Project Report

Data Security And Encryption

Submitted to: Tanuj Kumar

Submitted By:

Vipul Warik- 18BCE7157

Sangam Mahajan- 18BCE7005

Kunal Singh Chauhan- 18BCE7159

Audumber Chaudhari-18BCE7032

Gurunath Deshmukh- 18BCE7108

Anubhav Tyagi-18BEC7038

Deepanshu Tyagi-18BEC7027

Contents

Abstract

Security has become a wide necessity in day-today life. Data security is the most obliged security of all. The data in our system is opened to high potential risks. Due to various security reasons we adopt diverse methods.
Huge amount of data is energetically updated in today's world. In cloud computing there are a lot of important problems which include issues of privacy, security, secrecy, communications capacity, government surveillance, consistency, and responsibility.Although most of the web based applications requires a security for the data number of symmetric and Asymmetric algorithms approaches with maximum protection for the data to be transferred.We portray new public key cryptosystems which create steady size cipher texts with the end goal.

Introduction

Definition - What does Data Security mean?

Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites. Data security also protects data from corruption. Data security is an essential aspect of IT for organizations of every size and type.

Data security is also known as information security (IS) or computer security.

Examples of data security technologies include backups, data masking and data erasure. A key data security technology measure is encryption, where digital data, software/hardware, and hard drives are encrypted and therefore rendered unreadable to unauthorized users and hackers.

One of the most commonly encountered methods of practicing data security is the use of authentication. With authentication, users must provide a password, code, Data security is also very important for health care records, so health advocates and medical practitioners in the U.S. and other countries are working toward

implementing electronic medical record (EMR) privacy by creating awareness about patient rights related to the release of data to laboratories, physicians, hospitals and other medical facilities.biometric data, or some other form of data to verify identity before access to a system or data is granted.

Working

- **Cyphr Package**

This package tries to smooth over some of the differences in encryption approaches (symmetric vs. asymmetric, sodium vs. openssl) to provide a simple interface for users who just want to encrypt or decrypt things.

The scope of the package is to protect data that has been saved to disk. It is not designed to stop an attacker targeting the R process itself to determine the contents of sensitive data. The package does try to prevent you accidentally saving to disk the contents of sensitive information, including the keys that could decrypt such information.

This vignette works through the basic functionality of the package. It does not offer much in the way of an introduction to encryption itself; for that see the excellent vignettes in the `openssl` and `sodium` packages (see `vignette("crypto101")` and `vignette("bignum")` for information about how encryption works). This package is a wrapper around those packages in order to make them more accessible.

Use of R programming

Program: R is a clear and accessible programming tool

Transform: R is made up of a collection of libraries designed specifically for data science

Discover: Investigate the data, refine your hypothesis and analyze them

Model: R provides a wide array of tools to capture the right model for your data

Communicate: Integrate codes, graphs, and outputs to a report with R Markdown or build Shiny apps to share with the world

How R programming will be used in this projection(Algorithm)

Step-by-step

This Code has 3 Methods(Function)

1. **Ztest()**

This function basically validates whether the data provided is worth encrypting or not. If the data is valid with 5% level of significance then it starts encrypting it.

- First of all it reads the .csv file which is sent to it via argument
- Also 1st and 2nd columns are picked automatically by the function
- It also provides user option to carry out 1 or 2 tailed test through passing argument 1 or 2
- Then z test is carried out according to general formula used to carry out z test
- If the Z Test is qualified by the data with 5% level of Significance then encode function is called
- Else no action is performed

Function code:

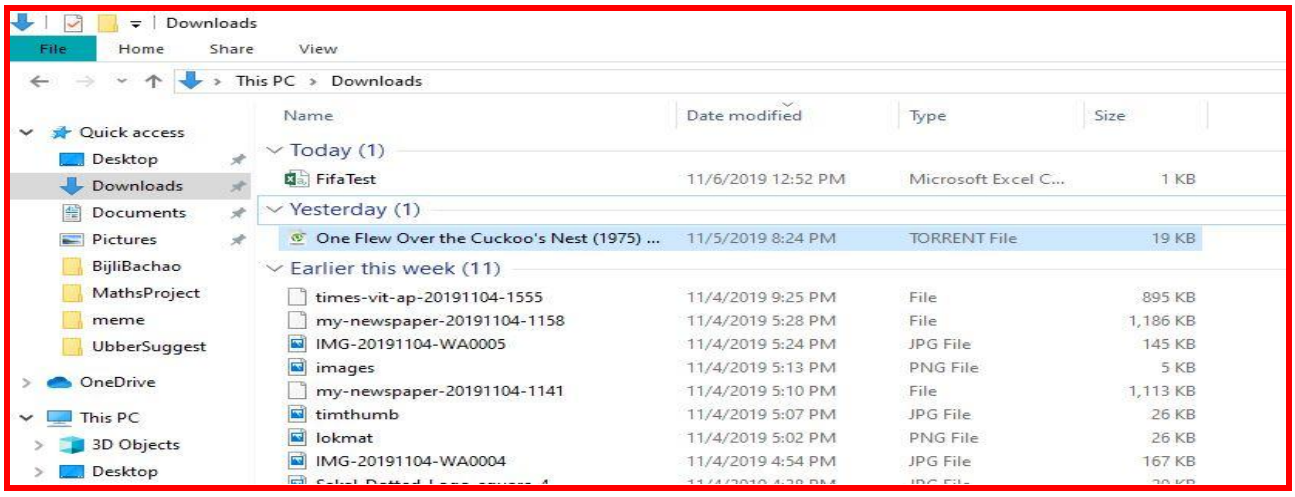
```
ztest<-function(tail,name){
data = read.csv(name)
d=as.matrix(data)
dim=dim(d)
```

```
n=dim[1]
panA=data[,1]
panB=data[,2]
t=tail
 meanA=mean(panA)
     meanB=mean(panB)
     #calculating population sd
      numA = sum((panA-meanA)**2)
      numB = sum((panB-meanB)**2)
     sdA = sqrt(numA/n)
     sdB = sqrt(numB/n)
     deno1 = (sdA**2)/n
     deno2 = (sdB**2)/n
     deno=deno1+deno2
     z=((meanA-meanB)-0)/sqrt(deno)
if(t==1){
   p.value=1-pnorm(z)
}
else{
p.value=1-2*pnorm(z)
}
print("p value is")
print(p.value)
if(p.value<0.05)
{print("Sorry Your Data Cannot be Encrypted As it Failed To
Qualify The Z-Test With 5% Level of Significance")
}
else{
print("Z-Test Qualified...Data Encryption Started")
k<-encode(name)
return(k)
}
}
```

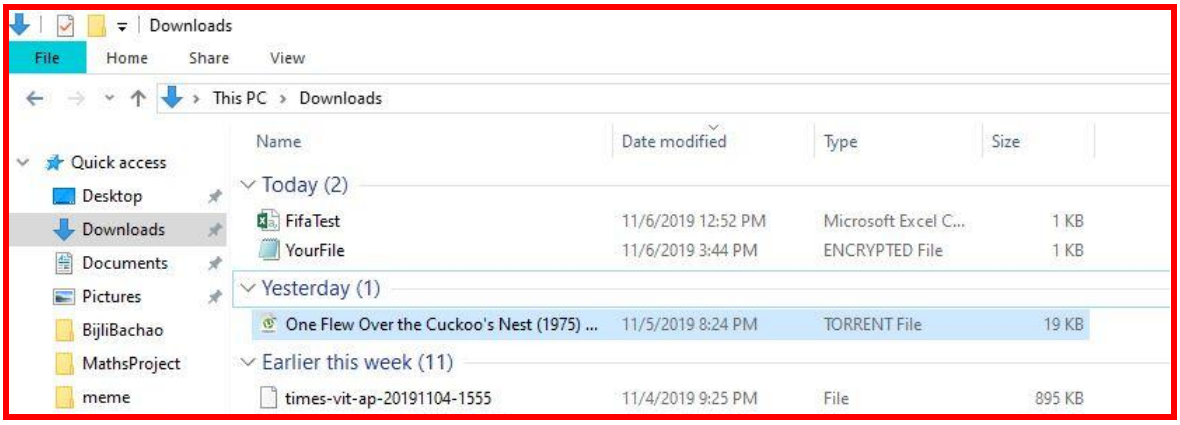
2. **Encode()**

   This is called only when data is validated in function Z test. **We use CYPHR package for encryption.** After installing cyphr package we need to load it in our workspace. Using **library(cyphr)**

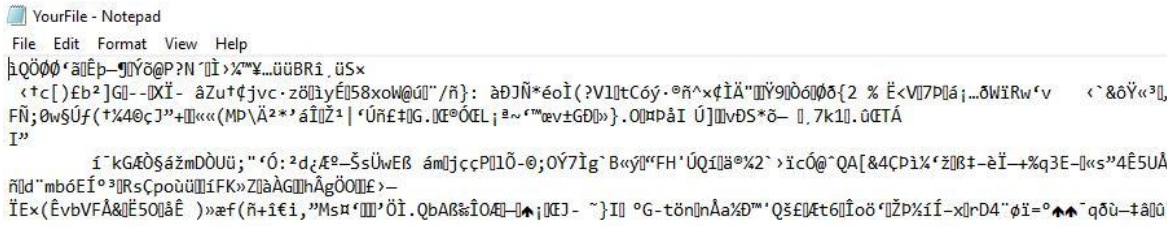
- This function also reads the .csv file once again
- Then a key is generated by using **sodium** function in cyphr package. Best part is that this is new every time and also unique
- After this encoding starts till then there is no file in dir except the data file



