»;

e) проверяет факультативно, что r^A повторно не использован. Это может быть достигнуто, например, введением в r^A последующей части, которая проверяется локальной реализацией на уникальность ее значений.

r^A имеет силу до истечения даты, указанной t^A. r^A всегда сопровождается последовательной частью, которая указывает, что A не должен повторять маркер в течение временного диапазона t^A, и поэтому проверка значения самого r^A не требуется.

В любом случае для стороны B целесообразно хранить в течение временно́го диапазона t^A последовательную часть вместе с отметкой времени t^A в открытом состоянии и вместе с хешированной частью маркера.

10.3 Двунаправленная аутентификация

При двунаправленной аутентификации (рисунок 8) выполняются следующие шаги:

1) Как и в 10.2.
2) Как и в 10.2.
3) Как и в 10.2.
4) B создает r^B (неповторяющийся номер), используемый для тех же целей, что и r^A.
5) B посылает к A маркер последующей аутентификации:

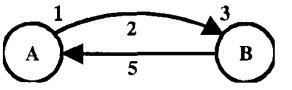


Рисунок 8 — Двунаправленная аутентификация

B {t^B, r^B, A, r^A},

где t^B — отметка времени, определенная тем же самым способом, что и t^A.

Как вариант, если аутентификация отправителя данных «sgnData» должна обеспечиваться цифровой подписью, маркер будет иметь вид:

B {t^B, r^B, A, r^A, sgnData}.

В тех случаях, когда передаваемая информация должна впоследствии использоваться в виде личного ключа (эта информация обозначается «encData»), маркер будет иметь вид:

B {t^B, r^B, A, r^A, sgnData, Ap[encData]}.

Использование «encData» в виде личного ключа предполагает, что он должен очень вни-

мательно выбираться, например, быть строгим ключом для любой используемой криптосистемы, как указано в поле «sgnData» маркера.

6) Пользователь А выполняет следующие действия:

a) проверяет подпись и, тем самым, целостность подписанной информации;

b) проверяет, что он сам является назначенным получателем;

c) проверяет, что отметка времени имеет значение «текущее»;

d) проверяет факультативно, что r^A повторно не использован (см. 10.2, шаг 3d).

10.4 Трехнаправленная аутентификация

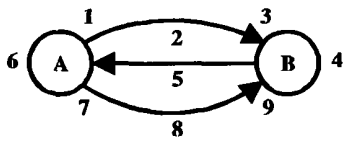


Рисунок 9 — Трехнаправленная аутентификация

При трехнаправленной аутентификации (рисунок 9) выполняются следующие шаги:

1) Как и в 10.3.

2) Как и в 10.3. Отметка времени t^A может быть нулевой.

3) Как для 10.3 за исключением того, что отметка времени не должна проверяться.

4) Как и в 10.3.

5) Как и в 10.3. Отметка времени t^B может быть нулевой.

6) Как и в 10.3 за исключением того, что отметка времени не должна проверяться.

7) Проверяет, идентичен ли полученный r^A переданному r^A.

8) А посылает к В следующий маркер аутентификации:

$$A\ \{r^B,\ B\}.$$

9) В выполняет следующие действия:

a) проверяет подпись и, тем самым, целостность подписанной информации;

b) проверяет, что полученный r^B идентичен r^B, переданному стороной В.

11 Административное управление ключами и сертификатами

11.1 Генерация пары ключей

Полная стратегия административного управления защитой при реализации должна определять жизненный цикл пары ключей, а это выходит за рамки основ аутентификации. Однако для полной защиты важно, чтобы все личные ключи оставались известны только тому пользователю, кому они принадлежат.

Данные ключа не просты для запоминания пользователю — человеку, поэтому приемлемым способом их хранения является обычный транспортабельный метод. Одним из возможных механизмов может быть использование «Smart Card». В них можно хранить секретный и (факультативно) ключи общего пользования, сертификат пользователя и копии общих ключей уполномоченных по сертификации.

Использование такой карты должно быть дополнительно защищено, например, по меньшей мере, использованием личного номера идентификации, повышающим защиту такой системы тем, что от пользователя требуется обладание такой картой и знание доступа к ней. Однако выбор точного способа хранения таких данных не входит в предмет рассмотрения настоящего стандарта.

Существует три способа, с помощью которых может быть создана пара ключей пользователя:

a) пользователь создает свою собственную пару ключей. Этот метод имеет преимущество в том, что личный ключ пользователя никогда не будет освобожден для другого логического объекта, но требует определенного уровня компетентности пользователя, как описано в приложении D;

b) пара ключей создается третьим участником. Этот третий участник должен освободить личный ключ для данного пользователя физически защищенным способом, затем уничтожить всю информацию, относящуюся к созданию пары ключей, и сами ключи. Должны быть приняты соответствующие физические меры безопасности, гарантирующие, что третий участник и операции над данными не будут подвергнуты вмешательству;

c) пара ключей создается уполномоченным по сертификации. Это особый случай способа b), и здесь применимы другие соображения.

Примечание — Уполномоченный по сертификации уже проявил доверительные функциональные возможности относительно пользователя и должен действовать в зависимости от необходимых физических мер защиты. Этот метод имеет преимущество в том, что он не требует передачи к УС защищенных данных для сертификации.

Используемая криптосистема налагает конкретные (технические) ограничения на создание ключей.

