PGP — Urbanculture

Криптоконспирология

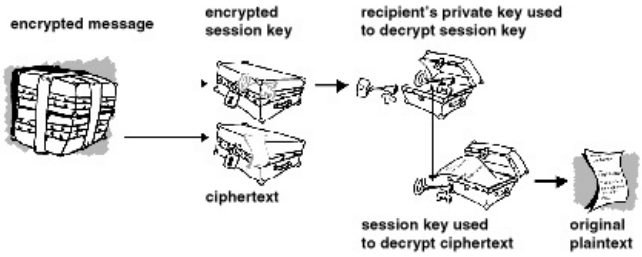
«Людям необходима конфиденциальность. PGP распространяется, как огонь в прериях, раздуваемый людьми, которые беспокоятся о своей конфиденциальности в этот информационный век. Сегодня организации по охране прав человека используют программу PGP для защиты своих людей за рубежом. Организация Amnesty International также использует её. »

— Филипп Циммерманн

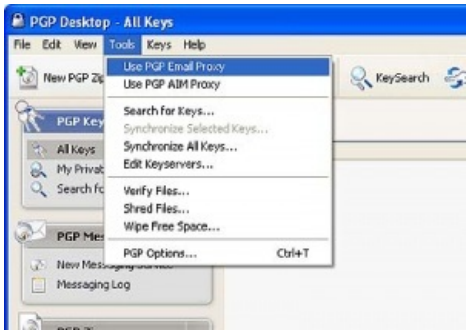
PGP — криптографическая система, основанная на криптографии с открытым ключом. Можно использовать как программу или как библиотеку, во втором случае можно самому написать программу, использующую PGP. В настоящее время считается одним из самых надёжных средств шифрования и ЭЦП. Существуют бесплатные и коммерческие версии. Изначально программа была бесплатной с открытым исходным кодом. Проанализировав код, ведущие криптоаналитики признали эффективность применения данного средства шифрования. Кроме самой PGP, существуют аналоги в виде filecrypt, gnupg (GPG), использующие те же алгоритмы. Все реализации полностью совместимы между собой — то, что зашифровано одной программой, может быть расшифровано другой при наличии ключа. Первые версии программы созданы Филиппом Циммерманном в 1991 году. Нынче технологии шифрования данных на основе PGP или совместимые с ней реализованы для всех ОС.

Как это работает

Основная статья: ***Криптография***



До 70-х годов 20 века было известно только симметричное шифрование, при котором для расшифровывания используется тот же ключ, что и для шифрования. Общим недостатком таких алгоритмов является необходимость безопасной передачи ключа через надежный (непрослушиваемый) канал. Прорывом было открытие асимметричного шифрования, когда ключ для шифрования может быть подслушан, но его знание не поможет криптоаналитику дешифровать сообщение. Для расшифровки используется закрытый ключ, который создается и хранится только у адресата. Асимметричные шифры работают медленнее, чем симметричные и требуют ключ большей длины для обеспечения такой же криптостойкости.



В современных версиях используют гибридное шифрование на основе обоих способов. Информацию вначале сжимают, затем шифруют одноразовым симметричным ключом, а тот, в свою очередь, асимметричным. Передаются зашифрованный текст и зашифрованный одноразовый ключ. (Сжатие нужно для устранения избыточности текста, позволяющей проводить анализ на основе наиболее часто встречающихся фрагментов.) Расшифровка идёт в обратном порядке: с помощью закрытого ключа расшифровывают одноразовый, а посредством оного уже расшифровывают весь текст [1]. Преимущество такого метода в том, что асимметричные алгоритмы менее стойки, чем симметричные, и при большом объёме данных легче подвергаются анализу [2]; кроме того, шифрование симметричным способом идёт значительно быстрее. Надёжность одноразового ключа по сравнению с паролем высока, ибо он создаётся не человеком, а генератором случайных или псевдослучайных чисел. Также симметричный ключ в данном случае не может быть скомпрометирован, так как используется только один раз и не демонстрируется отправителю или получателю — его знает лишь программа шифрования, в зашифрованном виде передающая его программе адресата внутри самого сообщения. При этом одноразовый ключ не содержит количество информации,

достаточное для успешного взлома или подбора закрытого ключа асимметричного алгоритма. Также такой одноразовый ключ не может быть подобран по словарю, в отличие от пароля, придуманного человеком.

