**Lab 1 - Encryption and Decryption**

Please write a program to encrypt and decrypt passwords. You’re welcome to use the following information for your program. Please test your program 10 times for 10 different passwords.

You may reference <http://www.code2learn.com/2011/06/encryption-and-decryption-of-data-using.html>

There are many problems when you try encrypting a string such as password, credit card numbers, and phone numbers.   
1. Which algorithm to use.  
2. How to store the generated Key in the database.  
3. Should I use MD5, AES etc?  
Here is the question to all your answers. After spending sometime on this, I finally got the best algorithm that a person can use to encrypt and decrypt data while he/she also wants to store those encrypted strings and later on want to decrypt it while retrieving the data.  
Many people face problem while decrypting the encrypted data as the KEY used for encryption if stored as String in database then it becomes very tough to use that string as the KEY. So, below is the code where you only need to store the encrypted code and not the key. The decryption will take place as when wanted.  
For encryption, we must use a secret key along with an algorithm. In the following example, we use an algorithm called **AES 128** and the bytes of the word "**TheBestSecretKey**" as the secret key (the best secret key we found in this world). AES algorithm can use a key of 128 bits (16 bytes \* 8); so we selected that key.

**package** nomad;

**import** java.security.**\***;

**import** java.security.spec.InvalidKeySpecException;

**import** javax.crypto.\*;

**import** sun.misc.**\***;

**public** **class** AESencrp {

**private** **static** **final** **String** ALGO = "AES";

**private** **static** **final** byte[] keyValue =

**new** byte[] { 'T', 'h', 'e', 'B', 'e', 's', 't','S', 'e', 'c', 'r','e', 't', 'K', 'e', 'y' };

**public** **static** **String** encrypt(**String** Data) **throws** **Exception** {

**Key** key = generateKey();

Cipher c = Cipher.getInstance(ALGO);

c.init(Cipher.ENCRYPT\_MODE, key);

byte[] encVal = c.doFinal(Data.getBytes());

**String** encryptedValue = **new** BASE64Encoder().encode(encVal);

**return** encryptedValue;

}

**public** **static** **String** decrypt(**String** encryptedData) **throws** **Exception** {

**Key** key = generateKey();

Cipher c = Cipher.getInstance(ALGO);

c.init(Cipher.DECRYPT\_MODE, key);

byte[] decordedValue = **new** BASE64Decoder().decodeBuffer(encryptedData);

byte[] decValue = c.doFinal(decordedValue);

**String** decryptedValue = **new** **String**(decValue);

**return** decryptedValue;

}

**private** **static** **Key** generateKey() **throws** **Exception** {

**Key** key = **new** SecretKeySpec(keyValue, ALGO);

**return** key;

} }

We use "generateKey()" method to generate a secret key for AES algorithm with a given key.  
Below is the code how you can use the above encryption algorithm.

**package** nomad;

**public** **class** Checker {

**public** **static** void main(**String**[] args) **throws** **Exception** {

**String** password = "mypassword";

**String** passwordEnc = AESencrp.encrypt(password);

**String** passwordDec = AESencrp.decrypt(passwordEnc);

**System**.out.println("Plain Text : " + password);

**System**.out.println("Encrypted Text : " + passwordEnc);

**System**.out.println("Decrypted Text : " + passwordDec);

} }

### NOTE : I have got emails from user saying that the above code gives error when using in ECLIPSE.

### Error like: Access restriction: The type BASE64Decoder is not accessible due to restriction on required library C:\Program Files\Java\jre6\lib\rt.jar; So, to avoid this, do the following:

**\***Go to **Window**-->**Preferences**-->**Java**-->**Compiler**-->**Error/Warnings**.

**\***Select **Deprecated and Restricted API**. Change it to warning.

**\***Change **forbidden and Discouraged Reference** and change it to warning. (or as your need.)

Note: 12-12-2013: One of our readers has given a solution about some issue:

Issue: Issue of Access Restriction

Solution: Removing JRE system Library then adding it back from Build Path settings in the project properties.

==========================================================================.

