Инновационный Евразийский университет

Кафедра
«Энергетика, металлургия и информационные технологии»

СЛАЙД-ЛЕКЦИЯ

по дисциплине

«Основы информационной безопасности»

Тема: Криптографическая система RSA

Образовательные программы:
6В06101 «Информатика»
6В06102 «Информационные системы»
6В06103 «Вычислительная техника и программное обеспечение»

Разработчик:
старший преподаватель, м.и. И.И. Ляшенко

План лекции:

1. Введение

2. Алгоритм шифрования RSA

3. Расшифрование RSA

4. Особенности применения криптографической системы RSA

1. Введение

RSA – криптографическая система открытого ключа, обеспечивающая такие механизмы защиты как шифрование и аутентификация – установление подлинности. Криптосистема RSA разработана в 1977 году и названа в честь ее разработчиков: Rivest, Shamir и Adleman.

На данный момент асимметричное шифрование на основе открытого ключа RSA использует большинство продуктов на рынке информационной безопасности.

Его *криптостойкость* основывается на сложности разложения на множители больших чисел, а именно - на исключительной трудности задачи определить секретный ключ на основании открытого, так как для этого потребуется решить задачу о существовании делителей целого числа.

В настоящее время Лаборатория RSA рекомендует для обычных задач ключи размером 1024 бита, а для особо важных задач – 2048 битов.

2. Алгоритм шифрования RSA

✔**Сгенерировать *открытый* и *секретные* ключи:**
Возьмем два больших простых числа *p* и *q*.
Определим *n* как результат умножения *p* на *q*

$$n = p*q$$

Выберем случайное число, которое назовем *d*. Это число должно быть *взаимно простым* (не иметь ни одного общего делителя, кроме 1) с результатом умножения (p-1)*(q-1).

Определим такое число *e* (1< e < (p-1)*(q-1)), не имеющее общих делителей кроме 1 (*взаимно простое*) с числом (p-1)*(q-1), для которого является истинным следующее соотношение

$$(e*d) \bmod ((p-1)*(q-1))=1.$$

Назовем *открытым ключом* числа *e* и *n*, а *секретным* - *d* и *n*.

- **Разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа**

$$M(i)=0,1,2..., n-1 \text{ (т.е. только до } n-1).$$

- **Зашифровать текст, рассматриваемый как последовательность чисел M(i) по формуле**

$$C(i)=(M(i)^e) \bmod n.$$

3. Расшифрование RSA

Чтобы расшифровать данные, используя секретный ключ {*d*, *n*}, необходимо выполнить следующие вычисления:

$$M(i) = (C(i)^d) \bmod n.$$

В результате будет получено множество чисел M(i), которые представляют собой исходный текст.

Пример.

Рассмотрим небольшой пример, иллюстрирующий применение алгоритма RSA.

Пусть требуется зашифровать сообщение «CAB». Для простоты будем использовать маленькие числа (на практике применяются гораздо большие). Пошагово проследим процессы шифрования и дешифрования.

1 шаг. Выберем $p = 3$ и $q = 11$.

2 шаг. Определим $n = 3 * 11 = 33$.

3 шаг. Найдем $(p – 1)(q – 1) = 20$. Следовательно, в качестве d можно взять число, взаимно простое с 20, например, $d = 3$.

4 шаг. Выберем число e. В качестве такого числа может быть взято любое число, для которого выполняется соотношение (e*3) mod 20 = 1, например 7.

5 шаг. **Представим шифруемое сообщение как последовательность целых чисел в диапазоне от 0 до 32 (до *n-1*!). Буква A=1, B=2, C=3 с помощью отображения: A -> 1, B-> 2, C-> 3. Тогда исходное открытое сообщение принимает вид M = (3,1,2).**

6 шаг. **Зашифруем сообщение с помощью ключа {7, 33}:**

$s1 = (3^7) \pmod{33} = 2187 \pmod{33} = 9$

$s2 = (1^7) \pmod{33} = 1 \pmod{33} = 1$

$s3 = (2^7) \pmod{33} = 128 \pmod{33} = 29$

Зашифрованное сообщение после этого примет вид S = (9, 1, 29).