Применяемые алгоритмы

- RSA — примечателен тем, что одна и та же пара ключей может использоваться и для асимметричного шифрования, и для ЭЦП.
- DSA
- Схема Эль-Гамаля (Elgamal)

Практическое применение

Тут понятно и ежу. Всё, что не должно быть известно третьим лицам, подлежит шифрованию. PGP и её аналоги дают достаточную (но не абсолютную) защиту конфиденциальности передаваемой информации, делая невозможным её анализ программами, собирающими персональные данные, и системами контроля типа eshelon и СОРМ. Кроме шифрования, PGP используется для создания электронной подписи документа — хэша на основе всего документа и закрытого ключа его владельца. Таким образом, обладатель публичного ключа может проверить, не был ли изменён документ при пересылке и в действительности ли он составлен обладателем ключа закрытого. При этом по имеющемуся хэшу невозможно установить сам закрытый ключ, но и невозможно создать подпись, аналогичную обладателю закрытого ключа. Электронная подпись позволяет обеспечить целостность документа, а также точно установить его владельца.

PGP с командной строки

На примере GNU Privacy Guard (свободная замена PGP).

Зашифруем файл **симметричным** шифром (программа спросит пароль):

```
gpg -c file.txt
```

Образуется файл *file.txt.gpg*.

Расшифруем файл *file.txt.gpg*:

```
gpg file.txt.gpg
```

При этом программа создаст файл *file.txt*.

Создадим пару ключей для **асимметричного** шифрования и ЭЦП:

```
gpg --gen-key
```

Всё, о чем спросит, можно оставить без изменений (нажимая ввод), кроме имени, адреса и комментария. По желанию можно указать пароль.

Пример

```
$ gpg --gen-key
gpg (GnuPG) 1.4.10; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection?
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y
```

```
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: BCPG v1.45

lQHsBFEkt+QBBADMrS0Hi0jJg4htS3/e8w8aSJXHjDdB++1itUuyW+FXJZkJg7e8
xwDcX87oPWC/QLn6ikduESIbvEn1jTqksaBzWxz+Q946DmVrAB+qPef+BzHrgzN/
ayYaessrijsyVTvOy4PBH2IOR/Y5ms/e6xmyyjGKFlO5yuWeyQIpdM9ClQARAQAB
/wMDAhqT7j4TXQBpYFqYqfAB36v5KvPqlkIInOsw8syhRn+KUv3BChTwq8tEO3v+
ZG/+6BDiOel6Rf54qdNrl3nVFYXAWrIfvfNJbcW7Ir8OWHTvI4fap1QpadT8rjFV
XhUBDdHiAmbBVEVR2bWlh+oykaTa9L2nSXhsRRIrmqe8+/s1ebY1/8mZndrraVVu
Cgch4blpeulKUZbbo2mB5+em3mPUG493OZmVdqogciWfnWkAvL2u7J3geNlF6CAm
Lv74UuUobC6D9+mht3/RB3kEQBA7R/a7Hy/nOTmnhv1h2M7GWQQzE34tXqdTg16P
b1qdZniqSBgEY3YOCYPQzgWMtYDiq5RYia2m/1Vd57kLeSkOIIBzmoV3jwh9uvcB
4ybzIoheu79DFXKmw8DYip2Mr6l4kz2pFI9WTrTCcVVUtzGx46/ChTLrFxAp6su5
MwKlav51FfGpo1GSGtRStFROZXNOIGtle5Bmb3IgcHVibGljYXRpb24gKNGN0YLQ
viDRgtC1OYHRgtC+OLLRi9C5INC6OLvRjtGHKSA8dGVzdGtleXB1YkBleGFtcGxl
Lm9yZz6InAQTAQIABgUCUS54AQAKCRB/PxXhdiENqSFtBAC9ptNXixoYFiajvQV2
V8D4t7lDmFfzAJyHX5TIzz5sCfFqQLke7fy2UbYyu2wKMT47rdhm4pQBV/wdvGP8
IlwoWQVBKAE0zocCepDYF+ink7BZoJhXg6OY9mS/tesOoSxl6MyfRagHB6+vjrwc
Bef2qtP5Mh8HJ2H1h4dsNphT24iwBBWBAgAaBQJRJLgBApsPBosJCAcDAgSVAggD
BJYCAwEACgkQfz8V4XYhDamBWAP/cgn2n0uTOyQZkuPgJlvUuh+QkIAxYIpFkXR8
tWQO6MtMacCmVG31p1fbi8OJf15ftIPs47OKUCCh83OpwVr9aKOXvKeTK1vOE0rT
yQyYDtwh3kcN1AJu+AWuaDhDOaGbSh7IF8X4s3GhkuResUzayt2eJT6/8teQB/Ox
9DgRbc4=
=zNWb
-----END PGP PRIVATE KEY BLOCK-----
```

