УДК 004.3

Н. А. Жиляк, кандидат технических наук, доцент (БГТУ);
Н. П. Цыганенко, студент (БГТУ)

ШИФРОВАНИЕ XML-ДОКУМЕНТА СРЕДСТВАМИ .NET

В статье приводятся результаты исследования средств для шифрования XML-документа, предоставляемых компанией Microsoft в рамках платформы .Net Framework. Данное исследование позволяет сформировать на основании выявленных преимуществ и недостатков требования к собственному программному продукту для шифрования XML-данных. Также приводится структура проектируемого приложения и результаты патентных исследований в данной области.

In article results of research tools for encrypting XML document provided by Microsoft as part of the. Net Framework platform. This research allows to generate the requirements for their own software products for encrypting XML data based on the identified strengths and weaknesses. Also shows the structure of the designed application and the results of patent research in this field.

Введение. В современном мире формат представления данных Extensible Markup Language (XML) становится одним из стандартов при передаче данных по сети Интернет. Главным преимуществом XML по сравнению с другими форматами электронных документов является то, что в нем описание внешнего представления документа отделено от структуры документа и его содержания. XML является гибким языком, который можно использовать для различных целей, при этом он способен обеспечить взаимодействие со многими системами и базами данных.

Формат представления данных XML имеет широкий круг применения. Например, среди наиболее типичных примеров можно выделить конфигурационные файлы (в .Net Framework – web.config, app.config, machine.config), отчетную документацию (с использованием технологии XSLT), новостные ленты (технология RSS), web-сервисы (протокол SOAP, язык описания сервисов WSDL), приложения WPF и Silverligth (язык XAML).

Среди основных технологий, обладающих огромным потенциалом и использующих XML-формат, выделяют web-сервисы, использование которых с каждым днем растет. В связи с этим возникает естественная необходимость в шифровании информации от злоумышленников.

Структура XML-документа позволяет одинаково легко зашифровать его как целиком, так и частично. Например, в целях экономии ресурсов и времени можно зашифровать только конфиденциальные данные документа.

Шифрование XML-файла имеет некоторые отличительные особенности от других механизмов обеспечения безопасности передаваемых данных (таких как IPSec, TLS, SSL и прочие):

– обеспечивает безопасность данных, а не сеанса;

– данные могут быть частично зашифрованы, что позволяет в рамках одного документа разграничивать доступ между разными лицами [1].

В консорциуме W3C была создана рабочая группа, которая специально занималась вопросами шифрования XML-данных. В 2001 г. выпущена первая спецификация, носящая название XML Encryption Syntax and Processing (синтаксис и выполнение шифрования XML). Последние изменения сделаны 24 января 2013 г.

Спецификация носит рекомендательный характер. Она определяет основные правила шифрования и дешифрования XML-документов, синтаксис шифрования, пространство имен XML Encryption, элементы шифрования и используемые классы XML Encryption.

Рассматриваемая спецификация W3C определяет три возможных варианта шифрования: весь файл, тег или содержимое тега.

Частичное шифрование XML-документа может использоваться как с целью разграничения доступа между пользователями с разными ключами, так и для уменьшения количества передаваемой информации. При шифровании, как правило, объем данных на выходе превышает объем данных на входе. Поэтому целесообразным видится шифрование только той части файла, которая носит конфиденциальный характер [2].

Компания Microsoft создала пространство имен System Security Cryptography XML в рамках .Net Framework. Пространство реализует спецификацию W3C, но не обеспечивает высокоуровневую поддержку работы с XML-документами. Т. е. при необходимости программист сам должен проектировать и реализовывать обертку необходимой части файла, .Net предоставляет только общие классы.

Основная часть. Для осуществления шифрования и дешифрования средствами System Security Cryptography XML платформы .Net требуется наличие сертификата, соответствующего стандарту X.509.

X.509 – стандарт, определяющий форматы данных и процедуры распределения открытых ключей с помощью сертификатов с цифровыми

подписями, которые предоставляются сертификационными органами (CA). RFC 1422 создает основу для PKI на базе X.509.

В RFC 5280 определен сертификат X.509 версии 3 и список отзыва сертификатов (CRL) версии 2.

Для технологии открытых ключей необходимо, чтобы пользователь открытого ключа был уверен, что этот ключ принадлежит именно тому удаленному субъекту (пользователю или системе), который будет использовать средства шифрования или цифровой подписи [3]. Такую уверенность дают сертификаты открытых ключей, т. е. структуры данных, которые связывают величины открытых ключей с субъектами. Эта связь достигается цифровой подписью доверенного CA под каждым сертификатом. Сертификат имеет ограниченный срок действия, указанный в его подписанном содержании. Поскольку пользователь сертификата может самостоятельно проверить его подпись и срок действия, сертификаты могут распространяться через незащищенные каналы связи и серверные системы, а также храниться в кэш-памяти незащищенных пользовательских систем. Содержание сертификата должно быть одинаковым в пределах всего PKI. В настоящее время в этой области предлагается общий стандарт для Интернета с использованием формата X.509 v3: номер версии; серийный номер; эмитент; субъект; открытый ключ субъекта (алгоритм, ключ); период действия; дополнительные (необязательные) значения; алгоритм подписи сертификата; значение подписи сертификата. X.509-сертификаты хранятся, как правило, в виде DER- (стандартное расширение .cer) или PEM-файлов.