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

$m1 = (9^3) \pmod{33} = 729 \pmod{33} = 3,$
$m2 = (1^3) \pmod{33} = 1 \pmod{33} = 1,$
$m3 = (29^3) \pmod{33} = 24389 \pmod{33} = 2.$

Как можем видеть, дешифрование шифртекста S = (9, 1, 29) привело к исходному открытому тексту M = (3, 1, 2).

Замечание. **Делители *p* и *q* можно уничтожить или сохранить вместе с секретным ключом.**

Если бы существовали эффективные методы разложения на сомножители, то, разложив *n* на сомножители *p* и *q*, можно было бы получить секретный ключ *d*.

Таким образом, надежность криптосистемы RSA основана на *трудноразрешимой (практически неразрешимой)* задаче разложения n на сомножители, так как в настоящее время эффективного способа поиска сомножителей не существует.

4. Особенности применения криптографической системы RSA

Криптосистема RSA может использоваться для подтверждения подлинности или идентификации. Это возможно потому, что каждый зарегистрированный пользователь криптосистемы имеет свой уникальный секретный ключ, который (теоретически) больше никому недоступен. Именно это делает возможным положительную и уникальную идентификацию.

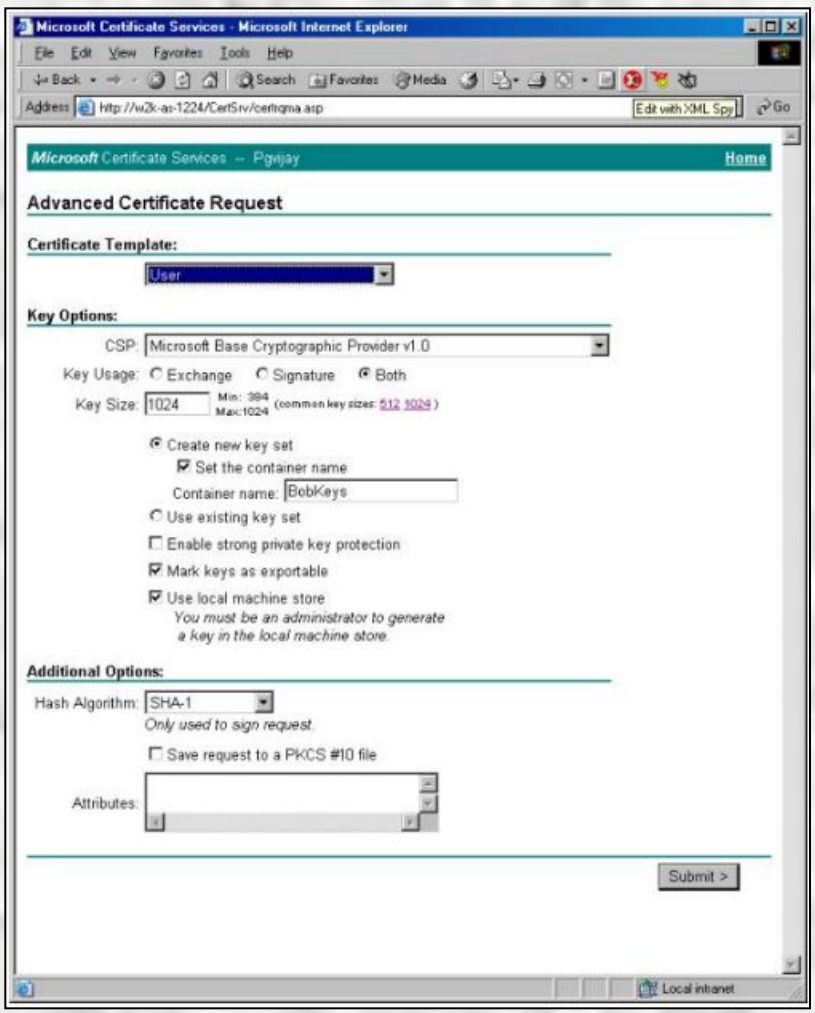
Скорость работы алгоритма RSA.

Как при шифровании и расшифровке, так и при создании и проверке подписи алгоритм RSA по существу состоит из возведения в степень, которое выполняется как ряд умножений.

В практических приложениях для открытого ключа обычно выбирается относительно небольшой показатель (*e*), а зачастую группы пользователей используют один и тот же открытый показатель (*e*), но каждый с различным модулем.

