

Sindy Morel	LAB7: Applying Encryption and Hashing Algorithms for Secure Communications	September 23, 2023
-------------	---	--------------------

LAB7: Applying Encryption and Hashing Algorithms for Secure Communications

Section 1: Hands-On Demonstration

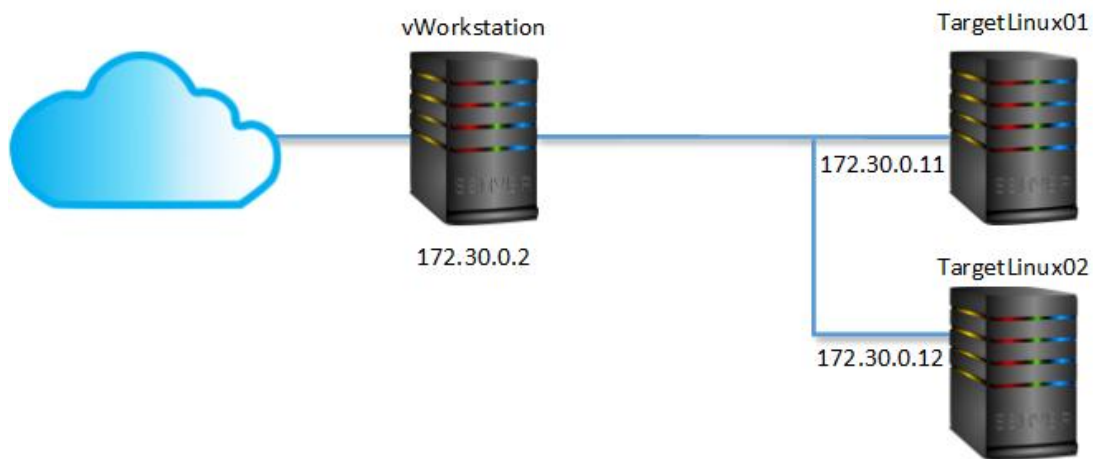
Topology.....	1
.....	2
Tools and Software.....	2
Hands-On Demonstration.....	2
Part 2: Create a MD5sum and a SHA1sum Hash String.....	3
Part 3: Modify a File and Verify Hash Values.....	6
Part 4: Generate GnuPG Keys.....	7
Part 5: Share a GnuPG Key.....	8
Part 6: Encrypt and Decrypt a ClearText Message.....	9

Sindy Morel	LAB7: Applying Encryption and Hashing Algorithms for Secure Communications	September 23, 2023
-------------	---	--------------------

Topology

This lab contains the following virtual devices. Please refer to the network topology diagram below.

- vWorkstation (Windows Server 2016)
- TargetLinux01 (Debian Linux)
- TargetLinux02 (Debian Linux)



Tools and Software

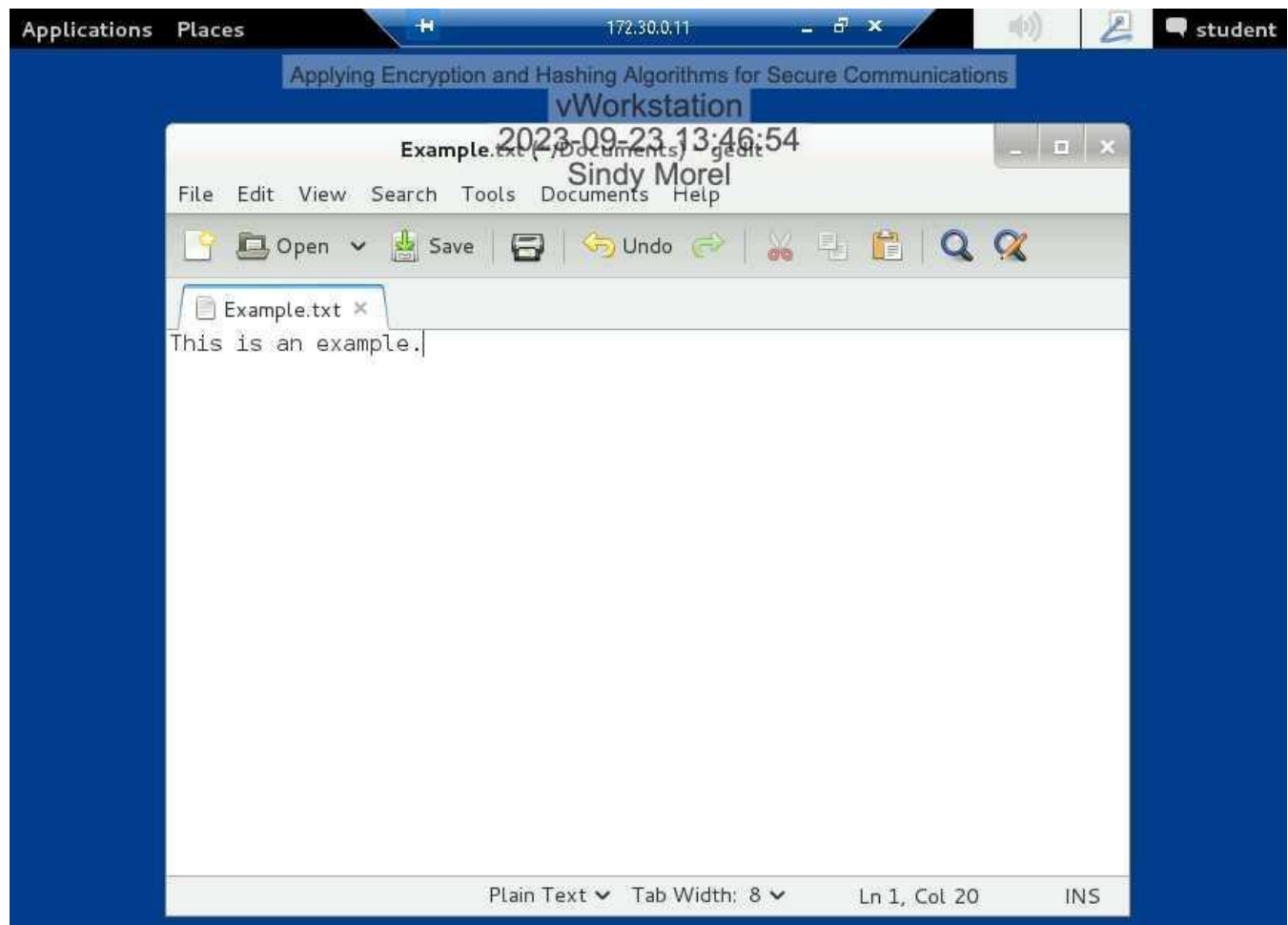
The following software and/or utilities are required to complete this lab. Students are encouraged to explore the Internet to learn more about the products and tools used in this lab.

