¨Применение криптографии в вопросах защиты данных, на примере рюкзачной системы шифрования.¨

Михаил Чегодарь.

http://mexmat.aaanet.ru/

Вот все говорят “безопасность”, а как на практике всё это применить? Именно практическому аспекту криптографии посвящена эта работа.

Сейчас вопросы защиты информации стоят наиболее остро. Это связано как с общим бурным развитием информационных технологий, так и с появлением новых языков и технологий программирования самостоятельно не способных решить все проблемы безопасности, кроме как административным методом.

1. К чему это?

Например, в XML-документе хранятся в открытом виде логины и пароли пользователей удаленной базы данных, что позволяет злоумышленнику, при условии чтения, им данного файла, осуществлять практически неограниченные изменения в базе данных.

В качестве основного средства ограничивающего доступ к конфиденциальной информации можно использовать метод хранения информации в закодированном виде.

Проблема решается следующим образом. Указанный XML документ кодируется и в закодированном виде хранится на сервере. При обращении к этому файлу на сервере запускается декодер, который декодирует и выполняет XML-документ на сервере, без создания открытой копии документа на жестком диске.

Остановимся подробнее на механизме доступа к данным.

Могу привести свой пример, для которого программа и была написана.

Интернет пользователь формирует запрос, через WWW-интерфейс к WEB-сайту. После запуска файла INDEX.php, данный скрипт формирует обычную HTML-страницу со статическими ссылками, заголовками и т.д. Между тегами “<?php” и “?>” расположен текст скрипта, который добавляет динамичность в страницу, т.к. “налету” обрабатывает файл students.xml, и если там появились новые узлы, тут же показывает их в таблице.

Заметим, что в файле students.xml хранятся в открытом виде логины и пароли пользователей базы данных.

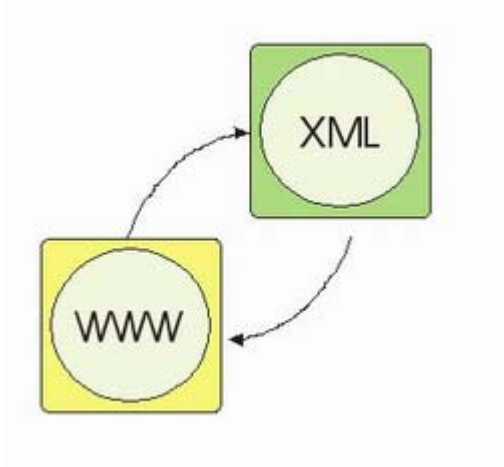

Рис 1.

После применения разработанной программы процесс доступа к данным несколько меняется (Рис 2.). После формирование запроса скриптом index.php на выполнение файла students.xml, последний файл автоматически декодируется на сервере лишь в том случае, если пользователь правильно ввёл свой секретный ключ, после чего выполняется. Конечно, даже если пользователь ввёл не верный секретный ключ, декодирование всё равно осуществляется, однако вместо XML-файла с логинами и паролями получается абракадабра и students.xml соответственно не обрабатывается. Пользователю становится доступна только информация сформированная после обработки файла students.xml.

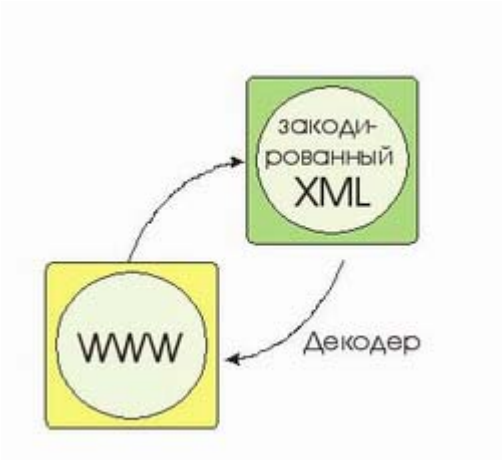

Рис 2.

2. Примененный алгоритм на основе "проблемы рюкзака".

"Проблема рюкзака" (или "ранца") может быть сформулирована следующим образом. Пусть задано множество натуральных чисел A = (a_1, a_2,..., a_n) и натуральное число S. Требуется установить, имеется ли такое подмножество множества A, сумма элементов которого была бы равна S. Эквивалентной является следующая формулировка: существует ли такой набор чисел x_i из

(0,1) , i<=n , для которого $\sum a_i * x_i = S$ (1<=i<=n).