11.2 Административное управление сертификатами

Сертификат логически увязывает ключ общего пользования и уникальное различительное имя соответствующего пользователя. Таким образом:

a) уполномоченный по сертификации должен быть убежден в идентичности пользователя до создания сертификата для него;

b) уполномоченный по сертификации не должен выдавать сертификаты двум пользователям с одинаковым именем.

Создание сертификата происходит автономно, и не должно выполняться путем использования механизма автоматического запроса/ответа. Преимущество такой сертификации состоит в том, что поскольку личный ключ уполномоченного по сертификации никому неизвестен кроме изолированного и физически защищенного УС, то личный ключ такого УС может быть получен только в результате его изучения путем угрозы самому УС, что делает компромисс маловероятным.

Важно, чтобы передача информации уполномоченному по сертификации не была поставлена под угрозу и были приняты подходящие физические меры защиты. С этой точки зрения:

a) было бы серьезным нарушением безопасности, если бы УС выдавали сертификат пользователю с ключом общего пользования, который можно исказить;

b) если реализованы средства создания пары ключей 11.1c), защита передачи не требуется;

c) если реализованы средства создания пары ключей согласно 11.1a) или 11.1b), пользователь может использовать различные методы (оперативные или автономные), чтобы сообщить свой ключ общего пользования уполномоченному по сертификации защищенным способом. Оперативные методы могут предоставить некоторую дополнительную гибкость для удаленных операций, выполняемых между пользователем и УС.

Сертификат — это общедоступная часть информации, и каких-либо конкретных мер защиты для его передачи в справочник не требуется. Поскольку он создается автономным уполномоченным по сертификации по поручению пользователя, которому будут переданы копии сертификата, пользователю необходимо только занести эту информацию в свою запись справочника и получить затем доступ к справочнику. Как вариант, УС мог бы закрепить сертификат за пользователем и в этом случае этому агенту должны быть предоставлены соответствующие права доступа.

Сертификаты должны иметь свой срок службы, по истечению которого они утрачивают свою силу. Для того, чтобы продлить услуги, УС должен гарантировать своевременную готовность новых сертификатов, заменяющих сертификаты, срок действия которых истек/истекает. Здесь следует отметить два момента:

- действительность сертификатов может быть обеспечена таким образом, что каждый становится действительным по истечения срока действия своего предшественника с возможным перекрытием срока их действия. Последнее предотвращает УС от необходимости устанавливать и назначать большое число сертификатов, которые могли бы действовать, начиная с одной даты истечения срока;
- сертификаты, срок службы которых истек, обычно удаляются из справочника. Вопрос стратегии защиты и ответственности УС состоит в том, чтобы сохранять прежние сертификаты на некоторый период времени, если обеспечиваются услуги данных «безотказность отправителя».

Сертификаты могут быть отменены до истечения их срока службы, например, если предполагается, что личный ключ пользователя находится под сомнением, или если пользователь уже не

имеет сертификата УС, или если предполагается, что сертификаты УС находятся под сомнением. Здесь следует отметить четыре момента:

- отмена сертификата пользователя или сертификата УС должна быть известна УС и в уместных случаях должен быть выдан новый сертификат. После этого УС может проинформировать владельца сертификата о произошедшей отмене с помощью некоторой автономной процедуры.
- УС должен поддерживать:

a) список выданных им и отмененных сертификатов с отметкой времени;

b) список аттестованных УС и отмененных сертификатов всех УС, известных данному УС, с отметкой времени.

Оба списка сертификатов должны иметь место, даже если они пустые;

- ответственность за поддержание записей справочника, подверженных воздействию отмененных УС из списка, возлагается на справочник и его пользователей, действующих в соответствии со стратегией защиты. Например, пользователь может модифицировать свою запись объекта, заменяя старый сертификат на новый. После этого последний может использоваться для аутентификации пользователя в справочнике;
- списки отмененных сертификатов («черные списки») хранятся в записях в виде атрибутов типа «CertificateRevocation» и «authorityRevocationList». Эти атрибуты могут обрабатываться с использованием тех же операций, что и другие атрибуты. Эти типы атрибутов определяются следующим образом:

```
certificateRevocationList       ATTRIBUTE : : = {
    WITH SYNTAX                 CertificateList
    ID                          id-at-certificateRevocationList }
authorityRevocationList         ATTRIBUTE : : = {
    WITH SYNTAX                 CertificateList
    ID                          id-at-authorityRevocationList }
CertificateList                 : : =   SIGNED {SEQUENCE {
    signature                   AlgorithmIdentifier,
    issuer                      Name
    thisUpdate                  UTCTime,
    nextUpdate                  UTCTime OPTIONAL,
    revokedCertificates         SEQUENCE OF SEQUENCE {
        userCertificate           CertificateSerialNumber,
        revocationDate            UTCTime} OPTIONAL}}
```

Примечания

1 Проверка полного списка сертификатов является локальным вопросом.

2 Если услуга данных «безотказность отправителя» зависит от ключей, обеспечиваемых УС, то эта услуга должна гарантировать, что все соответствующие ключи УС (отмененные или с истекшим сроком действия) и списки отмененных ключей с отметкой времени архивируются и аттестуются действующим уполномоченным.