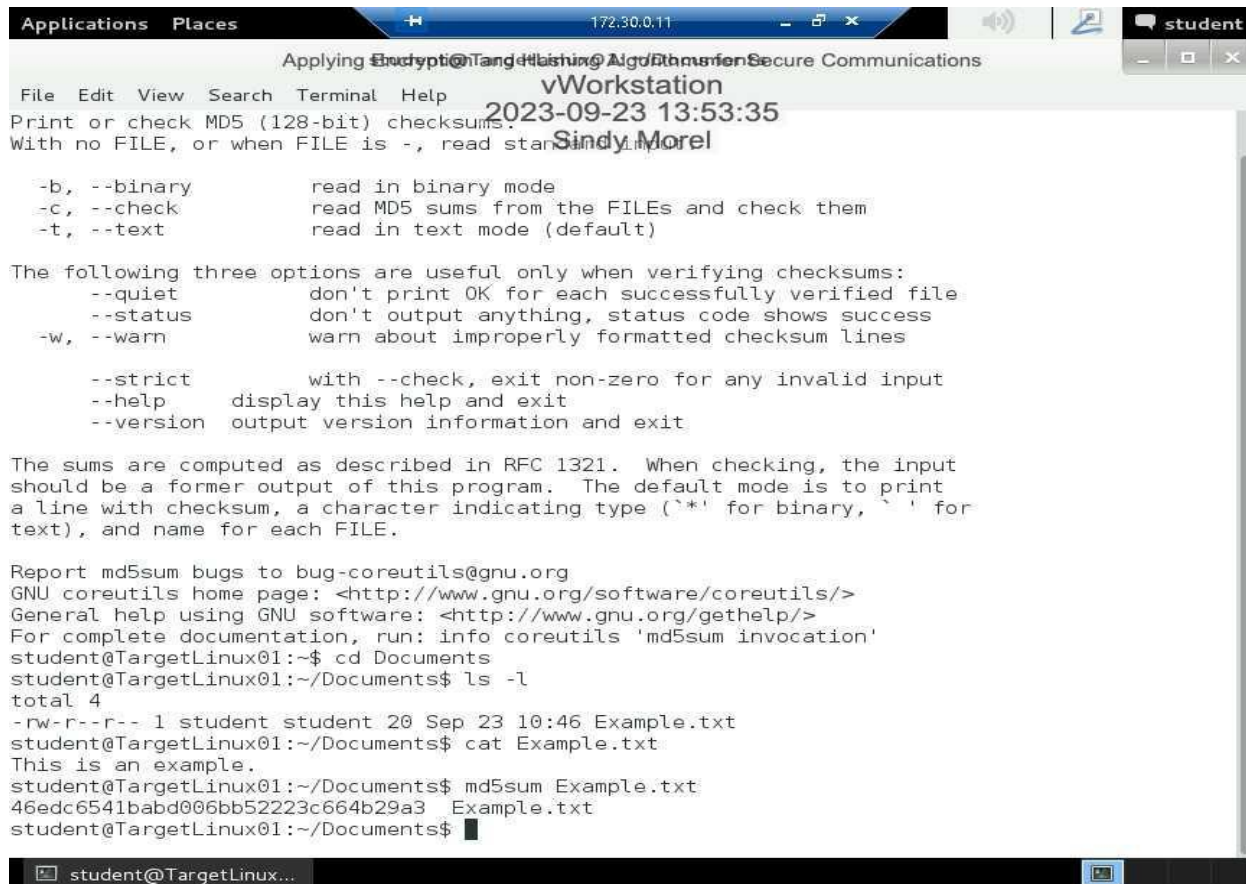
- GNU Privacy Guard (GnuPG or GPG)
- KeyTransfer
- WinSCP
- vi Editor

Sindy Morel	LAB7: Applying Encryption and Hashing Algorithms for Secure Communications	September 23, 2023
-------------	---	--------------------

Hands-On Demonstration

Part 1: Create a Text File on Linux



Part 2: Create a MD5sum and a SHA1sum Hash String

The screenshot shows a terminal window titled "Applying Encryption and Hashing Algorithms for Secure Communications" with a window manager bar at the top. The terminal displays the help text for the `md5sum` command, followed by a series of commands and their outputs. The user is logged in as "student" on a machine named "TargetLinux01".

```
File Edit View Search Terminal Help
Print or check MD5 (128-bit) checksums.
With no FILE, or when FILE is -, read standard input.

  -b, --binary      read in binary mode
  -c, --check       read MD5 sums from the FILEs and check them
  -t, --text       read in text mode (default)

The following three options are useful only when verifying checksums:
  --quiet          don't print OK for each successfully verified file
  --status         don't output anything, status code shows success
  -w, --warn       warn about improperly formatted checksum lines

  --strict         with --check, exit non-zero for any invalid input
  --help          display this help and exit
  --version       output version information and exit

The sums are computed as described in RFC 1321.  When checking, the input
should be a former output of this program.  The default mode is to print
a line with checksum, a character indicating type ('*' for binary, '-' for
text), and name for each FILE.

Report md5sum bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'md5sum invocation'
student@TargetLinux01:~$ cd Documents
student@TargetLinux01:~/Documents$ ls -l
total 4
-rw-r--r-- 1 student student 20 Sep 23 10:46 Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example.
student@TargetLinux01:~/Documents$ md5sum Example.txt
46edc6541babd006bb52223c664b29a3 Example.txt
student@TargetLinux01:~/Documents$
```