Данная проблема получила свое название в связи с тем, что поставленная задача может быть переформулирована также в следующем виде. Имеется набор предметов с известными весами и рюкзак, который может выдержать вес, не превышающий заданной величины. Можно ли выбрать набор предметов для погрузки в рюкзак так, чтобы они в точности имели максимально возможный вес.

"Проблема рюкзака" является весьма сложной, ее решение с полиномиальной сложностью в настоящее время не известно.

Идея построения системы шифрования на основе проблемы рюкзака заключается в выделении некоторого подкласса задач об укладке рюкзака, решаемых сравнительно легко, и "маскировки" задач этого класса (с помощью некоторого преобразования параметров) под общий случай.

Параметры подкласса определяют секретный ключ, а параметры модифицированной задачи – открытый ключ. В качестве легко решаемой задачи Р. Меркль и М. Хеллман в 1978 г. предложили задачу об укладке "супервозрастающего" рюкзака. Изложим ее суть.

Назовем супервозрастающей последовательность натуральных чисел $(b_1,b_2,..., b_n)$, обладающую свойством

$b_i>\sum b_j$, $1<=j<=i-1$, $2<i<n$.

Можно убедиться в том, что проблема рюкзака для супервозрастающей последовательности может быть решен помощью процедуры, состоящей в выполнении следующих шагов:

1. Положить i = n;
2. Если i>1, то положить x_i равным 1 и S равным $S – b_i$, если $S>b_i$, и положить x_i равным 0 в противном случае;
3. Положить i равным i–1 и возвратиться к шагу 2.

В системе, основанной на проблеме рюкзака, величина S является параметром системы.

Для вычисления открытого и соответствующего секретного ключа каждый из абонентов системы осуществляет следующую последовательность действий.

1. Выбирает супервозрастающую последовательность $(b_1, b_2,..., b_n)$, и модуль m такой, что $m>\sum b_i$, $1<=i<=n$.
2. Выбирает случайное число W, $1<W<m-1$, такое что НОД(W,m) =1
3. Выбирает случайную перестановку π чисел (1, 2,...,n).
4. Вычисляет $a_i =(W* b_{\pi(i)}) \bmod (n)$ для i=1…n.

Открытым ключом является набор $(a_1 ,a_2,...,a_n)$, секретным ключом – набор $(\pi, m, W, (b_1, b_2,…,b_n))$.

Чтобы зашифровать сообщение M, предназначенное для абонента A, абонент B осуществляет следующие шаги с помощью открытого ключа $(a_1, a_2,...,a_n)$ абонента A:

1. Представляет M в виде бинарной последовательности $M = M_1M_2 ...M_n$ длины n;

2. Вычисляет $C = \sum M_i * a_i$, i=1…n и направляет его к A .

Абонент A, получив C, вычисляет $H = (W^{-1} * C) \bmod (m)$, а затем, решая проблему рюкзака для супервозрастающей последовательности, находит числа z_i , i = (0,1) такие, что $H=\sum z_i * b_i$, (i=1,…,n).

Биты последовательности M_i вычисляются по формуле: $M_i=z_{\pi(i)}$, i=1,…,n

Корректность проведенной процедуры расшифрования вытекает из следующих рассуждений. Поскольку $H=W^{-1}*C=W^{-1}*\sum M_i*a_i = (\sum M_i*b_{\pi(i)}) \pmod m$

и $0<H<m$, то $H=\sum M_i*b_{\pi(i)}$, (i=1,…,n) , и, следовательно, алгоритм решения проблемы рюкзака действительно находит биты открытого текста, переставленные в соответствии с перестановкой π.

3. Реализация алгоритма.

В качестве примера, приведём программу которая шифрует/дешифрует текстовые файлы заменяя один символ другим, вычисляя его по алгоритму на основе "проблемы рюкзака". В данном случае, алгоритм применяется к байтам, т.е. символам текста.

Программа включает в себя универсальный кодер-декодер способный работать с любыми текстовыми файлами, и имеющий гибкую возможность манипулирования входными параметрами, что делает практически невозможным его использование посторонними лицами с целью дешифрования информации.

В процессе создания программы была использован простой и удобный интерфейс, позволяющий выполнить следующие действия:

1. Кодирование входного файла

-задание параметров его шифрования; т. е. секретного ключа (π, m, W, (b_1, b_2,…,b_n))**.** Естественно вы можете не все указанные параметры вводить каждый раз при кодировании, можно задать их статично в теле самой программы, однако не стоит забывать, что чем больше параметров вашего секретного ключа надо знать для кодирования/декодирования тем сложнее злоумышленнику дешифровать ваши данные.

