

به نام خدا



امنیت داده و شبکه

تمرین چهارم

استاد :

دکتر مهدی خرازی، دکتر مرتضی امینی، دکتر کامبیز میزانیان

نویسنده :

محمد هومان کشوری

شماره دانشجویی :

۹۹۱۰۵۶۶۷

سوال ۱

آ) برای اولین قسمت از پروتکل diffie-helman برای ساخت و تبادل کلید استفاده می‌کنیم.

$$C \rightarrow KDC : IP_S$$

$$KDC \rightarrow C : K_{C,S}$$

$$C \rightarrow S : E(K_{C,S}, \alpha || q || Y_C = (\alpha^{X_C} \bmod q))$$

$$S \rightarrow C : E(K_{C,S}, Y_S = (\alpha^{X_S} \bmod q))$$

$$K' = Y_C^{X_S} \bmod q = Y_S^{X_C} \bmod q$$

حال عملاً با دیفی هلمن توانستیم کلیدی ایجاد کنیم که KDC از آن اطلاعی ندارد اما طرفین آن را می‌دانند.

ب) در صورتی که KDC توان تغییر ترافیک را داشته باشد می‌تواند حمله Man-In-The-Middle را پیاده‌سازی کند و عملاً خود را برای S جای C و برای C جای S جا بزند.

$$C \rightarrow KDC : IP_S$$

$$KDC \rightarrow C : K_{C,S}$$

$$C \rightarrow S : E(K_{C,S}, \alpha || q || Y_C = (\alpha^{X_C} \bmod q))$$

$$KDC \text{ Intercept} - KDC \rightarrow S : E(K_{C,S}, \alpha || q || Y'_C = (\alpha^{X_{KDC}} \bmod q))$$

$$S \rightarrow C : E(K_{C,S}, Y_S = (\alpha^{X_S} \bmod q))$$

$$KDC \text{ Intercept} - KDC \rightarrow C : E(K_{C,S}, \alpha || q || Y'_S = (\alpha^{X_{KDC}} \bmod q))$$

ج) می‌دانیم که KDC با یکدیگر امکان تبانی ندارند پس منتظر از کلیدهای یکدیگر نیز خبر ندارند. تنها کاری که لازم است انجام دهیم این است که K' را با هر دو کلید رمز کنیم.

$$C \rightarrow S : E(K_{C,S,2}, E(K_{C,S,1}, K'))$$

حال در صورتی که یکی از KDC ها پیام را باز کند، نمی‌تواند متوجه پیام شود چون KDC_1 نمی‌تواند پیام بیرونی را باز کند و KDC_2 نیز نمی‌تواند پیام داخلی را باز کند.