При этом шифрование данных идет быстрее чем расшифровка, а проверка подписи – быстрее чем подписание.

Алгоритм RSA намного медленнее чем DES и другие алгоритмы блокового шифрования. Программная реализация DES работает быстрее по крайней мере в 100 раз и от 1000 до 10000 – в аппаратной реализации (в зависимости от конкретного устройства).



Процесс создания RSA 1024-битовой пары ключей и сохранение ключа в ключевой набор базовых поставщиков службы криптографии Microsoft Cryptographic Provider v1.0 BobKeys

Способы взлома криптосистемы RSA.

Существует несколько способов взлома RSA. Наиболее эффективная атака: найти секретный ключ, соответствующий необходимому открытому ключу. Это позволит нападающему читать все сообщения, зашифрованные открытым ключом и подделывать подписи.

Такую атаку можно провести, найдя главные сомножители общего модуля *n* – *p* и *q*. На основании *p*, *q* и *e*, нападающий может легко вычислить частный показатель *d*. Основная сложность – поиск главных сомножителей *n*; безопасность RSA зависит от разложения на сомножители.

Фактически, задача восстановления секретного ключа эквивалентна задаче разложения на множители модуля: можно использовать *d* для поиска сомножителей *n*, и наоборот, можно использовать *n* для поиска *d*.

Усовершенствование вычислительного оборудования само по себе не уменьшит стойкость криптосистемы RSA, если ключи будут иметь достаточную длину. Фактически же совершенствование оборудования увеличивает стойкость криптосистемы.

Другой способ взломать RSA состоит в том, чтобы найти метод вычисления корня степени *e* из *C (mod n)*.

Поскольку C(i)=(M(i)^e) mod *n*, то корнем степени *e* из C (*mod n*) является сообщение M. Вычислив корень, можно вскрыть зашифрованные сообщения и подделывать подписи, даже не зная частный ключ. Но в настоящее время неизвестны методы, которые позволяют взломать RSA таким образом.

Упомянутые атаки – единственные способы расшифровать все сообщения, зашифрованные данным ключом RSA.

Рекомендуемая длина ключа.

Размер ключа в алгоритме RSA связан с размером модуля *n*. Два числа *p* и *q*, произведением которых является модуль, должны иметь приблизительно одинаковую длину, поскольку в этом случае найти сомножители сложнее, чем в случае когда длина чисел значительно различается. Например, если предполагается использовать 768-битный модуль, то каждое число должно иметь длину приблизительно 384 бита.

Если два числа чрезвычайно близки друг к другу или их разность близка к некоторому предопределенному значению, то возникает потенциальная угроза безопасности, однако такая вероятность – близость двух случайно выбранных чисел – незначительна.

Как доказано Эвклидом более двух тысяч лет назад, существует бесконечное множество простых чисел. Поскольку алгоритм RSA оперирует с ключами определенной длины, то количество возможных простых чисел конечно, хотя и очень велико.

По теореме о Простых Числах количество простых чисел, меньших некоторого n приближается к $n = ln(n)$.

Следовательно, количество простых чисел для ключа длиной 512 битов или меньше приблизительно составляет 10^{150}. Это больше, чем количество атомов в известной Вселенной.

Контрольные вопросы:

1. На чем основывается криптостойкость RSA?
2. Сформулировать алгоритм шифрования RSA.
3. Каковы сравнительные характеристики RSA и DES-алгоритма?
4. Дать краткое описание способов взлома RSA.
5. Какова рекомендуемая длина ключа RSA?

Список используемых источников:

1. Бубнов А.А. Основы информационной безопасности. – М.: Академия, 2017. - 256 с.
2. Ерохин В.В. Безопасность информационных систем. - М. : Флинта, 2016. - 184 с.
3. Гашков С.Б. Криптографические методы защиты информации. - М.: Академия, 2010. - 300с.
4. Мельников В.П. Информационная безопасность. – М.: Академия, 2013. - 336 с.

Список используемых источников:

5. **Мельников В.П. Защита информации. – М.: Академия, 2014. - 304 с.**
6. **Бабаш А.В. Информационная безопасность. Лабораторный практикум. - М. : КНОРУС, 2013. - 136 с.**
7. **Платонов В.В. Программно-аппаратные средства защиты информации. – М.: Академия, 2014. - 336 с.**