УДК 681.3

Т.В. МОНАХОВА

ЗАЩИТА XML-СТРУКТУРИРОВАННЫХ ДАННЫХ

Монахова Т.В. **Защита XML-структурированных данных.**
Аннотация. В данной статье рассматривается структурирование данных при помощи языка XML. Кроме того, будут рассмотрены вопросы безопасности таких данных, поскольку представление данных в виде xml-документа связано с определёнными ограничениями и особенностями. В связи с этим мы затрагиваем некоторые вопросы шифрования, в частности, алгоритма CBC, применительно к технологии XML Encryption. Также упомянуты проблемы безопасности информации при использовании этого стандарта шифрования.
Ключевые слова: XML, XML Encryption, шифрование, безопасность данных.

Monakhova T.V. **XML-structured data protection.**
Abstract. The data structuring with the help of XML language are considered in this paper. The questions of such data protection will be considered besides, since the data representation in the form of xml document connected with certain limits and features. In this connection we will affect some of cryptography questions, algorithm CBC in particular, with reference to the XML Encryption technology. Information protection problems with this cryptography standard use will also considered.
Keywords: XML, XML Encryption, cryptography, data protection.

1. Введение. Довольно часто для структурирования данных применяется язык XML. В предложенной статье мы рассмотрим этот язык более внимательно, а также обсудим специфические аспекты безопасности данных, упорядоченных при помощи XML.

2. Структуризация данных с помощью языков XML, RDF и OWL. Как известно, на практике часто приходится иметь дело с совокупностями данных разных типов, описывающих один и тот же объект. Такие данные требуется структурировать и упорядочить для облегчения работы с ними и дальнейшего использования. Одним из распространённых методов данного преобразования является применение XML(Extensible Markup Language), представляющий собой язык разметки для так называемых древовидных, т.е. имеющих только один корневой элемент, структур. Основное правило этого языка заключается в том, что каждый элемент данных может быть представлен или тегом, или его атрибутом, в результате чего как раз и появляется возможность упорядочивать различные данные о каждом из объектов. Названия тегов записываются в угловых скобках, т.е. символах < и >, атрибуты же записываются после названия соответствующего тега, а их значения – в двойных кавычках. При этом получаем такого рода запись: <tag attribute=”value”> </tag>. Таким образом можно описать

данные довольно большой степени сложности, что объясняет популярность этого языка.

Кроме того, набор полученных данных можно также представить в виде онтологии на языке owl. Онтология определяет множество сущностей, описывающих и представляющих предметную область и логические выражения соотношений терминов друг с другом. Язык OWL применяется для описания классификаторов в базах данных и приложениях для совместного использования информации предметной области, позволяя описать объекты с различной сложностью структуры.

Как и для других способов представления данных, для облегчения описания онтологий существуют специальные средства для их построения и поддержки, к примеру, широко распространённая система Protege.

3. Защита структурированных данных(XML Encryption и XML Signature). Стандартизации языка XML способствовало то, что консорциум W3C выпустил рекомендации XML Signature и XML Encryption. Спецификация XML Signature определяет синтаксис и порядок применения электронной цифровой подписи и кодов аутентификации для данных в формате XML. Спецификация же XML Encryption описывает формат метаданных, используемых при шифровании. Далее мы рассмотрим использование цифровой подписи и защиту данных при помощи XML Encryption более подробно.

Впервые идея цифровой подписи как законного средства подтверждения подлинности и авторства электронного документа была выдвинута в работе У. Диффи и М. Хеллмана «Новые направления в криптографии» в 1976 году.

Как известно, применение электронной цифровой подписи имеет две основные цели: гарантирование подлинности информации и гарантирование того, что автором документа действительно является данное конкретное лицо. Для этого автор документа должен, используя своё секретное индивидуальное число (ключ, пароль и т.д.), определённым образом выполнять процесс «цифрового подписывания» документа. При этом подписывании каждый раз индивидуальный ключ соответствующим образом перемешивается с содержимым электронного документа. Полученное в результате такой процедуры число (последовательность определённой длины цифровых разрядов) и является цифровой подписью автора под данным конкретным документом. Из этого становится ясно, что процедуры подписывания и проверки цифровой подписи, в которых используется по одному из пары ключей, должны быть известны, но при этом должна обеспечиваться га-

рантированная невозможность восстановления ключа подписывания по ключу проверки.

