Chapter 8
Securing Your R cloud

Analytics deals with huge amounts of confidential data. For regulatory and business continuity, it is preferable that analytical environments be secure from operational risk including cyber threats. Some veteran IT administrators may think that their certifications in Networking and Legacy Operating Systems Engineering are not relevant in the cloud environment, but the basic principles remain the same.

8.1 Ensuring R Code Does not Contain Your Login Keys

Occasionally we have to connect to services from R that ask for login details, such as databases but you may not want to store your login details in the R source code file, instead the user would prefer to enter the my login details when I execute the code. The following code can do that https://gist.github.com/mages/2aed2a053e355e3bfe7c#file-getlogindetails-r while the corresponding note is at http://lamages.blogspot.be/2014/07/simple-user-interface-in-r-to-get-login.html

```
getLoginDetails <- function(){
## Based on code by Barry Rowlingson
## http://r.789695.n4.nabble.com/tkentry-that-exits-after-RETURN-tt854721.html#none
require(tcltk)
tt<-tktoplevel()
tkwm.title(tt,"Get login details")
Name <- tclVar("Login ID")
Password <- tclVar("Password")
entry.Name <-tkentry(tt,width="20",textvariable=Name)
entry.Password <-tkentry(tt,width="20", show="*",textvariable=Password)
tkgrid(tklabel(tt,text="Please enter your login details."))
tkgrid(entry.Name)
tkgrid(entry.Password)
```

A Ohri, *R for Cloud Computing*, DOI 10.1007/978-1-4939-1702-0_8

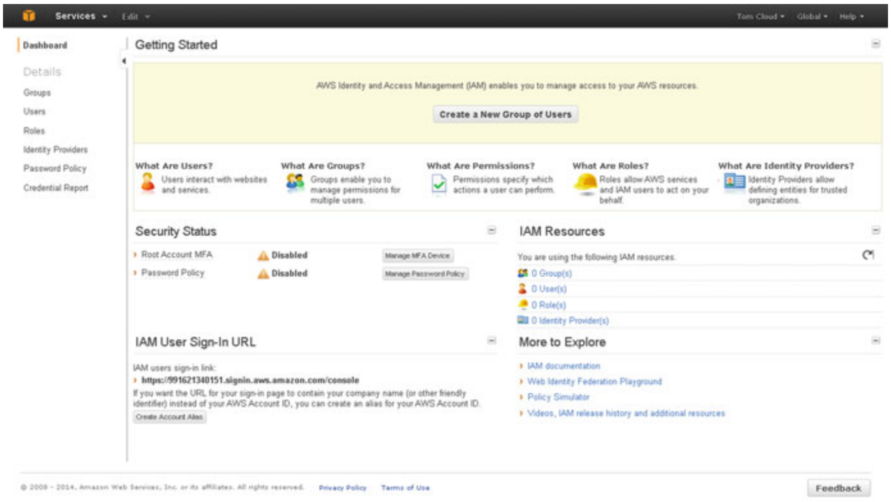
```
OnOK <- function() {
tkdestroy(tt)
}
OK.but <-tkbutton(tt,text=" OK ",command=OnOK)
tkbind(entry.Password, "<Return>",OnOK)
tkgrid(OK.but)
tkfocus(tt)
tkwait.window(tt)
invisible(c(loginID=tclvalue(Name), password=tclvalue(Password)))
}
credentials <- getLoginDetails()
## Delete credentials
##rm(credentials)
```

8.2 Setting Up Access Control (User Management Rights)

For a network to be secure, different users need to have different rights based on their need to know, their potential for administrative overlap, and their job functions. Accordingly we give the example of user management in the AWS cloud.

8.2.1 Amazon User Management

- Go to the IAM console (https://console.aws.amazon.com/iam/home).



Getting Started

AWS Identity and Access Management (IAM) enables you to manage access to your AWS resources.

What Are Users? Users interact with websites and services.

What Are Groups? Groups enable you to manage permissions for multiple users.

What Are Permissions? Permissions specify which actions a user can perform.

What Are Roles? Roles allow AWS services and IAM users to act on your behalf.

