

# MARYMOUNT UNIVERSITY

**Assignment:** IT557; Monitoring, Auditing, and Penetration Testing

**Assigned:** Sep. 16, 2018

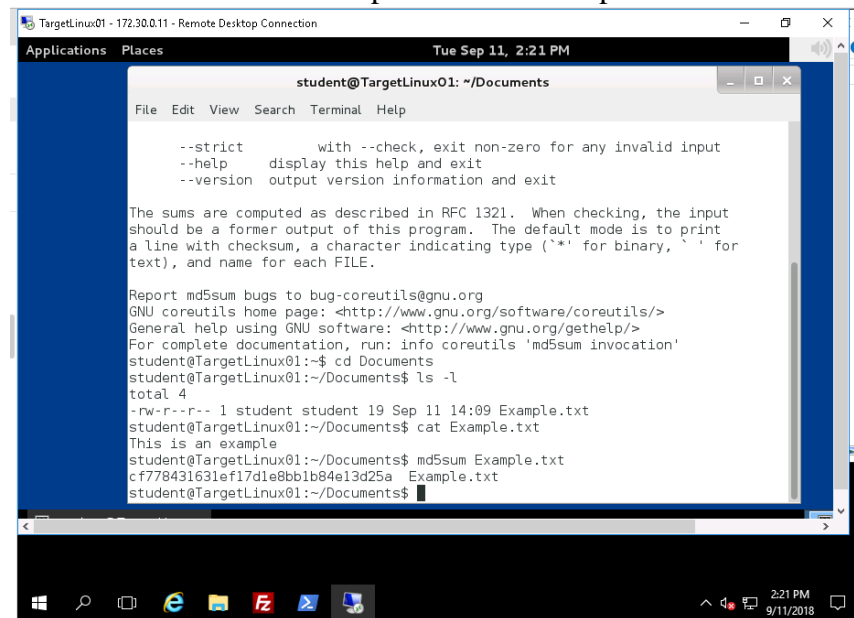
**Instructor:** Professor Ali Bicak

**Student Name:** George Boakye

## LAB REPORT FILE (LAB2)

### SECTION 1

#### Part 2: Step7: md5 hash Output



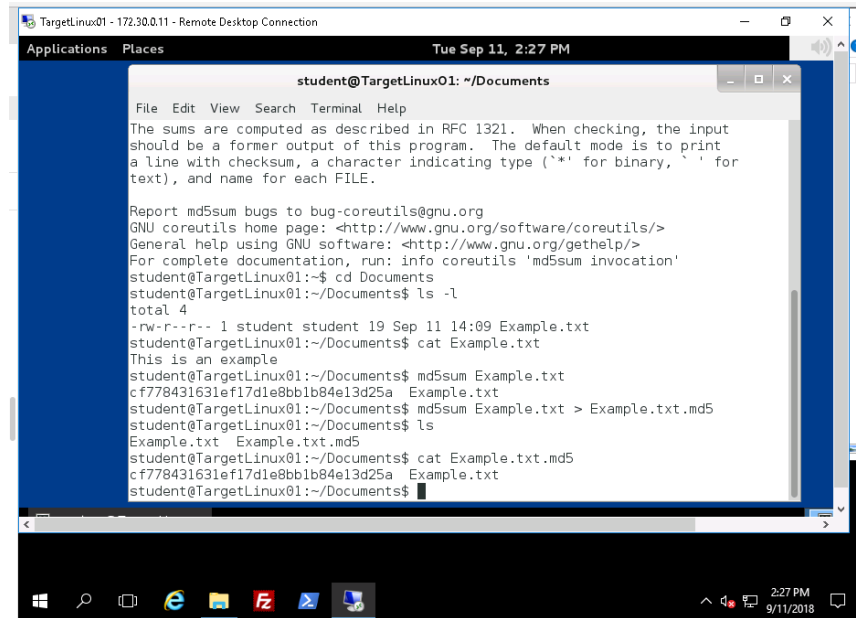
The screenshot shows a remote desktop connection to a machine named 'TargetLinux01' with IP '172.30.0.11'. The window title is 'TargetLinux01 - 172.30.0.11 - Remote Desktop Connection'. The desktop environment is Linux, with a terminal window open in the user's home directory. The terminal shows the output of the 'md5sum' command on a file named 'Example.txt'. The output is a single line: 'cf778431631ef17d1e8bb1b84e13d25a Example.txt'. The terminal also shows the command 'ls -l' which lists the file 'Example.txt' with permissions '-rw-r--r--', owner 'student', group 'student', size '19', and date 'Sep 11 14:09'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The desktop background is blue. The taskbar at the bottom shows the Windows logo, a search icon, and several application icons. The system clock in the bottom right corner shows '2:21 PM' and '9/11/2018'.

```
student@TargetLinux01: ~/Documents
--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 19 Sep 11 14:09 Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example
student@TargetLinux01:~/Documents$ md5sum Example.txt
cf778431631ef17d1e8bb1b84e13d25a Example.txt
student@TargetLinux01:~/Documents$
```

## Part 2: Step11: Example.txt.md5 Output



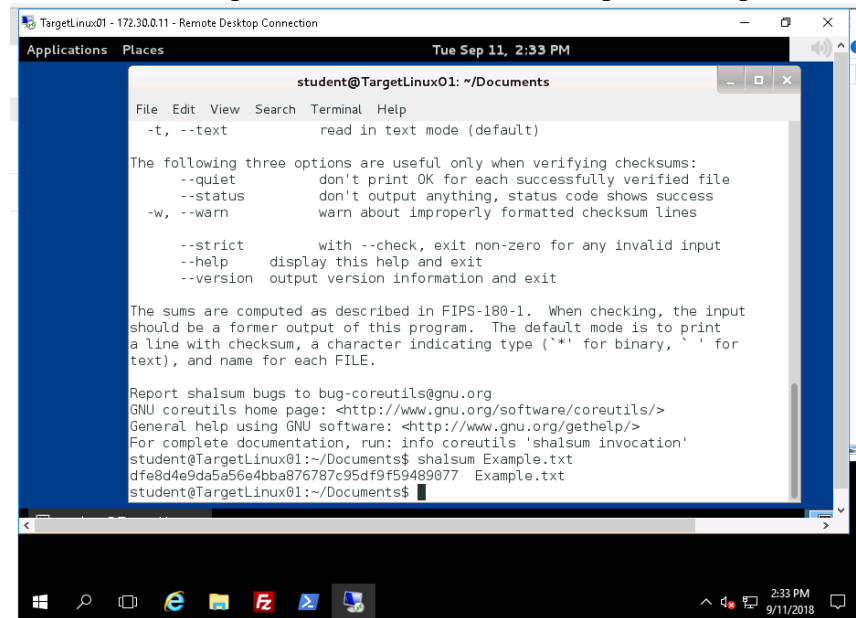
The screenshot shows a terminal window titled "student@TargetLinux01: ~/Documents". The terminal displays the output of the `md5sum` command. It starts with a header explaining the format of the output, followed by the command `md5sum Example.txt` which produces the output `cf778431631ef17d1e8bb1b84e13d25a Example.txt`. The user then runs `md5sum Example.txt > Example.txt.md5` to save the output to a file. Finally, the user runs `cat Example.txt.md5` to display the contents of the file, which shows the same checksum and filename.

```
student@TargetLinux01: ~/Documents
File Edit View Search Terminal Help

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:~/Documents$ cd Documents
student@TargetLinux01:~/Documents$ ls -l
total 4
-rw-r--r-- 1 student student 19 Sep 11 14:09 Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example
student@TargetLinux01:~/Documents$ md5sum Example.txt
cf778431631ef17d1e8bb1b84e13d25a 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
cf778431631ef17d1e8bb1b84e13d25a Example.txt
student@TargetLinux01:~/Documents$
```