4. Применение XML Encryption. Как уже было сказано, эта спецификация описывает формат метаданных, используемых при шифровании, т.е. идентификаторы ключей, алгоритмов, схем шифрования и т.д. Наиболее важным элементом является <CipherValue>, содержащий шифрованный текст (см. листинг 1).

```
<?xml version='1.0' encoding='utf-8'?>
<EncryptedData
Type='http://www.w3.org/2001/04/xmlEnc#Element'
xmlns='http://www.w3.org/2001/04/xmlEnc'>
 <EncryptionMethod
Algoritm='http://www.w3.org/2001/04/xmlenc#aes128-cbc'>
</EncryptionMethod>
<KeyInfo xmlns='http://www.w3.org/2000/09/xmldsig#'>
 <KeyName>John Smith</KeyName>
</KeyInfo>
<CipherData>
 <CipherValue>A123456…</CipherValue>
</CipherData>
</EncryptedData>
```

Листинг 1. Пример использования Xml Encryption.

Обработка полученного шифрованного сообщения осуществляется следующим образом. В первую очередь во всём документе производится поиск элементов <EncryptedData>. Каждый из таких элементов содержит метаданные с информацией о ключах, которая разбирается, обрабатывается и используется для построения ключа дешифрования данных. Затем содержимое <CipherValue> извлекается и обрабатывается для получения шифрованного текста. Для дешифровки полученного текста используется алгоритм, информация о котором содержится в элементе <EncryptionMethod>. После дальнейшей обработки открытый текст вставляется в xml-документ.

5. Алгоритм шифрования CBC. В соответствии со стандартом Xml Encryption при шифровании информации в документах xml применяется определённый набор алгоритмов шифрования. Довольно часто это так называемые блочные шифры AES и 3DES. Блочными шифрами называют последовательность (с возможным повторением и чередованием) основных методов преобразования, применяемую к блоку

(части) шифруемого текста. На практике блочные шифры встречаются чаще, чем простые преобразования того или иного класса в силу их более высокой криптостойкости. Именно на этом классе шифров основаны российский и американский стандарты шифрования.

Таким образом, блочный шифр преобразует блок открытого текста в блок (обычно длиной 16 байт, то есть 128 бит) в блок шифрованного текста. Если данные занимают больше одного блока, приходится использовать алгоритмы, самым популярным из которых в настоящее время является CBC.

Принцип его работы показан на рисунке 1. Для первого блока открытого текста случайным образом выбирается вектор инициализации, после чего над этим вектором и первым блоком открытого текста выполняется операция XOR, т.е. исключающее «или», результат которой шифруется с помощью блочного алгоритма. В качестве вектора инициализации для последующих блоков открытого текста используется предыдущий блок шифрованного текста.

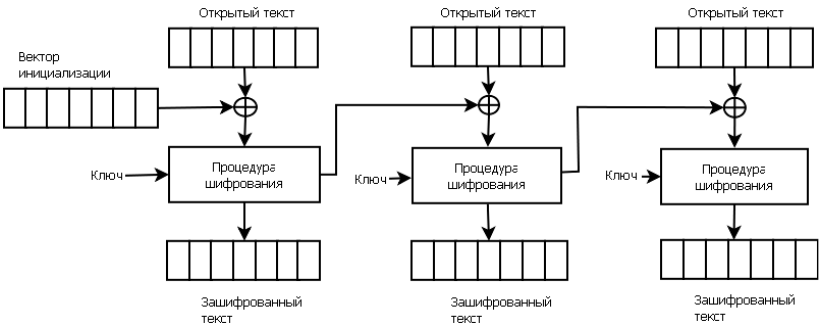

Рис. 1. Шифрование с использованием CBC.

Дешифрование же текста, как несложно догадаться, происходит в обратном порядке.

