The Usage of SHA1 function for Piracy Detection in Cloud Storage

David Setyanugraha 13511003
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung Jl. Ganesha 10 Bandung 40132, Indonesia
13511003@std.stei.itb.ac.id

***Abstract*—Nowdays, piracy files have been distributed through many different media. One of those media is through cloud storage or file hosting service. Many methods have been used to validate each file. The method like finding checksum with SHA1 of each file has been used extensively. SHA-1 stands for secure hash algorithm-1. SHA-1 is the most widely used of the existing SHA hash functions. For long time, SHA-1 forms part of several widely used security applications and protocols. The idea is to connect the SHA-1 hash function with the cloud storage. The paper represents the usage of SHA-1 function in detecting the piracy of files distributed in cloud storage. In short, it will be proposed a system to prevent user share the pirated file in cloud storage.**

***Keywords: SHA-1*, piracy, cloud storage**

I. INTRODUCTION

Nowadays, Cloud storage has become one of the choices for storage. It's a model of data storage where digital data is stored in logical pools, the physical storage spans multiple servers. It is a service model which data is maintained, managed, and backed up remotely and made available to users over a network (typically on internet). [3] The focus of this research is on file storage. Many commercial and open source file storage have been unveiled. File hosting storage allows users to create special folder on their computers. Each cloud storage also has their own architecture.

Every day, millions of computer users share files online. Many of those files are protected by copyright laws. Three main issue about cloud storage can be broken down into access and integrity security, data encryption, and ownership security.[1] Piracy file has become one of the major issues in file sharing service. The problem is about how to manage every data being shared is a piracy or not. Every hosting service has their mechanism to prevent this issue. The research focus on the utilization of SHA-1 in detecting the piracy of files distributed through cloud storages.

II. STUDY LITERATURE

A. Cloud Storage

Cloud Storage is storage where the user can put their file remotely. The advantages using cloud storage is that you can always get it on multiple devices. That means the photos, files, or songs are being back-up on those various services. So if the hard drive crashed, you would have a backup of all your files instantaneously. Cloud storage is based on highly virtualized infrastructure. Cloud Storage is made up of many distributed resources, but still acts as one – often reffered to as federated storage clouds [1]. There are many concerns about these cloud storage. The concerns are like security of stored data in transit, rich resource for hackers and NSA, Piracy, and legal aspect.

Many researchs have been done to build a secure cloud storage service. At high level, several architectures that combine recent and non-standard cryptographic primitives in order to achieve our goal. Every service should provide at least confidentiality and integrity. Confidentiality means the cloud storage provider does not learn any information about customer data. Integrity means any unauthorized modification of customer data by cloud storage provider can be detected by the customer.

B. SHA hash Function

National Security Agency (NSA) has claimed SHA as the standard one way hash function. SHA is built from MD4 by Ronald L. Rivest from MIT. SHA-1 algorithm is an algorithm to compute a fixed-length digital representation (called as a message digest) of an input data sequence (the message) of any length. Any change to a message will result in a different message digest. SHA algorithm get input message with max 2^{64} bit and produce message digest with 160 bit length longer than message digest from MD5.

Below is the pseudocode of SHA-1 hash function:

```
add some extra data to the end of the input
set the initial sha-1 values

for each 64-byte chunk do
  extend the chunk to 320 bytes of data

  perform first set of operations on chunk[i] (x20)
  perform second set of operations on chunk[i] (x20)
  perform third set of operations on chunk[i] (x20)
  perform fourth set of operations on chunk[i] (x20)
end

return sha-1 values as a hash
```

There are mainly six steps in SHA-1 Hash Function. These six steps will be executed every time the hash function executed. These six steps are the reason why SHA is considered as complex hash function. Every byte of data is executed using byte operations. Each step of SHA hash function will be explained briefly in the box below.

Step 1: Append Padding Bits
Message is "padded" with a 1 and as many 0's as necessary to make the message length to 64 bits fewer than an even multiple of 512

Step 2: Append Length
64 bits are appended to the end of the padded message.

Step 3: Prepare Processing Functions
SHA1 requires 80 processing functions as defined.

Step 4: Prepare Processing Constants
SHA1 requires 80 processing constant words

Step 5: Initialize Buffers
SHA1 requires 160 bits or 5 buffers of words (32 bits):

Step 6: Processing Message in 512-bit blocks
This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks.