The screenshot shows a terminal window titled "Applying Encryption and Hashing Algorithms for Secure Communications" within a "vWorkstation" environment. The terminal displays the help text for the `md5sum` command, followed by a series of commands and their outputs. The user navigates to the `Documents` directory, lists files, cat's `Example.txt`, calculates its MD5 checksum, saves it to `Example.txt.md5`, and then cat's the checksum file.

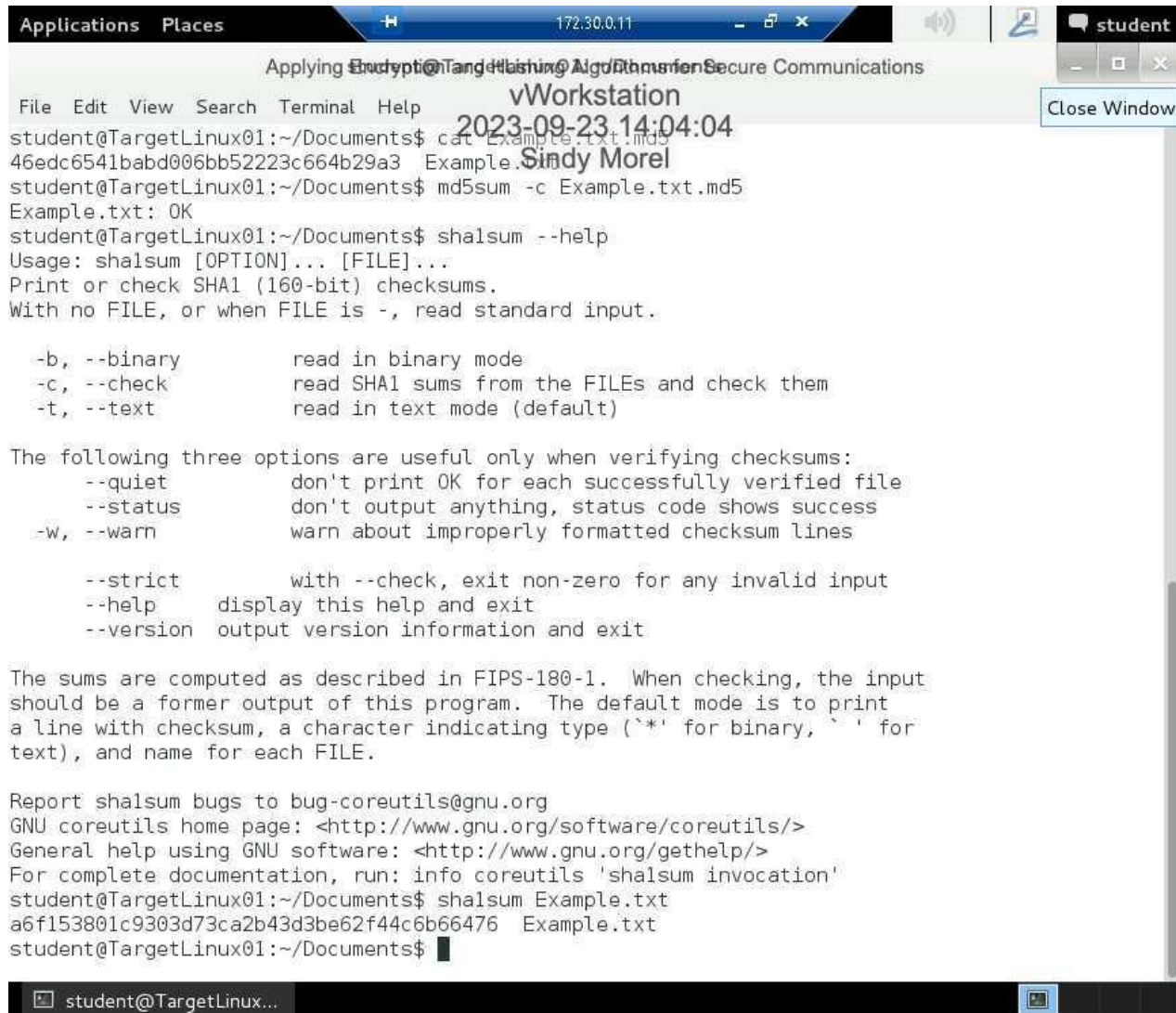
```
Applications  Places  172.30.0.11  student
Applying Encryption and Hashing Algorithms for Secure Communications
vWorkstation
2023-09-23 14:00:49
Sindy Morel
File Edit View Search Terminal Help
-t, --text      read in text mode (default)

The following three options are useful only when verifying checksums:
--quiet        don't print OK for each successfully verified file
--status       don't output anything, status code shows success
-w, --warn      warn about improperly formatted checksum lines

--strict       with --check, exit non-zero for any invalid input
--help         display this help and exit
--version      output version information and exit

The sums are computed as described in RFC 1321.  When checking, the input
should be a former output of this program.  The default mode is to print
a line with checksum, a character indicating type (* for binary, ` ' for
text), and name for each FILE.

Report md5sum bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'md5sum invocation'
student@TargetLinux01:~$ cd Documents
student@TargetLinux01:~/Documents$ ls -l
total 4
-rw-r--r-- 1 student student 20 Sep 23 10:46 Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example.
student@TargetLinux01:~/Documents$ md5sum Example.txt
46edc6541babd006bb52223c664b29a3  Example.txt
student@TargetLinux01:~/Documents$ md5sum Example.txt > Example.txt.md5
student@TargetLinux01:~/Documents$ ls
Example.txt  Example.txt.md5
student@TargetLinux01:~/Documents$ cat Example.txt.md5
46edc6541babd006bb52223c664b29a3  Example.txt
student@TargetLinux01:~/Documents$
```



```
Applications Places 172.30.0.11 student
Applying Encryption and Hashing Algorithms for Secure Communications
vWorkstation
2023-09-23 14:04:04
File Edit View Search Terminal Help
student@TargetLinux01:~/Documents$ cat Example.txt.md5
46edc6541babd006bb52223c664b29a3 Example.txt
student@TargetLinux01:~/Documents$ md5sum -c Example.txt.md5
Example.txt: OK
student@TargetLinux01:~/Documents$ shasum --help
Usage: shasum [OPTION]... [FILE]...
Print or check SHA1 (160-bit) checksums.
With no FILE, or when FILE is -, read standard input.

  -b, --binary      read in binary mode
  -c, --check       read SHA1 sums from the FILES and check them
  -t, --text       read in text mode (default)

The following three options are useful only when verifying checksums:
  --quiet          don't print OK for each successfully verified file
  --status         don't output anything, status code shows success
  -w, --warn       warn about improperly formatted checksum lines

  --strict        with --check, exit non-zero for any invalid input
  --help          display this help and exit
  --version       output version information and exit

The sums are computed as described in FIPS-180-1. When checking, the input
should be a former output of this program. The default mode is to print
a line with checksum, a character indicating type ('*' for binary, ' ' for
text), and name for each FILE.

Report shasum bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'shasum invocation'
student@TargetLinux01:~/Documents$ shasum Example.txt
a6f153801c9303d73ca2b43d3be62f44c6b66476 Example.txt
student@TargetLinux01:~/Documents$
```