Кроме предписания использования CBC, рассматриваемый стандарт определяет схему дополнения данных до полного блока, т.е. неполный блок данных с помощью произвольных значений дополняется до полного, а в его последний байт вписывается количество добавленных произвольных значений. Таким образом, если последний блок содержит всего три байта, то добавляется 12 произвольных байт и один байт со значением 0x05. Если же последний блок полон, т.е. содержит 16 байт, к нему добавляется ещё один блок, 15 байт которого задаются произвольно, а последний, 16-й, имеет значение 0x10.

6. Возможные уязвимости в указанных стандартах. К сожалению, названные нами ранее стандарты XML Signature и XML Encryption не могут считаться панацеей от несанкционированного доступа к защищаемым с их использованием данных и, как следствие – от злонамеренных действий, производимых над этими данными.

К примеру, использование XML Signature ненадёжно вследствие давно существующих схем хакерских атак. Одной из таких атак является XML Signature Wrapping, позволяющая модифицировать шифрованный текст в обход цифровой подписи и/или MAC. Соответственно, этот стандарт не может считаться надёжной защитой данных.

Перейдём к рассмотрению XML Encryption. Как известно, эта технология довольно часто используется в веб-сервисах на основе Apache Axis2, что и позволяет провести рассмотренную в [3] атаку с применением оракулов.

7. Атака на XML Encryption с использованием оракулов. Рассмотрим указанную атаку более подробно. В первую очередь имеющийся шифрованный текст изменяется с применением произвольной битовой маски и вектора инициализации. После чего полученное сообщение отсылается на сервер. Ответ сервера анализируется. Если получен ответ «true», то переданный блок шифрованного текста соответствует реальному шифрованному тексту. В случае ответа «false» текст снова изменяется и отсылается на сервер. Таким образом получается весь шифрованный текст и вектор инициализации последнего блока, на основе которых получается исходный открытый текст. Оракулом же называют объект, отвечающий на запросы злоумышленника. В данном контексте в роли оракула выступает сервер с веб-сервисом, в ответ на переданный ему запрос на обработку данных возвращающий либо сообщение об ошибке, либо результат обработки данных.

8. Заключение. Как было рассмотрено в статье, существуют два стандарта для защиты данных, представленных в формате XML. Это XML Signature, регламентирующий применение электронной цифровой подписи, и XML Encryption, описывающий синтаксис криптографических алгоритмов. К сожалению, ни один из этих методов защиты нельзя назвать стопроцентным вследствие подверженности разного рода атакам со стороны злоумышленника. Весьма действенной мерой по исправлению такой ситуации могло бы стать изменение стандарта XML Encryption с целью смены режима шифрования, но этим должен заняться консорциум W3C. Со стороны же пользователя в данном случае предпочтительна оценка рисков утраты данных, подлежащих за-

щите, и при необходимости дополнение рассмотренных методов другими, например, возможно применение обфускации.

Литература

1. *Монахова Т.В.* Онтологическая модель описания экспериментальных данных//Труды СПИИРАН. СПб.: СПИИРАН, 2012.
2. *Алгазинов Э.К., Сирота А.А.* Анализ и компьютерное моделирование информационных процессов и систем. М.: Диалог-МИФИ, 2009. 416 с.
3. *Plaintext* Взлом XML Encryption. Лёгкий способ дешифрования закрытой XML-информации.//Хакер. М.: Гейм Лэнд, 01.2012.
4. *Партыка Т.Л., Попов И.И.* Информационная безопасность. М.: ФОРУМ, 2012. 432 с.

Монахова Татьяна Вячеславовна - аспирант лаборатории вычислительных систем и проблем защиты информации СПИИРАН, младший научный сотрудник Четвёртого центрального научно-исследовательского института Министерства Обороны Российской Федерации(4 ЦНИИ МО РФ). Область научных интересов: моделирование информационных систем. Число научных публикаций - 3. panthernator@yandex.ru; 4ЦНИИ МО РФ, ул. М.К. Тихонравова, 39, Юбилейный, 141091, РФ. Научный руководитель – В.И. Воробьёв.

