УДК 004.33

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПРОТОКОЛОВ OCSP И TSP

Мерзликин Н. Ю., Платонов В. Ю., Лукьянов В. С., Быков Д. В.

ФБГУ ВПО «Волгоградский государственный технический университет», Волгоград, Россия (400131, г. Волгоград, пр. Ленина, 28), e-mail: nik-merzlikin@yandex.ru

В статье описывается особенности реализации протоколов удостоверяющего центра OCSP и TSP. Для построения алгоритма реализации этих протоколов были описаны каждый из них. В описании OSCP было представлено взаимодействие между доверяющей стороной и OSCP-сервером. Также описана процедура взаимодействия между TSP клиентом и доверенной третьей стороной TSA. В результате, в ходе проведенных исследований была спроектирована общая структура библиотеки для протоколов TSP и OCSP. Ключевым компонентом структуры является ASN.1 декодер. Реализация спроектированной библиотеки классов позволит серьезно повысить уровень безопасности мобильных приложений.

Ключевые слова: протокол TSP, электронное правительство, протокол OCSP, удостоверяющий центр, декодер, TSA.

IMPLEMENTATION FEATURES OF OCSP AND TSP PROTOCOLS

Merzlikin N. J., Platonov V. J., Lukyanov V. S., Bykov D. V.

Volgograd State Technical University, Volgograd, Russia (400131, Volgograd, Lenin's avenue, 28), e-mail: nik-merzlikin@yandex.ru

There is a growing need for services provided by the certifying centers. This is due to the fact that information technologies play an increasingly prominent place in human life. The introduction of e-government, the introduction of universal electronic cards, personal digital electronic signature for every citizen of the Russian Federation – all of which can use the certification authorities in their work. This paper describes the features of the protocols of the certification center OCSP and TSP. The algorithm implementation of these protocols have been described each of them. The description was provided OSCP interaction between relying party and OSCP-server. Also describes how the interaction between the TSP client and a trusted third party TSA. As a result, the research has been designed for the general structure of the library of protocols TSP and OCSP. A key component of the structure is the ASN.1 decoder. Implementation of the class library designed will greatly enhance the security of mobile applications.

Keywords: protocol TSP, e-government, protocol OCSP, certification center, decoder, TSA.

Введение

В настоящее время растет потребность к сервисам, предоставляемым удостоверяющими центрами. Это связано с тем, что информационные технологии занимают все большее место в жизни человека. Внедрение электронного правительства, введение универсальных электронных карт, личной цифровой электронной подписи для каждого гражданина РФ – все это может использовать удостоверяющие центры в своей работе.

Удостоверяющие центры поддерживают ряд дополнительных протоколов, которые позволяют получать расширенную информацию о цифровых сертификатах. Примерами таких протоколов являются протоколы OCSP и TSP. Рассмотрим их возможности более подробно.

Протокол OCSP

Протокол OCSP (Online Certificate Status Protocol) – протокол получения статуса сертификата в реальном времени применяется для предоставления пользователям УЦ актуальной информации о статусах сертификатов ключей подписи. По протоколу OCSP можно получить информацию об изменении статуса цифрового сертификата в реальном времени. Протокол OCSP необходим при работе с важной информацией, например, при обмене ценными бумагами крупного достоинства или масштабных фондовых продаж.

Протокол OCSP работает по принципу «запрос-ответ». OCSP-клиент генерирует OCSP-запрос и отправляет его на сервер. OCSP-сервер получает этот запрос, проверяет статус сертификата, генерирует OCSP-ответ и отсылает его клиенту. Рассмотрим подробнее структуру OCSP запросов и ответов.

OCSP-запрос состоит из номера версии протокола, типа запроса на обслуживание и одного или нескольких идентификаторов сертификатов. Идентификатор сертификата включает хэш-коды отличительного имени и открытого ключа издателя сертификата, а также серийный номер сертификата. В запросе иногда могут присутствовать необязательные дополнения.