The screenshot shows a vWorkstation terminal window titled "Applying Encryption and Hashing Algorithms for Secure Communications". The terminal displays the help text for the `shasum` command. The window's title bar includes "Applications", "Places", a network address "172.30.0.11", and a "student" user indicator. The terminal output is as follows:

```
Usage: shasum [OPTION]... [FILE]...
Print or check SHA1 (160-bit) checksums.
With no FILE, or when FILE is -, read standard input.

  -b, --binary      read in binary mode
  -c, --check        read SHA1 sums from the FILEs and check them
  -t, --text        read in text mode (default)

The following three options are useful only when verifying checksums:
  --quiet           don't print OK for each successfully verified file
  --status          don't output anything, status code shows success
  -w, --warn        warn about improperly formatted checksum lines

  --strict          with --check, exit non-zero for any invalid input
  --help            display this help and exit
  --version         output version information and exit

The sums are computed as described in FIPS-180-1.  When checking, the input
should be a former output of this program.  The default mode is to print
a line with checksum, a character indicating type ('*' for binary, ' ' for
text), and name for each FILE.

Report shasum bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'shasum invocation'
student@TargetLinux01:~/Documents$ shasum Example.txt
a6f153801c9303d73ca2b43d3be62f44c6b66476  Example.txt
student@TargetLinux01:~/Documents$ shasum Example.txt > Example.txt.sh1
student@TargetLinux01:~/Documents$ ls
Example.txt  Example.txt.md5  Example.txt.sh1
student@TargetLinux01:~/Documents$ cat Example.txt.sh1
a6f153801c9303d73ca2b43d3be62f44c6b66476  Example.txt
student@TargetLinux01:~/Documents$
```



The screenshot shows a terminal window titled "Applying Encryption and Hashing Algorithms for Secure Communications" running on a vWorkstation. The terminal displays the usage and options for the `shasum` command. It lists options like `-b` for binary mode, `-c` for checking SHA1 sums, and `-t` for text mode. It also shows three useful options for verifying checksums: `--quiet`, `--status`, and `--warn`. The terminal then shows the command being run on a file named `Example.txt`, displaying the resulting SHA1 checksum. Finally, it shows the command being run with the `-c` option to verify the checksum, resulting in an "OK" status.

```
Applying Encryption and Hashing Algorithms for Secure Communications
vWorkstation
2023-09-23 14:06:49
Sindy Morel

File Edit View Search Terminal Help
With no FILE, or when FILE is -, read standard input:

-b, --binary      read in binary mode
-c, --check       read SHA1 sums from the FILEs and check them
-t, --text       read in text mode (default)

The following three options are useful only when verifying checksums:
--quiet          don't print OK for each successfully verified file
--status         don't output anything, status code shows success
-w, --warn       warn about improperly formatted checksum lines

--strict         with --check, exit non-zero for any invalid input
--help           display this help and exit
--version        output version information and exit

The sums are computed as described in FIPS-180-1.  When checking, the input
should be a former output of this program.  The default mode is to print
a line with checksum, a character indicating type ('*' for binary, ' ' for
text), and name for each FILE.

Report shasum bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'shasum invocation'
student@TargetLinux01:~/Documents$ shasum Example.txt
a6f153801c9303d73ca2b43d3be62f44c6b66476 Example.txt
student@TargetLinux01:~/Documents$ shasum Example.txt > Example.txt.shal
student@TargetLinux01:~/Documents$ ls
Example.txt Example.txt.md5 Example.txt.shal
student@TargetLinux01:~/Documents$ cat Example.txt.shal
a6f153801c9303d73ca2b43d3be62f44c6b66476 Example.txt
student@TargetLinux01:~/Documents$ shasum -c Example.txt.shal
Example.txt: OK
student@TargetLinux01:~/Documents$
```