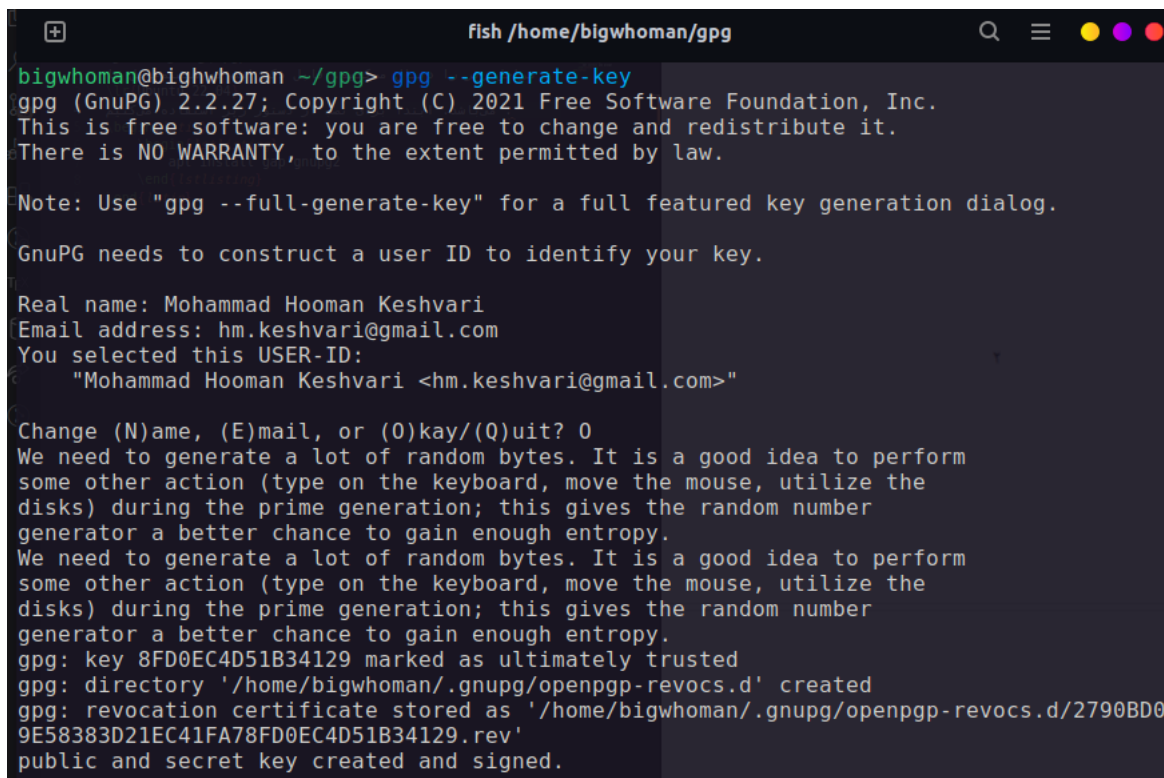
سوال ۲

برای راه اندازی pgp مراحل زیر را دنبال می کنیم. قابل ذکر است که سیستم عامل ما Ubuntu 22.04 می باشد. ابتدا برای نصب از دستور زیر استفاده می کنیم:

```
1 apt install gap gnupg2
```

2

سپس باید جفت کلید خود را جنریت کنیم.



```
fish /home/bigwhoman/gpg
bigwhoman@bigwhoman ~/gpg> gpg --generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 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.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Mohammad Hooman Keshvari
Email address: hm.keshvari@gmail.com
You selected this USER-ID:
    "Mohammad Hooman Keshvari <hm.keshvari@gmail.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? 0
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 8FD0EC4D51B34129 marked as ultimately trusted
gpg: directory '/home/bigwhoman/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/bigwhoman/.gnupg/openpgp-revocs.d/2790BD0
9E58383D21EC41FA78FD0EC4D51B34129.rev'
public and secret key created and signed.
```

بعد از جنریت کردن جفت کلیدهای خود، باید کلید عمومی را با زدن دستور زیر import کنیم.

```
1 gpg --import Reza_0xCFBEEE88_public.asc
```

سپس با زدن دستورات زیر ابتدا plain text را امضا کرده و سپس آنرا encrypt می کنیم.

```
bigwhoman@bighwhoman /m/b/l/S/t/S/D/HW4 (main)> echo -e "Subject: Mohammad Hooman Keshvari\n\n99105667"
| gpg --clearsign --armor > message
bigwhoman@bighwhoman /m/b/l/S/t/S/D/HW4 (main)> gpg --encrypt --armor --Recipient Reza message
gpg: 5F4AB93836D3E77F: There is no assurance this key belongs to the named user

sub cv25519/5F4AB93836D3E77F 2023-12-26 Reza <reza.saeedi9@yahoo.com>
Primary key fingerprint: 981C 65B7 BA35 83E2 6981 A39B B334 7414 CFBE EE88
Subkey fingerprint: 4395 AAE0 6458 29FC 1ACF 8443 5F4A B938 36D3 E77F

It is NOT certain that the key belongs to the person named
in the user ID. If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
```

