



Academic Year: 2025-26

Semester: V

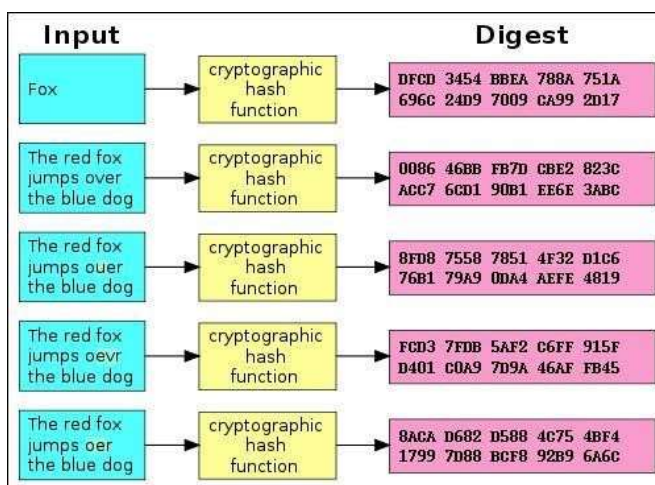
Class / Branch: TE IT Subject: Security Lab (SL)

Subject Lab Incharge: Prof. Apeksha Mohite

## Experiment No. 12

1. **Aim: To study and test message integrity by using MD5, SHA-1 for varying message sizes.**
2. **Software Required : Ubuntu 14.04 OS**
3. **Theory :**

Hashes are the products of cryptographic algorithms designed to produce a string of characters. Often these strings have a fixed length, regardless of the size of the input data. Take a look at the above chart and you'll see that both "Fox" and "The red fox jumps over the blue dog" yield the same length output.



Now compare the second example in the chart to the third, fourth, and fifth. You'll see that, despite a very minor change in the input data, the resulting hashes are all very different from one another. Even if someone modifies a very small piece of the input data, the hash will change dramatically.

MD5, SHA-1, and SHA-256 are all different hash functions.



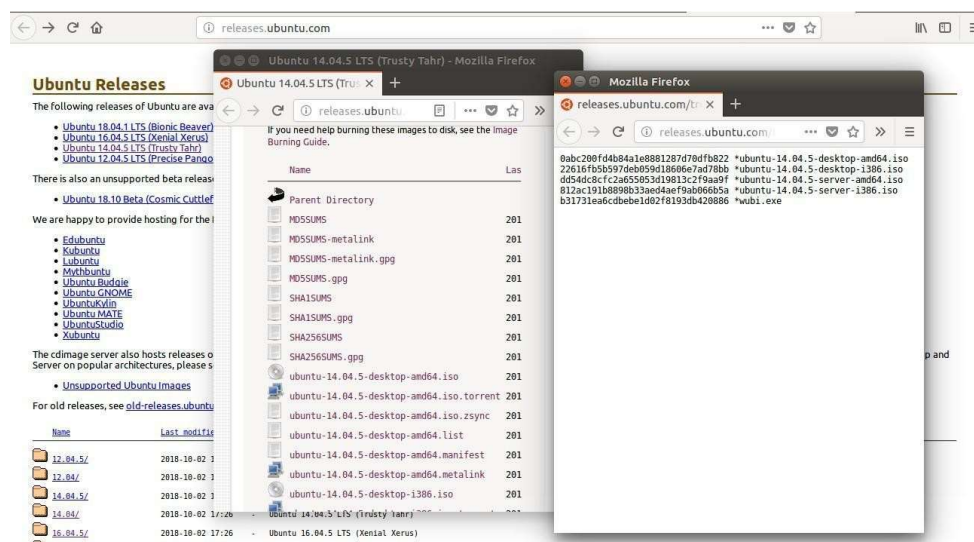
PARSHVANATH CHARITABLE TRUST'S  
**A. P. SHAH INSTITUTE OF TECHNOLOGY**  
Department of Information Technology  
(NBA Accredited)



Here is the comparison between MD5 and SHA1. You can get a clear idea about which one is better.