## Part 2: Step15: sha1sum hash for Example.txt Output



The screenshot shows a terminal window titled "student@TargetLinux01: ~/Documents". The terminal displays the output of the `sha1sum` command. It starts with a header explaining the format of the output, followed by the command `sha1sum Example.txt` which produces the output `dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt`. The user then runs `sha1sum Example.txt` again to verify the output.

```
student@TargetLinux01: ~/Documents
File Edit View Search Terminal Help

-t, --text          read in text mode (default)

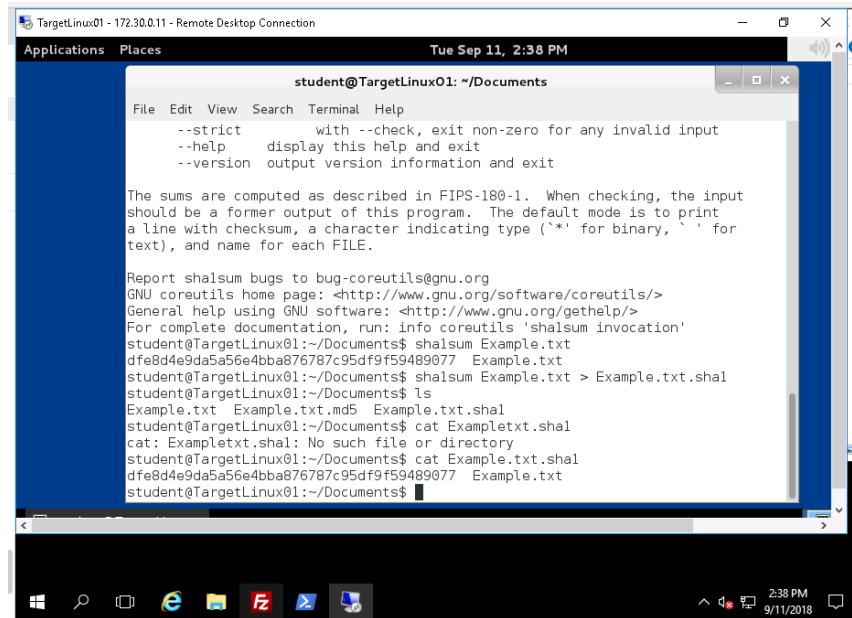
The following three options are useful only when verifying checksums:
--quiet             don't print OK for each successfully verified file
--status            don't output anything, status code shows success
-w, --warn          warn about improperly formatted checksum lines

--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 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 sha1sum 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 'sha1sum invocation'
student@TargetLinux01:~/Documents$ sha1sum Example.txt
dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt
student@TargetLinux01:~/Documents$ sha1sum Example.txt
dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt
student@TargetLinux01:~/Documents$
```

## Part 2: Step19: Example.txt.shal Output



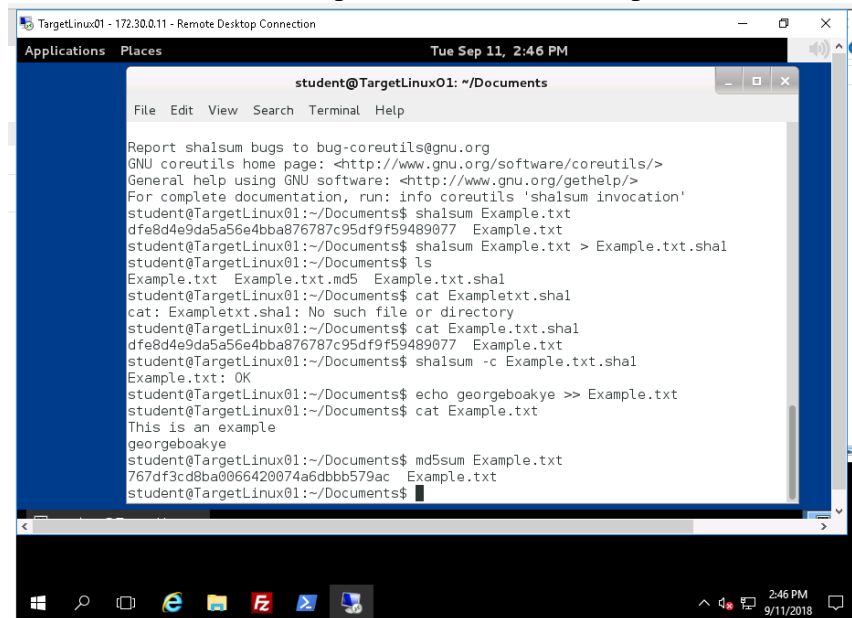
The screenshot shows a remote desktop connection to a machine named 'TargetLinux01' with IP '172.30.0.11'. The window title is 'TargetLinux01 - 172.30.0.11 - Remote Desktop Connection'. The desktop environment is Linux, with a terminal window open. The terminal window title is 'student@TargetLinux01: ~/Documents'. The terminal output shows the help for 'shalsum', which is a utility for computing and verifying checksums. The output includes the following text:

```
File Edit View Search Terminal Help
--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
dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt
student@TargetLinux01:~/Documents$ shalsum Example.txt > Example.txt.shal
student@TargetLinux01:~/Documents$ ls
Example.txt Example.txt.md5 Example.txt.shal
student@TargetLinux01:~/Documents$ cat Example.txt.shal
cat: Example.txt.shal: No such file or directory
student@TargetLinux01:~/Documents$ cat Example.txt.shal
dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt
student@TargetLinux01:~/Documents$
```

## Part 3: Step4: New md5 hash Output

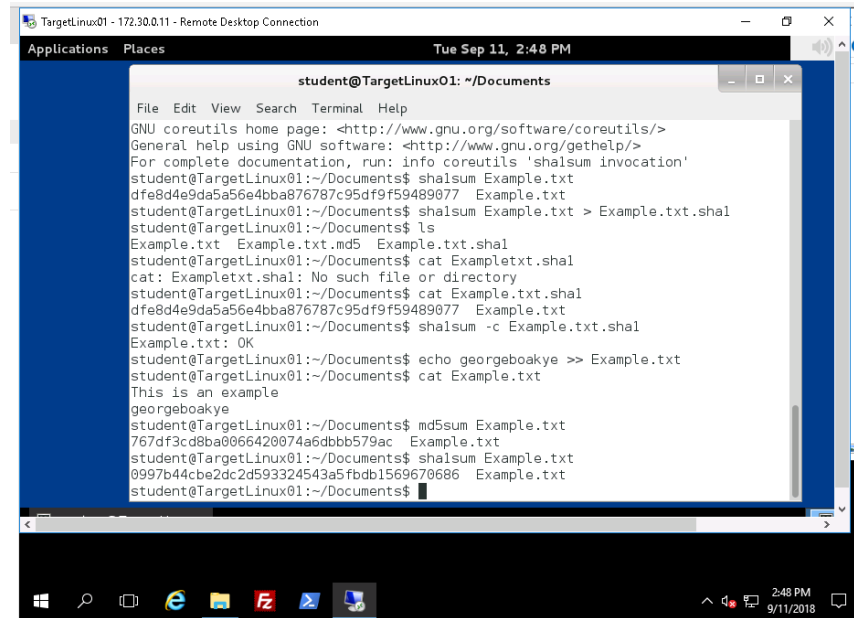


The screenshot shows a remote desktop connection to a machine named 'TargetLinux01' with IP '172.30.0.11'. The window title is 'TargetLinux01 - 172.30.0.11 - Remote Desktop Connection'. The desktop environment is Linux, with a terminal window open. The terminal window title is 'student@TargetLinux01: ~/Documents'. The terminal output shows the help for 'shalsum', which is a utility for computing and verifying checksums. The output includes the following text:

```
File Edit View Search Terminal Help

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
dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt
student@TargetLinux01:~/Documents$ shalsum Example.txt > Example.txt.shal
student@TargetLinux01:~/Documents$ ls
Example.txt Example.txt.md5 Example.txt.shal
student@TargetLinux01:~/Documents$ cat Example.txt.shal
cat: Example.txt.shal: No such file or directory
student@TargetLinux01:~/Documents$ cat Example.txt.shal
dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt
student@TargetLinux01:~/Documents$ shalsum -c Example.txt.shal
Example.txt: OK
student@TargetLinux01:~/Documents$ echo georgeboakye >> Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example
georgeboakye
student@TargetLinux01:~/Documents$ md5sum Example.txt
767df3cd8ba0066420074a6dbb579ac Example.txt
student@TargetLinux01:~/Documents$
```

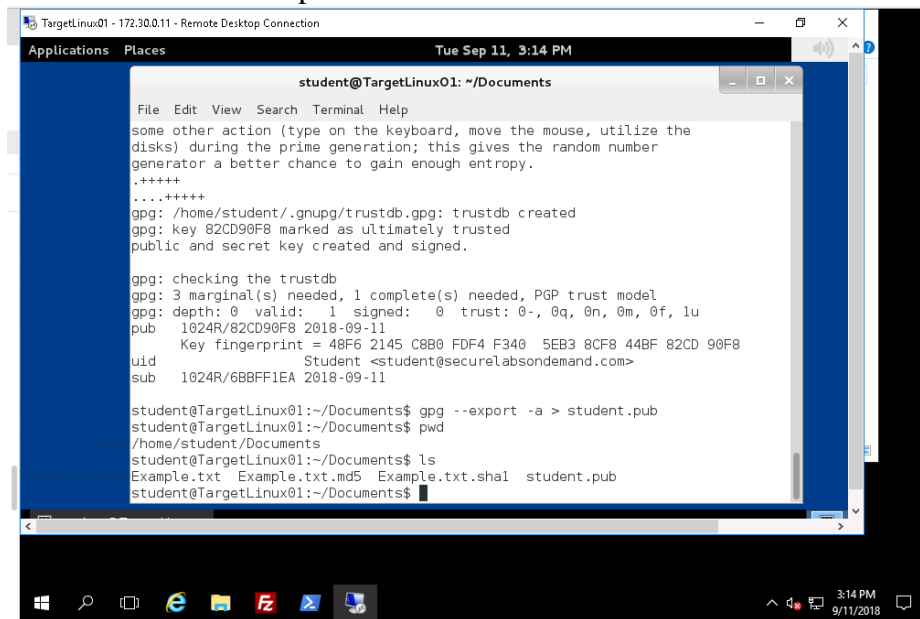
### Part 3: Step6: New sha1 hash Output



A terminal window titled 'student@TargetLinux01: ~/Documents' showing the following commands and output:

```
File Edit View Search Terminal Help
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
dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt
student@TargetLinux01:~/Documents$ shalsum Example.txt > Example.txt.shal
student@TargetLinux01:~/Documents$ ls
Example.txt Example.txt.md5 Example.txt.shal
student@TargetLinux01:~/Documents$ cat Example.txt.shal
cat: Example.txt.shal: No such file or directory
student@TargetLinux01:~/Documents$ cat Example.txt.shal
dfe8d4e9da5a56e4bba876787c95df9f59489077 Example.txt
student@TargetLinux01:~/Documents$ shalsum -c Example.txt.shal
Example.txt: OK
student@TargetLinux01:~/Documents$ echo georgeboakye >> Example.txt
student@TargetLinux01:~/Documents$ cat Example.txt
This is an example
georgeboakye
student@TargetLinux01:~/Documents$ md5sum Example.txt
767df3cd8ba0066420074a6dbb579ac Example.txt
student@TargetLinux01:~/Documents$ shalsum Example.txt
0997b44cbe2dc2d59324543a5fbd1569670686 Example.txt
student@TargetLinux01:~/Documents$
```

### Part 4: Step13: /home/student/documents folder



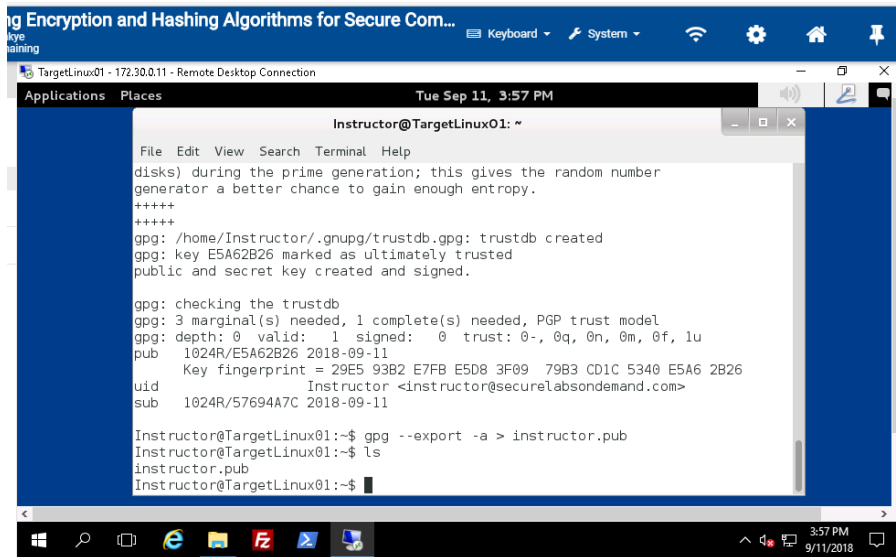
A terminal window titled 'student@TargetLinux01: ~/Documents' showing the following commands and output:

```
File Edit View Search Terminal Help
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 82CD90F8 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/82CD90F8 2018-09-11
Key fingerprint = 40F6 2145 C8B0 FDF4 F340 5EB3 8CF8 44BF 82CD 90F8
uid Student <student@securelabsondemand.com>
sub 1024R/6BBFF1EA 2018-09-11

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.shal student.pub
student@TargetLinux01:~/Documents$
```