ПРИЛОЖЕНИЕ А
(обязательное)

ОСНОВЫ АУТЕНТИФИКАЦИИ В АСН.1

В данном приложении приведены определения всех типов, значений и классов информационных объектов АСН.1, содержащихся в настоящем стандарте, в виде модуля АСН.1 «AuthentificationFramework» (Основы-Аутентификации).

```
AuthentificationFramework {joint-iso-ccitt ds(5) module(1) authentificationFramework(7) 2}
DEFINITIONS ::=
BEGIN
-- EXPORTS ALL -
-- Определенные в этом модуле типы и значения экспортируются для использования в других модулях
-- АСН.1, содержащихся в спецификациях справочника, и в других прикладных программах, которые,
-- в свою очередь, будут использовать их для доступа к услугам справочника. Другие прикладные програм-
-- мы могут использовать эти типы и значения для своих собственных целей, но это не должно
-- препятствовать расширениям и модификациям, необходимым при обслуживании или усовершенствова-
-- нии услуг справочника.
IMPORTS
    id-at, informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1)
        usefulDefinitions(0) 2}
    Name, ATTRIBUTE
        FROM InformationFramework informationFramework
    ub-user-password
        FROM UpperBounds upperBounds
    AuthentificationLevel
        FROM BasicAccessControl basicAccessControl
    UniqueIdentifier, octetStringMatch
        FROM SelectedAttributeTypes selectedAttributeTypes;
-- Типы --
    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,
                -- при его наличии, должна быть версия v2 --}}
        subjectUniqueIdentifier             [2] IMPLICIT UniqueIdentifier OPTIONAL,
                -- при его наличии, должна быть версия v2 --}}
    Version                     ::=     INTEGER { v1(0), v2(1) }
    CertificateSerialNumber     ::=     INTEGER
    AlgorithmIdentifier         ::=     SEQUENCE {
        algorithm               ALGORITHM.&id ({SupporteadAlgorithms}),
        parameters              ALGORITHM.&Type ({SupporteadAlgorithms} {Algorithms}) OPTIONAL}
    - Определение следующего набора информационных объектов отложено, возможно, до разработки стан-
    - дартизованных профилей или заявок о соответствии реализации протоколу. Такой набор необходим для
    - определения табличных ограничений на компоненту «параметры» атрибута AlgorithmIdentifier.
    - SupporteadAlgorithms ALGORITHM ::= {...I...}
    Validity                    ::=     SEQUENCE {
        notBefore       UTCTime,
        notAfter        UTCTime }
    SubjectPublicKeyInfo        ::=     SEQUENCE {
        algorithm                       AlgorithmIdentifier,
        subjectPublicKey                BIT STRING }
    Certificates                ::=     SEQUENCE {
        userCertificate                 Certificate,
```

```
        certificationPath                ForwardCertificationPath  OPTIONAL}
    ForwardCertificationPath  : : =      SEQUENCE  OF CrossCertificates
    CertificationPath         : : =      SEQUENCE  {
        userCertificate                  Certificate,
        theCACertificates                SEQUENCE  OF  CertificatePairOPTIONAL }
    CrossCertificates         : : =      SET OF Certificate
    CertificateList           : : =      SIGNED {SEQUENCE{
        signature                        AlgorithmIdentifier,
        issuer                           Name,
        this Update                      UTCTime,
        nextUpdate                       UTCTime OPTIONAL,
        revokedCertificates              SEQUENCE OF SEQUENCE  {
            userCertificate                  CertificateSerialNumber,
            revocationDate                   UTCTime  } OPTIONAL}}
    CertificatePair           : : =      SEQUENCE (
        forward       [0]    Certificate OPTIONAL,
        reverse       [1]    Certificate OPTIONAL
            - - Должна иметь место, по меньшей мере, одна из пар  - -}
        - - Типы атрибутов - -
    userPassword                         ATTRIBUTE     : : = {
      WITH SYNTAX                        OCTET STRING (SIZE
                                         (O . . ub-user-password))
      EQUALITY MATCHING  RULE  octetStringMatch
      ID                                     id-at-userPassword  }
userCertificate                ATTRIBUTE     : : = {
    WITH SYNTAX                Certificate
    ID                         id-at-userCertificate  }
cACertificate                  ATTRIBUTE     : : = {
    WITH SYNTAX                Certificate
    ID                         id-at-cAcertificate  }
authorityRevocationList        ATTRIBUTE     : : = {
    WITH SYNTAX                CertificateList
    ID                         id-at-authorityRevocationList  }
certificateRevocationList      ATTRIBUTE     : : = {
    WITH SYNTAX                CertificateList
    ID                         id-at-CertificateRevocationList  }
crossCertificatePair           ATTRIBUTE     : : = {
          WITH SYNTAX              CertificatePair
          ID                       id-at-crossCertificatePair  }
    - - Классы информационных объектов - -
ALGORITHM : : =  TYPE-IDENTIFIER
    - - Параметризованные типы - -
HASHED {ToBeHashed}                  : : =       OCTET STRING (CONSTRAINED-BY {
- - должно быть результатом применения процедуры хеширования к октетам значения параметра ToBeHashed,
    закодированным в соответствии с базовыми правилами кодирования (см. 8.7 }) - -
ENCRYPTED {ToBeEnciphered}           : : =       BIT STRING (CONSTRAINED BY  {
- - должно быть результатом применения процедуры шифрования к октетам значения параметра ToBeEnci p-
    hered, закодированным в соответствии с базовыми правилами кодирования (см. 8.7)  }) - -
SIGNED  {ToBeSigned}                 : : =       SEQUENCE  {
        toBeSigned                               ToBeSigned,
        COMPONENTS OF                            SIGNATURE {ToBeSigned}}
SIGNATURE {OfSignature}              : : =       SEQUENCE  {
        algorithmIdentifier                      AlgorithmIdentifier,
        encrypted                                ENCRYPTED {HASHED {OfSignature}}}
- - Присвоения объектных идентификаторов - -
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}
END
```