Keys For Comparison	MD5	SHA
Security	Less Secure than SHA	High Secure than MD5
Message Digest Length	128 Bits	160 Bits
Attacks required to find out original Message	$2^{128}$ bit operations required to break	$2^{160}$ bit operations required to break
Attacks to try and find two messages producing the same MD	$2^{64}$ bit operations required to break	$2^{80}$ bit operations required to break
Speed	Faster, only 64 iterations	Slower than MD5, Required 80 iterations
Successful attacks so far	Attacks reported to some extents	No such attach report yet

Software creators often take a file download—like a Linux .iso file, or even a Windows .exe file—and run it through a hash function. They then offer an official list of the hashes on their websites.



That way, you can download the file and then run the hash function to confirm you have the real, original file and that it hasn't been corrupted during the download process. As we saw



PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

**Department of Information Technology**

**(NBA Accredited)**



above, even a small change to the file will dramatically change the hash.

These can also be useful if you have a file you got from an unofficial source and you want to confirm that it's legitimate. Let's say you have a Linux .ISO file you got from somewhere and you want to confirm it hasn't been tampered with. You can look up the hash of that specific ISO file online on the Linux distribution's website. You can then run it through the hash function on your computer and confirm that it matches the hash value you'd expect it to have. This confirms the file you have is the exact same file being offered for download on the Linux distribution's website, without any modifications.

### **Verify Data Integrity :**

The checksum is used to verify the correctness of a file. It can be described as a digital fingerprint of a file. By verifying the Checksum value we can determine the correctness of a file while it's been transferred from one location to another. The checksum is a long string of data containing various letters and numerals. All popular software downloading websites provides a checksum value for the downloaded file with which we can confirm our data by verifying the checksum value.

### **Generating Checksums**

A checksum is generated by a checksum algorithm. It generates a checksum value by taking the file as input. MD5 and SHA (Secure Hash Algorithms) are the most popular algorithms used for generating the checksums.

#### **Command-line Checksum tools**

Almost all Linux distribution provides the command line tools for various checksum algorithms. You can generate and verify checksum with them. Some of the standard command-line checksum tools used nowadays are the followings:

MD5 checksum tool is called: md5sum

SHA-1 checksum tool is called: sha1sum

SHA-256 checksum tool is called: sha256sum SHA-384 checksum tool is called: sha384sum

SHA-224 checksum tool is called: sha224sum SHA-512 checksum tool is called: sha512sum



PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

**Department of Information Technology**

**(NBA Accredited)**



md5sum: MD5 is an algorithm that is used to verify data integrity through the creation of a 128-bit message digest from a data input that is claimed to be as unique to that specific data as a fingerprint to a specific individual.

On Linux, access a Terminal and run the following commands to view the hash for a file :

```
apsit@apsit-HP-Notebook: ~/Music
apsit@apsit-HP-Notebook:~$ cd /home/apsit/Music
apsit@apsit-HP-Notebook:~/Music$ echo This is demo of md5sum>example.txt
apsit@apsit-HP-Notebook:~/Music$ md5sum example.txt
2bdb073a79fa278cb34a466d94ac784c  example.txt
apsit@apsit-HP-Notebook:~/Music$
```

Even a small change to the file will dramatically change the hash. We try to make changes and view the hash values again.

```
apsit@apsit-HP-Notebook: ~/Music
apsit@apsit-HP-Notebook:~$ cd /home/apsit/Music
apsit@apsit-HP-Notebook:~/Music$ echo This is demo of md5sum>example.txt
apsit@apsit-HP-Notebook:~/Music$ md5sum example.txt
2bdb073a79fa278cb34a466d94ac784c  example.txt
apsit@apsit-HP-Notebook:~/Music$ echo This is to check message integrity >example.txt
apsit@apsit-HP-Notebook:~/Music$ cat example.txt
This is to check message integrity
apsit@apsit-HP-Notebook:~/Music$ md5sum example.txt
458209d843ab0d8c41358d26311737d0  example.txt
apsit@apsit-HP-Notebook:~/Music$
```

shasum:

SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function designed by the United States National Security Agency. SHA-1 produces a 160-bit (20-byte) hash value known as a message digest. Please see the sha1 hash value for the same file.

```
apsit@apsit-HP-Notebook: ~/Music
apsit@apsit-HP-Notebook:~/Music$ shasum example.txt
a1617450c7b5e21efa3b1b76724fa4569121e60d  example.txt
apsit@apsit-HP-Notebook:~/Music$ echo testing sha1 >example.txt
apsit@apsit-HP-Notebook:~/Music$ shasum example.txt
d9a786e86480cd108a912abea3069cf9e369d602  example.txt
apsit@apsit-HP-Notebook:~/Music$
```

sha256sum/sha512sum/sha224sum/sha384sum:





PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

**Department of Information Technology**

**(NBA Accredited)**



SHA-2 is a family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words whereas SHA-512 uses 64-bit words. There are also truncated versions of each standard, known as SHA-224, SHA-384, SHA-512/224 and SHA-512/256. SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash. Hash is a one-way function, which cannot be decrypted back. We can generate the hash value using this SHA-256 algorithm for the same file using the command below:

```
apsit@apsit-HP-Notebook: ~/Music
apsit@apsit-HP-Notebook:~/Music$ sha256sum example.txt
ecc28c251bf3522e66157bdc5c43617d37a3c05e58ac48204d67c660b38666c0 example.txt
apsit@apsit-HP-Notebook:~/Music$ sha224sum example.txt
da33a14765ebc7c7288234e617b91d2af7c508f17b12d744d6f9ed21 example.txt
apsit@apsit-HP-Notebook:~/Music$ sha512sum example.txt
4aebec20fd4e0cbdf8c4b0664e954a519256d3226eb84fbf245cd09507c0e125a006d7757ec241a47
9729a87531a54c4d1eb4d672ea9163047d639ba373295727 example.txt
apsit@apsit-HP-Notebook:~/Music$ sha384sum example.txt
7f6bda478d1f3dfd1cd4b0ba5ca5f85fcb5b94ffe24614ad20689afb2aae4ed3ca6044b4cc48c242
0b206d4458e7c517 example.txt
apsit@apsit-HP-Notebook:~/Music$
```

You can confirm the correctness of your downloaded ISO by comparing the checksum value here. It appears to be same, which means you've downloaded the exact file.

If you delete or change even one character from any one of the text files inside the iso image, the checksum algorithm will generate a totally different checksum value for that changed iso image. And that will definitely not match with the checksum provided on the download page.

```
apsit@apsit-HP-Notebook: ~/Music
apsit@apsit-HP-Notebook:~/Music$ ls
example.txt  ubuntu-14.04.5-desktop-amd64.iso
apsit@apsit-HP-Notebook:~/Music$ md5sum ubuntu-14.04.5-desktop-amd64.iso
0abc200fd4b84a1e8881287d70dfb822 ubuntu-14.04.5-desktop-amd64.iso
apsit@apsit-HP-Notebook:~/Music$
```



While hashes can help you confirm a file wasn't tampered with, there's still one avenue of



PARSHVANATH CHARITABLE TRUST'S

**A. P. SHAH INSTITUTE OF TECHNOLOGY**

**Department of Information Technology**

**(NBA Accredited)**



attack here. An attacker could gain control of a Linux distribution's website and modify the hashes that appear on it, or an attacker could perform a man-in-the-middle attack and modify the web page in transit if you were accessing the website via HTTP instead of encrypted HTTPS.

That's why modern Linux distributions often provide more than hashes listed on web pages. They cryptographically sign these hashes to help protect against attackers that might attempt to modify the hashes.

4. **Conclusion** : We have seen how checksum are generated for MD5 and SHA. You can make use of this Checksum method as a redundancy check to detect errors in data. Hence, ensure the integrity of data portions for data transmission or storage.