What Are Identity Providers? Identity Providers allow defining entities for trusted organizations.

Security Status

Root Account MFA	Disabled	Manage MFA Device
Password Policy	Disabled	Manage Password Policy

IAM Resources

You are using the following IAM resources.

0 Group(s)
0 User(s)
0 Role(s)
0 Identity Provider(s)

IAM User Sign-In URL

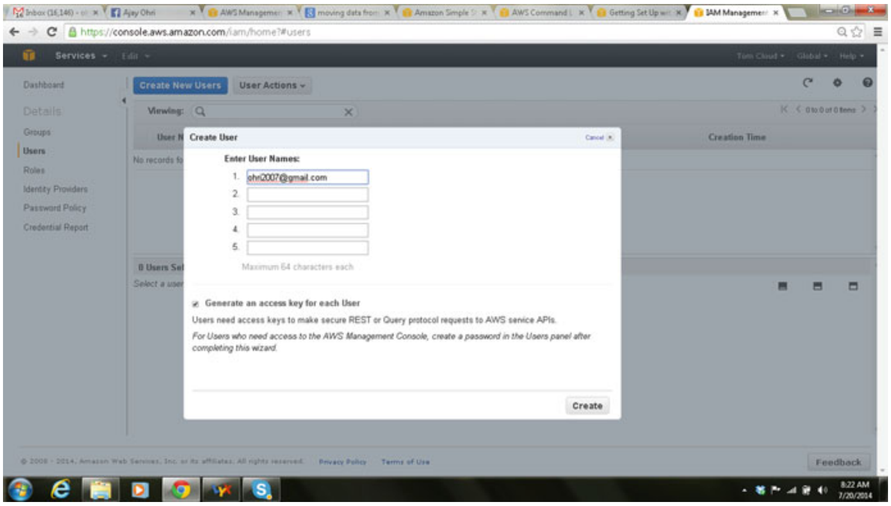
IAM users sign-in link:
https://991621340151.signin.aws.amazon.com/console
If you want the URL for your sign-in page to contain your company name (or other friendly identifier) instead of your AWS Account ID, you can create an alias for your AWS Account ID.

More to Explore

- IAM documentation
- Web Identity Federation Playground
- Policy Simulator
- Videos, IAM release history and additional resources

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

- From the navigation menu, click Users.



Create User

Enter User Names:

1. ohri2007@gmail.com
2.
3.
4.
5.

Maximum 64 characters each

[✓] Generate an access key for each User

Users need access keys to make secure REST or Query protocol requests to AWS service APIs.

For Users who need access to the AWS Management Console, create a password in the Users panel after completing this wizard.

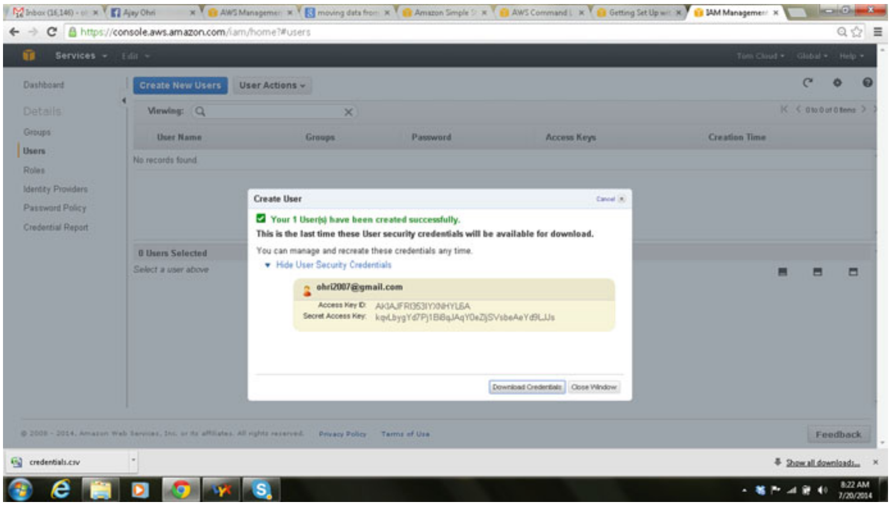
- Select your IAM user name.
  - Click User Actions, and then click Manage Access Keys.
- Click Create Access Key. Your keys will look something like this:
  - Access key ID example: AKIAIOSFFORE7EXAMPLE

    Secret access key example: wJalrXUtnYETAK7MDENG/bPxRfiCYEX-AMPLEKEY

- Click Download Credentials, and store the keys in a secure location.
- You can deactivate these keys (using Manage Access Keys from Security Credentials) and easily Delete Users as well once your project requirements are over.