OCSP-ответ состоит из идентификатора сертификата, статуса сертификата ("нормальный", "аннулированный" или "неизвестный") и срока действия ответа, связанного с идентификатором каждого указанного в исходном запросе сертификата. Если сертификат имеет статус аннулированного, то отображается время аннулирования и может быть указана причина аннулирования (необязательно). Срок действия задается интервалом от текущего обновления до следующего обновления. Ответ может содержать необязательные дополнения, а также код ошибки, если обработка запроса не была завершена корректно.

Рис. 1 иллюстрирует взаимодействие между доверяющей стороной и OCSP-респондером (OSCP-сервером). OCSP-ответы должны быть заверены цифровой подписью, гарантирующей, что ответ исходит от доверенного субъекта и не был изменен при передаче.

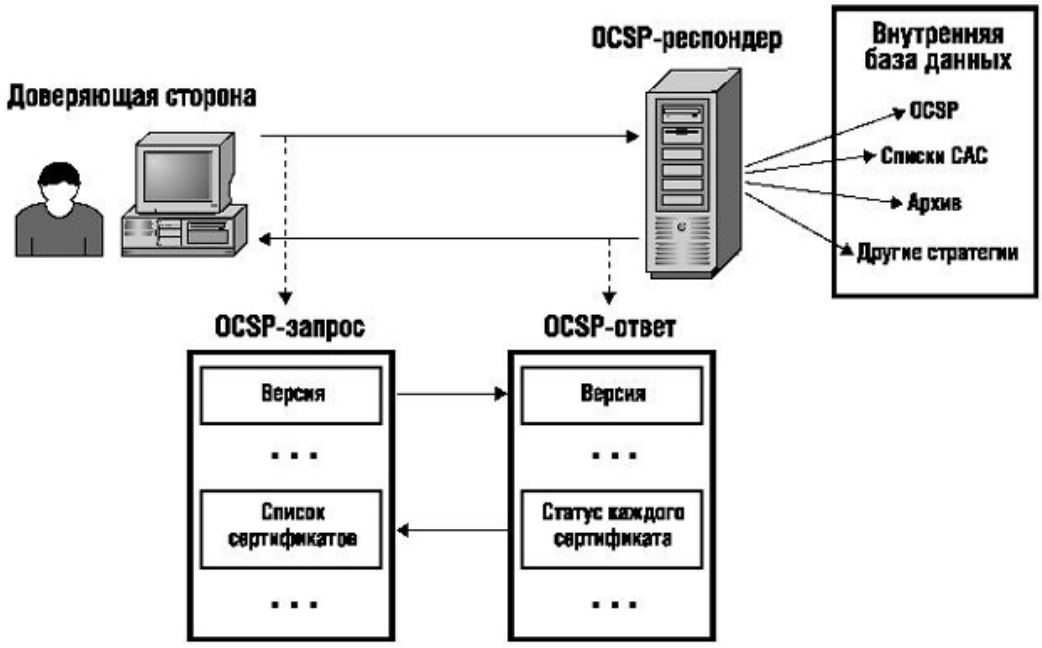

Рис 1. Процедура проверки статуса сертификата по протоколу OSCP

Протокол TSP

TSP протокол представляет собой набор методов, позволяющих установить, является ли электронный документ созданным или подписанным в (или раньше) определенное время. На практике, в большинстве случаев системы с метками времени используют в качестве доверенной третьей стороной – Time-Stamping Authority (TSA). TSA является доверенной третьей стороной, которая создает временную метку для того, чтобы указать, что данный документ существовал в определенный момент времени. Для того чтобы связать документ с конкретным моментом времени, сервис Time Stamp (TSA) должен быть использован.

Это может быть использовано, например, для проверки цифровой подписи, которая была применена к сообщению перед аннулированием сертификата, что позволяет отменить сертификат открытого ключа, который будет использоваться для проверки подписей, созданных до аннулирования.

Значение метки времени становится ясно, когда есть необходимость законного использования электронных документов длительное время. Без фиксации времени мы не можем доверять подписанным документам в тех случаях, когда они подписаны с помощью криптографических примитивов, которые стали ненадежными, и в тех случаях, когда подписавший сам отказывается от подписи, утверждая, что он случайно потерял свой ключ. В последние годы, особенно в контексте правового регулирования использования цифровых подписей, организационные и правовые аспекты временных меток стали предметом всемирного внимания. В дополнение к определению обязанностей владельца подписи,

обязанности и ответственность третьих лиц (TSA) тоже указаны. Следовательно, существует растущий интерес к системам с метками времени, которые должны доверять TSA.