ПРИЛОЖЕНИЕ В
(справочное)

ТРЕБОВАНИЯ К ЗАЩИТЕ ИНФОРМАЦИИ

Примечание — Более подробная информация относительно требований к защите информации приведена в ГОСТ Р ИСО 7498—2.

Многие применения ВОС, а также службы, определенные МККТТ и не-МККТТ, предъявляют требования к защите информации. Такие требования возникают из необходимости защитить передачу информации от большого числа потенциальных угроз.

В.1 **Угрозы**

К некоторым общеизвестным угрозам относятся:

a) перехват идентичности — наблюдается злоупотребление идентичностью одного или нескольких пользователей, участвующих в обмене данными;

b) маскирование — попытка пользователя выдать себя за другого пользователя для получения доступа к информации или приобретения дополнительных привилегий;

c) повторные действия — регистрация и последующее повторение действий по обмену данными в последующее время;

d) перехват данных — просмотр данных пользователя несанкционированным пользователем в процессе обмена;

e) манипулирование — замена, вставка, удаление или нарушение последовательности данных пользователя несанкционированным пользователем в процессе обмена данными;

f) самоотказ — отрицание пользователем своего частичного или полного участия в обмене;

g) отклонение услуги — предотвращение или прерывание обмена, или задержка операций, критичных ко времени.

Примечание — Эта угроза защиты является одной из самых общих угроз и зависит от конкретного применения или намерения несанкционированного разрушения, и поэтому явно не рассматривается в основах аутентификации;

h) ошибочная маршрутизация — ошибочная маршрутизация пути обмена данными, предназначенного для одного пользователя к другому пользователю

Примечание — Ошибочная маршрутизация обычно возможна на уровнях 1—3 ВОС, поэтому она не рассматривается в основах аутентификации. Однако можно избежать последствий ошибочной маршрутизации, используя соответствующие услуги защиты, предусмотренные в основах аутентификации;

i) анализ трафика — просмотр информации, относящейся к обмену между пользователями (например, отсутствие, наличие, частота, направление, последовательность, тип, объем, и т. д.).

Примечание — Угрозы анализа трафика обычно не ограничиваются определенным уровнем ВОС, поэтому в общем случае анализ трафика не рассматривается в основах аутентификации. Однако от анализа трафика можно частично защититься генерацией дополнительного нераспознаваемого трафика (заполнением трафика), используя шифрованные или случайные данные.

В.2 **Услуги защиты**

Чтобы защититься от угроз, необходимо использовать различные услуги защиты. Услуги защиты, предусмотренные основами аутентификации, реализуются с помощью механизмов защиты, описанных в В.3 данного приложения.

a) Аутентификация равноправного логического объекта — эта услуга обеспечивает подтверждение того, что пользователь в определенном сеансе обмена является заявленным. Могут быть запрошены две различные услуги аутентификации равноправного логического объекта:

- аутентификация одного равноправного логического объекта (аутентификация логического объекта отправителя или получателя);
- взаимная аутентификация, когда оба обменивающихся пользователя выполняют аутентификацию друг с другом.

При запросе услуги аутентификации равноправного логического объекта оба пользователя договариваются, должны ли быть защищены их идентичности или нет.

Услуга аутентификации равноправного логического объекта поддерживается основами аутентификации. Она может быть использована для защиты от маскирования и повторных действий, касающихся идентичности пользователей.

b) Управление доступом — эта услуга может быть использована для защиты от несанкционированного использования ресурсов. Она обеспечивается справочником или другой прикладной программой и поэтому не относится к основам аутентификации.

c) Конфиденциальность данных — эта услуга может быть использована для обеспечения защиты данных от несанкционированного раскрытия информации и поддерживается основами аутентификации. Она может быть использована для защиты от перехвата данных.

d) Целостность данных — эта услуга обеспечивает подтверждение целостности данных при обмене и поддерживается основами аутентификации. Она может использоваться для выявления угрозы манипулирования данными и защиты от такой угрозы.

e) Безотказность — эта услуга обеспечивает подтверждение целостности и источника данных — то и другое в непредсказуемых взаимоотношениях, — что может быть удостоверено любым третьим участником в любой момент времени.

B.3 **Механизмы защиты**

Механизмы защиты выполняют услуги защиты, описанные в B.2:

a) обмен аутентификацией — существуют два вида механизма аутентификации, обеспечиваемых основами аутентификации:
   - простая аутентификация, рассчитанная на отправителя, поставляющего свое имя и пароль, которые проверяются получателем, и
   - строгая аутентификация, рассчитанная на использование криптографических методов защиты обмена действительной информацией. В основах аутентификации строгая аутентификация строится по асимметричной схеме.

Механизм обмена аутентификацией используется для обеспечения услуги аутентификации равноправного логического объекта.

b) Шифрование — основы аутентификации предусматривают шифрование данных во время передачи. Могут быть использованы симметричная или асимметричная схемы. Обмен необходимыми ключами в любом случае выполняется либо во время предыдущего обмена аутентификацией, либо автономно в любое время до предполагаемого обмена. Последний случай не входит в предмет рассмотрения основ аутентификации. Механизм шифрования поддерживает услугу конфиденциальности данных.

c) Целостность данных — этот механизм использует шифрование сжатой строки тех данных, которые должны быть переданы. Это сообщение вместе с открытыми данными посылается получателю. Чтобы убедиться в целостности данных, получатель повторяет сжатие и последующее шифрование открытых данных и сравнивает свой результат с результатом отправителя.

Механизм целостности данных может быть обеспечен шифрованием сжатых открытых данных по асимметричной или симметричной схеме. (По симметричной схеме сжатие и шифрование данных могут быть выполнены одновременно). Такой механизм явно не обеспечивается основами аутентификации. Однако он полностью обеспечивается в рамках механизма цифровой подписи (см. ниже), использующего асимметричную схему.

Механизм целостности данных обеспечивает услугу целостности данных и частично обеспечивает услугу безотказности (для полного удовлетворения своих требований такой услуге также необходим механизм цифровой подписи).

d) Цифровая подпись — этот механизм использует шифрование с помощью личных ключей отправителей сжатой строки тех данных, которые должны быть переданы. Цифровая подпись вместе с открытыми данными передается получателю. Подобно случаю с механизмом целостности данных, это сообщение обрабатывается получателем, чтобы убедиться в его целостности. Механизм цифровой подписи проверяет также подлинность отправителя и однозначные отношения между отправителем и переданными данными.

Основы аутентификации обеспечивают механизм цифровой подписи, используя асимметричную схему.

Механизм цифровой подписи обеспечивает услугу целостности данных, а также и услугу безотказности.

B.4 **Защита от угроз с помощью услуг защиты**

В таблице B.1 перечисленны угрозы и соответствующие услуги защиты от них. Наличие пометки «*» указывает, что соответствующая услуга обеспечивает защиту от определенной угрозы.

B.5 **Согласование услуг и механизмов защиты**

Для обеспечения средств защиты во время сеанса обмена данными необходимо согласование контекста, в котором требуются услуги защиты. Это влечет за собой согласование типа механизмов защиты и параметров, необходимых для обеспечения таких услуг защиты. Процедуры, требуемые для согласования механизмов и параметров, могут быть выполнсны либо как неотъемлемая часть нормальной процедуры установления связи, либо как отдельный процесс. Более подробно такие процедуры согласования не определяются в данном приложении.

Таблица В1 — Угрозы и защита

Угрозы	Аутентификация логического объекта	Конфиденциальность данных	Целостность данных	Безотказность
Перехват идентичности	* (Если требуется)			
Перехват данных		*		
Маскирование	*			
Повторные действия	* (Идентичность)		* (Данные)	*
Обработка			*	
Самоотрицание				*

ПРИЛОЖЕНИЕ С
(справочное)

ВВЕДЕНИЕ В КРИПТОГРАФИЮ КЛЮЧЕЙ ОБЩЕГО ПОЛЬЗОВАНИЯ

В общепринятых криптографических системах ключ, используемый для шифрования информации отправителем секретного сообщения, тот же, что и используемый для дешифрования сообщения законным получателем.

Однако в криптографических системах ключей общего пользования (ККОП) ключи вводятся парами: один для шифрования, другой для дешифрования. Каждая пара ключей связана с конкретным пользователем X. Один из ключей, общеизвестный как ключ общего пользования (X_o), может быть использован любым пользователем для шифрования данных. Только X, который обладает дополнительным личным ключом ($X_л$), может расшифровать данные. (В нотации это представлено в виде: $D = X_л[X_o\ [D]]$). Вычислительным путем невозможно получить личный ключ, зная ключ общего пользования. Любой пользователь может таким образом обмениваться частью информации, обнаружить которую может только X, зашифровав ее с помощью $X_л$. С помощью некоторого расширения два пользователя могут обмениваться секретно, используя ключ общего пользования друг друга для шифрования данных, как показано на рисунке С.1.

Пользователь А имеет ключ общего пользования A_o и личный ключ $A_л$, а пользователь В имеет другой набор ключей: B_o и $B_л$. А и В знают ключи общего пользования друг друга, но не знают личный ключ другой стороны. Поэтому А и В могут обмениваться секретной информацией друг с другом, используя следующие шаги (см. рисунок В.1).

1 Пользователь А желает послать некоторую секретную информацию x к В. Для этого А шифрует x, используя ключ шифрования В и посылает зашифрованную информацию e к В. Это представлено в виде:
$e = B_p[x]$.

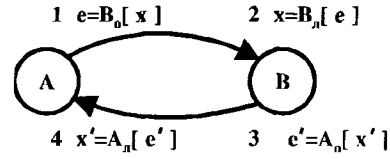


Рисунок С.1 — Использование ККОП для обмена секретной информации

2 Пользователь В может теперь расшифровать зашифрованную информацию e для получения информации x, используя секретный ключ расшифрования $B_л$. Заметим, что поскольку В является единственным владельцем ключа $B_л$ и этот ключ никогда не может быть раскрыт или передан, то для любой другой стороны невозможно получить информацию x. Владелец $B_л$ определяет идентичность В. Операцию расшифрования можно представить в виде:
$x = B_л[e]$ или $x = B_л[B_o[x]]$.