File verification is the process of using an algorithm for verifying the authenticity of computer file. This can be done by comparing two files bit-by-bit, but requires two copies of the same file. One of the popular approaches is called checksums (hashes) / message digests of files for later comparison. A checksum file like SHA-1 is a small file that contains the checksum of other files. Every file will have unique SHA-1 checksum. This characteristic of file will be used on the system for detecting the piracy files.

III. PROPOSED SYSTEM

The system proposed is to detect the piracy file in cloud storage. The system requires the cloud storage company to have an agreement with companies who own the original file. It can prevent the file saved in cloud storage to be shared in public. Every time the user press share button in their cloud storage system, it will trigger the validation system that has been made. Share button is a feature for every cloud storage system to enable the file distributed in the web.

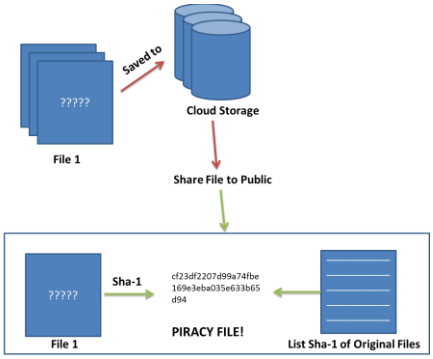

Figure 1. Piracy file detection system

The system detects piracy from a file simply from the comparison between SHA1 checksum generated for the file and SHA1 checksum from the list of original files. The agreements between the cloud storage company and the owner of the files are also the success key for this system. Every company who doesn't want their file being shared publicly must give the SHA1 code for the cloud storage company. The prototype of system has been implemented using Java language. The system is built using small part of storage. We don't focus on the cloud storage. We assume every cloud storage has their own unique architecture. The system will be easily adapted to every part of cloud storage. The prototype of cloud storage validation system has been built to test the system. As can be seen in figure 2, the prototype of the system will contain text area of list SHA1 file and file to be shared.

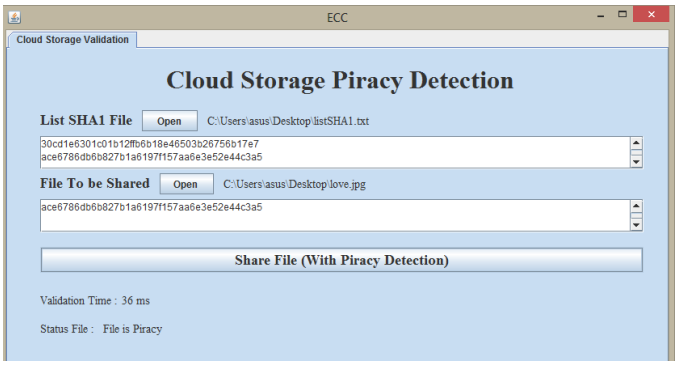

Figure 2. The prototype of the system

To improve the performance and security of list SHA1 file, there will be many lists of SHA1 files categorized by its file extension. For instance, files with extension MP3 will have their own SHA1 list. The illustration of the list SHA1 file can be seen on the diagram below.

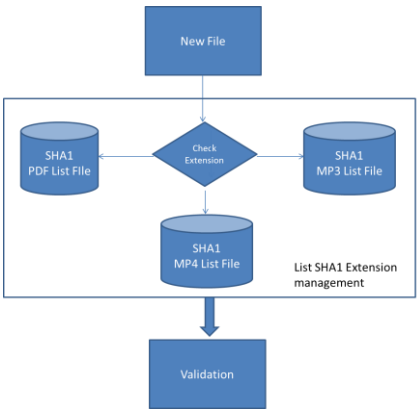

Figure 3. List SHA1 extension management

Each new file will enter the extension checking process before the validation being conducted. After the extension validation done, each file will be detected the piracy status using main algorithm.

The main idea of the progam can be simplified using the algorithm below. The core system is written in java programming language.There are four main steps for the core system. First, initalized every variable needed. Second, check the checksum of each file. The execution time for checking checksum SHA1 of the file depends on how large the file. Third, it's the process searching SHA1 list by looping through each loop of SHA1 list file. Finally, the system will judge if

the file is a piracy or not. If it is a piracy then the system will cancelled the whole share execution.

```
Public static void main(String[] args) {

      /**Initialize every variable needed*/
      ArrayList<String> fileEks = ReadFromExternalFile();
      SHA1 hash = new SHA1();
      String fileName = {{Path with filename}};

      /**Check SHA1 for file*/
      System.out.println("SHA1 for "+fileName);
      String sha1String = getSHA1ForFile(hash, fileName);

      /**Searching SHA1 on  -List-*/
      boolean isPiracy = false;
      for (String fileEk : fileEks) {
            System.out.println(fileEk);
            if (sha1String.equals(fileEk)){
              isPiracy = true;
              break;
            }
      }
      /**Print out the result*/
      if (isPiracy){
         System.out.println("It is a Piracy !");
         /**Execute the share cancel method*/
      }else {
         System.out.println("It isn't a Piracy!");
         /**continue to the next step in cloud storage*/
      }
}
```