Part 3: Modify a File and Verify Hash Values



The screenshot shows a terminal window titled "Applying Encryption and Hashing Algorithms for Secure Communications" with a "vWorkstation" icon. The terminal output includes the following text:

```
File Edit View Search Terminal Help
The following three options are useful only when verifying checksums:
  --quiet      don't print OK for successfully verified file
  --status     don't output anything, status code shows success
  -w, --warn   warn about improperly formatted checksum lines

  --strict     with --check, exit non-zero for any invalid input
  --help      display this help and exit
  --version   output version information and exit

The sums are computed as described in FIPS-180-1. When checking, the input
should be a former output of this program. The default mode is to print
a line with checksum, a character indicating type ('*' for binary, ' ' for
text), and name for each FILE.

Report shasum bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'shasum invocation'
student@TargetLinux01:~/Documents$ shasum Example.txt
a6f153801c9303d73ca2b43d3be62f44c6b66476 Example.txt
student@TargetLinux01:~/Documents$ shasum Example.txt > Example.txt.shal
student@TargetLinux01:~/Documents$ ls
Example.txt Example.txt.md5 Example.txt.shal
student@TargetLinux01:~/Documents$ cat Example.txt.shal
a6f153801c9303d73ca2b43d3be62f44c6b66476 Example.txt
student@TargetLinux01:~/Documents$ shasum -c Example.txt.shal
Example.txt: OK
student@TargetLinux01:~/Documents$ echo sindymorel >> Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example.
sindymorel
student@TargetLinux01:~/Documents$ md5sum Example.txt
eal04498baee238b9ea83c16599eceae Example.txt
student@TargetLinux01:~/Documents$
```



The screenshot shows a terminal window titled "Applying Encryption and Hashing Algorithms for Secure Communications" within a "vWorkstation" environment. The terminal displays the help output for the "shalsum" command, which includes options like --status, --warn, --strict, --help, and --version. Below the help text, a paragraph explains that sums are computed as described in FIPS-180-1 and that the default mode is to print a line with checksum, a character indicating type (* for binary, ' for text), and name for each FILE. The terminal then shows a series of commands and their outputs: reporting shalsum bugs, getting GNU coreutils help, running 'info coreutils shalsum invocation', calculating the sha1 checksum for 'Example.txt', listing files, displaying the contents of 'Example.txt.sha1', verifying the checksum with '-c', echoing the filename 'sindymorel', and finally calculating the md5sum and shalsum for 'Example.txt'.

```
Applying Encryption and Hashing Algorithms for Secure Communications
vWorkstation
2023-09-23 14:13:12
Sindy Morel

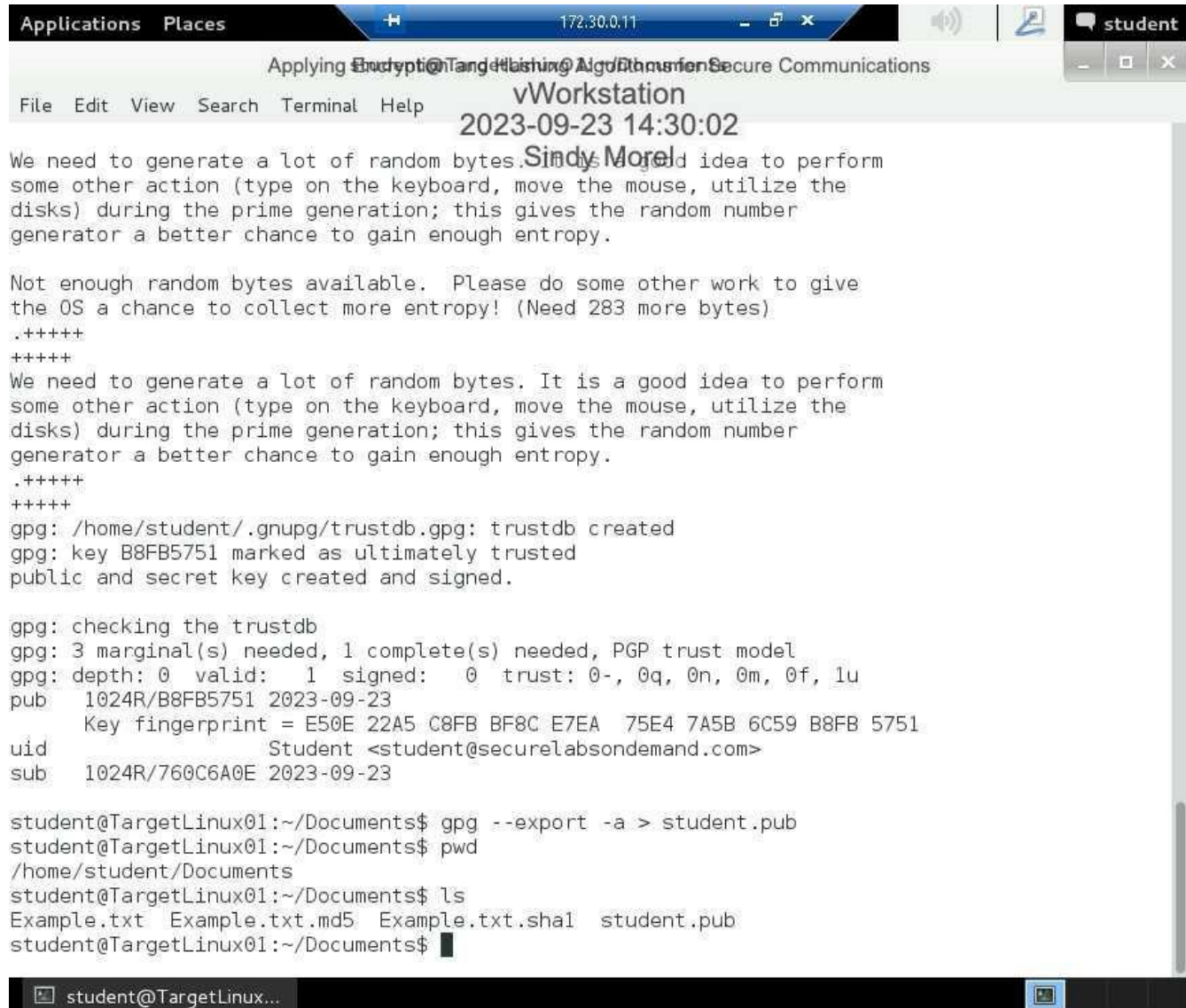
File Edit View Search Terminal Help
--status      don't output anything, status code shows success
-w, --warn    warn about improper or truncated checksum lines

--strict      with --check, exit non-zero for any invalid input
--help        display this help and exit
--version     output version information and exit

The sums are computed as described in FIPS-180-1.  When checking, the input
should be a former output of this program.  The default mode is to print
a line with checksum, a character indicating type (* for binary, ' for
text), and name for each FILE.

Report shalsum bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'shalsum invocation'
student@TargetLinux01:~/Documents$ shalsum Example.txt
a6f153801c9303d73ca2b43d3be62f44c6b66476 Example.txt
student@TargetLinux01:~/Documents$ shalsum Example.txt > Example.txt.sha1
student@TargetLinux01:~/Documents$ ls
Example.txt Example.txt.md5 Example.txt.sha1
student@TargetLinux01:~/Documents$ cat Example.txt.sha1
a6f153801c9303d73ca2b43d3be62f44c6b66476 Example.txt
student@TargetLinux01:~/Documents$ shalsum -c Example.txt.sha1
Example.txt: OK
student@TargetLinux01:~/Documents$ echo sindymorel >> Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example.
sindymorel
student@TargetLinux01:~/Documents$ md5sum Example.txt
eal04498baee238b9ea83c16599eceae Example.txt
student@TargetLinux01:~/Documents$ shalsum Example.txt
3fba1e4f37934b3186437754a5e991103e47b71d Example.txt
student@TargetLinux01:~/Documents$
```