با دستورات بالا یک فایل message.asc ساخته می‌شود که حاوی پیام و امضا شده آن است که در سمت گیرنده با دستور زیر می‌توان به خود پیام رسید.

```
1 gpg --decrypt --output message.txt message.asc
```

در صورتی که امضا درست باشد باید با زدن دستور زیر خروجی مانند خروجی زیر را مشاهده کنید.
ساخته می‌شود که ابتدا باید مطمئن شویم درست امضا شده :

```
bigwhoman@bighwhoman /m/b/l/S/t/S/D/HW4 (main)> gpg --verify message.txt
gpg: Signature made 0330+ ۱۲:۵۴:۳۷ , ۲۴ شنبه ۰۶ زانویه ۱۴۰۳
gpg: using RSA key 2790BD09E58383D21EC41FA78FD0EC4D51B34129
gpg: Good signature from "Mohammad Hooman Keshvari <hm.keshvari@gmail.com>" [ultimate]
```

با زدن دستور زیر نیز کلید عمومی خود را در فایل keshvari.asc قرار می‌دهیم و آنرا به ایمیل ضمیمه می‌کنیم.

```
1 gpg --armor --output keshvari.asc --export "Mohammad Hooman Keshvari"
```

سوال ۳

- آ) جست و جوی فراگیر : در این حمله، حمله کننده عملاً تمامی ترکیبات ممکن برای شکستن رمزنگاری را امتحان می کند.
- متن اصلی معلوم : در این حمله، حمله کننده یک plain text را به همراه رمز شده آن دارد و عملاً می تواند به رابطه بین این دو پی ببرد.
- حمله تکرار : حمله ای است که حمله کننده در آن یک بسته را گرفته و آنرا عیناً دوباره ارسال می کند. در هر دو سرویس AH و ESP در IPSec پارامترهای SA مربوطه یک Sequence Number Counter گذاشته می شود و همچنین یک Anti Replay Window که پنجره بسته های ارسال شده را دارد و اگر فرض کنیم بسته های m تا n را پوشش می دهد و آمار آنها را دارد، در صورتی که بسته $n + 1$ بیاید، پنجره را یکی به جلو می برد. پس می توان مطمئن بود که بسته های بزرگ تر از n که هنوز نیامده اند. بسته های بین m تا n نیز وضعیت آنها مشخص است و در صورتی که بسته کمتر از m بیاید، آنرا تکراری در نظر می گیریم و اجازه عبور نمی دهیم.
- شوند رمز عبور : حمله ای که در آن، حمله کننده با استفاده از ابزارهای متفاوت اطلاعات حساس و پسوردها را در حین عبور در شبکه شنود می کند. پروتکل ESP در IPSec به این علت که عملاً قسمت data را به صورت کامل encrypt می کند، بجز خود و مقصد، هیچ کس در این بین نمی تواند از محتویات بسته اطلاع یابد. در بین مودهای انتقال و تونل عملاً مود انتقال چون end-to-end است بهتر است چون در مود تونل عملاً چون دو سمت تونل یک دور کل بسته را باز می کنند، خود می توانند محتویات data را ببینند مگر اینکه محتویات data را نیز به صورت جداگانه رمز کنیم.
- جعل IP : حمله ای که در آن، حمله کننده قسمت IP بسته ها را عوض کرده تا بنظر برسد مبدا یا مقصد متفاوتی دارند. در پروتکل AH، عملاً بخش هایی از سرایند IP و بخش data در IP یک MAC گرفته می شود که در صورت تغییر هر کدامیک از آنها متوجه تغییر آن می شویم. این کارکرد برای هر دو مود تونل و انتقال در AH صادق است.
- سرقت IP : حمله ای که در آن حمله کننده کنترل شبکه فرد را به دست می گیرد و و عملاً خود را به جای وی جا می زند.
- حمله SYN flooding : حمله ای است که حمله کننده در مرحله سوم handshake در لایه انتقال، به جای فرستادن ACK برای کارگزار، دوباره یک SYN می فرستد و عملاً باعث ایجاد یک حمله dos می شود. IPSec نمی تواند جلوی SYN flooding را بگیرد چرا که این حمله عملاً در لایه انتقال تعریف می شود اما IPSec در بین لایه های IP و انتقال تعریف می شود.