- If you or your users prefer that they login to your AWS management console, this can be done as follows:
  - For ease of the users we can make the sign in page to a custom url

- Password can be auto generated (which sometimes creates problems). We can specify the password also.



Create User

Your 1 User(s) have been created successfully.

This is the last time these User security credentials will be available for download.

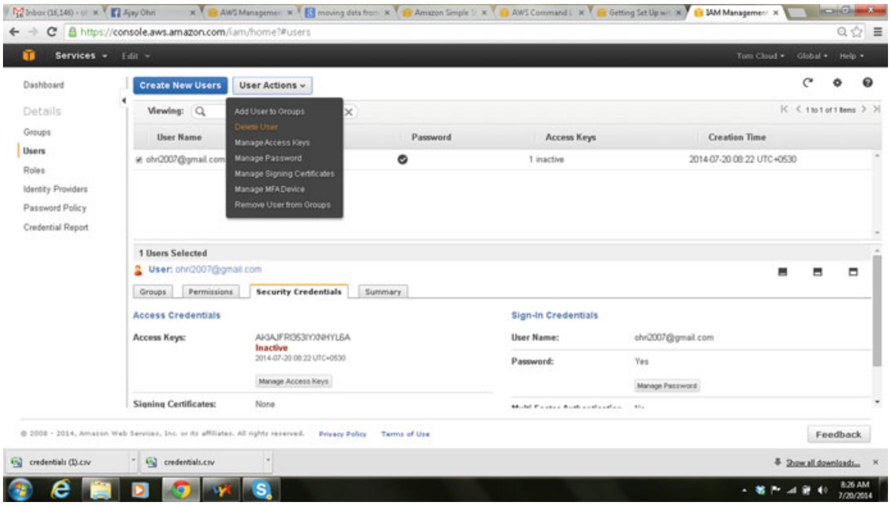
You can manage and recreate these credentials any time.

▾ Hide User Security Credentials

ohri2007@gmail.com

Access Key ID: AKIAJFRI353IYXNHYL6A

Secret Access Key: kqvLbygYd7Pj1BiBqJAqY0eZjSVsbeAeYd9LJJs



User Name	Password	Access Keys	Creation Time
ohri2007@gmail.com	✓	1 inactive	2014-07-20 08:22 UTC+0530