Part 4: Generate GnuPG Keys



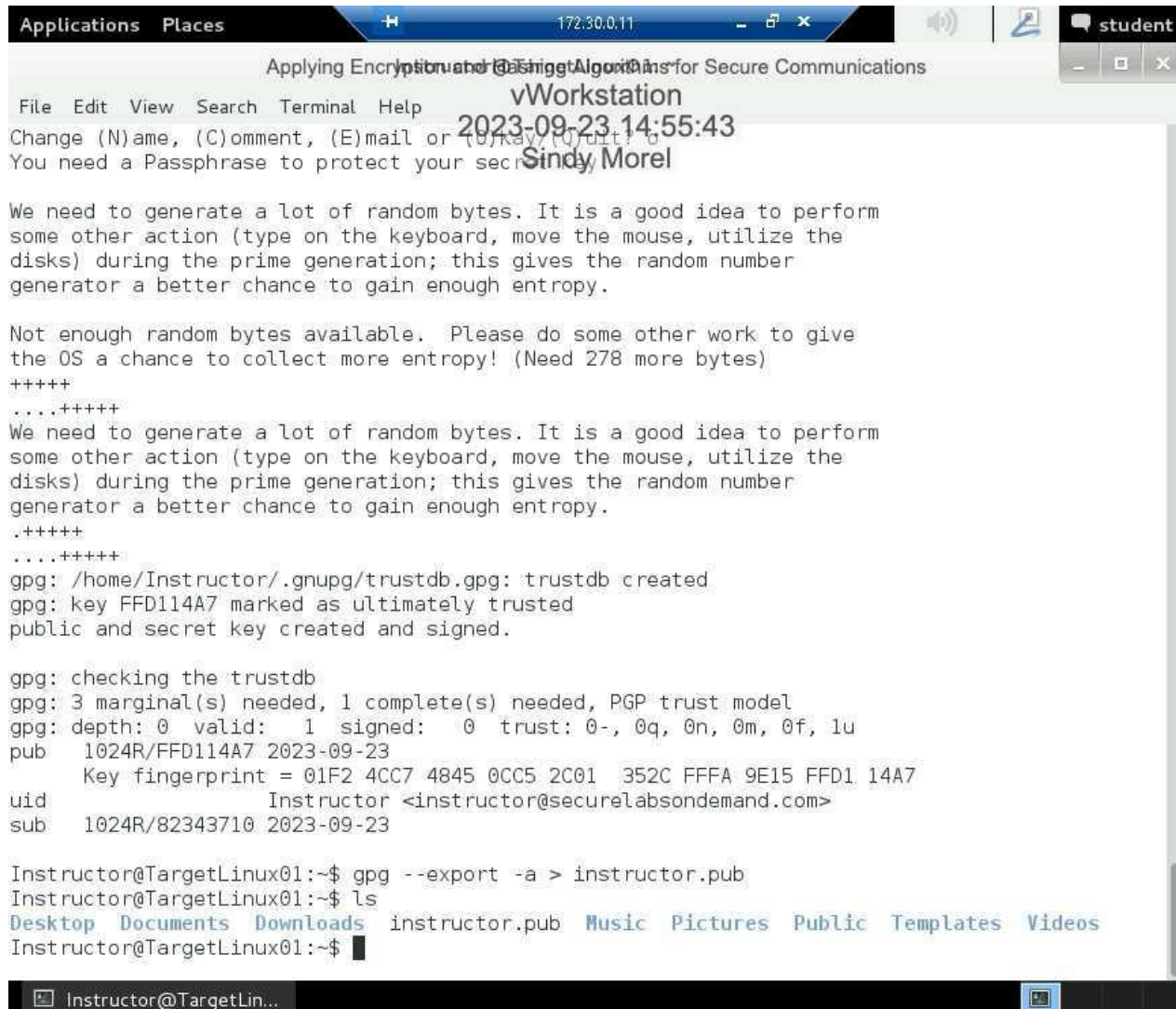
The screenshot shows a terminal window titled "vWorkstation" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar at the bottom showing "student@TargetLinux...". The terminal output is as follows:

```
2023-09-23 14:30:02
Sindy Morel
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.

Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 283 more bytes)
.+++++
+++++
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: /home/student/.gnupg/trustdb.gpg: trustdb created
gpg: key B8FB5751 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 1024R/B8FB5751 2023-09-23
    Key fingerprint = E50E 22A5 C8FB BF8C E7EA 75E4 7A5B 6C59 B8FB 5751
uid                               Student <student@securelabsondemand.com>
sub 1024R/760C6A0E 2023-09-23

student@TargetLinux01:~/Documents$ gpg --export -a > student.pub
student@TargetLinux01:~/Documents$ pwd
/home/student/Documents
student@TargetLinux01:~/Documents$ ls
Example.txt Example.txt.md5 Example.txt.sh1 student.pub
student@TargetLinux01:~/Documents$
```



```
Applications Places 172.30.0.11 student
Applying Encryption and Hashing Algorithms for Secure Communications
vWorkstation
2023-09-23 14:55:43
Sindy Morel

File Edit View Search Terminal Help
Change (N)ame, (C)omment, (E)mail or (U)RL, (O)utput
You need a Passphrase to protect your secret key.

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.

Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 278 more bytes)
+++++
....+++++
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: /home/Instructor/.gnupg/trustdb.gpg: trustdb created
gpg: key FFD114A7 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 1024R/FFD114A7 2023-09-23
    Key fingerprint = 01F2 4CC7 4845 0CC5 2C01 352C FFFA 9E15 FFD1 14A7
uid      Instructor <instructor@securelabsondemand.com>
sub 1024R/82343710 2023-09-23

Instructor@TargetLinux01:~$ gpg --export -a > instructor.pub
Instructor@TargetLinux01:~$ ls
Desktop Documents Downloads instructor.pub Music Pictures Public Templates Videos
Instructor@TargetLinux01:~$
```


