ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АСИММЕТРИЧНОГО АЛГОРИТМА ШИФРОВАНИЯ ДАННЫХ RSA

ВВЕДЕНИЕ

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании.

Для современных криптографических систем защиты информации сформулированы следующие общепринятые требования:

- зашифрованное сообщение должно поддаваться чтению только при наличии ключа;
- число операций, необходимых для определения использованного ключа шифрования по фрагменту шифрованного сообщения и соответствующего ему открытого текста, должно быть не меньше общего числа возможных ключей;
- число операций, необходимых для расшифровывания информации путем перебора всевозможных ключей должно иметь строгую нижнюю оценку и выходить за пределы возможностей современных компьютеров (с учетом возможности использования сетевых вычислений);
- знание алгоритма шифрования не должно влиять на надежность защиты;
- незначительное изменение ключа должно приводить к существенному изменению вида зашифрованного сообщения даже при использовании одного и того же ключа;
- структурные элементы алгоритма шифрования должны быть неизменными;
- дополнительные биты, вводимые в сообщение в процессе шифрования, должен быть полностью и надежно скрыты в шифрованном тексте;
- длина шифрованного текста должна быть равной длине исходного текста;
- не должно быть простых и легко устанавливаемых зависимостью между ключами, последовательно используемыми в процессе шифрования;
- любой ключ из множества возможных должен обеспечивать надежную защиту информации;
- алгоритм должен допускать как программную, так и аппаратную реализацию, при этом изменение длины ключа не должно вести к качественному ухудшению алгоритма шифрования.

1. СИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ

Как бы ни были сложны и надежны криптографические системы - их слабое мест при практической реализации - проблема *распределения ключей*. Для того, чтобы был возможен обмен конфиденциальной информацией между двумя субъектами ИС, ключ должен быть сгенерирован одним из них, а затем каким-то образом опять же в конфиденциальном порядке передан другому. Т.е. в общем случае для передачи ключа опять же требуется использование какой-то криптосистемы.

Для решения этой проблемы на основе результатов, полученных классической и современной алгеброй, были предложены системы с открытым ключом. Суть их состоит в том, что каждым адресатом ИС генерируются два ключа, связанные между собой по определенному правилу. Один ключ объявляется открытым, а другой закрытым. Открытый ключ публикуется и доступен любому, кто желает послать сообщение адресату. Секретный ключ сохраняется в тайне.

Исходный текст шифруется открытым ключом адресата и передается ему. Зашифрованный текст в принципе не может быть расшифрован тем же открытым ключом. Дешифрование сообщение возможно только с использованием закрытого ключа, который известен только самому адресату.

Криптографические системы с открытым ключом используют так называемые необратимые или односторонние функции, которые обладают следующим свойством: при заданном значении x относительно просто вычислить значение f(x), однако если y=f(x), то нет простого пути для вычисления значения x.

Множество классов необратимых функций и порождает все разнообразие систем с открытым ключом. Однако не всякая необратимая функция годится для использования в реальных ИС. В самом определении необратимости присутствует неопределенность. Под *необратимостью* понимается не теоретическая необратимость, а практическая невозможность вычислить обратное значение используя современные вычислительные средства за обозримый интервал времени.

Поэтому чтобы гарантировать надежную защиту информации, к системам с открытым ключом (СОК) предъявляются два важных и очевидных требования:

1. Преобразование исходного текста должно быть необратимым и исключать его восстановление на основе открытого ключа.
2. Определение закрытого ключа на основе открытого также должно быть невозможным на современном технологическом уровне. При этом желательна точная нижняя оценка сложности (количества операций) раскрытия шифра.

Алгоритмы шифрования с открытым ключом получили широкое распространение в современных информационных системах. Так, алгоритм RSA стал мировым стандартом де-факто для открытых систем.

Вообще же все предлагаемые сегодня криптосистемы с открытым ключом опираются на один из следующих типов необратимых преобразований: Разложение больших чисел ан простые множители. Вычисление логарифма в конечном поле. Вычисление корней алгебраических уравнений.

Здесь же следует отметить, что алгоритмы криптосистемы с открытым ключом (СОК) можно использовать в различных назначениях:
1. как самостоятельные средства защиты передаваемых и хранимых данных;
2. как средства для распределения ключей.

Алгоритмы СОК более трудоемки, чем традиционные криптосистемы. Поэтому часто на практике рационально с помощью СОК распределять ключи, объем которых как информации незначителен. А потом с помощью обычных алгоритмов осуществлять обмен большими информационными потоками.

2. АЛГОРИТМ RSA

Несмотря на довольно большое число различных СОК, наиболее популярна - криптосистема RSA, разработанная в 1977 году и получившая название в честь ее создателей: Рона Ривеста , Ади Шамира и Леонарда Эйдельмана. Они воспользовались тем фактом, что нахождение больших простых чисел в вычислительном отношении осуществляется легко, но разложение на множители произведения двух таких чисел практически невыполнимо.

Доказано (теорема Рабина), что раскрытие шифра RSA эквивалентно такому разложению. Поэтому для любой длины ключа можно дать нижнюю оценку числа операций для раскрытия шифра, а с учетом производительности современных компьютеров оценить и необходимое на это время. Возможность гарантированно оценить защищенность алгоритма RSA стала одной из причин популярности этой СОК на фоне десятков других схем.

В настоящее время алгоритм RSA широко используется в банковских компьютерных сетях, особенно для работы с удаленными клиентами (обслуживание кредитных карточек). Указанный алгоритм используется во многих стандартах, среди которых SSL, S-HHTP, S-MIME, S/WAN, STT и PCT.

