

GnuPG: Руководство К Применению



Владимир Иванов
ivlad@unixgods.net

Зачем нужно шифрование?

- Право на частную жизнь
- Право на конфиденциальность переписки
 - Логично распространить его на электронные коммуникации
- Проблемы протокола SMTP

Зачем нужно шифрование?

- Системы СОРМ, Echelon
 - Провайдеры Internet не имеют технической возможности гарантировать конфиденциальности электронной почты
- Борьба с терроризмом приводит к ослаблению контроля за спецслужбами

Симметричные шифры

- Пусть даны функции $E(T,k)$ и $D(C,k)$, причем $D(E(T,k),k)=T$
- Назовем k ключом, T – открытым текстом, C – шифротекстом, E – функцией шифрования, D – функцией дешифрования
- Пусть Алиса и Боб договорились о конкретном значении k , тогда они могут установить конфиденциальный канал связи

Симметричные шифры

- Простой пример: шифр Юлия Цезаря
- Алгоритм: сдвиг номера буквы алфавита на величину ключа, например, для значения ключа равного 3, вместо А записывается D, вместо В – Е и т.д. (сложение по модулю 26)
- Таким образом, фраза “VENI, VEDI, VICI” записывается как “YHQL, YHGL, YLFL”

Симметричные шифры

- Невскрывааемый шифр: одноразовый блокнот
- Изобретен в 1917 году Major Joseph Mauborgne и Gilbert Vernam
- Широко использовался (используется?) в разведке
- Основной недостаток: длина ключа равна длине сообщения

Симметричные шифры

- Если Виктория желает общаться с Бобом и Алисой, причем так, что бы третье лицо не могло читать переписку любой пары, потребуется два дополнительных ключа
- Для n пользователей необходимо $n(n-1)/2$ ключей; например, для 100 пользователей необходимо 4950 ключей

Симметричные шифры

- Шифр DES
- Разработан IBM под именем Lucifer, в 1977 году после некоторых модификаций принят как стандарт
- Представляет собой блочный шифр с размером блока 64 бита; длина ключа равняется 56 битам, ключ обычно сохраняется как 64 бита, каждый восьмой бит не используется

Симметричные шифры

- 3DES, как попытка продлить жизнь DES
- 2DES и атака «встреча посередине»
- ГОСТ 28147-89, проблема S-блоков
- IDEA, использовался в оригинальном PGP но не используется в GnuPG
- AES, может использоваться для защиты TOP SECRET, блок 128 бит, ключ 128, 192 или 256 бит

Режимы шифрования

- ECB: независимое шифрование блоков
- CFB: шифруется синхропосылка;
Результат шифрования складывается по модулю 2 с первым блоком открытого текста (получается первый блок шифротекста) и снова подвергается зашифрованию. Полученный результат складывается со вторым блоком открытого текста и т.д.

Режимы шифрования

- OFB: сначала зашифрованию подвергается синхропосылка. Результат складывается по модулю 2 с первым блоком открытого текста - получается первый блок шифротекста; шифра получается путем многократного шифрования синхропосылки

Режимы шифрования

- СВС: очередной блок открытого текста складывается по модулю 2 с предыдущим блоком шифртекста, после чего подвергается зашифрованию в режиме ЕСВ;

Ассиметричные шифры

- Пусть даны функции $E(T,k)$ и $D(T,k)$
- Пусть даны k и k' , взаимосвязанные, таким образом, что $D(E(T,k),k')=T$
- Зная k , мы не можем вычислить k' и наоборот
- Назовем k открытым ключом, а k' - закрытым

Алгоритм **ElGamal**

- Основан на трудности дискретного логарифмирования в конечном поле
- Выбираем простое p , случайные $g < p$ и $x < p$
- Вычисляем $y = g^x \bmod p$
- Открытый ключ: y, g, p
- Закрытый ключ: x

Алгоритм **ElGamal**

- Подпись:
 - Подписываем сообщение M
 - Выбираем случайное k , взаимно простое с $p-1$
 - Вычисляем $a = g^k \bmod p$
 - Вычисляем b такое, что $M = (xa + kb) \bmod (p-1)$
 - Подпись: a, b

Алгоритм ElGamal

- Проверка подписи:
 - Даны a , b – подпись, M – сообщение, y , g , p – открытый ключ
 - Если $y^a * a^b \bmod p = g^M \bmod p$, то подпись верна

