How to Decrypt MD5 in Java? (Real Solution with Code Sample)

infosecscout.com/decrypt-md5-in-java

Patrick Fromaget

Everybody wants a solution to decrypt MD5 hashs, and Java programmers are also interested 🙂
In this article, you'll learn how to do this, and also discover a few things about the MD5 algorithm.

The MD5 cryptographic algorithm only works one way. It's possible to crypt a word into MD5 with Java, but there is no reverse function. In case of a password verification, the best practice is to also crypt to entered password and compare the result with the original one.

In this tutorial, I'll start by a quick reminder about the MD5 algorithm. Then we'll see how to verify passwords in Java, without having to decrypt them.
And just in case you are here for this, we'll conclude with a solution to really decrypt MD5 hashs with Java.

By the way, if you are interested in how MD5 decryption really works, I highly encourage you to take a look at my e-book "The Secrets of MD5 Decryption"here. It explains everything you need to know, going directly to the point with practical examples you can

test on your computer. You don't need any hardware to get started, just a few tips I give in this book.

Master Linux Commands
Your essential Linux handbook
Want to level up your Linux skills? Here is the perfect solution to become efficient on Linux. **20% off today!**

Download now

Reminder about the MD5 algorithm

I often start with an introduction about the MD5 algorithm on this blog because most people have difficulties to understand the logic behind the MD5 algorithm.

MD5 encryption

So, **MD5 is a cryptographic algorithm that generate a string with 32 hexadecimal characters, whatever the word or text length you try to encrypt.**
Even an ISO file from several gigabytes can be hashed in 32 characters with the MD5 algorithm.

Pseudo-code example:

```
MD5("MD5Online") = d49019c7a78cdaac54250ac56d0eda8a
```

This information provides the following algorithm results:

- **The output is always 32 characters long**, but you can hash anything in 32 characters.
- So, the MD5 algorithm output is not unique. Two word or files can have the same MD5 hash.
- Given this information, it's not possible to reverse a hash to the original word.

If you are interested to understand all the details, I recommend reading this page (Wikipedia) or picking a course or book from my resource page.

MD5 decryption

So, why the MD5 algorithm is so fascinating if decrypting hashes is not possible?
The MD5 algorithm has a weakness we can exploit, each time you create a MD5 hash of a word, you get the same result.
As this algorithm was the principal one in the world a few decades ago, many databases exists with the corresponding word for each MD5 they know.

So, there is no decryption algorithm for MD5, but there is a solution.
For example, you now know that the MD5 hash from “MD5Online” is d49019c7a78cdaac54250ac56d0eda8a.
If someone is looking for the word corresponding to this hash, there is a good chance that “MD5Online” was the original password.

That’s what is used for MD5 decryption in general.
And especially **on MD5Online.org, we have a huge database with over a trillion hashes stored inside**. You can access this database with our tools.
There are other solutions, but it’s the main one.

How to encrypt and decrypt a MD5 password with Java?

Theory (pseudo-code)

In any language, the MD5 function is really fast to encrypt a password.
So, you can use it in your application without any performance issue.

That’s the reason why some developers are using the MD5 algorithm to encode passwords in their database.
To verify the login credentials, they just encrypt the typed password in MD5 and compare this hash to the one stored in the database.
If there is a match, we consider that the login is valid (even if the encryption is not unique, it’s not a big deal).

The pseudo-code can look like this:

```
IF (MD5(PASSWORD_ATTEMPT) == DATABASE_PASSWORD)
THEN LOGIN_SUCCESS();
ELSE LOGIN_ERROR();
```

We’ll now see how to do this in Java.

Java examples

Encrypting password in MD5 with Java

Unfortunately, Java doesn’t include a built-in function to encrypt passwords in MD5. But don’t worry, there is way, we just need to import the java.security library.
By the way, this method also works for SHA-1 and SHA-256.