3 Теперь В может точно также послать А некоторую секретную информацию x', используя секретный ключ А, A_o:

$e' = A_o[x']$.

4 А получает x' путем дешифрования e':

$x' = A_д[e']$ или $x' = A_д\{A_o[x']\}$.

Таким способом А и В обмениваются секретной информацией x и x'. Эта информация не может быть получена никем другим, кроме А и В, при условии, что их личные ключи не раскрыты.

Такой обмен, как и передача секретной информации между сторонами, может быть использован для проверки их идентичности. В частности, А и В идентифицируются по обладанию ими секретными ключами дешифрования $A_д$ и $B_д$ соответственно. А может определить, обладает ли В секретным ключом дешифрования $B_д$, получая возвращаемую часть своей информации x в сообщении x'. Это показывает А, что обмен происходит с обладателем $B_д$. В аналогично может проверить идентичность А.

Некоторые ККОП обладают тем свойством, что шаги дешифрования и шифрования могут быть реверсивными как в $D = X_o[X_д[D]]$. Это позволяет любому пользователю (который обладает X_o) читать часть информации, которая могла быть отправлена только X. Следовательно, эта возможность может быть использована при выполнении сертификации источника информации, и является основой для цифровых подписей. Только те ККОП, которые обладают таким свойством (перестановки), пригодны для использования в этих основах аутентификации. Один из таких алгоритмов приведен в приложении D.

ПРИЛОЖЕНИЕ D
(справочное)

КРИПТОСИСТЕМА КЛЮЧА ОБЩЕГО ПОЛЬЗОВАНИЯ RSA

Примечания

1 Криптосистема, определенная в настоящем приложении, известна под названием RSA-Rivst-Shamir-Adleman (Ривс-Шамир-Адлеман).

2 Более подробная информация относительно этой криптосистемы приведена в приложении J.

D.1 **Назначение и область применения**

Полное рассмотрение криптосистемы RSA в данном приложении невозможно. Приведенное, однако, краткое ее описание предложено по методу, основанному на использовании модульного экспонирования.

D.2 **Определения**

Ключ общего пользования — пара параметров, состоящая из показателя степени общего пользования и арифметического модуля.

Примечание — Элемент данных в АСН.1 «subjectPublicKey», определенный в виде битовой строки (см. Приложение А), должен интерпретироваться в случае RSA как:

SEQUENCE (INTEGER, INTEGER),

где первое целое число — арифметический модуль, а второе — показатель степени общего пользования. Последовательность представляется базовыми правилами кодирования АСН.1.

Личный ключ — пара параметров, состоящая из секретного показателя степени и арифметического модуля.

D.3 **Символы и сокращения**

X, Y — блоки данных, которые арифметически меньше модуля

n — арифметический модуль

e — показатель степени общего пользования

d — секретный показатель степени

p, q — простые числа, произведение которых образует арифметический модуль (n)

Примечание — Простые числа предпочтительно иметь в количестве двух, хотя не запрещается использовать модуль с тремя и более первичными факторами.

lcm — наименьший множитель

mod n — арифметический модуль n.

D.4 **Описание**

Асимметричный алгоритм использует мощную функцию для преобразования блоков данных так, что

$Y = X^e \bmod n$ при $0 \le X < n$

$X = Y^d \bmod n$ при $0 \le Y < n$

Эти условия могут быть выполнены, например, при

$ed \bmod \text{lcm}(p-1, q-1) = 1$ или

$ed \bmod (p-1)(q-1) = 1$

Чтобы этот процесс действовал, блок данных должен интерпретироваться как целое число. Это достигается рассмотрением всего блока данных в виде упорядоченной последовательности битов (длиной λ). При этом

целое число формируется как сумма битов после назначения первому биту веса $2\lambda^{-1}$ к первому биту и делением на 2 каждого последующего бита (последний бит имеет вес 1).

Длина блока данных должна иметь наибольшее число октетов, содержащих меньше битов, чем модуль. Неполные блоки должны заполняться любым приемлемым способом. Может быть добавлено любое число блоков дополнительного заполнения.

D.5 **Требования к защите**

D.5.1 Длины ключей

Признано, что приемлемая длина ключа изменяется со временем и зависит от стоимости и доступности аппаратных средств, временных затрат, достижений в технологии и требуемой степени защиты. Рекомендуется, чтобы изначально принималась длина n — 512 битов, но значение этой длины может быть предметом дальнейшего изучения.

D.5.2 Генерация ключа

Защита RSA полагается на трудность факторизации параметра n. Существует несколько алгоритмов выполнения этой операции и, чтобы предотвратить использование какого-нибудь известного в настоящее время способа, значения p и q должны выбираться с особым вниманием согласно следующим правилам (см приложение J «Библиография»):

a) они должны выбираться по случайному закону;

b) они должны быть большими по значению;

c) они должны быть простыми числами;

d) $|p - q|$ должен иметь большое значение;

e) $(p + 1)$ должен иметь большую величину первичного фактора;

f) $(q + 1)$ должен иметь большую величину первичного фактора;

g) $(p - 1)$ должен иметь большую величину первичного фактора, например, r;

h) $(q - 1)$ должен иметь большую величину первичного фактора, например, s;

i) $(r - 1)$ должен иметь большую величину первичного фактора;

j) $(s - 1)$ должен иметь большую величину первичного фактора.

