We certify that we are the original authors of this paper. No part of it is plagiarized.

Kai Li, Xintang He, Zheng Liu
Names of the authors are listed alphabetically.

```
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.HashMap;

public class Regulation {

    private String origin_address;
    private String encrypted_address;

    private Regulation(String origin_address) {
        this.origin_address = origin_address;
    }

    private void encrypted_address(int target_length) {
        if (this.encrypted_address == null) {
            String diff = getSHA256(this.origin_address.substring(0,
target_length - this.origin_address.length()));
            char[] c = new char[target_length];
            for(int i = 0; i < diff.length(); i++){
                c[2*i+1] = diff.charAt(i);
                c[2*i] = this.origin_address.charAt(i);
            }
            for(int i = 2*diff.length() ; i < target_length; i++){
                c[i] = this.origin_address.charAt(i);
            }
            StringBuilder sb = new StringBuilder();
            for(char ch : c){
                sb.append(c);
            }
            this.encrypted_address = sb.toString();
        } else {
            return;
        }
    }

    private String get_Original() {
        if (this.isValid()) {
            return this.origin_address;
        } else {
            return "Error: Illegal Address.";
        }
    }

    private boolean isValid() {
        int flag = 0;
        int j = 0;
        for(int i = 0; i < this.origin_address.length(); i++){
            if(flag < 2){
                if(this.origin_address.charAt(i) ==
this.encrypted_address.charAt(j)){
                    j ++;
                    flag = 0;
                }
                flag ++;
            }else{
                return false;
            }
        }
        return true;
    }

    public String getSHA256(String str) {
        MessageDigest messageDigest;
        String encodestr = "";
        try {
            messageDigest = MessageDigest.getInstance("SHA-256");
            messageDigest.update(str.getBytes("UTF-8"));
            encodestr = byte2Hex(messageDigest.digest());
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        return encodestr;
    }

    private String byte2Hex(byte[] bytes) {
        StringBuffer stringBuffer = new StringBuffer();
        String temp = null;
        for (int i = 0; i < bytes.length; i++) {
            temp = Integer.toHexString(bytes[i] & 0xFF);
            if (temp.length() == 1) {
                stringBuffer.append("0");
            }
            stringBuffer.append(temp);
        }
        return stringBuffer.toString();
    }

}
```