If you want to try it, here is the code to use to encrypt a string into MD5:

```
import java.security.*;
import java.math.*;

public class tests {

   public static String MD5(String s) throws Exception {
      MessageDigest m=MessageDigest.getInstance("MD5");
      m.update(s.getBytes(),0,s.length());
      return new BigInteger(1,m.digest()).toString(16);
   }

   public static void main(String args[]) throws Exception {
      //Encode
      String md5 = MD5("MD5Online");
      System.out.println("MD5 hash: "+md5);
   }
}
```

Yes, other language like PHP and Python are way easier when you need to do something with MD5 hashs, but that's the best way I found to do this in Java.

Obviously, this class only display the MD5 hash on your screen, you can adjust it to do anything you want (like storing the password in a database). The important thing to remember here is the MD5 function you need to create before using MD5 hashs in your code.

MD5 password verification in Java

Once the password is encrypted and stored in the database, you can use a simple condition to check that the login attempt you try to validate is correct.
The idea is to compare the input password to the stored password for this user:

```
#The first part depends on the framework you are using
#Let's say you get a password in clear format from the request:
String password = "MD5Online";

#The second part depends on the database you are using
#But your password is hashed in the database,
#so you get a string like:
String stored_password = "d49019c7a78cdaac54250ac56d0eda8a";

if(MD5(password).equals(stored_password)) {
   #Authentication success, do whatever was asked
   System.out.println("OK");
}
else {
   #Doesn't match, display error and ask the password again
   System.out.println("KO");
}
```

Obviously, you need the MD5 function presented earlier for this code to work.

I hope you understand the idea.
Just get the two passwords in MD5 format and compare them with a simple condition.
You never need to decrypt the one stored in the database, except for hacking, that's what we'll see in the next part.

The only solution to decrypt passwords in Java

If you are still reading these lines, that's because you are here to learn how to really decrypt a list of MD5 passwords and get the plain text as a result.
I have a solution for you.

MD5Online.org is offering an API you can use in Java (or any other language that can handle HTTP requests), to try to decrypt each one of your hashes with our database.
This is a paid service, but it's really affordable, and avoid having a huge server at home that do brute-force all day.
If you want further information, check this page that explains everything.

Once your account is created with a few credits to test (the first package costs €1), **you can get your API key in your account and try this script in Java:**

```
import java.security.*;
import java.math.*;
import java.net.*;
import java.io.*;

public class tests {

     public static String MD5(String s) throws Exception {

        MessageDigest m=MessageDigest.getInstance("MD5");
        m.update(s.getBytes(),0,s.length());
        return new BigInteger(1,m.digest()).toString(16);

     }

     public static String Decrypt(String md5_hash) throws Exception {

          String api_key = "YOUR_VIP_KEY";
          URL md5online = new URL("https://www.md5online.org/api.php?
d=1&p="+api_key+"&h="+md5_hash);
          BufferedReader in = new BufferedReader(new
InputStreamReader(md5online.openStream()));

          String result = "";
          String inputLine;
          while ((inputLine = in.readLine()) != null)
             result = result+inputLine;
          in.close();

          return result;
    }

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

           //Encode
           String md5 = MD5("MD5Online");
           System.out.println("MD5 hash: "+md5);

           //Decode
           String word = Decrypt(md5);
           System.out.println("API Result: "+word);

    }
}
```

The “&d=1” parameter in the URL is here to display any error message.
Also, feel free to contact me if you don’t know how to fix it.

Anyway, this code is working fine on my side:

```
MD5 hash: d49019c7a78cdaac54250ac56d0eda8a
API Result: MD5Online
```

Conclusion

That’s it, you now know how to decrypt MD5 passwords in Java, with the two solutions depending on your situation:

- If the goal is to verify the password, you don’t need to decrypt them at all (and you know how to do this)
- If your main purpose is try to hack passwords and find the corresponding word, you can use our API at MD5Online.

If this tutorial was useful for you, please share it on your favorite social network!

Whenever you’re ready for more security, here are things you should think about: