Encryption mechanism used in the password hashing technique

Health2me utilises a comprehensive technique for password encryption which is compliant with HIPAA policies. It uses PBKDF2 algorithm which uses SHA256 as an underlying one way hash function.

A brief summary on PBKDF2

- PBKDF2 (Password-Based Key Derivation Function 2) is a key derivation function that is part of RSA Laboratories' Public-Key Cryptography Standards (PKCS) series, specifically PKCS #5 v2.0, also published as Internet Engineering Task Force's RFC 2898.

- **P**BKDF2 applies a pseudorandom function, such as a cryptographic hash (Health2me use SHA256) to the input password or passphrase along with a salt value and repeats the process many times to produce a derived key, which can then be used as a cryptographic key in subsequent operations.
- The added computational work which is known as key stretching makes password cracking much more difficult.
- As per the standard recommendation Health2me uses a minimum number of iterations is 1000, but the parameter can be increased based on requirement.
- Having a salt added to the password reduces the ability to use pre computed hashes (rainbow tables) for attacks, and means that multiple passwords have to be tested individually, not all at once. The standard recommends a salt length of at least 64 bits.

Detail approach (Code Implementation)

Please refer to the below link which mentions in detail about the hashing techniques and best practices to be followed for implementing password hashing

https://crackstation.net/hashing-security.htm

The actual code written in PHP called PasswordHash.php is provide by the author of the above website which follows http://creativecommons.org/licenses/by-sa/3.0/deed.en_US licensing format. That means it can be used or formatted for any purpose provided the author has been attributed in the code.

Health2me uses PasswordHash.php file as an API and implement below methods for encryption and password matching in the code.

Encryption

Method to encrypt the a password : create_hash(password)
Returns a random strings of character for a every new password.

Example:
Input password: 11111111 ,
Hash Results :
sha256:1000:ZSUhoNCeEiUhBeTUDn8CAQMN66KOVPn+:mHCZT5OmTOTu0PZyaI4LKdd+baMfnEnl

Method to match a password : validate_password(origin_password, hash_results)

Returns true or false depending on the whether matching happened or not.

Channel protection and network protocols used for the secure communication

Health2me uses various channels through which users can upload health records/data.All the channel are compliant with HIPAA security requirement.
These are primarily as below:

1) Cloud Channel (Getting files from in cloud storage services such as DropBox)
2) Email Channel (Receiving an Email securely)
3) Mobile Channel (Android or Ios app sending the file captured securely to the Health2me central server)
4)Web Channel (Upload records directly through web application)

The main web application(web channel) is using HTTPS protocol, which is standard protocol providing encrytion using standard TLS/SSL sockets over http protocol.

The cloud channel such as Dropbox for business provides 256 AES encryption for storing the files online.

Mobile Channel such as Android or IOS application uses HTTPS protocol to send files to the Health2me Server.

At-Rest Encryption details for storing the files (Health2me details)

At present the files present in the health2me central server is kept in Packages_Encrypted and PackagesTH_Encrypted using standard AES based openSSL encryption protocol.

For more details on Open SSL, click on the here

For Encryption following commands are being used

```
echo [password] | openssl.exe aes-256-cbc -pass stdin -salt -in [file_source]-out
[file_destination]
)
```

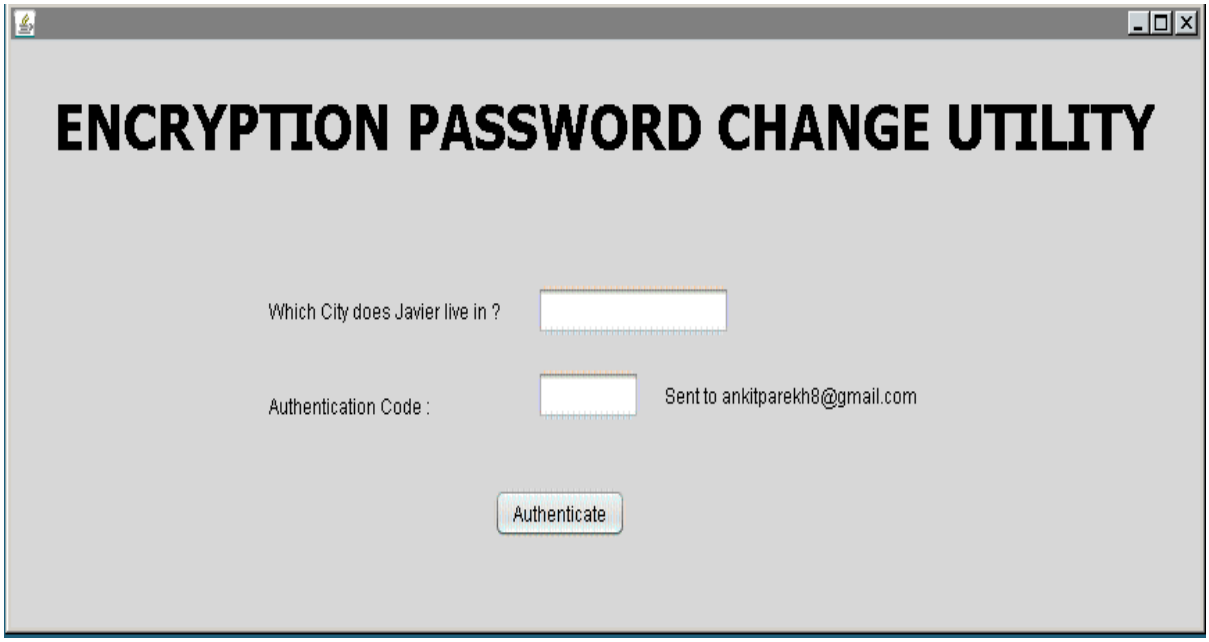
For Decryption following commands are being used

```
@echo off
echo [password] |openssl.exe aes-256-cbc -pass stdin -d -in [file_source] -out
[file_destination]
```

The passwords are stored in the database table called **encryption_pass** from where the application takes the latest password and encrypts or decrypts the files.

A standalone application has been created to reset passwords used for encryption and decryption process.

A screenshot of the application is as below



* Steps for details on how certificates are configured and added in the https

HTTPS certificate details

Health2me main web application uses SSL certificate signed and verified by certificate authority Comodo group based in New Jersey, USA which provides AES-256 bit encryption compliant as per HIPAA policy.

For more details, see the below mail details

Your COMODO SSL Certificate for www.health2.me is attached!

Dear Carlos Ubinas,
Thank you for placing your order. We are pleased to announce that your COMODO SSL Certificate for www.health2.me has been issued.
To help reduce domain name mismatch warnings, we have also included the domain name health2.me in your certificate.
We strongly recommend that you click here for instructions to ensure that your certificate is installed and your webserver is configured correctly.
Attached to this email you should find a .zip file containing:

- Your COMODO SSL Certificate - www_health2_me.crt
- Your Apache "bundle" file - www_health2_me.ca-bundle

You can also find your COMODO SSL Certificate for www.health2.me in text format at the bottom of this email.
Should you have any questions or issues you would like to discuss, please do not hesitate to contact us.

Kind Regards,

Comodo Security Services
Support Telephone: +1.703.581.6361
Support Website: http://support.comodo.com
Validation Docs Fax: US and Canada +1.866.831.5837 / Worldwide +1.801.303.9291

The SSL folder residing in C:\apache24\conf contains below files which have been modified to facilitate the inclusion of comodo certificates.

- ssl.key (contains the server.key given by comodo for health2.me)
- ssl.crt (contains the actual certificate provided by comodo for health2.me)
- httpd (This file is used to enable and configure the https)