Метка времени – цифровая аттестация TSA, которая идентифицирует электронный документ, которая была получена от TSA в определенное время.

Схему работы протокола TSP можно представить следующим образом:

1. Создание уникального тега. TSP используется для отметки времени данных, таких как файл или электронной почты. Первый шаг состоит в вычислении уникального тега для этого файла. Наиболее часто используемые алгоритмы включают MD5, MD2, MD4, SHA1, SHA, MDC2. Эти алгоритмы вычисляют хэш-функции, которые однозначно идентифицируют цифровые данные.
2. Отправление запроса к TSA. Клиент посылает тег сообщения TSA. Сообщение упаковано в структуру, состоящую из:
   - алгоритм кодирования сообщения;
   - тег сообщения.
3. TSA считывает информацию из сокета, извлекает оттуда сообщение с тегом. TSA поставит к сообщению дату, время и создать отметку времени для клиента. TSA будет использовать свой закрытый ключ, чтобы подписать эту отметку времени.
4. Отправление отметки времени клиенту.
5. Проверка данных. Клиент, который получил отметку времени, может ее проверить. Клиент расшифровывает отметку, извлекает из нее тег сообщения и сравнивает с оригинальным тегом сообщения. Если они не соответствуют, значит, отметка времени испорчена или она не для этого файла [1].

Особенности реализации протоколов OCSP и TSP

Протоколы OCSP и TSP позволяют серьезно повысить уровень безопасности как в настольных приложениях, так и при использовании их на мобильных устройствах. Мобильные устройства стали неотъемлемой частью повседневной жизни человека. При работе на них с важной или конфиденциальной информацией возникает необходимость ее защиты. Поэтому актуальным является проектирование библиотек классов, для реализации функций протоколов OCSP и TSP на мобильных платформах. Это позволит обеспечить максимальную безопасность приложений при работе с защищенными данными.

В ходе проведенных исследований была спроектирована следующая общая структура библиотеки для протоколов OCSP и TSP:

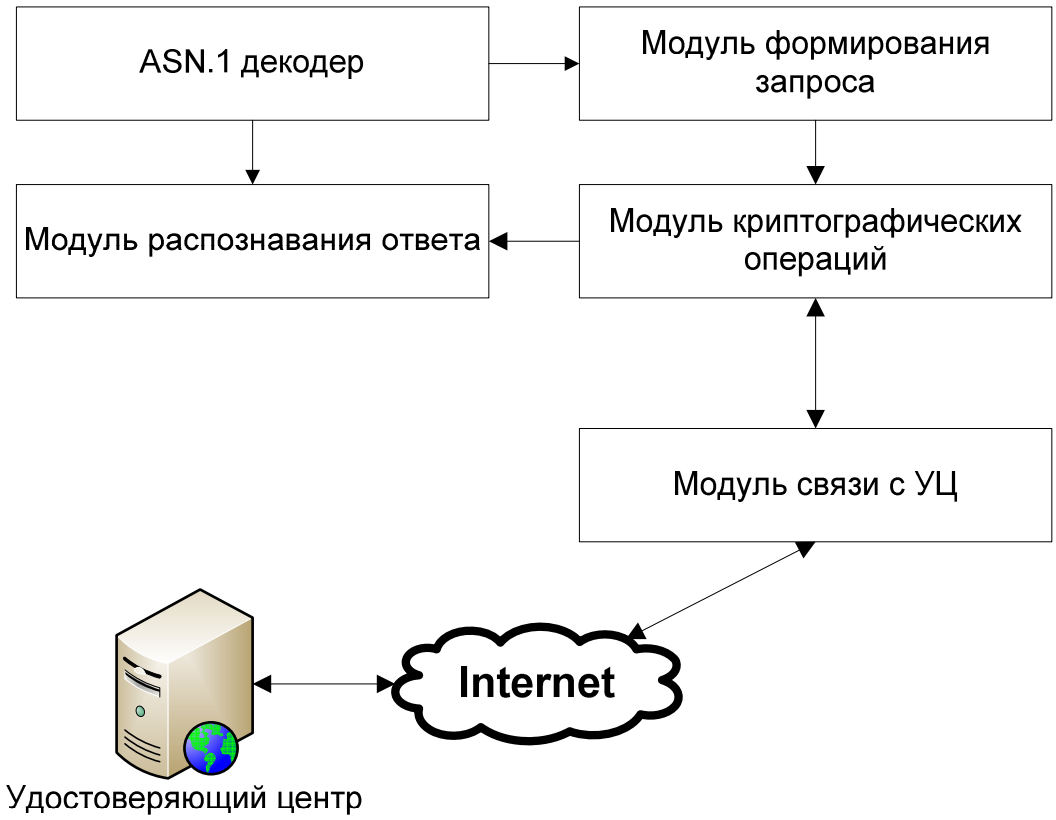

Рис. 2. Структурная схема библиотеки для работы по протоколам TSP и OCSP

Основной частью библиотеки является ASN.1 декодер. Обмен сообщениями с серверами OCSP и TSP производится по нотации ASN.1 по правилам кодирования структур данных DER (Distinguished Encoding Rules).

Для начала поясним, для чего же создавался этот стандарт. В мире существует множество различных компьютеров. И кроме того существует множество стандартов представления данных в этих компьютерах. ASN.1 создавался как некий общий стандарт, позволяющий описывать произвольную информацию, которая бы понималась любым компьютером, имеющим представление об этом стандарте. В стандарте ASN.1 поэтому предъявляются жесткие правила кодирования даже на уровне отдельных битов информации, а также взаимного их расположения. Дополнительно нужно сказать, что стандарт ASN.1 кодирует информацию не в виде текста, а виде двоичных последовательностей. Сейчас уже появились вариации форматов кодирования, позволяющие представлять данные и в виде текста (XML), но обзор этих форматов выходит за рамки данной статьи. Данные, закодированные в формате ASN.1, представляют из себя последовательность байт (или "октетов"), которые идут один за другим, без каких либо разрывов. Последовательность, закодированную в ASN.1, можно передавать по линиям связи, сохранять в файл – блок закодированной информации в ASN.1 уже содержит необходимое описание его общей длины и содержимого.

Для возможности подобного описания содержащейся в закодированном блоке информации применяется определенная общая структура каждого блока. Каждый блок содержит минимум 3 обязательных части (в отдельных случаях остаются только первые два блока, но эти случаи описываются отдельно):

1. Часть идентификатора блока (до нескольких октетов);
2. Часть общей длины блока (до нескольких октетов);
3. Часть, содержащая собственно значение, которое переносит этот блок (до нескольких октетов);

ASN.1 определяет стандарт записи, описывающий структуры данных для представления, кодирования, передачи и декодирования данных. Он обеспечивает набор формальных правил для описания структуры объектов, которые не зависят от конкретной машины. ASN.1 описывается стандартом X.690. Правила кодирования, описанные в X.690, служат для представления структур данных, описанных по правилам ASN.1, в виде последовательностей байт. Правила кодирования, описанные в X.690, служат для представления структур данных, описанных по правилам ASN.1, в виде последовательностей байт. Такие последовательности удобнее передавать по линиям связи или сохранять в файлы, чем делать те же операции с самими структурами.

Приведем описание правил кодирования типа INTEGER:

Общее описание типа:

- Номер тэга – 2;
- Форма кодирования значения – примитивная (не конструктивная форма).

В ASN.1 могут быть закодированы как положительные, так и отрицательные числа. Каждое целое число практически не ограничено в своей величине (то есть в ASN.1 могут быть закодированы сколь угодно большие (по модулю) целые числа).

Каждое целое число кодируется последовательностью октетов, каждый октет представляет собой 8 бит информации. Каждый октет представляет собой «вес» перед соответствующей степенью числа 256, участвующий в разложении кодируемого числа по основанию 256. То есть для кодирования числа исходное число сначала разлагается по основанию 256, и затем значения «весов» перед соответствующими степенями 256 кодируются в качестве октетов. Например, число 8388607 будет кодироваться по следующей схеме:

- Разложим число по основанию 256: 838860710 = 127*2562 + 255*2561 + 255*2560;
- Получаем, что «веса» при соответствующих степенях 256 будут: 127, 255 и 255;

- Переводим каждое число в последовательность бит, а затем кодируем эту последовательность бит, группируя в группы по 4 бита. Получаем следующие значения для «весов»: 7F FF FF;

Теперь рассмотрим более сложный пример кодирования составной структуры ASN.1:

```
SampleProtocol DEFINITIONS ::= BEGIN
   SampleQuestion ::= SEQUENCE {
      trackingNumber INTEGER,
      question       IA5String
   }
END
```

Допустим, что необходимо закодировать следующие значения:

```
myQuestion SampleQuestion ::= {
   trackingNumber     5,
   question           "Anybody there?"
}
```

По правилам DER эта структура кодируется следующим образом:

30 13 02 01 05 16 0e 41 6e 79 62 6f 64 79 20 74 68 65 72 65 3f, где

30 – значение SEQUENCE

13 – длина значения в октетах

02 – значение INTEGER

01 – длина значения в октетах

05 – само значение

16 – значение IA5String

0e – длина значения в октетах

41 6e 79 62 6f 64 79 20 74 68 65 72 65 3f – значение ("Anybody there?").

Структуры ASN.1, необходимые для работы по протоколам OCSP и TSP, описаны в RFC 2560 и RFC 3161. При реализации библиотеки классов основной задачей является написание корректного ASN.1 DER декодера, позволяющего кодировать запросы к серверу и распознавать ответы от него.

Обмен данных с сервером можно осуществлять по протоколу HTTP. Запрос, использующий метод POST, конструируется следующим образом: заголовок Content-Type имеет значение application/ocsp-request или application/tsp-request, а тело сообщения является бинарным значением DER-представления OCSPRequest (TSPRequest).

Ответ OCSP, основанный на HTTP, создается в соответствии с заголовками HTTP, за которыми следует бинарное значение DER-представления OCSPResponse (TSPResponse).

Заголовок Content-Type имеет значение application/ocsp-response или application/tsp-response. Заголовок Content-Length должен специфицировать длину ответа. Остальные заголовки HTTP могут присутствовать и могут игнорироваться, если запрашивающим они не распознаются.

Реализация библиотеки классов для работы по протоколам OCSP и TSP позволит серьезно повысить уровень безопасности мобильных приложений, работающих с конфиденциальными данными.

Список литературы

1. Использование TSP [Электронный ресурс] / КриптоПро. – 2013. – Режим доступа: http://www.cryptopro.ru/products/pki/tsp/usage
2. Механизмы онлайновых запросов [Электронный ресурс] / inSSL – все о SSL технологиях. – 2013. – Режим доступа : http://www.inssl.com/main-info-cat.html.
3. On-line протокол статуса сертификата [Электронный ресурс] / Негосударственное образовательное частное учреждение «Национальный Открытый Университет «ИНТУИТ». – 2013. – Режим доступа: http://www.intuit.ru/studies/courses/59/59/ lecture/904?page=1.
4. Online Certificate Status Protocol [Электронный ресурс] / RFC. – 2013. – Режим доступа: http://www.ietf.org/rfc/rfc2560.txt
5. Time-Stamp Protocol [Электронный ресурс] / RFC. – 2013. – Режим доступа: http://www.ietf.org/rfc/rfc3161.txt
6. ITU-T Recommendation X.690 [Электронный ресурс] / ASN.1 encoding rules. – 2013. – Режим доступа: http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.690-200811-I!!PDF-E&type=items

Рецензенты:

Камаев Валерий Анатольевич, д-р техн. наук, профессор, заведующий кафедрой САПР и ПК федерального государственного бюджетного образовательного учреждения высшего профессионального образования ВолгГТУ, г. Волгоград.

Муха Юрий Петрович, д-р техн. наук, профессор, заведующий кафедрой «Вычислительная техника» федерального государственного бюджетного образовательного учреждения высшего профессионального образования ВолгГТУ, г. Волгоград.