Лабораторная работа №7

Цифровая подпись RSA

Цель: Приобрести практические навыки в реализации цифровой подписи RSA по стандарту CAdES и PKCS 7.

Введение

Цифровая подпись является некоторой фиксированной строкой, сформированной с помощью специального алгоритма на основе подписываемого документа, позволяющей предотвратить возможность отказа от авторства подписанного документа, а также присвоения авторства для подписанного документа.

Существует 3 подхода к построению схем цифровой подписи, но наиболее распространенным является подход на основе ассиметричных шифров. Рассмотрим цифровую подпись на основе шифра RSA.

Пусть , где p,q - простые числа.

Выбирается пара чисел e и d такие, что ).

Открытым ключом цифровой подписи (ключом проверки подписи) является пара (n,e). Закрытым ключом (ключом формирования подписи, ключом подписи) является набор (p, q, d). Обозначим через - значение хеш-функции от сообщения.

Формирование цифровой подписи производится по следующей формуле:

,

где mod - операция взятия остатка от деления на число; c - шифртекст; m - открытый текст.

Проверка цифровой подписи производится по следующей формуле.

.

Если выполняется последнее равенство, то цифровая подпись считается верной. Стандартом, который регулирует структуру цифровой подписи для документа, является стандарт PKCS 7, рассмотренный немного ранее. Согласно стандарта PKCS 7 подписанные данные имеют следующую структуру: {CMSVersion, DigestAlgorithmIdentifiers, EncapsulatedContentInfo, CertificateSet OPTIONAL, RevocationInfoChoises OPTIONAL, SignerInfos}, где

CMSVersion - версия используемого синтаксиса. Определяется по достаточно алгоритму, описанному в стандарте. Зависит от использования сертификатов открытых ключей и списка отозванных сертификатов. Так как в рамках работы не используются сертификаты, то значение поля = 1.

DigestAlgorithmIdentifiers - идентификатор алгоритма хеширования. В зависимости от реализации имеет разный вид. В рамках рассматриваемой работы можно рассматривать его вид, как AB, где A={GOST, SHA} - алгоритм хеширования используемый в подписи; B={256,512} - размер вырабатываемого хеша. Например, GOST512.

EncapsulatedContentInfo - встраиваемая информация о подписываемом контенте. Имеет следующую структуру: {ContentType, OCTET STRING OPTIONAL}, где

ContentType - вид подписываемой информации;

OCTET STRING OPTIONAL - опциональная строка данных;

CertificateSet OPTIONAL - опциональные сведения о сертификате открытого ключа, с помощью которого можно проверить подпись.

RevocationInfoChoises OPTIONAL - опциональные сведения о списке отозванных сертификатов, с помощью которого можно проверить подлинность сертификата открытого ключа.

SignerInfos - информация о подписывающем документ. Данная информация имеет следующую структуру: {CMSVersion, SignerIdentifier, DigestAlgorithmIdentifier, SignedAttributes OPTIONAL, SignatureAlgorithmIdentifier, SignatureValue, UnsignedAttributes OPTIONAL}, где

CMSVersion - версия используемого синтаксиса. Такая же, как и в основном теле.

SignerIdentifier - идентификатор автора. Обычно указывается серийный номер открытого ключа из сертификата открытых ключей. В данном случае указывается имя и фамилия подписывающего.

DigestAlgorithmIdentifier - идентификатор алгоритма хеширования. В зависимости от реализации имеет разный вид. В рамках рассматриваемой работы можно рассматривать его вид, как AB, где A={GOST, SHA} - алгоритм хеширования используемый в подписи; B={256,512} - размер вырабатываемого хеша. Например, GOST512.

SignedAttributes OPTIONAL - опциональное поле подписанных атрибутов.

SignatureAlgorithmIdentifier - идентификатор алгоритма подписи. Принимает значение, RSAdsi или DSAdsi (алгоритм Эль-Гамаля), либо идентификатор другого алгоритма.

SignatureValue - значение цифровой подписи в шестнадцатиричном виде.

UnsignedAttributes OPTIONAL - опциональное поле неподписанных атрибутов. В рамках рассматриваемой работы будет использоваться данное поле. Поле имеет структуру: {OBJECT IDENTIFIER, SET OF AttributeValue}, где OBJECT IDENTIFIER - идентификатор объекта;

SET OF AttributeValue - набор атрибутов.

Ключи хранятся согласно стандартам PKCS 8 и PKCS 12.

Стандарт синтаксиса криптографических сообщений расширенного для электронных подписей (CAdES), представляет собой расширение стандарта PKCS 7, устраняющее основные недочеты стандарта PKCS 7. В частности, таким недочетом является отсутствие указания времени создания подписи. Будем рассматривать стандарт CAdES-T, который как раз и включает в себя процедуру установки штампа времени в рамках данного алгоритма.

В данном случае поле неподписанных атрибутов будет иметь OBJECT IDENTIFIER = signature-time-stamp и SET OF AttributeValue={хеш от подписанного пользователем сообщения (подпись конкатенируется с документом), timestamp, подпись центра штампов времени от хеша подписанного пользователем документа и timestamp, удостоверение центра штампов времени}, где timestamp имеет следующий формат: {UTCTime, GeneralizedTime}, где

UTCTime - время по Гринвичу в формате YYMMDDHHMMSSZ, где YY - год; MM- месяц; DD- день; HH - часы; MM - минуты; SS - секунды;

GeneralizedTime - время центра штампов времени в формате YYMMDDHHMMSSZ.

Подпись центра штампов времени - подпись в формате CAdES. Центр штампов времени выступает центром штампов времени относительно самого себя;

Удостоверение центра штампов времени - сертификат центра штампов времени. В работе в этом поле содержится открытый ключ центра штампов времени по стандарту PKCS 8.

Протокол получения штампа времени следующий:

1. Подписанный документ с конкатенированной цифровой подписью (без заполненного поля неподписанных атрибутов, в которое будет включаться метка времени), хеш от данного документа.

2. Центр штампов времени: Проверяет корректность хеша.

SET OF AttributeValue.

3. Подписант: проверяет значение подписи центра штампов времени. Если все корректно, то подпись доформируется за счет включения поля времени временной метки.

1. Рабочее задание

1.1 Реализовать схему цифровой подписи RSA в формате CAdES.

2. Требования к реализации

2.1 Должен присутствовать выбор между хеш-функциями, которые будут использоваться. в рамках схемы цифровой подписи.

2.2 Должен быть реализована служба центра штампов времени на отдельной виртуальной машине, которая проверяет наличие сгенерированного открытого и закрытого ключа (соответствующих стандартам PKCS 8 и PKCS 12). Ключи должны записываться в отдельные файлы.

2.3 Штамп времени должен получаться по сети, а не локально.

2.4 Должна быть реализована процедура генерации ключа для пользователя.

2.5 Должна поддерживаться кодировка Unicode

2.6 Должен производиться выбор между процедурой генерации ключей, процедурой формирования подписи, процедурой проверки подписи.

2.7 Должна быть реализована процедура чтения из файла для подписываемого документа, ключей, подписи.

2.8 Процедура проверки подписи должна выполнятся самостоятельно без выбора пользователем алгоритмов использованных для создания подписи. Результат проверки должен выводится на экран в формате:

Результат проверки подписи

Алгоритм хеширования, использованный при создании подписи

Алгоритм подписи

Автор подписи

Время создания подписи

2.9 Нельзя пользоваться готовыми реализациями ГОСТ Р 34.11-2012, SHA-256, SHA-512, RSA.

Подписать исходный код своей электронной подписью, выслать на адрес ladg91@mail.ru с темой lab7, после получения подписанного преподавателем исходного кода, прийти и защитить его.