Пример закрытого ключа

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: BCPG v1.45

mI0EUSS37QEEAMytLQeLSMmDiG1Lf97zDxpIlceMNOH77WK1S7Iz4VclmQmDt7zF
YNxfzug8WL9AufqKR24Tkhu8SfwNOqSxoHNbHP5D3joOZWsAH6o95/4HMeuDM39r
Jhp6yyuKOzJW2/fLg8EfYjRH9jmaz97rGbLKMYoWXRLK5Z7JAil0z0KVABEBAAG0
VHRlc3QgaZV5IGZvciBwdWJsaWNhdGlvbiAoOY3RgtC+INGC0LXRgdGC0L7QstGL
OLkg0LrQu9GOOYcpIDx0ZXN0a2V5cHVi0GV4YWlwbGUub3JnPoicBBMBAgAGBQJR
JLgBAAoJEH8/FeF2IQ2pIWOEAL2m01eLGhgWJqO9BXZXwPi3uUOYV/NAnIdflNjP
NKwJ8WpAuR7t/LZRtjK7bAoxPjut2Gbi1AFX/B28Y/w1VahZBUEoATTOhwJ6kNgX
6KeTsFngmFeDo5j2ZL+16w6hLGXozJ9FqAcHr6+OvBwF5/aqO/keHwcnYeWHh2w2
mFPbiLAEEwECABoFAlEkuAECmw8GiwkIBwMCBJUCCAMEigIDAQAKCRB/PxXhdiEN
qYEwA/9yCfafS5M7JBmRQ+ArW9S6H5CQgDFgikWRdHy1ZDToy0xpwKZUbfWnV9uL
w41+VJ+Dg+zjvQpQIKHzc6nBWv1oo5e8p5MrW/QTStPJDJgO3CHeRw3UAm74Ba5o
OEPRoZtKHsgXxfizcaGS5F6xTNrK3Z41Pr/y15AH/THOOBFtzg==
=Do38
-----END PGP PUBLIC KEY BLOCK-----
```

Пример открытого ключа из той же пары

```
You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: test
Name must be at least 5 characters long
Real name: test test
Email address: test@example.com
Comment: no comment
You selected this USER-ID:
    "test test (no comment) <test@example.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

gpg: key BC0A6855 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   2048R/BC0A6855 2013-03-03
      Key fingerprint = 7A9D 15DD 2E7C 44C7 CACA  6D59 27CC 8672 BC0A 6855
uid                  test test (no comment) <test@example.com>
sub   2048R/4C920DE9 2013-03-03
```

Хеш публичного ключа: BC0A6855, фингерпринт: 7A9D 15DD 2E7C 44C7 CACA 6D59 27CC 8672 BC0A 6855

Посмотрим ещё раз на фингерпринт нашего ключа:

```
gpg --fingerprint
```

Пример

```
$ gpg --fingerprint
/home/guest/.gnupg/pubring.gpg
------------------------------
pub   2048R/BC0A6855 2013-03-03
      Key fingerprint = 7A9D 15DD 2E7C 44C7 CACA  6D59 27CC 8672 BC0A 6855
uid                  test test (no comment) <test@example.com>
sub   2048R/4C920DE9 2013-03-03
```

Укажем в настройках сервер ключей. Для этого добавим в файл *.gnupg/gpg.conf*:

```
keyserver hkp://pool.sks-keyservers.net
```

где pool.sks-keyservers.net — адрес сервера ключей.

Вместо этого можно каждый раз указывать сервер ключей в командной строке: --keyserver pool.sks-keyservers.net

Отправим наш ключ в **сервер ключей**:

```
gpg --send-keys хеш_ключа
```

Пример

```
$ gpg --send-keys BC0A6855
gpg: sending key BC0A6855 to hkp server pool.sks-keyservers.net
```

Серверы ключей обмениваются ключами между собой, поэтому достаточно загрузить ключ на один из них, а скоро он окажется доступен на всех серверах.

У сервера ключей может быть веб-интерфейс, через который можно загружать и скачивать ключи. При поиске по хешу нужно перед хешом приписывать «0x».

Пример
результаты поиска ключа с хешом BC0A6855

На другом компьютере получим ключ с сервера ключей:

```
gpg --recv-keys хеш_ключа
```

Пример

```
$ gpg --recv-keys BC0A6855
gpg: requesting key BC0A6855 from hkp server pool.sks-keyservers.net
gpg: key BC0A6855: public key "test test (no comment) <test@example.com>" imported
gpg: Total number processed: 1
```

```
gpg:               imported: 1  (RSA: 1)
```

Теперь авторство файлов, подписанных этим ключом, будет считаться достоверным.

Альтернативный способ передачи публичного ключа — через файл:

```
gpg --output файл_с_ключем.gpg --export
```

Можно добавить опцию --armor, чтобы файл получился текстовым.

Пример

```
$ gpg --output BC0A6855.gpg --armor --export
```

Скопируем файл на другой компьютер и импортируем его:

```
gpg --import файл_с_ключем.gpg
```

Пример

```
$ gpg --import /tmp/BC0A6855.gpg
gpg: key BC0A6855: public key "test test (no comment) <test@example.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
```

Теперь нужно подписать ключ:

```
gpg --edit BC0A6855 sign
```

На втором компьютере **зашифруем** файл открытым ключом с первого:

```
gpg --recipient получатель --encrypt файл
```

Получателя можно указывать разными способами, например через хеш его ключа, имя или e-mail. Чтобы скрыть получателя, используйте --hidden-recipient вместо --recipient.

Пример

```
$ gpg --recipient BC0A6855 --encrypt 1.txt
```

Получится файл 1.txt.pgp, который нужно отправить на первый компьютер.

На первом компьютере расшифруем файл:

```
gpg файл
```

Пример

```
$ gpg 1.txt.gpg
gpg: encrypted with 2048-bit RSA key, ID 4C920DE9, created 2013-03-03
      "test test (no comment) <test@example.com>"
```

Подпишем файл:

```
gpg --sign 1.txt # бинарный файл 1.txt.gpg, включает исходный файл
gpg --clearsign 1.txt # текстовый файл 1.txt.asc, включает исходный файл
gpg --detach-sig 1.txt # бинарный файл 1.txt.asc, не включает исходный файл (только подпись)
```

Проверим подпись: gpg --verify файл_с_подписью # проверяет подпись gpg --decrypt файл_с_подписью --output новый_файл # проверяет подпись и извлекает подписанный файл

Пример

```
$ gpg --verify 1.txt.sig
gpg: Signature made Sun 03 Mar 2013 04:42:13 PM MSK using RSA key ID BC0A6855
gpg: Good signature from "test test (no comment) <test@example.com>"
$ gpg --verify 1.txt.asc
gpg: Signature made Sun 03 Mar 2013 04:40:31 PM MSK using RSA key ID BC0A6855
gpg: Good signature from "test test (no comment) <test@example.com>"
$ gpg --verify 1.txt.gpg
gpg: Signature made Sun 03 Mar 2013 04:39:05 PM MSK using RSA key ID BC0A6855
gpg: Good signature from "test test (no comment) <test@example.com>"
$ gpg --output 1.txt --decrypt 1.txt.gpg
gpg: Signature made Sun 03 Mar 2013 04:39:05 PM MSK using RSA key ID BC0A6855
gpg: Good signature from "test test (no comment) <test@example.com>"
$ cat 1.txt
hello
$ gpg --output a.txt --decrypt 1.txt.asc
gpg: Signature made Sun 03 Mar 2013 04:40:31 PM MSK using RSA key ID BC0A6855
gpg: Good signature from "test test (no comment) <test@example.com>"
$ cat a.txt
```

```
hello
```

Опции -c, --encrypt и --sign можно использовать совместно.

Ещё
Удалить чей-то публичный ключ:

```
gpg --delete-keys кто-то
```

Протестировать импорт ключа (вхолостую):

```
gpg --dry-run --import file.gpg
```

См. также

- Jabber/XMPP
- RetroShare

Ссылки

- Установка и использование PGP
- Русская энциклопедия
- Статья в википедии
- Сервер обмена ключами

Примечания

1. ↑ https://www.pgpru.com/biblioteka/osnovy/vvedenievkripto/glava1/kakdejjstvuetpgp
2. ↑ по надёжности 1024-битный асимметричный ключ сравним с 128-битным симметричным

Рекомендуется ознакомиться с обсуждением этой статьи	