- After encoding the encoded file is added to the directory



- The Encrypted file cannot be read with in any case.
- Encrypted if opened will look something like the one below



- If the End User have the key He/She will be able to decode the file

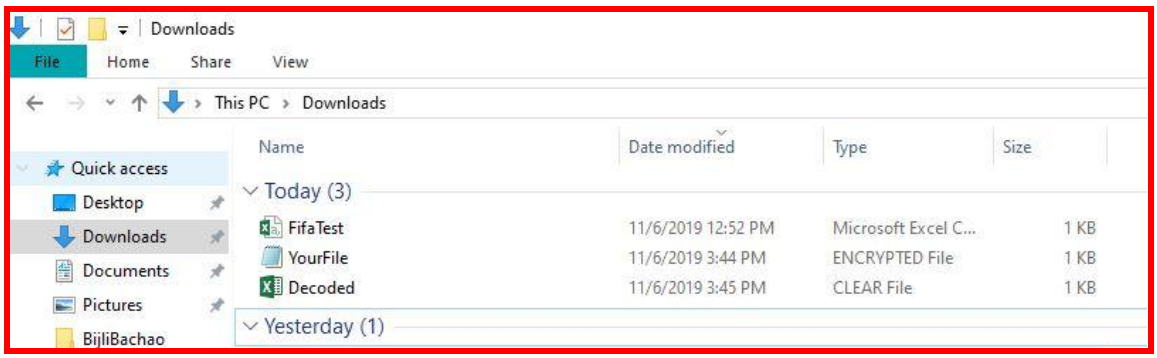
Function code:

```
encode<-function(name)
{
data = read.csv(name)

  k <- sodium::keygen()
  key <- cyphr::key_sodium(k)
print(k)
```

```
  cyphr::encrypt_file(name, key, "YourFile.encrypted")
return(key)


}
```

3. Decode()

- This function will take the encoded file and decode it
- Key must be given as an argument
- Decoded file will be Added to the Directory automatically



Function code:

```
decode<-function(key){
cyphr::decrypt_file("YourFile.encrypted",key, "Decoded.clear")
}
```

Conclusion:

- As the data has to pass the z test before encryption the project makes sure that data is relevant and valid.
- Unique key is generated every time when encryption is necessary
- The is big enough and hence cannot be regenerated by any method or also cannot be predicted
- Project is capable of handling any number of data without making changes in the code
- End user doesn't have to be familiar with R programming. Just some basic computer knowledge would make it.
- Encrypted files can be transferred to any device or over the internet without the fear of getting stolen.

References:

- **www.kaggle.com**
- **https://www.techopedia.com/definition/26464/data-security**
- **https://cran.r-project.org/web/packages/cyphr/readme/README.html**
- **http://ropensci.github.io/cyphr/articles/cyphr.html**

Appendix:

code:

```
 library(cyphr)
 #Z-Test Function
ztest<-function(tail,name){
data = read.csv(name)
d=as.matrix(data)
dim=dim(d)
n=dim[1]
panA=data[,1]
panB=data[,2]
t=tail
 meanA=mean(panA)
    meanB=mean(panB)
    #calculating population sd
     numA = sum((panA-meanA)**2)
     numB = sum((panB-meanB)**2)
    sdA = sqrt(numA/n)
```

```
    sdB = sqrt(numB/n)
    deno1 = (sdA**2)/n
    deno2 = (sdB**2)/n
    deno=deno1+deno2
    z=((meanA-meanB)-0)/sqrt(deno)
if(t==1){
  p.value=1-pnorm(z)
}
else{
p.value=1-2*pnorm(z)
}
print("p value is")
print(p.value)
if(p.value<0.05)
{print("Sorry Your Data Cannot be Encrypted As it Failed To Qualify The
Z-Test With 5% Level of Significance")
}
else{
print("Z-Test Qualified...Data Encryption Started")
k<-encode(name)
return(k)
}
}

 #Encode Function
encode<-function(name)
{
data = read.csv(name)

  k <- sodium::keygen()
  key <- cyphr::key_sodium(k)
print(k)
  cyphr::encrypt_file(name, key, "YourFile.encrypted")
```

```
return(key)


}

 #Decode Function
decode<-function(key){
cyphr::decrypt_file("YourFile.encrypted",key, "Decoded.clear")
}
```