1 Users Selected

User: ohri2007@gmail.com

Access Credentials

Access Keys: AKIAJFRI353IYXNHYL6A Inactive 2014-07-20 08:22 UTC+0530

Signing Certificates: None

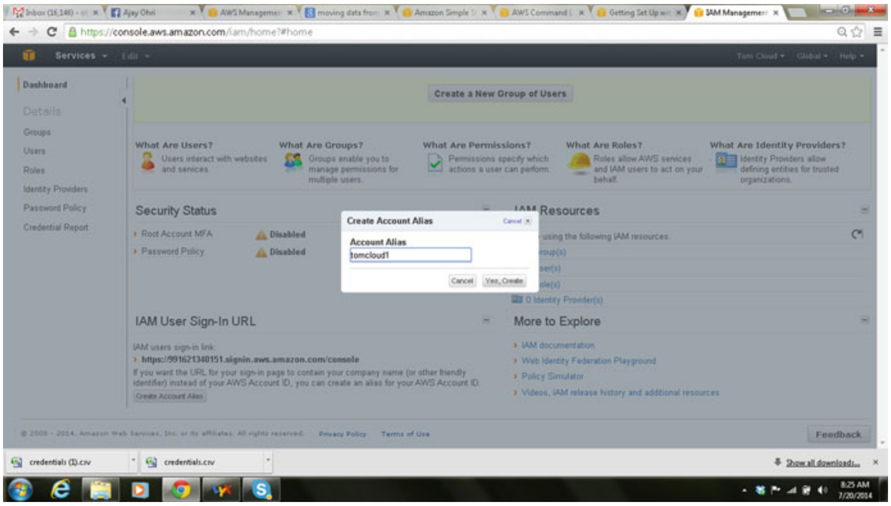
Sign-In Credentials

User Name: ohri2007@gmail.com

Password: Yes

8.3 Setting Up Security Control Groups for IP Address Level Access (Security Groups)

To truly secure your network, it should allow logins from certain IP addresses only, even if the security key and password credentials are given. This is of particular use in case of laptop theft.


https://console.aws.amazon.com/iam/home?#home

Create a New Group of Users

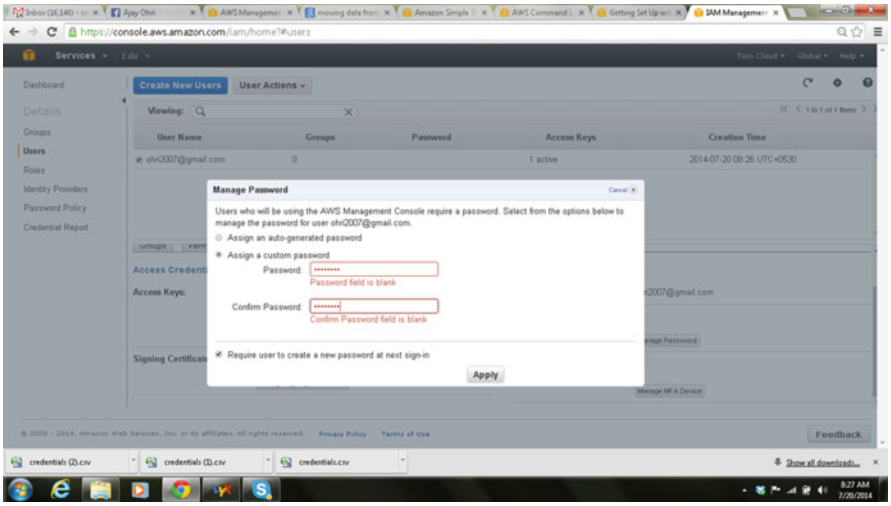
What Are Users? Users interact with websites and services.
What Are Groups? Groups enable you to manage permissions for multiple users.
What Are Permissions? Permissions specify which actions a user can perform.
What Are Roles? Roles allow AWS services and IAM users to act on your behalf.
What Are Identity Providers? Identity Providers allow defining entities for trusted organizations.

Security Status
- Root Account MFA — Disabled
- Password Policy — Disabled

Create Account Alias
Account Alias: tomcloud1
Cancel | Yes, Create

IAM User Sign-In URL
IAM users sign-in link:
https://991621340151.signin.aws.amazon.com/console
If you want the URL for your sign-in page to contain your company name (or other friendly identifier) instead of your AWS Account ID, you can create an alias for your AWS Account ID.
Create Account Alias

More to Explore
- IAM documentation
- Web Identity Federation Playground
- Policy Simulator
- Videos, IAM release history and additional resources


https://console.aws.amazon.com/iam/home?#users

User Name	Groups	Password	Access Keys	Creation Time
ohri2007@gmail.com	0		1 active	2014-07-20 08:26 UTC+0530

Manage Password
Users who will be using the AWS Management Console require a password. Select from the options below to manage the password for user ohri2007@gmail.com.
- ○ Assign an auto-generated password
- ● Assign a custom password
  - Password: •••••••••
  - Password field is blank
  - Confirm Password: •••••••••
  - Confirm Password field is blank