Monakhova Tatyana Vyacheslavovna – post-graduated student, Laboratory of Computing Systems and Information Protection Problems, SPIIRAS. Junior researcher, 4 Central Science Institute of the Ministry of Defense, Russia. Research interests: information systems modeling. The number of publications — 3. panthernator@yandex.ru; 4 CSI MD, Tikhonravova st., 39, Yubilejnyj, 141091, Russia. The supervisor of studies is V.I. Vorobev

Рекомендовано СПИИРАН, лабораторией информационно-вычислительных систем, заведующий лабораторией Воробьев В.И., д-р техн. наук, проф.
Статья поступила в редакцию 25.12.2012.

РЕФЕРАТ

Монахова Т.В. **Защита XML-структурированных данных.**

В данной статье рассматривается структурирование данных при помощи языка XML, а также некоторые специфические вопросы безопасности таких данных.

Данные, с которыми приходится работать на практике, часто представляют собой наборы данных различных типов, характеризующих различные или же сходные объекты, или один и тот же объект при различных условиях. Очевидно, что для упрощения обработки такие данные необходимо структурировать.

Для этого часто применяются онтологии OWL или язык XML. Рассмотрим более подробно представление данных при помощи XML.

Этот язык представления данных позволяет описать древовидную структуру любой степени сложности. Данные при этом описываются при помощи тегов и атрибутов. Названия тегов записываются в угловых скобках, а значения атрибутов – в двойных кавычках после знака равенства.

Консорциум W3C также разработал два стандарта – XML Signature и XML Encryption. Первый из этих стандартов определяет синтаксис и порядок применения электронной цифровой подписи и кодов аутентификации для данных в формате XML. Стандарт же XML Encryption описывает синтаксис криптографических алгоритмов, разработанных для произвольных XML-структурированных данных.

Стандарт XML Encryption основывается на использовании блочных шифров в режиме CBC. Блочный шифр преобразует блок открытого текста, обычно длиной в 16 байт, в блок шифрованного текста. Если данные занимают больше одного блока, используются алгоритмы, самым популярным из которых является CBC. Принцип работы этого алгоритма заключается в следующем: для первого блока открытого текста случайно выбирается вектор инициализации, затем над этим вектором и первым блоком выполняется операция XOR. В качестве вектора инициализации для последующих блоков открытого текста используется предыдущий блок шифрованного текста.

К сожалению, оба описанных стандарта не гарантируют защиты данных от действий злоумышленников, поскольку разработаны широко известные атаки XML Signature Wrapping, позволяющая модифицировать шифрованный текст в обход подписи, и атаки с применением так называемых оракулов, в случае с XML Encryption – сервера с веб-сервисом, в ответ на запрос на обработку данных возвращающего либо сообщение об ошибке, либо результат обработки данных.

SUMMARY

Monakhova T.V. **XML-structured data protection.**

A structuring of data with XML language and some specific security questions of this data will considered in this paper.

The data with working in practice are often representing sets of different types data which describing similarity objects or the same object at the different conditions. It is obvious that such data must be structured for its processing simplification.

OWL ontologies or XML language are often use for it. We will consider in more detail the data representation with XML.

This data representation language make it possible to describe tree-like structure of any complexity degree. Data with XML are described with tags and attributes. The tags name are wrote in broken brackets and the attributes value are wrote in double quotes after the equal sign.

W3C consortium also worked out two standards, named XML Signature and XML Encryption. The first standard are set electronic digital signature application syntax and order and authentification codes for XML formatted data. The XML Encryption standard are describes a syntax of cryptographic algorithms which are developed for the arbitrary XML-structured data.

The XML Encryption standard based on using of block codes in CBC mode. The block code is transform of a plain text block, at 16 bytes usually, in a coded text block. If the data has more then one block special algorithms are used and the most popular of them is CBC. This algorithm work principle is consists in following: for a first plain text block are randomly choose an initialization vector and with this vector and the first block the XOR operation is carried out then. By the initialization vector way for followings plain text blocks a previous coded text block is used.

Unfortunately, both of described standard are not guarantee a data protection from malefactor's actions because widely known attacks – the XML Signature Wrapping that allows to modify a coded text passing signature and the oracle use attacks, in this case, a web-service server that sends either an error message or a data processing result in response to data processing request – are worked out.