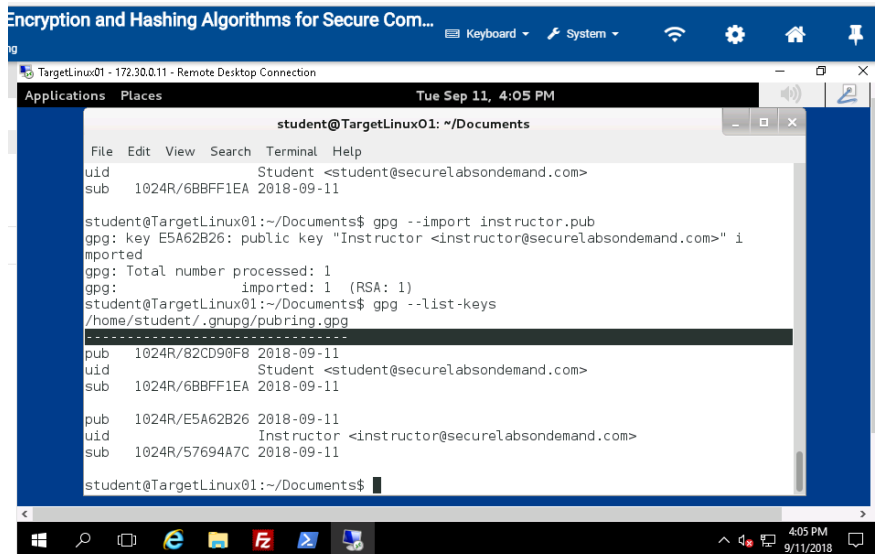
## Part 4: Step 21: instructor folder



The screenshot shows a terminal window titled "Instructor@TargetLinux01: ~". The user is running GPG commands to generate a key pair, check the trust database, and export the public key. The output shows a key with ID 1024R/E5A62B26, created on 2018-09-11, with a fingerprint of 29E5 93B2 E7FB E5D8 3F09 79B3 CD1C 5340 E5A6 2B26. The key is marked as "ultimately trusted". The user then exports the public key to "instructor.pub".

```
Instructor@TargetLinux01: ~  
File Edit View Search Terminal Help  
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 E5A62B26 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/E5A62B26 2018-09-11  
Key fingerprint = 29E5 93B2 E7FB E5D8 3F09 79B3 CD1C 5340 E5A6 2B26  
uid Instructor <instructor@securelabsondemand.com>  
sub 1024R/57694A7C 2018-09-11  
  
Instructor@TargetLinux01:~$ gpg --export -a > instructor.pub  
Instructor@TargetLinux01:~$ ls  
instructor.pub  
Instructor@TargetLinux01:~$
```

## Part 5: Step 6: Exchange PGP Keys



The screenshot shows a terminal window titled "student@TargetLinux01: ~/Documents". The user is importing the instructor's public key and then listing the keys. The output shows the imported key with ID 1024R/E5A62B26, created on 2018-09-11, with a fingerprint of 29E5 93B2 E7FB E5D8 3F09 79B3 CD1C 5340 E5A6 2B26. The key is marked as "ultimately trusted". The user then lists the keys, showing both the student's and instructor's keys.

```
student@TargetLinux01: ~/Documents  
File Edit View Search Terminal Help  
uid Student <student@securelabsondemand.com>  
sub 1024R/6BBFF1EA 2018-09-11  
  
student@TargetLinux01:~/Documents$ gpg --import instructor.pub  
gpg: key E5A62B26: public key "Instructor <instructor@securelabsondemand.com>" i  
mported  
gpg: Total number processed: 1  
gpg: imported: 1 (RSA: 1)  
student@TargetLinux01:~/Documents$ gpg --list-keys  
/home/student/.gnupg/pubring.gpg  
-----  
pub 1024R/82CD90F8 2018-09-11  
uid Student <student@securelabsondemand.com>  
sub 1024R/6BBFF1EA 2018-09-11  
  
pub 1024R/E5A62B26 2018-09-11  
uid Instructor <instructor@securelabsondemand.com>  
sub 1024R/57694A7C 2018-09-11  
  
student@TargetLinux01:~/Documents$
```

[illegible]

The screenshot shows a Windows desktop with a blue taskbar at the bottom. The main window is titled "Encryption and Hashing Algorithms for Secure Com..." and shows a remote desktop connection to "TargetLinux01 - 172.30.0.11 - Remote Desktop Connection". The terminal window, titled "Instructor@TargetLinux01: ~", shows the following commands and output:

```

File Edit View Search Terminal Help
-rw-r--r-- 1 Instructor Instructor 1037 Sep 11 15:56 instructor.pub
-rw-r--r-- 1 Instructor Instructor 675 Mar 27 2017 .profile
student@TargetLinux01:/home/Instructor$ 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 57694A7C, created 2018-09-11 (main key ID E5A62B26)

gpg: Invalid passphrase; please try again ...

You need a passphrase to unlock the secret key for
user: "Instructor <instructor@securelabsondemand.com>"
1024-bit RSA key, ID 57694A7C, created 2018-09-11 (main key ID E5A62B26)

gpg: encrypted with 1024-bit RSA key, ID 57694A7C, created 2018-09-11
      "Instructor <instructor@securelabsondemand.com>"
this is a clear-text message from georgeboakye
Instructor@TargetLinux01:~$
  
```

The Windows taskbar at the bottom shows the Start button, search icon, task view icon, and several application icons (Edge, File Explorer, Firefox, Mail, etc.). The system tray in the bottom right corner shows the time as 4:18 PM on 9/11/2018.