После создания ключа общего пользования и личного ключа, например «X_o» и «$X_д$», как определено в разделах 3 и 4 настоящего стандарта, содержащих d, e и n, значения p и q вместе со всеми другими данными, полученными в результате умножения $(p - 1)(q - 1)$, и первичные факторы, имеющие большие значения, должны быть аннулированы. Однако, сохраняя p и q локально, можно повысить производительность дешифрования в два — четыре раза. Считается, что сохранение p и q должно решаться локально.

Должно быть обеспечено соотношение $e > \log_2(n)$. Если оно не обеспечено, то простая операция извлечения корня целочисленной степени e блока шифротекста может раскрывать открытый текст.

D.6 **Показатель степени общего пользования**

Показатель степени общего пользования (e) должен быть общим для всей функциональной среды, чтобы минимизировать длину той части ключа общего пользования, которая фактически должна быть распределена, с целью уменьшения объема передачи и сложности преобразований.

Показатель степени e должен быть достаточно большим, но до такой степени, чтобы экспонирование можно было выполнять достаточно эффективно относительно времени обработки и емкости памяти. Если требуется фиксированный показатель степени общего пользования e, следует учитывать значительные преимущества использования числа Ферми F_4.

$$F_4 = 2^{2^4} + 1$$

= 65537 десятичное число, и

= 1 0000 0000 0000 0001 двоичное число

Примечания

1 Несмотря на то, что модуль n и показатель степени e являются общими, модуль не должен быть той частью, которая является общей для группы пользователей. Зная модуль n, показатель степени общего пользования e и секретный показатель степени d, достаточно определить факторизацию n. Поэтому, если бы модули были общими, то каждый смог бы вывести свои коэффициенты, определив тем самым секретный показатель степени каждого.

2 Фиксированный показатель степени должен быть большим и первичным, но он должен также обеспечивать эффективную обработку. Число Ферми F_4 удовлетворяет этим требованиям, например, аутентификация требует только 17 умножений и в среднем в 30 раз быстрее, чем дешифрование.

D.7 **Соответствие**

Хотя данное приложение и определяет алгоритм для общих и секретных функций, но оно не определяет сам метод вычислений, поэтому возможны различные результаты, которые согласуются с этим приложением и взаимно совместимы.

ПРИЛОЖЕНИЕ Е
(справочное)

ХЕШ-ФУНКЦИИ

Требования к хеш-функциям

Путем использования хеш-функции как однонаправленной функции защиты оказывается невозможно получить один и тот же результат хеширования при различных комбинациях входных сообщений.

Строгая хеш-функция должна удовлетворять следующим требованиям:

a) она должна быть однонаправленной, то есть при любом заданном результате хеширования вычислительным способом должно быть невозможно построить входное сообщение, хеширование которого дало бы такой же результат;

b) она должна быть свободна от конфликтов, то есть вычислительным способом должно быть невозможно построить два различных входных сообщения, хеширование которых дало бы одинаковый результат.

ПРИЛОЖЕНИЕ F
(справочное)

ЗАЩИТА ОТ УГРОЗ МЕТОДОМ СТРОГОЙ АУТЕНТИФИКАЦИИ

Метод строгой аутентификации, описанный в настоящей спецификации справочника, предлагает защиту от угроз, как описано в приложении В для строгой аутентификации.

Кроме того, существует следующий ряд потенциальных угроз, которые специфичны для самого метода строгой аутентификации.

Нарушение личного ключа пользователя — один из основных принципов строгой аутентификации состоит в том, что личный ключ пользователя остается защищенным. Существует ряд практических методов, доступных пользователю, чтобы сохранить свой личный ключ способом, обеспечивающим адекватную защиту. Последствия нарушения ограничены подверсией обмена данными с участием такого пользователя.

Нарушение личного ключа УС — личный ключ УС остается защищенным, что также является основным принципом строгой аутентификации. Применяют методы физической защиты и принцип «необходимо знать». Последствия нарушения ограничиваются подверсией обмена данными любого участвующего пользователя, аттестованного таким УС.

Ошибочное вовлечение УС в выработку недействительного сертификата означает, что УС самостоятельно вырабатывают некоторую защиту. Обязанность УС состоит в том, чтобы до выдачи сертификата убедиться в действительности предоставляемых строгих удостоверений личности. Последствия нарушения ограничиваются под версией обмена данными любого участвующего пользователя, которому был выдан сертификат, и любого другого пользователя, на которого повлиял недействительный сертификат.

Секретное соглашение между нарушителем УС и пользователем — атака на такое соглашение может нарушить этот метод. Это приведет к предательству УС, обладающего доверием. Последствия наличия УС-нарушителя ограничиваются подверсией обмена данными любого участвующего пользователя, аттестованного таким УС.

Подделка сертификата — метод строгой аутентификации защищает от подделки сертификата наличием подписи УС. Этот метод зависит от сохранения секретности личного ключа УС.

Подделка маркера — метод строгой аутентификации защищает от подделки маркера при наличии подписи отправителя. Этот метод зависит от сохранения секретности личного ключа отправителя.