Алгоритм ElGamal

- Пример:
 - $p=11, g=2, x=8$
 - $y=g^x \bmod p = 2^8 \bmod 11 = 3$
 - $M=5, k=9$
 - $a=g^k \bmod p = 2^9 \bmod 11 = 6$
 - $M=(ax+kb) \bmod (p-1), 5=(8*6+9b) \bmod 10, b=3$
 - $y^a * a^b \bmod p = g^M \bmod p, 3^6 * 6^3 \bmod 11 = 2^5 \bmod 11$

Алгоритм **ElGamal**

- Шифрование
 - Шифруем сообщение M , выбираем случайное k , взаимно простое с $p-1$
 - Вычисляем $a = g^k \bmod p$
 - Вычисляем $b = (y^k * M) \bmod p$
 - Шифротекст: a, b

Алгоритм ElGamal

- Расшифрование:

- $M = (b/a^x) \bmod p$

- Пояснение:

$$\begin{aligned} M &= (b/a^x) \bmod p = (y^k * M / g^{xk}) \bmod p \\ &= (g^{kx} * M / g^{kx}) \bmod p = M \end{aligned}$$

Алгоритм **RSA**

- Основан на трудоемкости факторизации больших чисел
- Назван в честь разработчиков **R**ivest, **S**hamir и **A**dleman
- Является стандартом de-facto в коммерческих системах
- Не используется в GnuPG

Алгоритм **RSA**

- Ключи:
 - Открытый: $n=p \cdot q$, p, q - большие простые числа, e – взаимно простое с $(p-1)(q-1)$
 - Закрытый: $d=e^{-1} \bmod ((p-1)(q-1))$
- Зашифрование: $c=m^e \bmod n$
- Расшифрование: $m=c^d \bmod n$

Алгоритм **RSA**

- Пример:
 - $p=23$, $q=41$; $n=p*q=943$
 - $(p-1)(q-1)=880$; $e=7$
 - $M=35$
 - d : $d*e=1 \pmod{(p-1)(q-1)}$; $d=503$
 - $c=M^e \pmod n = 35^7 \pmod{943} = 545$
 - $m=c^d \pmod n = 545^{503} \pmod{943} = 35$

Понятие хеша

- Хешем называется «однонаправленная» функция, по значению которой нельзя восстановить ее аргументы
- Алгоритмы хеширования: MD5, SHA, ГОСТ 34.11-94

PGP и GnuPG

- Запрет на экспорт алгоритмов шифрования из США
- Создание PGP Филиппом Циммерманом и публикация исходных текстов в виде книги
- Коммерциализация PGP
- Стандарт OpenPGP
- GNU Privacy Guard

Создание ключей

- Команда **gpg --gen-key**
- Ответить на вопросы
- Выбрать «хорошую» ключевую фразу
- Выбор длины ключа и срока действия ключа

Отзывающий сертификат

- Команда **gpg --output revoke.txt --gen-revoke keyid**
- Распечатать сертификат и хранить под замком

Работа с ключами

- Просмотр: **gpg --list-keys**
- Экспорт: **gpg --output key.gpg --export keyid**
- Импорт: **gpg --import key.gpg**
- Послать на keyserver: **gpg --send-key**
- Получить: **gpg --recv-key**

Шифрование

- Зашифрование: **gpg --encrypt --recipient**
- Расшифрование: **gpg --decrypt**

Подписи

- Подпись: **gpg --sign**
- «Чистая» подпись: **gpg --clearsign**
- Отделенная подпись: **gpg --detach-sig**
- Проверка: **gpg --verify**

Интеграция

- Mutt
- KMail
- Evolution
- Mozilla/Thunderbird (Enigmail)
- Outlook/Outlook Express/The Bat
- Jabber/ICQ
- rpm

Сеть доверия

- Доверие владельцу
 - Команда: **gpg --edit-key**
 - Команда gpg: **trust**
- Доверие ключу
 - Подписан достаточным числом ключей
 - Собой, полностью доверенным или 3 частично доверенными
 - Длина цепочки не превышает 5 ключей

Другие аспекты

- Расширение сети доверия
- Важность keysigning party
- Публикация хеша ключа
- Правовые аспекты использования GnuPG