## SECTION 2

### Part 2: Step 6: Example2.txt.md5

```
Applying Encryption and Hashing Algorithms for Secure Com...
george Boakye
5 minutes remaining

student@TargetLinux01: ~/Documents
student@TargetLinux01:~/Documents$ ls
clear.txt      Example.txt      Example.txt.sha1  instructor.pub  student2.pub
Example2.txt    Example.txt.md5  instructor2.pub   shasum         student.pub
student@TargetLinux01:~/Documents$ cat Example2.txt
This file is from georgeboakye
student@TargetLinux01:~/Documents$ md5sum Example2.txt
9b537607f1f9c0838bc0df13f8f51a0b  Example2.txt
student@TargetLinux01:~/Documents$ md5sum Example2.txt > Example2.txt.md5
student@TargetLinux01:~/Documents$ ls
clear.txt      Example2.txt.md5  Example.txt.md5   instructor2.pub  shasum         student.pub
Example2.txt    Example.txt       Example.txt.sha1  instructor.pub   student2.pub
student@TargetLinux01:~/Documents$ cat Example2.txt.md5
9b537607f1f9c0838bc0df13f8f51a0b  Example2.txt
student@TargetLinux01:~/Documents$
```

### Part 2: Step 14: Example2.txt.sha256

```
Applying Encryption and Hashing Algorithms for Secure Com...
george Boakye
2 minutes remaining

student@TargetLinux01: ~/Documents
student@TargetLinux01:~/Documents$ ls
clear.txt      Example.txt      Example.txt.sha1  instructor.pub  student2.pub
Example2.txt    Example.txt.md5  instructor2.pub   shasum         student.pub
student@TargetLinux01:~/Documents$ cat Example2.txt
This file is from georgeboakye
student@TargetLinux01:~/Documents$ md5sum Example2.txt
9b537607f1f9c0838bc0df13f8f51a0b  Example2.txt
student@TargetLinux01:~/Documents$ md5sum Example2.txt > Example2.txt.md5
student@TargetLinux01:~/Documents$ ls
clear.txt      Example2.txt.md5  Example.txt.md5   instructor2.pub  shasum         student.pub
Example2.txt    Example.txt       Example.txt.sha1  instructor.pub   student2.pub
student@TargetLinux01:~/Documents$ cat Example2.txt.md5
9b537607f1f9c0838bc0df13f8f51a0b  Example2.txt
student@TargetLinux01:~/Documents$ md5sum -c Example2.txt.md5
Example2.txt: OK
student@TargetLinux01:~/Documents$ sha256sum Example2.txt
6c1a412fcc054722e8ba8e44abc8086a6f7943e8f97786acc0e245eb9eb13f71  Example2.txt
student@TargetLinux01:~/Documents$ sha256sum Example2.txt > Example2.txt.sha256
student@TargetLinux01:~/Documents$ ls
clear.txt      Example2.txt.md5  Example.txt       Example.txt.sha1  instructor.pub  student2.pub
Example2.txt    Example2.txt.sha256  Example.txt.md5  instructor2.pub   shasum         student.pub
student@TargetLinux01:~/Documents$ cat Example2.txt.sha256
6c1a412fcc054722e8ba8e44abc8086a6f7943e8f97786acc0e245eb9eb13f71  Example2.txt
student@TargetLinux01:~/Documents$
```

### Part 3: Step 7: Modified MD5suma and SHA256 hash

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
3 minutes remaining

student@TargetLinux01: ~/Documents
student@TargetLinux01:~/Documents$ ls
clear.txt.txt  Example2.txt.md5  Example.txt.md5  instructor2.pub  shasum  student.pub
Example2.txt  Example.txt  Example.txt.sha1  instructor2.pub  student2.pub
student@TargetLinux01:~/Documents$ cat Example2.txt.md5
9b537607f1f9c0838bc0df13f8f51a0b  Example2.txt
student@TargetLinux01:~/Documents$ md5sum -c Example2.txt.md5
Example2.txt: OK
student@TargetLinux01:~/Documents$ sha256sum Example2.txt
6c1a412fcc054722e8ba8e44abc8086a6f7943e8f97786acc0e245eb9eb13f71  Example2.txt
student@TargetLinux01:~/Documents$ sha256sum Example2.txt > Example2.txt.sha256
student@TargetLinux01:~/Documents$ ls
clear.txt.txt  Example2.txt.md5  Example.txt  Example.txt.sha1  instructor2.pub  shasum  student2.pub
Example2.txt  Example2.txt.sha256  Example.txt.md5  instructor2.pub  shasum  student2.pub
student@TargetLinux01:~/Documents$ cat Example2.txt.sha256
6c1a412fcc054722e8ba8e44abc8086a6f7943e8f97786acc0e245eb9eb13f71  Example2.txt
student@TargetLinux01:~/Documents$ sha256sum -c Example2.txt.sha256
Example2.txt: OK
student@TargetLinux01:~/Documents$ cat Example2.txt
This file is from georgeboakye
student@TargetLinux01:~/Documents$ echo "This example is testing hash values." >> Example2.txt
student@TargetLinux01:~/Documents$ cat Example2.txt
This file is from georgeboakye
This example is testing hash values.
student@TargetLinux01:~/Documents$ md5sum Example2.txt > Example2.txt.md5
student@TargetLinux01:~/Documents$ sha256sum Example2.txt > Example2.txt.sha256
student@TargetLinux01:~/Documents$ cat Example2.txt.md5
f8441d2cde276168c2541192c0713fc0  Example2.txt
student@TargetLinux01:~/Documents$ cat Example2.txt.sha256
4ccc2bfa5dcbe0608482a1f50c437ea2b33af4dc3a36dda3decae886a01db330  Example2.txt
student@TargetLinux01:~/Documents$
```

### Part 4: Step 17: Instructor's public key ring

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
45 minutes remaining

Instructor@TargetLinux02: ~
.....+++++
Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 73 more bytes)
.....+++++
gpg: key 3A7E460B 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: 2  signed: 0  trust: 0-, 0q, 0n, 0m, 0f, 2u
pub 2048R/3A7E460B 2018-09-12
Key fingerprint = 7766 D8A2 1BDC 6A0F CD3A 51D9 0A1C 58D8 3A7E 460B
uid Instructor2 <instructor@securelabsondemand.com>
sub 2048R/05FB333D 2018-09-12

Instructor@TargetLinux02:~$ gpg --export -a > instructor2.pub
Instructor@TargetLinux02:~$ gpg --list-keys
/home/Instructor/.gnupg/pubring.gpg
-----
pub 2048R/38FD6AB2 2018-09-12
uid Instructor2 <instructor@securelabsondemand.com>
sub 2048R/524FC7EC 2018-09-12

pub 2048R/3A7E460B 2018-09-12
uid Instructor2 <instructor@securelabsondemand.com>
sub 2048R/05FB333D 2018-09-12

Instructor@TargetLinux02:~$
```