Основными математическими результатами, положенными в основу этого алгоритма являются: малая теорема Ферма и функция Эйлера [1-3].

Открытый текст шифруется блоками, каждый из которых содержит двоичное значение, меньшее некоторого заданного числа n. Это значит, что

длина блока должна быть меньше или равна $\log_2(n)$. На практике длина блока выбирается равной 2^k битам, где 2^k<n<2^{k+1}.

Шифрование и дешифрование для блока открытого текста *M* и блока шифрованного текста *C* можно представить в виде:

$C = M^e \bmod n$

$M = C^d \bmod n = M^{ed} \bmod n = M \bmod n$.

Отправитель и получатель должны знать значение *n*. Отправитель знает значение *e* и только получатель знает значение *d*.

Таким образом, открытым ключом является $\{e, n\}$, а личным закрытым ключом $\{d, n\}$.

При этом должны быть выполнены следующие требования:

- Должны существовать такие значения *e, d* и *n,* при которых выполняется $M^{ed} \bmod n = M \bmod n$ для всех значений *M* < n.
- Должны относительно легко вычисляться M^e и C^d для всех значений *M* < n.
- Должно быть практически невозможно определить *d* по имеющимся *e* и *n.*

Схема шифрования выглядит следующим образом:

1. Выбираются два простых числа *p* и *q* . (Например, *p=7 и q=17*).
2. Вычисляется *n* = *p* · *q*. (*n* = 119).
3. Определяется φ(*n*)=(*p* – 1)(*q* – 1). (φ(*n*)=96).
4. Выбор числа *e*, взаимно простого с φ(*n*), причем *e* < φ(*n*). (*e* = 5).
5. Вычисляется $d = e^{-1} \bmod \varphi(n)$.

   (Определяется такое *d*, что *d* · *e* = 1mod 96 и *e* < 96.
   Соответствующим значением будет *d* = 77, так как
   77 ·5 = 385 = 4 · 96+1).
6. Открытым ключом является $\{e, n\}$. ($\{e, n\}$=$\{5, 119\}$).
7. Закрытым ключом является $\{d, n\}$. ($\{d, n\}$=$\{77, 119\}$).
8. Шифрование $C = M^e \bmod n$
9. Дешифрование $M = C^d \bmod n$

Рассмотрим небольшой пример, иллюстрирующий применение алгоритма RSA.

Пример

Зашифруем сообщение "CAB".

Для простоты будем использовать маленькие простые числа (на практике применяются гораздо большие). Выберем $p = 3$ и $q = 11$. Определим $n = 3 \cdot 11 = 33$. Найдем $(p-1)(q-1) = 20$.

Следовательно, в качестве *e*, взаимно просто с φ(*n*)=20, например, $e = 7$. Выберем число *d*. В качестве такого числа может быть взято любое число, для которого удовлетворяется соотношение $(d \cdot e)(\bmod 20) = (d \cdot 7)(\bmod 20) = 1$, например *d* = 3.

Представим шифруемое сообщение как последовательность целых чисел с помощью отображения: $A \to 1, B \to 2, C \to 3$. Тогда сообщение принимает вид (3,1,2).

Зашифруем сообщение с помощью ключа {7,33}:

$$(3^7)(\mathrm{mod}\,33) = 2187(\mathrm{mod}\,33) = 9,$$

$$(1^7)(\mathrm{mod}\,33) = 1(\mathrm{mod}\,33) = 1,$$

$$(2^7)(\mathrm{mod}\,33) = 128(\mathrm{mod}\,33) = 29.$$

Расшифруем полученное зашифрованное сообщение (9,1,29) на основе закрытого ключа {3,33}:

$$(9^3)(\mathrm{mod}\,33) = 729(\mathrm{mod}\,33) = 3,$$

$$(1^3)(\mathrm{mod}\,33) = 1(\mathrm{mod}\,33) = 1,$$

$$(29^3)(\mathrm{mod}\,33) = 24389(\mathrm{mod}\,33) = 2.$$

Итак, в реальных системах алгоритм RSA реализуется следующим образом: каждый пользователь выбирает два больших простых числа, и в соответствии с описанным выше алгоритмом выбирает два простых числа e и d. Как результат умножения первых двух чисел (p, q) устанавливается n, $\{e, n\}$ образует открытый ключ, а $\{d, n\}$ - закрытый (хотя можно взять и наоборот).

Открытый ключ публикуется и доступен каждому, кто желает послать владельцу ключа сообщение, которое зашифровывается указанным алгоритмом. После шифрования, сообщение невозможно раскрыть с помощью открытого ключа. Владелец же закрытого ключа без труда может расшифровать принятое сообщение.

3. ЗАДАНИЕ

1. Создать программную реализацию алгоритма RSA.
2. Исследовать зависимость времени шифрования и дешифрования файлов от размера файла и длины ключа, результаты представить в графическом или табличном виде.
3. Сформировать и представить преподавателю отчет по результатам выполнения лабораторной работы.

ЛИТЕРАТУРА

1. http://www.ssl.stu.neva.ru/psw/crypto.html – Перевод на русский язык книги: Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley& Sons, Inc. 1994; (Шнайер Брюс. Прикладная криптография. Протоколы, алгоритмы и исходные тексты на языке С, 2-е издание.)
2. Спицын В.Г., Столярова Н.А. Защита информации и информационная безопасность. Учеб. пособие. - Томск: Изд. ТУСУР, 2002, 158 с.
3. Столлинг В. Криптография и защита сетей: принципы и практика. – М.: “Вильямс”, 2001, 672 с.