ب) برای راه اندازی پروتکل IPSec بر روی سیستم عامل لینوکس، ابتدا مطابق زیر پکیج های خواسته شده را نصب می کنیم.

```
1 sudo apt update
2 sudo apt install strongswan
```

سپس برای تغییر تنظیمات باید این دو فایل زیر را تغییر دهیم :

● /etc/ipsec.conf

● /etc/ipsec.secret

محتویات تنظیمات ماشین مجازی اول با ای پی 192.168.2.189 :

```
1 config setup
2     uniqueids=no
3     charondebug=ike 2
4
5
6 # Add connections here.
7
8 conn %default
9     ikelifetime=60m
```

```

10 keylife=20m
11 keyingtries=1
12 ike=aes128-sha256-modp1024 , aes256-sha384-modp1024 !
13 keyexchange=ikev2
14
15 conn ubuntu
16     authby=secret
17     leftsubnet=192.168.2.0/24
18     leftsendcert=never
19     right=192.168.2.188
20     rightsubnet=192.168.2.0/24
21     auto=start

```

محتویات تنظیمات ماشین مجازی اول با ای پی 192.168.2.188 :

```

1 config setup
2     uniqueids=no
3     charondebug=ike 2
4
5
6 # Add connections here.
7
8 conn %default
9     ikelifetime=60m
10    keylife=20m
11    keyingtries=1
12    ike=aes128-sha256-modp1024 , aes256-sha384-modp1024 !
13    keyexchange=ikev2
14
15 conn ubuntu
16     authby=secret
17     leftsubnet=192.168.2.0/24
18     right=192.168.2.189
19     rightsubnet=192.168.2.0/24
20     auto=start

```

حال در نهایت باید فایل /etc/ipsec.secrets را تغییر دهیم تا دو ماشین مجازی بتوانند یکدیگر را authenticate کنند. محتوای ipsec.secrets در هر دو ماشین مجازی :

```

1 192.168.2.189 192.168.2.188 : PSK "123"

```

در بالا، یک کلید مشترک^۱ "۱۲۳" بین دو ماشین مجازی تنظیم شده است. حال با زدن دستورات زیر ipsec ران می‌شود.

```

1 sudo ipsec rereadsecrets
2 sudo ipsec start

```

در ماشین‌های مجازی با زدن دستور زیر می‌توانیم وضعیت دو ماشین را ببینیم :

```

1 sudo ipsec status

```

^۱Pre-Shared Key

```
ub-clone [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hoom@hoom:~$ sudo ipsec status
Security Associations (1 up, 0 connecting):
    ubuntu[3]: ESTABLISHED 86 seconds ago, 192.168.2.189[192.168.2.189]...192.168.2.188[192.168.2.188]
    ubuntu[5]:  INSTALLED, TUNNEL, reqid 2, ESP SPIs: c67e5106_i cd2c9253_o
    ubuntu[5]:  192.168.2.0/24 === 192.168.2.0/24
hoom@hoom:~$
```

```
ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
hoom@hoom:~$ sudo ipsec status
Security Associations (1 up, 0 connecting):
    ubuntu[1]: ESTABLISHED 3 minutes ago, 192.168.2.188[192.168.2.188]...192.168.2.189[192.168.2.189]
    ubuntu[1]:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: cd2c9253_i c67e5106_o
    ubuntu[1]:  192.168.2.0/24 === 192.168.2.0/24
hoom@hoom:~$ _
```