## Part 5: Step 12: Updated student's public key ring

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
36 minutes remaining

student@TargetLinux01: ~/Documents
gpg: unchanged: 1
student@TargetLinux01:~/Documents$ gpg --list-keys
/home/student/.gnupg/pubring.gpg
-----
pub   1024R/82CD90F8 2018-09-11
uid           Student <student@securelabsondemand.com>
sub   1024R/6BBFF1EA 2018-09-11

pub   1024R/E5A62B26 2018-09-11
uid           Instructor <instructor@securelabsondemand.com>
sub   1024R/57694A7C 2018-09-11

pub   2048R/82C49951 2018-09-11
uid           Student2 <student@securelabsondemand.com>
sub   2048R/A0B906ED 2018-09-11

pub   2048R/38FD6AB2 2018-09-12
uid           Instructor2 <instructor@securelabsondemand.com>
sub   2048R/524FC7EC 2018-09-12

pub   2048R/2627D45E 2018-09-12
uid           Student2 <student@securelabsondemand.com>
sub   2048R/736B7655 2018-09-12

pub   2048R/3A7E460B 2018-09-12
uid           Instructor2 <instructor@securelabsondemand.com>
sub   2048R/05FB333D 2018-09-12

student@TargetLinux01:~/Documents$
```

## Part 6: Step 19: Instructor decrypts message

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
23 minutes remaining

Instructor@TargetLinux02: ~
sub   2048R/524FC7EC 2018-09-12

pub   2048R/3A7E460B 2018-09-12
uid           Instructor2 <instructor@securelabsondemand.com>
sub   2048R/05FB333D 2018-09-12

Instructor@TargetLinux02:~$ ls -l
total 44
-rw-r--r-- 1 Instructor Instructor 256 Sep 11 16:10 cleartext.txt.gpg
drwxr-xr-x 2 Instructor Instructor 4096 Mar 27 2017 Desktop
drwxr-xr-x 2 Instructor Instructor 4096 Mar 27 2017 Documents
drwxr-xr-x 2 Instructor Instructor 4096 Mar 27 2017 Downloads
-rwxrwxrwx 1 Instructor Instructor 176 Apr 12 2017 entropy_loop.sh
-rw-r--r-- 1 Instructor Instructor 3360 Sep 12 15:02 instructor2.pub
drwxr-xr-x 2 Instructor Instructor 4096 Mar 27 2017 Music
drwxr-xr-x 2 Instructor Instructor 4096 Mar 27 2017 Pictures
drwxr-xr-x 2 Instructor Instructor 4096 Mar 27 2017 Public
drwxr-xr-x 2 Instructor Instructor 4096 Mar 27 2017 Templates
drwxr-xr-x 2 Instructor Instructor 4096 Mar 27 2017 Videos
Instructor@TargetLinux02:~$ gpg -d cleartext.txt.gpg

You need a passphrase to unlock the secret key for
user: "Instructor2 <instructor@securelabsondemand.com>"
2048-bit RSA key, ID 524FC7EC, created 2018-09-12 (main key ID 38FD6AB2)

gpg: encrypted with 2048-bit RSA key, ID 524FC7EC, created 2018-09-12
      "Instructor2 <instructor@securelabsondemand.com>"
This clear-text message is from georgeboakye
Instructor@TargetLinux02:~$
```

## SECTION 3

### Part 1

Differences between RSA and ECDSA encryption algorithms. Product that uses each type of encryption.

RSA and ECDSA are algorithms used by Asymmetric (public key) cryptographic systems to provide a mechanism for authentication (collectively called digital signature algorithms). They are mathematical problems that are complex and relatively simple to compute one way but impractical to reverse. Some ECDSA products are iPhone and iPad (Genkin, Pachmanov, & Pipman, 2016). RSA products include RSA SecureID Tokens, USB Flash Drives, and Smart Cards (RSA SecurID, 2018).

Here are the key differences (Naziridis, 2018):

**Standard maturity:** RSA was first standardized for SSL/TLS in 1994, while ECDSA was introduced in the specification of TLS v1.2 in 2008. This makes RSA extremely recognized and more used than ECDSA.

**Key-size to security-level ratio:** An RSA 2048-bit public key provides 112 bits security level bits while ECDSA requires only 224-bit sized public keys to provide same 112-bit security level. The implication is smaller key sizes such as ECDSA (224-bit) require less bandwidth to set up an SSL/TLS stream. This makes ECDSA certificates ideal for mobile applications.

**Performance & time complexity:** During implementations, RSA seems to be significantly faster than ECDSA in verifying signatures but slower in message signing.

## Part 2: A: Send.txt folder

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
hours remaining

student@TargetLinux01: ~/Documents
Using username "student".
Linux TargetLinux01 3.2.0-4-amd64 #1 SMP Debian 3.2.04-1 x86_64
#####
# Jones and Bartlett ISSA Labs #
# Hosted by: Hatsize #
# Created by: Security Centric #
#####

You have new mail.
Last login: Wed Sep 12 14:50:49 2018 from 172.30.0.2
student@TargetLinux01:~$ whoami
student
student@TargetLinux01:~$ pwd
/home/student
student@TargetLinux01:~$ cd /home/student/Documents
student@TargetLinux01:~/Documents$ echo "My name is George Boakye." > Send.txt
student@TargetLinux01:~/Documents$ cat Send.txt
My name is George Boakye.
student@TargetLinux01:~/Documents$
```

## Part 2: B: MD5 hash for Send.txt

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
hours remaining

student@TargetLinux01: ~/Documents
# Hosted by: Hatsize #
# Created by: Security Centric #
#####

You have new mail.
Last login: Wed Sep 12 14:50:49 2018 from 172.30.0.2
student@TargetLinux01:~$ whoami
student
student@TargetLinux01:~$ pwd
/home/student
student@TargetLinux01:~$ cd /home/student/Documents
student@TargetLinux01:~/Documents$ echo "My name is George Boakye." > Send.txt
student@TargetLinux01:~/Documents$ cat Send.txt
My name is George Boakye.
student@TargetLinux01:~/Documents$ ls
cleartext2.txt      Example2.txt      instructor2.pub    shasum
cleartext2.txt.gpg  Example2.txt.md5  Example.txt.md5   instructor.pub     student2.pub
cleartext.txt       Example2.txt.sha256 Example.txt.sha1   Send.txt          student.pub
student@TargetLinux01:~/Documents$ md5sum Send.txt
493b9bed9e227e45272392d2e64cab97  Send.txt
student@TargetLinux01:~/Documents$ md5sum Send.txt > Send.txt.md5
student@TargetLinux01:~/Documents$ ls
cleartext2.txt      Example2.txt      Example.txt      instructor2.pub    Send.txt.md5  student.pub
cleartext2.txt.gpg  Example2.txt.md5  Example.txt.md5  instructor.pub     shasum
cleartext.txt       Example2.txt.sha256 Example.txt.sha1  Send.txt          student2.pub
student@TargetLinux01:~/Documents$ cat Send.txt.md5
493b9bed9e227e45272392d2e64cab97  Send.txt
student@TargetLinux01:~/Documents$ md5sum -c Send.txt.md5
Send.txt: OK
student@TargetLinux01:~/Documents$
```