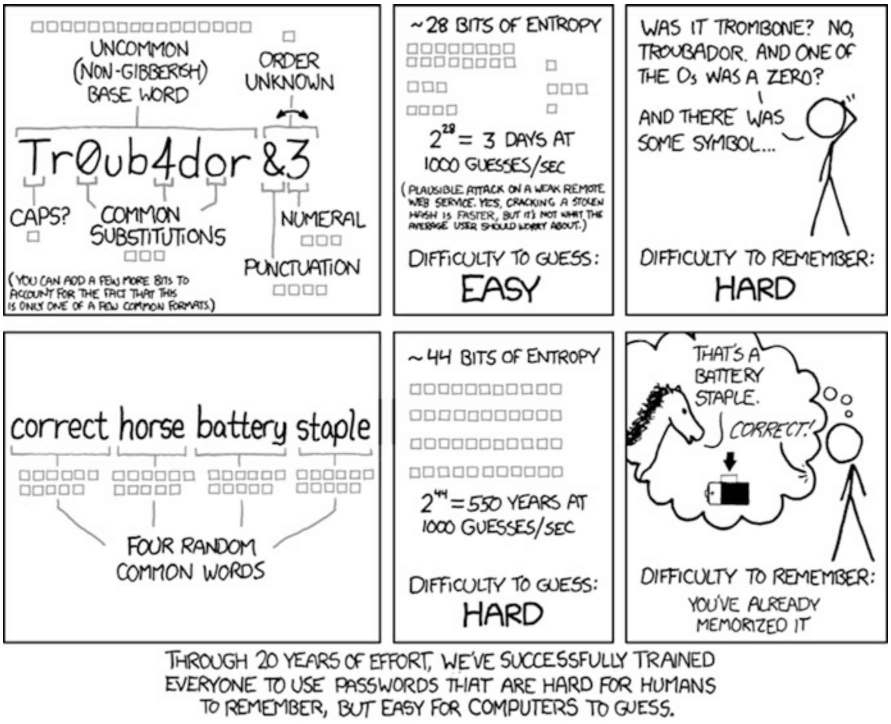
☑ Require user to create a new password at next sign-in

Apply

8.3.1 A Note on Passwords and Passphrases

A passphrase is easier to memorize and has more entropy (randomness) than a password which uses a combination of uppercase, lowercase, numbers, special symbols. A lucid example is given here to demonstrate this—perhaps you can share it with your IT administrator.

A salt is random data that is used as an additional input to a one-way function that hashes a password or passphrase. Salts also combat the use of rainbow tables


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor &3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

for cracking passwords. A rainbow table is a large list of pre-computed hashes for commonly used passwords. A dictionary attack is a technique for defeating encryption by trying to determine its decryption key or passphrase by trying hundreds or sometimes millions of likely possibilities, such as words in a dictionary.

8.3.2 A Note on Social Media's Impact on Cyber Security

Social media has basically created a lot more websites for where human users need to manually create and remember passwords. Unfortunately this has led to a lot more lax security as people basically use the same password in repeated locations, thus opening up places they can be attacked or breached by.

8.4 Monitoring Usage for Improper Access

Web analytics to monitor resource usage in the network and statistical tools to detect outliers in such activity is one solution. Anomaly/outlier detection systems looks for deviation from normal or established patterns within given data. In case of network security any threat will be marked as an anomaly.

You can use **wireshark** to capture network data (http://www.wireshark.org/download.html). A demo model to check for network intrusion (good or bad) is given by bigml at https://bigml.com/user/bigml/gallery/model/4f8a88921552687841000000

A public dataset for training purposes is the DARPA Intrusion Detection Data Set. http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/)