2. Расшифрование файла

-задание параметров расшифрования; т. е. тех же самых параметров указанных выше - (π, m, W, (b_1, b_2,…,b_n)).

Конечно, можно ещё немного схитрить. Например, при расшифравании запрашивать параметры, которые не требовалось вводить при кодировании (при кодировании они заданы заранее), но это уже дело вашей фантазии.

4. Немного о криптоанализе рюкзачной системы шифрования.

Заметим, что предложенная реализация алгоритма шифрует текстовые файлы заменяя один символ другим. Очевидно что в данной ситуации возможно успешное примение атаки на основе частотного анализа. Например по следующему простейшему алгоритму:

1. Подсчёт частоты встречаемости шифрообозначений, а также их некоторых сочетаний (например по 2, 3, 4 - символа).

2. Выявление шифрообозначений, заменяющих один символ соответствующим ему другим.

3. Выдвижение гипотез о значениях шифрообозначений и их проверка. Здесь можно использовать практически любую достоверную и не очень информацию об открытом тексте, например если заранее известен формат (XML) или набор используемых символов.

Следует уточнить, что реализация алгоритма шифрования поддаётся совершенствованию, и только для наглядности она представлена в данном виде. Приведём несколько советов, как можно это сделать:

1. Можно кодировать один символ нескольмими - WORDом например;

2. Ничто не мешает кодировать не по одному символу, а сразу несколько символов.

3. Приведенные выше методы, к сожалению не избавляют от раскрытия кода на основе статистического анализа. Для предотвращения этого можно добавить любую, хотя бы самую простую, зависимость одного шифруемого символа от другого, например соседнего (или нескольких символов).

Заметим, что вне зависимости от реализации рюкзачный алгоритм шифрования не идеален. Доказано, что существует алгоритм полиномиальной сложности, который может быть использован противником для получения открытого текста M по шифротексту C. Пусть противнику известна последовательность $\{a_i\}$, этот алгоритм находит пару таких целых чисел u_1, m_1, что отношение u_1/ m_1 близко к отношению u/m (где $u= W^{-1} \bmod (m)$, а W, m являются частью секретного ключа). Кроме того, числа $B_i = (u_i * a_i) \bmod (m)$, $1<i<n$, образуют супервозрастающую последовательность. Эта последовательность затем используется противником вместо $(b_1, b_2,\ldots,b_n)$ для дешифрования сообщения.

Понятие идеальности алгоритма в криптографии вообще очень сложное. В определённых случаях нам нужен DES, RSA или Blow-fish, а где-то подойдёт рюкзачная система шифрования. Не будем забывать о том, что, во-первых, злоумышленник не должен вообще знать о том какая криптосистема применяется для защиты данных, и, во-вторых, он не должен иметь возможности получать в каком либо виде криптотекст и уж тем более доступ к самой программе-кодеру. Но даже в этом случае нельзя быть на 100%

уверенным в защите ваших данных. Однако, это лучше чем не предпринимать вообще ничего:)

При грамотном администрировании, таком как ограничение доступа к данным (принцип наименьших привилегий), возможность выполнения программы на сервере только тому, кому это разрешено, можно существенно снизить все основные угрозы безопасности ваших данных.

Заключение.

В данной работе было представлено приложение, с помощью которого пользователь имеет возможность легко и быстро кодировать и декодировать текстовые документы, а также задавать параметры системы шифрования. На практике эта идея была реализована при помощи рюкзачной системы шифрования, и применена для реальной защиты базы данных от несанкционированного доступа.

Приложение может быть использовано как администраторами баз данных, так и системными администраторами, в качестве дополнения к стандартным средствам безопасности.

Для создания программы был использован пакет Borland C++, Version 5.01 фирмы Borland. После этапа отладки и компиляции программа была перенесена в среду Unix, что позволило применить её на Sun-сервере, для реально существующей базы данных.

Статья написана специально для UInC (http://www.uinc.ru)

Список литературы.

1. Алферов А.П. , Зубов А.Ю. , Кузьмин А.С. , Черемушкин А.В. , Основы криптографии: Учебное пособие. – М. Гелиос АРВ, 2001.

2. Саломаа А. Криптография с открытым ключом: Пер. с англ. - М.: Мир, 1995. - 318 с., ил. ISBN 5-03-001991-X

3. Введение в криптографию / Под редакцией В.В.Ященко // www.cryptography.ru

4. Атака на Internet // www.bugtraq.ru

5. Merkle R.C., Hellman M.E., On the security of multiple encryption // communications of the ACM. – 1981. – Vol. 24.