Part 5: Share a GnuPG Key



The screenshot shows a terminal window titled "vWorkstation" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar at the bottom displaying "student@TargetLinux...". The terminal output shows the following commands and results:

```
options are given and SOURCE and DEST are the same name for an existing,
regular file.

Report cp bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
For complete documentation, run: info coreutils 'cp invocation'
student@TargetLinux01:~/Documents$ cp /home/Instructor/instructor.pub /home/student/Documents/
instructor.pub
student@TargetLinux01:~/Documents$ ls
Example.txt Example.txt.md5 Example.txt.shal instructor.pub student.pub
student@TargetLinux01:~/Documents$ gpg --list-keys
/home/student/.gnupg/pubring.gpg
-----
pub   1024R/B8FB5751 2023-09-23
uid           Student <student@securelabsondemand.com>
sub   1024R/760C6A0E 2023-09-23

student@TargetLinux01:~/Documents$ gpg --import instructor.pub
gpg: key FFD114A7: public key "Instructor <instructor@securelabsondemand.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
student@TargetLinux01:~/Documents$ gpg --list-keys
/home/student/.gnupg/pubring.gpg
-----
pub   1024R/B8FB5751 2023-09-23
uid           Student <student@securelabsondemand.com>
sub   1024R/760C6A0E 2023-09-23

pub   1024R/FFD114A7 2023-09-23
uid           Instructor <instructor@securelabsondemand.com>
sub   1024R/82343710 2023-09-23

student@TargetLinux01:~/Documents$
```


Part 6: Encrypt and Decrypt a ClearText Message

The screenshot shows a terminal window titled "vWorkstation" with a menu bar (File, Edit, View, Search, Terminal, Help) and a toolbar. The terminal output lists file permissions for various system files and directories, including .dbus, Desktop, Documents, Downloads, .gconf, .gnome2, .gnupg, .gtk-bookmarks, .gvfs, .ICEauthority, instructor.pub, .local, .mission-control, Music, Pictures, .profile, Public, .pulse, .pulse-cookie, Templates, Videos, and .xsession-errors. The user "Instructor" is prompted to enter a password and then runs the command "gpg -d cleartext.txt.gpg". The output shows the decryption process, including the passphrase "Instructor <instructor@securelabsondemand.com>" and the decrypted message "this is a clear-text message from sindymorel".

```

drwx----- 3 Instructor Instructor 4096 Nov 23 2020 .dbus
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Desktop
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Documents
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Downloads
drwx----- 3 Instructor Instructor 4096 Nov 23 2020 .gconf
drwx----- 3 Instructor Instructor 4096 Nov 23 2020 .gnome2
drwx----- 2 Instructor Instructor 4096 Sep 23 11:45 .gnupg
-rw-r--r-- 1 Instructor Instructor 162 Nov 23 2020 .gtk-bookmarks
drwx----- 2 Instructor Instructor 4096 Nov 23 2020 .gvfs
-rw----- 1 Instructor Instructor 346 Nov 23 2020 .ICEauthority
-rw-r--r-- 1 Instructor Instructor 1037 Sep 23 11:55 instructor.pub
drwxr-xr-x 3 Instructor Instructor 4096 Nov 23 2020 .local
drwx----- 3 Instructor Instructor 4096 Nov 23 2020 .mission-control
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Music
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Pictures
-rw-r--r-- 1 Instructor Instructor 675 Mar 27 2017 .profile
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Public
drwx----- 2 Instructor Instructor 4096 Nov 23 2020 .pulse
-rw----- 1 Instructor Instructor 256 Nov 23 2020 .pulse-cookie
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Templates
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Videos
-rw----- 1 Instructor Instructor 5991 Nov 23 2020 .xsession-errors
Instructor@TargetLinux01:~$ su Instructor
Password:
Instructor@TargetLinux01:~$ gpg -d cleartext.txt.gpg

You need a passphrase to unlock the secret key for
user: "Instructor <instructor@securelabsondemand.com>"
1024-bit RSA key, ID 82343710, created 2023-09-23 (main key ID FFD114A7)

gpg: encrypted with 1024-bit RSA key, ID 82343710, created 2023-09-23
      "Instructor <instructor@securelabsondemand.com>"
this is a clear-text message from sindymorel
Instructor@TargetLinux01:~$