Создание тестового сертификата осуществляется через утилиту makecert.exe – инструмент для создания сертификатов, генерирует сертификаты X.509, предназначенные исключительно для тестирования. Этот инструмент создает пару из открытого и закрытого ключей для цифровой подписи и помещает ее в файл сертификата. Он также привязывает пару ключей к указанному имени издателя и создает сертификат X.509, который связывает заданное пользователем имя с открытым ключом пары.

Для работы с сертификатами стандарта X.509 в .Net используется пространство имен System Security Cryptography X509Certificates. Процесс получения экземпляра сертификата состоит из открытия для чтения локального хранилища, получения набора сертификатов и взятия первого сертификата, имеющего заданное имя.

Процесс шифрования XML-файла состоит из нескольких этапов. Первым делом проверяется корректность ввода полного пути к файлу (т. е. существование файла) и выдача предупреждения в случае ошибки. Далее вызывается функция для получения сертификата. Также осуществляется проверка на существование сертификата с указанным именем и в случае ошибки выдается предупреждение. Следующим шагом является загрузка XML-данных из файла в специальный класс XmlDocument [4]. Затем создается экземпляр класса EncryptedXml. Через него осуществляются все операции по шифрованию и дешифрованию XML-данных. Потом получается коллекция всех тегов с заданным именем. Далее выполняется цикл, который шифрует каждый найденный тег. Первым шагом цикла получается тег и приводится к типу XmlElement, вторым шагом тег шифруется с использованием сертификата, и третьим шагом зашифрованный тэг заменяет собой исходный. После шифрования XML-данных файл сохраняется.

Дешифрование происходит практически в автономном режиме, так как при шифровании в специальные теги была записана служебная информация. Создается объект класса EncryptedXml на основании переданного объекта класса XmlDocument и для него вызывается стандартная функция дешифрования, которая находит нужный сертификат и дешифрует зашифрованные теги.

Суммируя результаты изучения средств присутствующих в пространстве имен System Security Cryptography XML, можно отметить, что данные средства не совсем удобны для конечного пользователя (программиста). Процесс шифрования запутан, децентрализован и не стандартизирован. Плюс ко всему реализация шифрования и дешифрования является закрытой, что не дает 100%-й гарантии надежности использования функций. Конечно, необходимо четко выявить ряд достоинств и недостатков средств для шифрования XML-документов, предоставляемых .Net Framework:

– достоинства: полное следование стандарту; легкий для понимания процесс дешифрования;

– недостатки: нет высокоуровневой поддержки процессов шифрования и дешифрования; нет возможности выбора алгоритма шифрования; изначальная избыточная информация, вносимая стандартом; коды, реализующие шифрование и дешифрование, являются закрытыми, что не может дать полной уверенности в конфиденциальности шифруемых данных.

Для устранения недостатков предлагается разработать программное средство, лишенное их. Для этого необходимо:

– проанализировать стандарт W3C для нахождения и устранения избыточности;

– обеспечить высокоуровневую поддержку шифрования документа или его частей путем

проектирования и разработки специализированного класса;

– обеспечить поддержку различных алгоритмов шифрования, таких как DES, AES, RSA;

– разработать программное средство, демонстрирующее возможности реализованного класса;

– распространить реализованное программное средство с открытым исходным кодом.

Выводы. Центром планируемого программного продукта должна стать иерархия классов, которая через программный интерфейс или набор интерфейсов обрабатывает запросы клиентского (вызывающего) приложения.

В целях демонстрации возможности и удобности использования вышеописанной иерархии классов для шифрования XML-документов необходимо создать разнообразные типы клиентских (по отношению к иерархии классов) приложений. Типичными клиентскими приложениями для платформы .Net Framework являются консольное приложение, оконное приложение, веб-приложение и веб-сервис. Примерная схема работы с проектируемой иерархией классов:

– создается экземпляр интерфейса для работы с классами шифрования XML;

– интерфейсу передаются данные для шифрования, желаемый алгоритм и необходимые ключи; вызывается метод, который шифрует указанный тип тегов (для шифрования нескольких различных типов тегов можно сделать несколько вызовов функции шифрования или передать в метод список типов тегов);

– данные готовы для пересылки, хранения.

Для дешифрования необходимо будет повторно создать экземпляр интерфейса с теми же параметрами, которые использовались при шифровании, только вместо открытого XML-текста в экземпляр заносится закрытый текст. Далее дешифрование происходит в автоматическом режиме.

При исследовании актуальности и необходимости разработки описанного выше программного средства был проведен патентный поиск. Он осуществлялся с помощью базы данных, предоставляемой сайтом федерального института промышленной собственности (ФИПС – http://www.fips.ru).

Поиск по ключевым словам «шифрование xml» выдал три результата.

1. Повышение уровня автоматизации при инициализации компьютерной системы для доступа к сети.

2. Устройство и способ поддержки обмена содержимым между доменами с отличающимися DRM.

3. Система и способ, обеспечивающие распределенную архитектуру сварки.

Результат изучения документации по найденным патентам показал, что ни один из них не является специализированным средством для шифрования XML-данных. В связи с этим логично, что в настоящее время актуальность и необходимость в таком программном продукте высока.

Литература

1. Неволин А. О. Защита XML-данных. М.: LAP LAMBERT Academic Publishing, 2011. 124 с.

2. Петцольд Ч. Код. М.: Русская Редакция, 2006. 512 с.

3. Аршинов М. Н., Садовский А. Е. Коды и математика (Рассказы о кодировании). М.: Наука, 1999. 144 с.

4. C# для профессионалов / С. Робинсон [и др.]. М.: Лори, 2008. 1002 с.

Поступила 19.03.2014