**How to submit your Lab or Project Assignment (PA)?**

(1) Each program must be well-documented with block comments and proper line comments. The

beginning of each program must have a block comment to show your name, date, and purpose.

The following is an example of block and line comments.

/\*

Author: Dr. Simon Lin

Date: 1/10/2016

Purpose: To check whether a string is a palindrome.

\*/

int yearCount = 0 ; // yearCount to count # of years being processed so far

(2) You must submit the following items as attachments through sakai.apu.edu.

(a) All source programs (i.e., all **.java** files), and

(b) One WORD document (i.e., **.doc** or **.docx** file) containing all source programs’ listing and the

output of at least **3** test runs for each program (if applicable). For this lab, you must test it **10 times**. ==========================================================================.

Grading Rubric:

You got \_\_\_ points out of 100 for **CS250 Lab 1**. Thank you for your excellent/good work.

[ ] -10 points for each day late.

[ ] 20 points – Your program must be fully tested **10 times** with different data if possible.

[ ] 10 points – Your program must be well-documented.

[ ] 10 points – You must follow the program specification to develop your programs properly and completely.

[ ] 20 points – You must submit your WORD document.

[ ] -5 points if your program did not have block comment to show your name, date, and purpose.

[ ] 5 points – Your WORD document must show “**CS250 Lab 1**” and your full name on the **header**.

[ ] 5 points – Your WORD document must be page-numbered on the footer with format “Page 2 of 15” (for example).

==================================================================================.

Please use this Word document as a template for your Word document to be submitted. Please delete everything above.

Please copy all your source program into here:

/\*

Author: Gao, Jie

Date: 2/3/2016

Purpose: To check Encryption and Decryption.

\*/

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPasswordField;

**public** **class** Checker {

**public** **static** **void** main(String[] args) **throws** Exception {

JLabel label = **new** JLabel("Please enter your password:");

JPasswordField jpf = **new** JPasswordField();

JOptionPane.*showConfirmDialog*(**null**,

**new** Object[]{label, jpf}, "Password:",

JOptionPane.***OK\_CANCEL\_OPTION***);

String password = **new** String(jpf.getPassword());

String passwordEnc = AESencrp.*encrypt*(password);

String passwordDec = AESencrp.*decrypt*(passwordEnc);

System.***out***.println("Plain Text : " + password);

System.***out***.println("Encrypted Text : " + passwordEnc);

System.***out***.println("Decrypted Text : " + passwordDec);

}

}

Please copy all your testing output into here:

Plain Text : mypassword

Encrypted Text : 2t3jcv4Q9ze5CqJrLUnmIQ==

Decrypted Text : mypassword

Plain Text : aglndkkvdbke

Encrypted Text : XLZgW2VJiVZ9U9zQRome2g==

Decrypted Text : aglndkkvdbke

Plain Text : gregvdgsad

Encrypted Text : K5Toez+9zcQDB/lLc+CUgg==

Decrypted Text : gregvdgsad

Plain Text : thisismypassword

Encrypted Text : l6i54eL0U5BRDpFIFQcQtUhw/lFbiofDyxnO0LO/zsw=

Decrypted Text : thisismypassword

Plain Text : tahlfgaiujde

Encrypted Text : GrMPOIgElPNiUjAIoqzOag==

Decrypted Text : tahlfgaiujde

Plain Text : ghhrfejbv

Encrypted Text : 5CU21wuYVBqBTGI08di00w==

Decrypted Text : ghhrfejbv

Plain Text : asdgrredggthc

Encrypted Text : nbsU8iS2GEEIrFRQ08feDw==

Decrypted Text : asdgrredggthc

Plain Text : 111111111111

Encrypted Text : QrcCw9+FXK+ZjQBef5BahA==

Decrypted Text : 111111111111

Plain Text : djkgjdgfkeea

Encrypted Text : RzyUz0YcgAScB4u9/ruFDQ==

Decrypted Text : djkgjdgfkeea

Plain Text : 123456

Encrypted Text : oJfTtchpM9WC/4Oqpu7FZQ==

Decrypted Text : 123456