```

Sindy Morel

LAB7: Applying Encryption and Hashing Algorithms for Secure Communications

September 23, 2023

```
Applications Places 172.30.0.11 vWorkstation
Applying Encryption and Hashing Algorithms for Secure Communications
File Edit View Search Terminal Help
text.txt
student@TargetLinux01:~/Documents$ cat cleartext.txt
this is a clear-text message from sindymorel
student@TargetLinux01:~/Documents$ gpg -e cleartext.txt
You did not specify a user ID. (you may use "-r")

Current recipients:

Enter the user ID. End with an empty line: Instructor
gpg: 82343710: There is no assurance this key belongs to the named user

pub 1024R/82343710 2023-09-23 Instructor <instructor@securelabsondemand.com>
Primary key fingerprint: 01F2 4CC7 4845 0CC5 2C01 352C FFFA 9E15 FFD1 14A7
Subkey fingerprint: 58DB DF81 72D4 09A1 8FE7 05AF 45B3 4665 8234 3710

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

Current recipients:
1024R/82343710 2023-09-23 "Instructor <instructor@securelabsondemand.com>"

Enter the user ID. End with an empty line:
student@TargetLinux01:~/Documents$ ls
cleartext.txt      Example.txt      Example.txt.sha1  student.pub
cleartext.txt.gpg  Example.txt.md5  instructor.pub
student@TargetLinux01:~/Documents$ cat cleartext.txt.gpg
00000000Fe0470000000N\000000k001000p0f0(1000(000'0[0
                                , 'u000000100D/.00m0p0t000000;0000>0t0(90000500
000hs"0005R0F00o30000c00000000^000000n00enet ~}X000Zq00q0LlK00!0V00000000000yG00%0000GF00o
000k@000000[Z0t0p0000=]C#000000Q0H00000
0000Astudent@TargetLinux01:~/Documents$
```

The screenshot shows a vWorkstation window titled "vWorkstation" with a terminal running. The terminal output lists files and directories with permissions, owner, group, size, date, and name. The files are sorted by date. The terminal then shows the user switching to 'Instructor' and running the command 'gpg -d cleartext.txt.gpg'. The output indicates that the file was encrypted with a 1024-bit RSA key, ID 82343710, created on 2023-09-23. The message is: "this is a clear-text message from sindymorel".

```

Applications Places 172.30.0.11 student
Browse and run installed applications Encrypted and Hashing Algorithms for Secure Communications
vWorkstation
2023-09-23-15:36:10
Sindy Morel
File Edit View Search Terminal Help
drwx----- 3 Instructor Instructor 4096 Nov 23 2020 .dbus
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Desktop
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Documents
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Downloads
drwx----- 3 Instructor Instructor 4096 Nov 23 2020 .gconf
drwx----- 3 Instructor Instructor 4096 Nov 23 2020 .gnome2
drwx----- 2 Instructor Instructor 4096 Sep 23 11:45 .gnupg
-rw-r--r-- 1 Instructor Instructor 162 Nov 23 2020 .gtk-bookmarks
drwx----- 2 Instructor Instructor 4096 Nov 23 2020 .gvfs
-rw----- 1 Instructor Instructor 346 Nov 23 2020 .ICEauthority
-rw-r--r-- 1 Instructor Instructor 1037 Sep 23 11:55 instructor.pub
drwxr-xr-x 3 Instructor Instructor 4096 Nov 23 2020 .local
drwx----- 3 Instructor Instructor 4096 Nov 23 2020 .mission-control
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Music
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Pictures
-rw-r--r-- 1 Instructor Instructor 675 Mar 27 2017 .profile
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Public
drwx----- 2 Instructor Instructor 4096 Nov 23 2020 .pulse
-rw----- 1 Instructor Instructor 256 Nov 23 2020 .pulse-cookie
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Templates
drwxr-xr-x 2 Instructor Instructor 4096 Nov 23 2020 Videos
-rw----- 1 Instructor Instructor 5991 Nov 23 2020 .xsession-errors
Instructor@TargetLinux01:~$ su Instructor
Password:
Instructor@TargetLinux01:~$ gpg -d cleartext.txt.gpg

You need a passphrase to unlock the secret key for
user: "Instructor <instructor@securelabsondemand.com>"
1024-bit RSA key, ID 82343710, created 2023-09-23 (main key ID FFD114A7)

gpg: encrypted with 1024-bit RSA key, ID 82343710, created 2023-09-23
      "Instructor <instructor@securelabsondemand.com>"
this is a clear-text message from sindymorel
Instructor@TargetLinux01:~$

```