8.5 Basics of Encryption for Data Transfer (PGP- Public Key, Private Key)

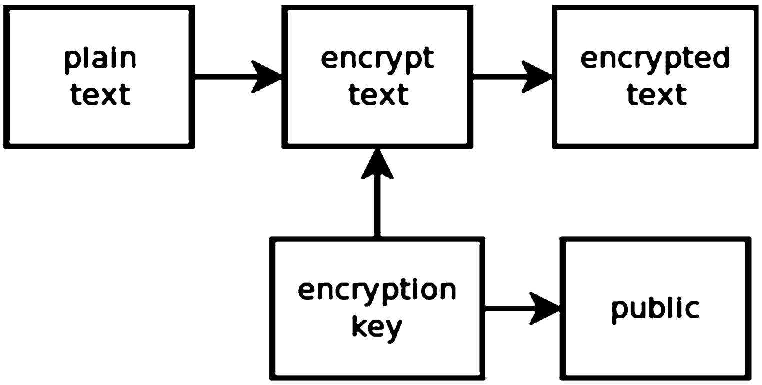
As a data scientist, we should know how to protect and defend our data. For sensitive data, it is best to encrypt it before transmission or storage. A public key is used to encrypt text, and a private key is used to decrypt text (we saw this earlier when we ran Puttygen to make keys before we transmit data for the cloud).

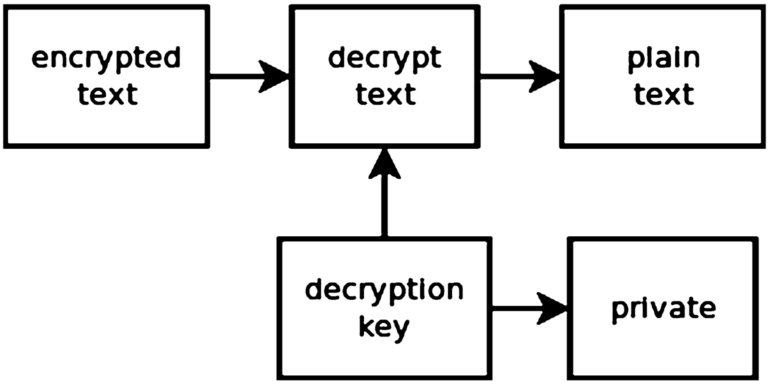
Here is a more lucid way of explaining how encryption works.

8.5.1 Encryption Software

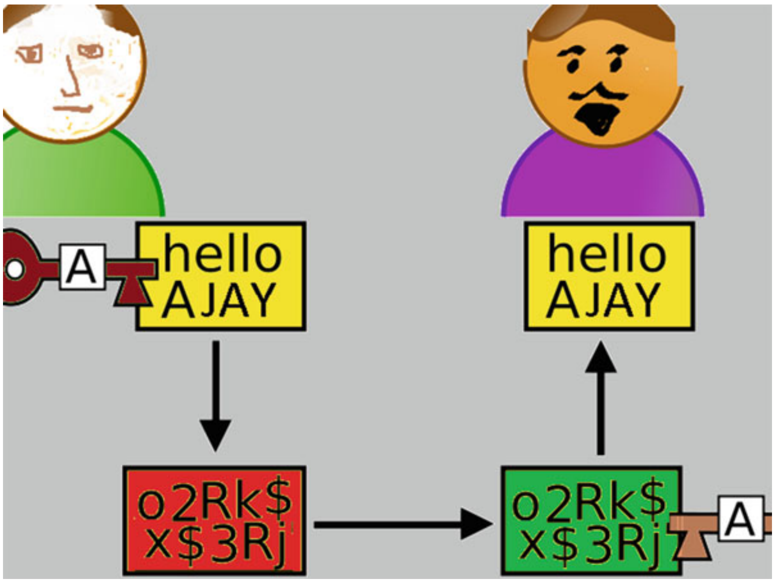
Software such as GNU PGP and openssl can help with encryption. GNU PGP is available at https://www.gnupg.org/ and the windows version is available at http://www.gpg4win.org/ GnuPG is a complete and free implementation of the OpenPGP standard as defined by RFC4880 (also known as PGP). GnuPG allows to encrypt and sign your data and communication, features a versatile key management system as well as access modules for all kinds of public key directories.

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. It is available at https://www.openssl.org/ and windows version is downloadable at http://slproweb.com/products/Win32OpenSSL.html





The digest package in R (http://cran.r-project.org/web/packages/digest/index.html) helps create hash digests of arbitrary R objects (using the md5, sha-1, sha-256 and crc32 algorithms) permitting easy comparison of R language objects, as well as a function 'hmac()' to create hash-based message authentication code.


A
hello AJAY
o2Rk$ x$3Rj
o2Rk$ x$3Rj
A
hello AJAY