IV. Experiment and Results

The experiment is conducted through some kinds of file. There are three types of file we tested. The files include mp3, mp4, and PDF. There will be different size for each file. Each test file will be labelled as small, medium or large size file. The purpose of the experiment is to find the relation between size of file with the performance of system. The reason why mp3, mp4, and PDF are chosen because those three files are the most popular file shared in cloud storage. Details size the test file can be seen in table below:

File Type	Small	Medium	Large
Mp3	1.27 MB	3.50 MB	11.0 MB
Mp4	17.5 MB	85.5 MB	238 MB
PDF	1MB	7.1MB	20MB

The experiment assumes that the list SHA1 original files have existed. Execution time also will become one of the parameter in the experiment. Execution time is the parameter for the system to work properly through many scenarios of files. The table below shows each scenario and result of the experiments done in the system. Expected column will be the real label of the file. Prediction column is the result of the system. The timer of execution time will be started from the process checking SHA1 file and stopped when the result of the prediction system is obtained. There will be 9 files total for the experiment done through the system. The experiment is conducted through the same capacity of computer.

Scenario	Expected	Prediction	Execution Time
MP3 Small	Piracy	Piracy	15 ms
MP3 Medium	notPiracy	notPiracy	56 ms
MP3 Large	Piracy	Piracy	270 ms
MP4 Small	Piracy	Piracy	28 ms
MP4 Medium	Piracy	Piracy	681 ms
MP4 Large	notPiracy	notPiracy	1717 ms
PDF Small	notPiracy	notPiracy	78 ms
PDF Medium	Piracy	Piracy	88 ms
PDF Large	notPiracy	notPiracy	332 ms

As can be seen from the table above, the proposed system has successfully predicted the expected type of file in all scenarios. However, the execution time varies. The longest execution time is on MP4 large file which has 238 MB size. The bigger size of the file, longer the execution time will be. From the results, we can see that MP4 file consume more time to be executed than other type of file. The MP4 files have also the longest execution time because the size of the file is bigger than others.

V. Security Analysis

From the perspective of security, the architecture of system fully depends on the strength of SHA1 hash function. In 2011, the best attack can produce hash collisions with a complexity between $2^{60.3}$ and $2^{65.3}$.[4] Collision attacks are the most common kind of attack against cryptographic hashes. The reason collision attacks are relatively easy is because any pair of inputs that hash to the same value are acceptable.There are no restrictions placed on what those inputs are, or what the value of the hash is.

The security of the list SHA1 original file storage is also important to be considered. The list SHA1 must be stored in a safer place. In order to make the system more powerful, the list about SHA1 original file must be added more. For more security, a token system can be also built to get access of list SHA1 function. This token must be randomly generated. We must guarantee only the certified system can access that data.

VI. Conclusion and Suggestion

This paper proposes the new system to detect piracy file in cloud storage. The proposed system has been proved sucessfully in detecting piracy in cloud storage. Several experiments using Java programming language have been done to prove the power of the system. From the experiment, we can conclude that the bigger the size of the file, the time needed to finish the system will also bigger. The main issue of the system is about how big the list SHA1 file will be. For future works, the writer suggests the usage of machine learning system to make the system more robust.

VII. REFERENCES

[1] http://abcnews.go.com/Technology/cloud-computing-storage-explained/story?id=16647561. Access Time 9 May 2015, 12.00 AM
[2]http://www.schneier.com/blog/archives/2012/10/when_will_we_se.html Access Time 9 May 2015, 8.00 PM
[3] http://www.webopedia.com/TERM/S/storage_cloud.html . Access Time 10 May 2015, 10.00 AM
[4] Marc Stevens (19 June 2012). "Attacks on Hash Functions and Applications"
[5] Vernik, Gil, et al. "Data On-boarding in Federated Storage Clouds." Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing. IEEE Computer Society, 2013.

VIII. PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Mei 2015

David Setyanugraha