## Part 2: C: Shared Student public key with Instructor

```

Instructor@TargetLinux01:~$ gpg --import george.pub
gpg: key 82C090F8: public key "Student <student@securelabsondemand.com>" imported
gpg: key E5A62B26: "Instructor <instructor@securelabsondemand.com>" not changed
gpg: key 82C49951: public key "Student2 <student2@securelabsondemand.com>" imported
gpg: key 38FD6AB2: public key "Instructor2 <instructor2@securelabsondemand.com>" imported
gpg: key 2627D45E: public key "Student2 <student2@securelabsondemand.com>" imported
gpg: key 3A7E4608: public key "Instructor2 <instructor2@securelabsondemand.com>" imported
gpg: key D26D053C: public key "Student <student@securelabsondemand.com>" imported
gpg: key 3421DD81: public key "Student <student@securelabsondemand.com>" imported
gpg: Total number processed: 8
gpg:      imported: 7      (RSA: 7)
gpg:      unchanged: 1
Instructor@TargetLinux01:~$ gpg --list-keys
/home/Instructor/.gnupg/pubring.gpg
-----
pub      1024R/ESA62B26 2018-09-11
uid
sub      1024R/S76947AC 2018-09-11

pub      1024R/E901CB9A 2018-09-13
uid
sub      1024R/5EE0C639 2018-09-13

pub      1024R/877F65328 2018-09-13
uid
sub      1024R/05F6A3CA 2018-09-13

pub      1024R/82CD90F8 2018-09-11
uid
sub      1024R/6B8FF1EA 2018-09-11

pub      2048R/82C49951 2018-09-11
uid
sub      2048R/A0B906ED 2018-09-11

pub      2048R/38FD6AB2 2018-09-12

```

## Part 2: D: Encrypting Send.txt with Instructor Account

```
Instructor@TargetLinux01:~$ gpg --e Send.txt
You did not specify a user ID. (you may use "--r")

Current recipients:

Enter the user ID. End with an empty line: Student
gpg: 6BBFF1EA: There is no assurance this key belongs to the named user

pub 1024R/6BBFF1EA 2018-09-11 Student <student@securelabsondemand.com>
Primary key fingerprint: 48F6 2145 C8B0 FDF4 F340 5EB3 8CF8 4ABF 62CD 90F8
Subkey fingerprint: C467 F2BE B148 EE8A 2DC3 113D EB84 DD97 6BBF F1EA

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/6BBFF1EA 2018-09-11 "Student <student@securelabsondemand.com>"

Enter the user ID. End with an empty line:
Instructor@TargetLinux01:~$ ls
cleartext.txt.gpg george.pub instructor.pub nina.pub Send.txt Send.txt.gpg
Instructor@TargetLinux01:~$ cat Send.txt.gpg
P.L. <* V. 6L88R OX C格.* S E *> R S S b 明 a?Fip Uq?Sgq
C) Sd_
?[-+]]] !Y.v v p P k - -</]:
;c- +
q' Z l ~ P.V
X Instructor@TargetLinux01:~$ PuTTYPaTty
```

## Part 2: E: Decrypting Send.txt.gpg with Student Account

```
student@TargetLinux01: ~/Documents
gpg: encrypted with 1024-bit RSA key, ID 6BBFF1EA, created 2018-09-11
"Student <student@securelabsondemand.com>"
gpg: public key decryption failed: bad passphrase
gpg: decryption failed: secret key not available
student@TargetLinux01:~/Documents$ gpg -d Send.txt.gpg

You need a passphrase to unlock the secret key for
user: "Student <student@securelabsondemand.com>"
1024-bit RSA key, ID 6BBFF1EA, created 2018-09-11 (main key ID 82CD90F8)

gpg: Invalid passphrase; please try again ...

You need a passphrase to unlock the secret key for
user: "Student <student@securelabsondemand.com>"
1024-bit RSA key, ID 6BBFF1EA, created 2018-09-11 (main key ID 82CD90F8)

gpg: Invalid passphrase; please try again ...

You need a passphrase to unlock the secret key for
user: "Student <student@securelabsondemand.com>"
1024-bit RSA key, ID 6BBFF1EA, created 2018-09-11 (main key ID 82CD90F8)

gpg: encrypted with 1024-bit RSA key, ID 6BBFF1EA, created 2018-09-11
"Student <student@securelabsondemand.com>"
gpg: public key decryption failed: bad passphrase
gpg: decryption failed: secret key not available
student@TargetLinux01:~/Documents$ gpg -d Send.txt.gpg

You need a passphrase to unlock the secret key for
user: "Student <student@securelabsondemand.com>"
1024-bit RSA key, ID 6BBFF1EA, created 2018-09-11 (main key ID 82CD90F8)

gpg: encrypted with 1024-bit RSA key, ID 6BBFF1EA, created 2018-09-11
"Student <student@securelabsondemand.com>"
My Name is George Boakye.
student@TargetLinux01:~/Documents$
```