Повторные действия маркера — одно- и двунаправленная аутентификация защищают от повторных действий маркера включением временной метки в маркер. Метод трехнаправленной аутентификации выполняет это на основе проверки случайных чисел.

Вторжение в криптографическую систему — вероятность эффективного криптоанализа системы основана на преимуществах вычислительной теории чисел и обусловливает необходимость разумно предсказуемой большой длины ключа.

ПРИЛОЖЕНИЕ G
(справочное)

КОНФИДЕНЦИАЛЬНОСТЬ ДАННЫХ

G.1 **Введение**

Процесс конфиденциальности данных может быть инициирован после обмена необходимыми ключами шифра. Конфиденциальность данных может быть обеспечена предшествующим обменом данными аутентификации в соответствии с разделом 9 или процессом обмена некоторым другим ключом, который, однако, не входит в предмет рассмотрения настоящего стандарта.

Конфиденциальность данных может быть обеспечена применением схемы асимметричного или симметричного шифрования.

G.2 **Конфиденциальность данных асимметричным шифрованием**

В этом случае конфиденциальность данных достигается средствами отправителя, шифрующего данные, которые должны быть переданы с использованием ключа общего пользования заданного получателя: получатель затем должен расшифровать их, используя свой личный ключ.

G.3 **Конфиденциальность данных симметричным шифрованием**

В этом случае конфиденциальность данных достигается использованием алгоритма симметричного шифрования. Его выбор не рассматривается в основах аутентификации.

Если обмен данными аутентификации в соответствии с разделом 9 выполнен двумя вовлеченными сторонами, то ключ для использования симметричного алгоритма может быть получен. Выбор личных ключей зависит от используемого метода преобразования. Стороны должны быть уверены, что они обладают строгими ключами. Настоящая спецификация справочника не определяет, как это осуществить, хотя ясно, что соответствующий метод должен быть согласован участвующими сторонами или определен в других стандартах.

ПРИЛОЖЕНИЕ H
(обязательное)

ЭТАЛОННОЕ ОПРЕДЕЛЕНИЕ ИДЕНТИФИКАТОРОВ ОБЪЕКТОВ АЛГОРИТМА

Данное приложение определяет идентификаторы объектов, используемых в алгоритмах аутентификации и шифрования, при отсутствии формального регистра. Предполагается использовать такой регистр, как только он станет доступным. Определения используют форму модуля ACH.1, «AlgorithmObjectIdentifiers».

```
AlgorithmObjectIdentifiers  joint-iso-ccitt ds(5) module(1)  algorithmObjectIdentifiers(8) 2}
DEFINITIONS : : =
BEGIN
- EXPORTS ALL -
- - Определенные в этом модуле типы и значения экспортируются для использования в других модулях
- - ACH.1, содержащихся в спецификациях справочника, и другими прикладными программами, которые
- - будут, в свою очередь, использовать их для доступа к услугам справочника. Другие прикладные програм-
- - мы могут использовать их для своих собственных целей, но это не препятствует расширениям и
- - модификациям, необходимым при обслуживании или усовершенствовании справочной службы.
IMPORTS
   algorithm, authenticationFramework
      FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1)
                              usefulDefinitions(0)  2}
   ALGORITHM
      FROM AuthentificationFramework authentificationFramework;
- - Категории идентификаторов объектов - -
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
- - Присвоения идентификаторов объектов - -
id a-rsa                          OBJECT IDENTIFIER  : : =  {id-ea 1}
- - Следующие присвоения идентификаторов объектов резервируют значения, предназначенные для предупреждающих функций
id-ha-sqMod-n          OBJECT IDENTIFIER    : : =  {id-ha 1}
id-sa-sqMod-nWithRSA   OBJECT IDENTIFIER    : : =  {id-sa 1}
END
```

ПРИЛОЖЕНИЕ J
(справочное)

БИБЛИОГРАФИЯ

1 Rivest, R. L., Shamir, A., and Adleman, L. A Method for Obtaining Digital Signatures and Public-key Cryptosystems, Communications of the ACM, 21,2 (Febrary 1978), 120—126

2 Gordon, J Strong RSA Keys, Electronics Letters, 20,5, 514—516. Quisquater, J. J., and Couvreur, c. Fast Decipherment Algorithm for RSA Public-key Cryptosystems, Electronics Letters, 18—21 (October 14, 1982), 905—907

УДК 681.324:006.354 ОКС 35.100.70 П85 ОКСТУ 4002

Ключевые слова: обработка данных, обмен информацией, взаимосвязь сетей, взаимосвязь открытых систем, справочники

Редактор *В. П. Огурцов*
Технический редактор *В. Н. Прусакова*
Корректор *О. Я. Чернецова*
Компьютерная верстка *З. И. Мартыновой*

Изд. лиц. № 021007 от 10.08.95. Сдано в набор 28.05.98. Подписано в печать 06.08.98. Усл. печ. л. 3,72. Уч.-изд. л. 3,54. Тираж 238 экз. С 938. Зак. 1138

ИПК Издательство стандартов,107076, Москва, Колодезный пер., 14.
Набрано в Калужской типографии стандартов на ПЭВМ.
Калужская типография стандартов, ул. Московская, 256.
ПЛР № 040138

gost.e-ecolog.ru