## Part 3: A: George.txt created

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
minutes remaining

student@TargetLinux01: /home/instructor
student@TargetLinux01:~/Documents$ echo "This is a test of AES256 encryption." > George.txt
student@TargetLinux01:~/Documents$ ls
clear.txt      Example2.txt      Example.txt      George.pub      instructor.pub  Send.txt.md5     student.pub
clear.txt.gpg  Example2.txt.md5  Example.txt.md5  George.txt      instructor2.pub Send.txt          student2.pub
clear.txt.txt  Example2.txt.sha256 Example.txt.shal  instructor2.pub Send.txt.gpg    student2.pub
student@TargetLinux01:~/Documents$ gpg --cipher-algo AES256 --symmetric George.txt
student@TargetLinux01:~/Documents$ ls
clear.txt      Example2.txt      Example.txt      George.pub      instructor2.pub  Send.txt.gpg     student2.pub
clear.txt.gpg  Example2.txt.md5  Example.txt.md5  George.txt      instructor2.pub  Send.txt.md5     student2.pub
clear.txt.txt  Example2.txt.sha256 Example.txt.shal  George.txt.gpg  Send.txt        shasum           student2.pub
student@TargetLinux01:~/Documents$ rm George.txt
student@TargetLinux01:~/Documents$ sudo cp George.txt.gpg /home/instructor
[sudo] password for student:
student@TargetLinux01:~/Documents$ cd /home/instructor
student@TargetLinux01:/home/instructor$ sudo chown Instructor:Instructor George.txt.gpg
student@TargetLinux01:/home/instructor$ ls -la
total 64
drwxr-xr-x 3 Instructor Instructor 4096 Sep 12 19:03 .
drwxr-xr-x 4 root      root      4096 Mar 27 2017 ..
-rw-r--r-- 1 Instructor Instructor 769 Sep 12 18:31 .bash_history
-rw-r--r-- 1 Instructor Instructor 220 Mar 27 2017 .bash_logout
-rw-r--r-- 1 Instructor Instructor 3392 Mar 27 2017 .bashrc
-rw-r--r-- 1 Instructor Instructor 256 Sep 11 16:14 clear.txt.txt.gpg
-rw-r--r-- 1 Instructor Instructor 10254 Sep 12 18:05 George.pub
-rw-r--r-- 1 Instructor Instructor 116 Sep 12 19:03 George.txt.gpg
drwx----- 2 Instructor Instructor 4096 Sep 12 18:22 .gnupg
-rw-r--r-- 1 Instructor Instructor 1037 Sep 11 15:56 instructor.pub
-rw-r--r-- 1 Instructor Instructor 2881 Sep 12 18:01 nina.pub
-rw-r--r-- 1 Instructor Instructor 675 Mar 27 2017 .profile
-rw-r--r-- 1 Instructor Instructor 26 Sep 12 18:14 Send.txt
```

### Part 3: B: AES256 encryption method used

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
2 minutes remaining

student@TargetLinux01: /home/instructor

clear.txt      Example2.txt      Example.txt      george.pub      instructor.pub  Send.txt.md5    student.pub
clear.txt2.txt Example2.txt.gpg   Example.txt.md5  Example.txt      George.txt      Send.txt        shasum
clear.txt2.txt Example2.txt.sha256 Example.txt.sha1 instructor2.pub  Send.txt.gpg   student2.pub
student@TargetLinux01:~/Documents$ gpg --cipher-algo AES256 --symmetric George.txt
student@TargetLinux01:~/Documents$ ls
clear.txt      Example2.txt      Example.txt      george.pub      instructor2.pub  Send.txt.gpg    student2.pub
clear.txt2.txt Example2.txt.gpg   Example.txt.md5  Example.txt      George.txt      instructor.pub  Send.txt.md5    student.pub
clear.txt2.txt Example2.txt.sha256 Example.txt.sha1 George.txt.gpg   Send.txt        shasum
student@TargetLinux01:~/Documents$ rm George.txt
student@TargetLinux01:~/Documents$ sudo cp George.txt.gpg /home/instructor
[sudo] password for student:
student@TargetLinux01:~/Documents$ cd /home/instructor
student@TargetLinux01:/home/instructor$ sudo chown Instructor:Instructor George.txt.gpg
student@TargetLinux01:/home/instructor$ ls -la
total 64
drwxr-xr-x 3 Instructor Instructor 4096 Sep 12 19:03 .
drwxr-xr-x 4 root      root      4096 Mar 27 2017 ..
-rw-r--r-- 1 Instructor Instructor 769 Sep 12 18:31 .bash_history
-rw-r--r-- 1 Instructor Instructor 220 Mar 27 2017 .bash_logout
-rw-r--r-- 1 Instructor Instructor 3392 Mar 27 2017 .bashrc
-rw-r--r-- 1 Instructor Instructor 256 Sep 11 16:14 clear.txt.txt.gpg
-rw-r--r-- 1 Instructor Instructor 10254 Sep 12 18:05 george.pub
-rw-r--r-- 1 Instructor Instructor 116 Sep 12 19:03 George.txt.gpg
drwxr-xr-x 2 Instructor Instructor 4096 Sep 12 18:22 .gpgpg
-rw-r--r-- 1 Instructor Instructor 1037 Sep 11 15:56 instructor.pub
-rw-r--r-- 1 Instructor Instructor 2881 Sep 12 18:01 nina.pub
-rw-r--r-- 1 Instructor Instructor 675 Mar 27 2017 .profile
-rw-r--r-- 1 Instructor Instructor 26 Sep 12 18:14 Send.txt
-rw-r--r-- 1 Instructor Instructor 237 Sep 12 18:22 Send.txt.gpg
student@TargetLinux01:/home/instructor$
```

### Part 3: C: Copied to Instructor Account, Changed Permission, & Decrypted

```
Applying Encryption and Hashing Algorithms for Secure Com...
George Boakye
6 minutes remaining

Instructor@TargetLinux01: ~

student@TargetLinux01:~/Documents$ cd /home/instructor
student@TargetLinux01:/home/instructor$ sudo chown Instructor:Instructor George.txt.gpg
student@TargetLinux01:/home/instructor$ ls -la
total 64
drwxr-xr-x 3 Instructor Instructor 4096 Sep 12 19:03 .
drwxr-xr-x 4 root      root      4096 Mar 27 2017 ..
-rw-r--r-- 1 Instructor Instructor 769 Sep 12 18:31 .bash_history
-rw-r--r-- 1 Instructor Instructor 220 Mar 27 2017 .bash_logout
-rw-r--r-- 1 Instructor Instructor 3392 Mar 27 2017 .bashrc
-rw-r--r-- 1 Instructor Instructor 256 Sep 11 16:14 clear.txt.txt.gpg
-rw-r--r-- 1 Instructor Instructor 10254 Sep 12 18:05 george.pub
-rw-r--r-- 1 Instructor Instructor 116 Sep 12 19:03 George.txt.gpg
drwxr-xr-x 2 Instructor Instructor 4096 Sep 12 18:22 .gpgpg
-rw-r--r-- 1 Instructor Instructor 1037 Sep 11 15:56 instructor.pub
-rw-r--r-- 1 Instructor Instructor 2881 Sep 12 18:01 nina.pub
-rw-r--r-- 1 Instructor Instructor 675 Mar 27 2017 .profile
-rw-r--r-- 1 Instructor Instructor 26 Sep 12 18:14 Send.txt
-rw-r--r-- 1 Instructor Instructor 237 Sep 12 18:22 Send.txt.gpg
student@TargetLinux01:/home/instructor$ su Instructor
Password:
Instructor@TargetLinux01:~$ gpg --output George.txt.gpg --decrypt George.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
File 'George.txt.gpg' exists. Overwrite? (y/N) n
Enter new filename: DecryptedGeorge.txt.gpg
Instructor@TargetLinux01:~$ ls
clear.txt      DecryptedGeorge.txt.gpg george.pub  George.txt.gpg  instructor.pub  nina.pub  Send.txt  Send.txt.gpg
Instructor@TargetLinux01:~$ cat DecryptedGeorge.txt.gpg
This is a test of AES256 encryption.
Instructor@TargetLinux01:~$
```

### Reference

- ask ubuntu. (2018, September 13). Retrieved from ask ubuntu: <https://askubuntu.com/questions/60712/how-do-i-quickly-encrypt-a-file-with-aes>
- Genkin, D., Pachmanov, L., & Pipman, I. (2016, August 18). *ECDSA Key Extraction from Mobile Devices via Nonintrusive Physical Side Channels*. Retrieved from <https://eprint.iacr.org/2016/230.pdf>
- Naziridis, N. (2018, June 27). *Comparing ECDSA vs RSA*. Retrieved from SSL.com: <https://www.ssl.com/article/comparing-ecdsa-vs-rsa/>
- RSA SecurID. (2018, September 2). Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/RSA\\_SecurID](https://en.wikipedia.org/wiki/RSA_SecurID)

