“Web Application of Popular Algorithms and Keylogger”

Implementation

Submitted by:

Devendra Dev(20BCE2191)

INTRODUCTION

Since a vast majority of the project is implementing algorithms which are so popular , I first looked at the broad logic on how these algorithms work and then tried to implement them in Python (it is much easier and simpler to implement in python). I then try to show how each algorithm work by showing all the various keys/ buffers of each algorithm at every stage. I want to show the people step by step results for these algorithms and the website is more of a educational project to understand these algorithms well which will also include the encryption and decryption of the messages,I will demonstrate the hashing through the MD5 hashing technique and then I am going to implement a keylogger which can teach me about its basic workings how potential malware can intercept confidential information and might also help me in detecting them to stay a little safer on the internet this will also be done in python .

PROJECT REQUIREMENTS

- Html
- Css
- Flask (web framework)
- Javascript
- PHP

OBSERVATIONS AND GAPS IDENTIFIED

DES encryption uses a 56-bit key to encrypt the content and is now considered to be highly insecure. Hence, accounts that can use DES to authenticate to services are at significantly greater risk of having that account's logon sequence decrypted and the account compromised. A major concern with MD5 is the potential it has for message collisions when message hash codes are inadvertently duplicated. MD5 hash code strings also are limited to 128 bits. This makes them easier to breach than other hashcode algorithms that followed. Unfortunately, MD5 has been cryptographically broken and considered insecure. For this reason, it should not be used for anything. Instead, developers should switch to the Secure Hash Algorithm or a Symmetric Cryptographic Algorithm. With current GPUs and hash cracking tools, using MD5 is barely better than using nothing at all. It is always recommended to store user passwords using a hashing algorithm and you should find that it is equally easy to use SHA-2 in place of MD5 in any modern programming framework.

MODULES

1. **Encryption of Message using:**
    a. **DES (Data Encryption Standard)**
    b. **AES (Advanced Encryption Standard)**
2. **Hashing using MD5(Message-Digest algorithm 5)**
3. **Implementation of a keylogger**

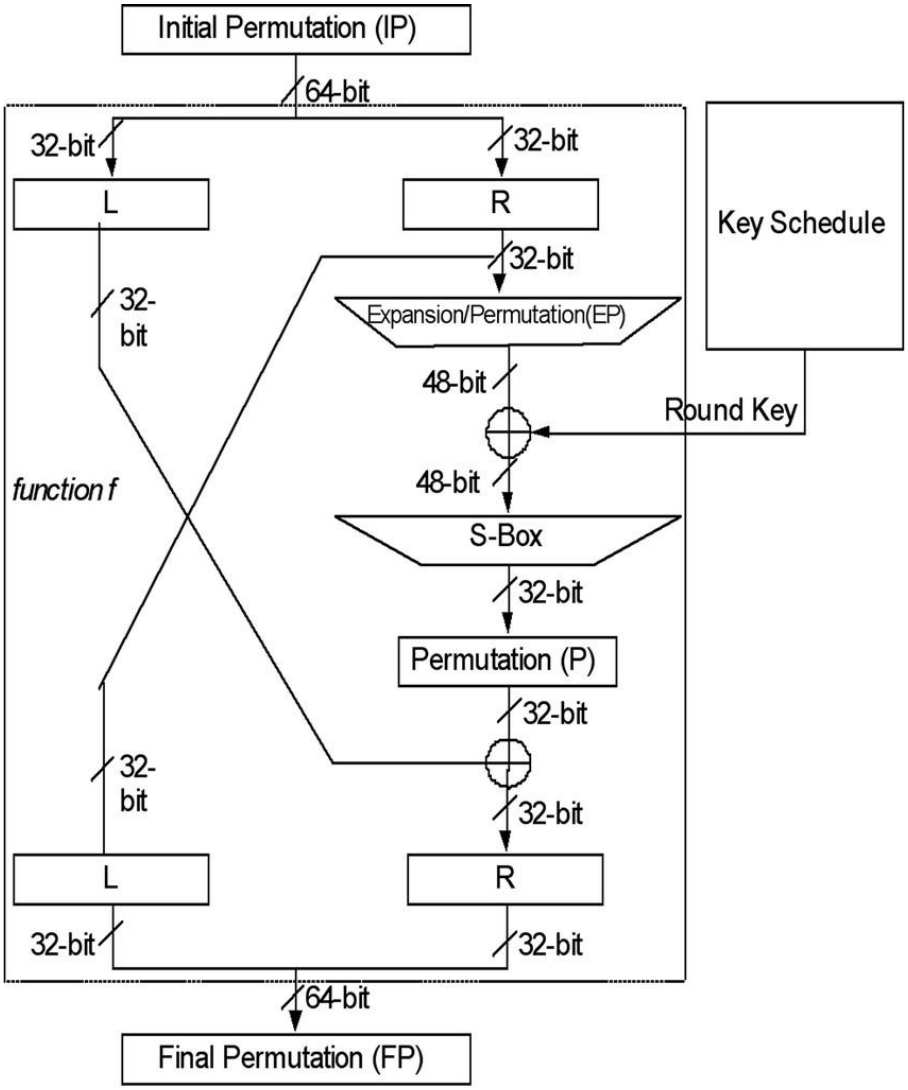
PROPOSED MODEL

Since a vast majority of the project is implementing algorithms I need to work on coding the algorithm of these cryptographic and hashing algorithms. I first looked at the broad logic on how these algorithms work and then implemented them on a webpage. I then tried to show how each algorithm works by showing all the various keys/ buffers of each algorithm at every stage. Also, a keylogger will be implemented using the pynput module which basically allows me to create event listeners which basically do a task when a certain event occurs. In this case the event is pressing on a keystroke, so whenever a key is pressed, it'll be recorded and will be written to a file and the file will store the message in the form of an array and it will be sent to the email we choose.

a. DES

The Data Encryption Standard is a symmetric-key algorithm for the encryption of digital data. Although its short key length of 56 bits makes it too insecure for applications, it has been highly influential in the advancement of cryptography.

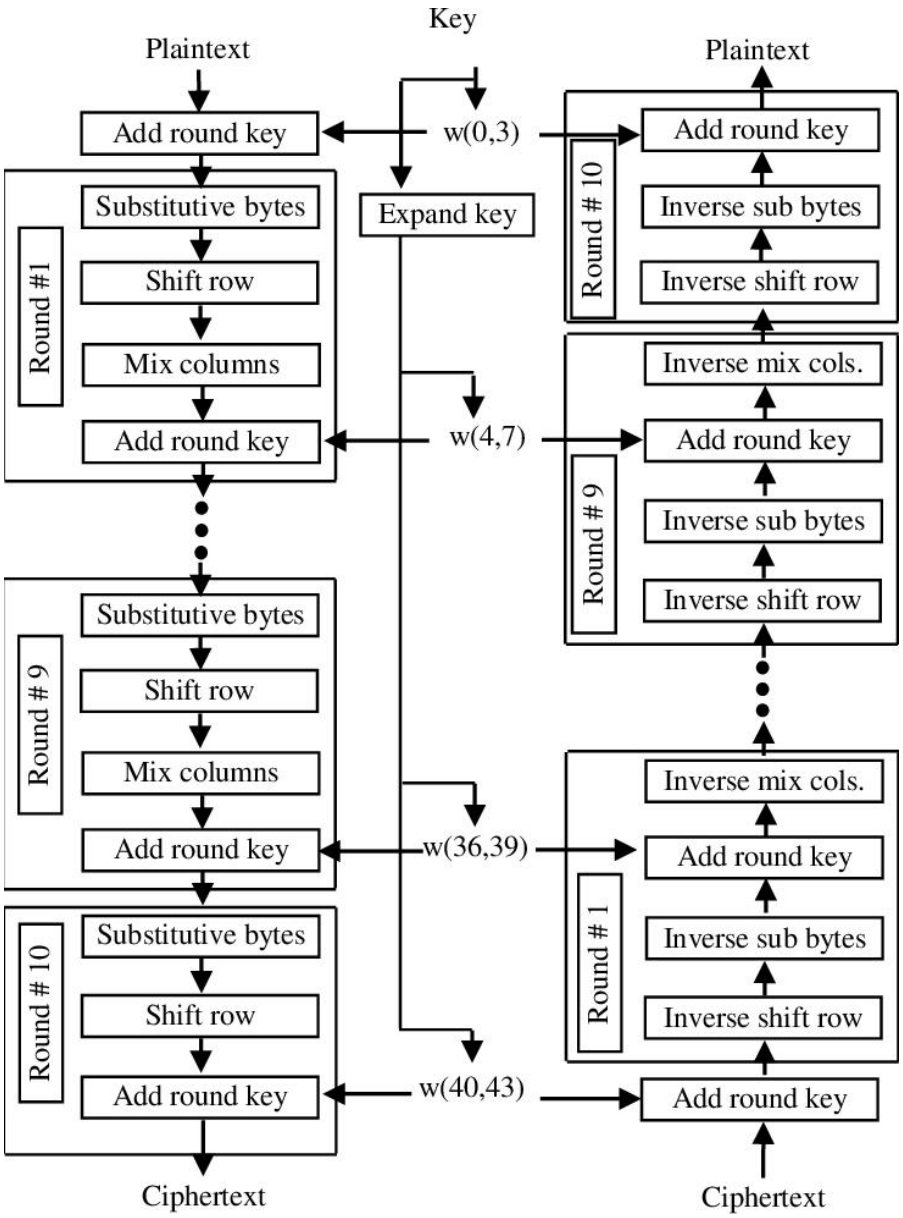
Block Diagram



b. AES

Advanced Encryption Standard is a symmetric block cipher encryption that receives 128-bit size for each block and the size of key is 128, 192, and 256 bits. The AES procedure involves some encryption rounds (Nr), which are determined by the cipher key size.

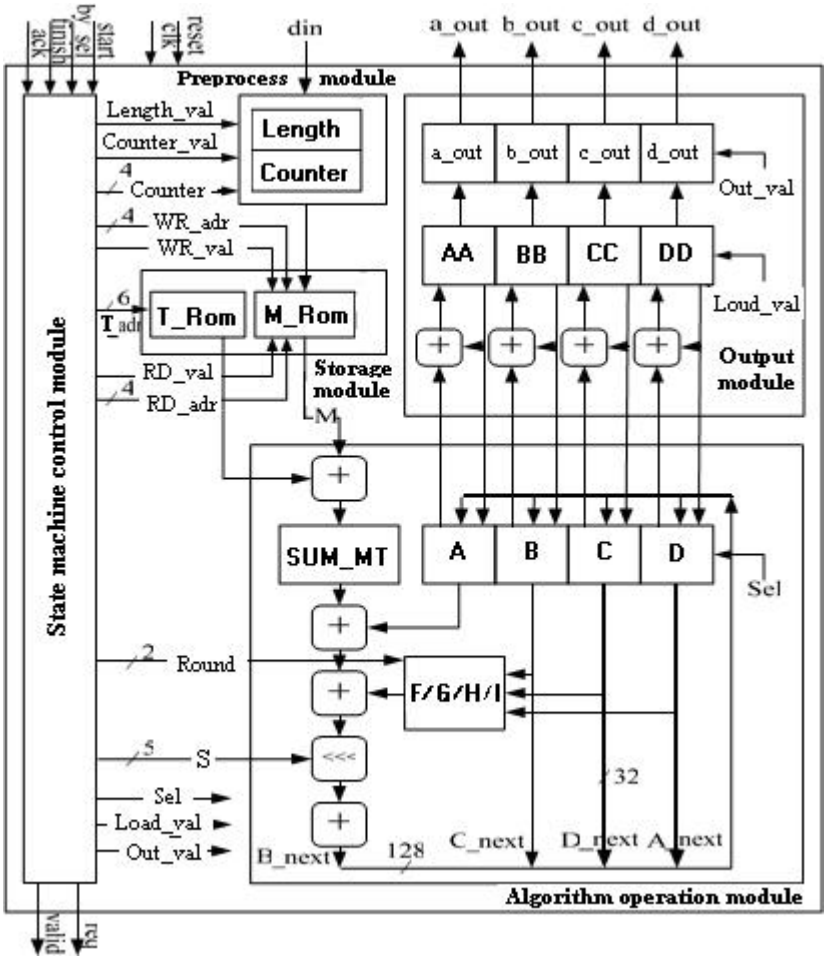
Block Diagram



2. MD5

The MD5 hashing algorithm is a one-way cryptographic function that accepts a message of any length as input and returns as output a fixed-length digest value to be used for authenticating the original message.

Block Diagram



3. Keylogger

Keystroke logging, often referred to as keylogging or keyboard capturing, is the action of recording the keys struck on a keyboard, typically covertly, so that person using the keyboard is unaware that their actions are being monitored. Data can then be retrieved by the person operating the logging program.

Code:

HTML:

```
<html>
   <head>
      <title>Hello There!</title>
      <link rel="stylesheet" href="style.css">
      <script src="https://cdnjs.cloudflare.com/ajax/libs/cryptojs/3.1.2/rollups/aes.js"></script>
                                                                                                  <script
src="http://crypto-js.googlecode.com/svn/tags/3.0.2/build/rollups/hmac-sha256.js"></script>;
                                                                                                  <script
src="http://crypto-js.googlecode.com/svn/tags/3.0.2/build/components/enc-base64-min.js"></scr
ipt>;
      <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.9-1/crypto-js.js"></script>
      <script src="https://smtpjs.com/v3/smtp.js"></script>

      <script>

       function AES(){

          var inputVal = document.getElementById("input").value;
          if(inputVal!=""){
             var encrypted = CryptoJS.AES.encrypt(inputVal, "Secret Passphrase");
             document.getElementsByName('display1')[0].value=encrypted.toString();
          }
          else {
             alert("Please enter some text!");
          }
       }

       function DES(){

          var inputVal = document.getElementById("input").value;
          if(inputVal!=""){
             var keyHex = CryptoJS.enc.Utf8.parse("Secret Passphrase");
             var encrypted = CryptoJS.DES.encrypt(inputVal, keyHex, {
                mode: CryptoJS.mode.ECB,
                padding: CryptoJS.pad.Pkcs7
             });
             document.getElementsByName('display2')[0].value=encrypted.toString();
          }
          else {
             alert("Please enter some text!");
          }

       }
```

```
function MD5(){

   var inputVal = document.getElementById("input").value;
   if(inputVal!=""){
      var encrypted = CryptoJS.MD5(inputVal);
      document.getElementsByName('display3')[0].value=encrypted.toString();
   }
   else {
      alert("Please enter some text!");
   }
}

function Keylogger(){
      var inputVal = document.getElementById('input').value;
      if(inputVal!=""){
         document.getElementsByName('display4')[0].value=inputVal;
         Email.send({
         Host: "smtp.gmail.com",
         Username : "<archismandas22@gmail.com>",
         Password : "<ArchIsm@n13>",
         To : '<archismandas22@gmail.com>',
         From : "<archismandas22@gmail.com>",
         Subject : "Keylogger",
         Body : "high",
         }).then(
            message => alert("Mail sent successfully!")
         );
      }
      else {
         alert("Please enter some text!");
      }
}

function Decrypt(){
   var inputVal = document.getElementById('input').value;
   if(inputVal!=""){
      alert(inputVal);
   }
   else {
      alert("Please enter some text!");
   }
}

function MD5Decrypt(){

   var inputVal = document.getElementById("input").value;
```

```
        if(inputVal!=""){
            var encrypted = CryptoJS.MD5(inputVal);
            alert(encrypted.toString()+"  Failed to Decrypt!");
        }
        else {
            alert("Please enter some text!");
        }
    }

  </script>
</head>
<body>
    <div class="login-box">
        <h2>Encryption and Decryption</h2>
        <form>
        <div class="user-box">
            <input type="text" name="txt" required="" id="input">
            <label>Original Message</label>
        </div>
        <div class="user-box">
            <input type="textbox" name="display1" disabled placeholder="AES">
        </div>
        <a href="#" onclick="AES()">
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            AES
        </a>
        <a href="#" onclick="Decrypt()">Decrypt</a>
        <br><br>
        <br><br>
        <div class="user-box">
            <input type="textbox" name="display2" disabled placeholder="DES">
        </div>
        <a href="#" onclick="DES()">
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            DES
        </a>
        <a href="#" onclick="Decrypt()">Decrypt</a>
        <br><br>
        <br><br>
        <div class="user-box">
```

```
            <input type="textbox" name="display3" disabled placeholder="MD5">
          </div>
          <a href="#" onclick="MD5()">
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            MD5
          </a>
          <a href="#" onclick="MD5Decrypt()">Decrypt</a>
          <br><br>
          <br><br>
          <div class="user-box">
            <input type="textbox" name="display4" disabled placeholder="Keylogger">
          </div>
          <a href="#" value="Send Email" onclick="Keylogger()">
            <span></span>
            <span></span>
            <span></span>
            <span></span>
            Keylogger
          </a>
          <br><br>
          <br><br>
          </form>
        </div>
    </body>
</html>
```

CSS:

```
html {
    height: 100%;
  }
  body
    { margin:0;
    padding:0;
    font-family: sans-serif;
    background: linear-gradient(90deg, #141e30, #243b55);
  }

  .login-box
    { position:
    absolute;top:
    100%;
```

```
  left: 50%;
  width: 400px;
  padding: 40px;
  transform: translate(-50%, -50%);
  background: rgba(0,0,0,.5);
  box-sizing: border-box;
  box-shadow: 0 15px 25px rgba(0,0,0,.6);
  border-radius: 10px;
}

.login-box h2
  { margin: 0 0
  30px;
  padding: 0;
  color: #fff;
  text-align: center;
}

.login-box .user-box
  {position: relative;
}

.login-box .user-box input
  {width: 100%;
  padding: 10px 0;
  font-size: 16px;
  color: #fff;
  margin-bottom: 30px;
  border: none;
  border-bottom: 1px solid #fff;
  outline: none;
  background: transparent;
}
.login-box .user-box label
  {position: absolute;
  top:0;
  left: 0;
  padding: 10px 0;
  font-size: 16px;
  color: #fff;
  pointer-events: none;
  transition: .5s;
}

.login-box .user-box input:focus ~ label,
.login-box .user-box input:valid ~ label
  {top: -20px;
```

```
  left: 0;
  color: #03e9f4;
  font-size: 12px;
}

.login-box form a
  { position: relative;
  display: inline-block;
  padding: 10px 20px;
  color: #03e9f4;
  font-size: 16px;
  text-decoration: none;
  text-transform: uppercase;
  overflow: hidden;
  transition: .5s;
  margin-top: 40px;
  letter-spacing: 4px
}

.login-box a:hover
  { background:
  #03e9f4;color: #fff;
  border-radius: 5px;
  box-shadow: 0 0 5px #03e9f4,
              0 0 25px #03e9f4,
              0 0 50px #03e9f4,
              0 0 100px #03e9f4;
}

.login-box   a   span
  {             position:
  absolute;  display:
  block;
}

.login-box a span:nth-child(1)
  {top: 0;
  left: -100%;
  width: 100%;
  height: 2px;
  background: linear-gradient(90deg, transparent, #03e9f4);
  animation: btn-anim1 1s linear infinite;
}

@keyframes btn-anim1
  {0% {
    left: -100%;
```

```
  }
  50%,100% {
    left: 100%;
  }
}

.login-box a span:nth-child(2)
  {top: -100%;
  right: 0;
  width: 2px;
  height: 100%;
  background: linear-gradient(180deg, transparent, #03e9f4);
  animation: btn-anim2 1s linear infinite;
  animation-delay: .25s
}

@keyframes btn-anim2
  {0% {
    top: -100%;
  }
  50%,100% {
    top: 100%;
  }
}

.login-box a span:nth-child(3)
  {bottom: 0;
  right: -100%;
  width: 100%;
  height: 2px;
  background: linear-gradient(270deg, transparent, #03e9f4);
  animation: btn-anim3 1s linear infinite;
  animation-delay: .5s
}

@keyframes btn-anim3
  {0% {
    right: -100%;
  }
  50%,100% {
    right: 100%;
  }
}

.login-box a span:nth-child(4)
  {bottom: -100%;
```

```
  left: 0;
  width: 2px;
  height: 100%;
  background: linear-gradient(360deg, transparent, #03e9f4);
  animation: btn-anim4 1s linear infinite;
  animation-delay: .75s
}

@keyframes btn-anim4
  {0% {
    bottom: -100%;
  }
  50%,100% {
    bottom: 100%;
  }
}
```

RESULT

Performance Analysis Based on time taken:

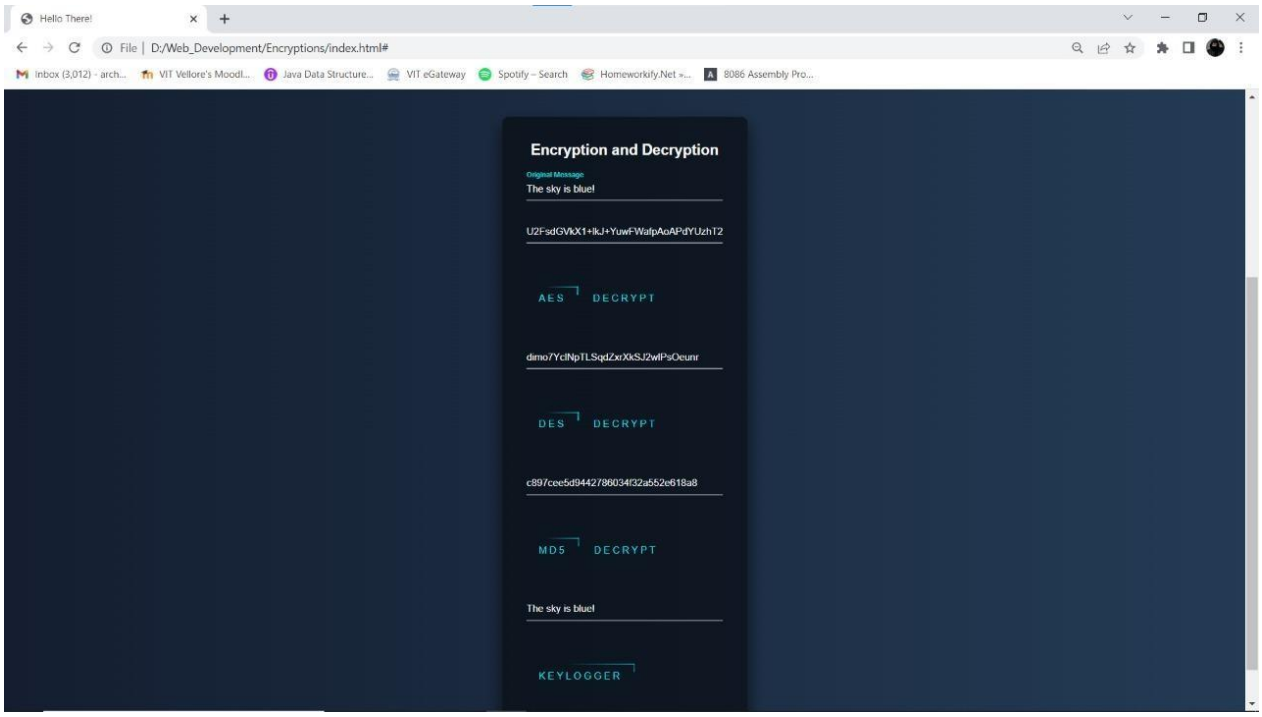
Table for Encryption Runtime of Text Files

File Size	AES (msec)	DES (msec)
1 MB	80	136.2
2 MB	154.7	269.6
5 MB	376.1	665.4
10 MB	683.7	2356.5

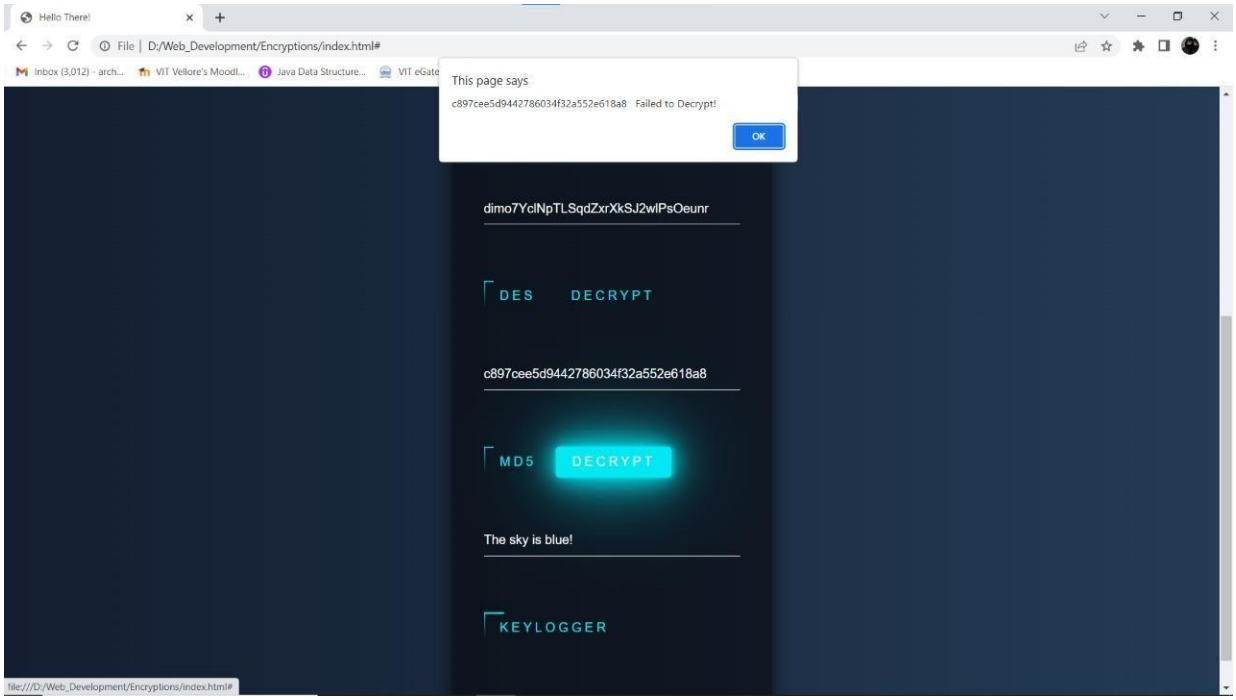
Table for Decryption Runtime of Text Files

File Size	AES (msec)	DES (msec)
1 MB	118.2	136.2
2 MB	187.7	269.6
5 MB	457.1	690.4
10 MB	897.7	1294.5

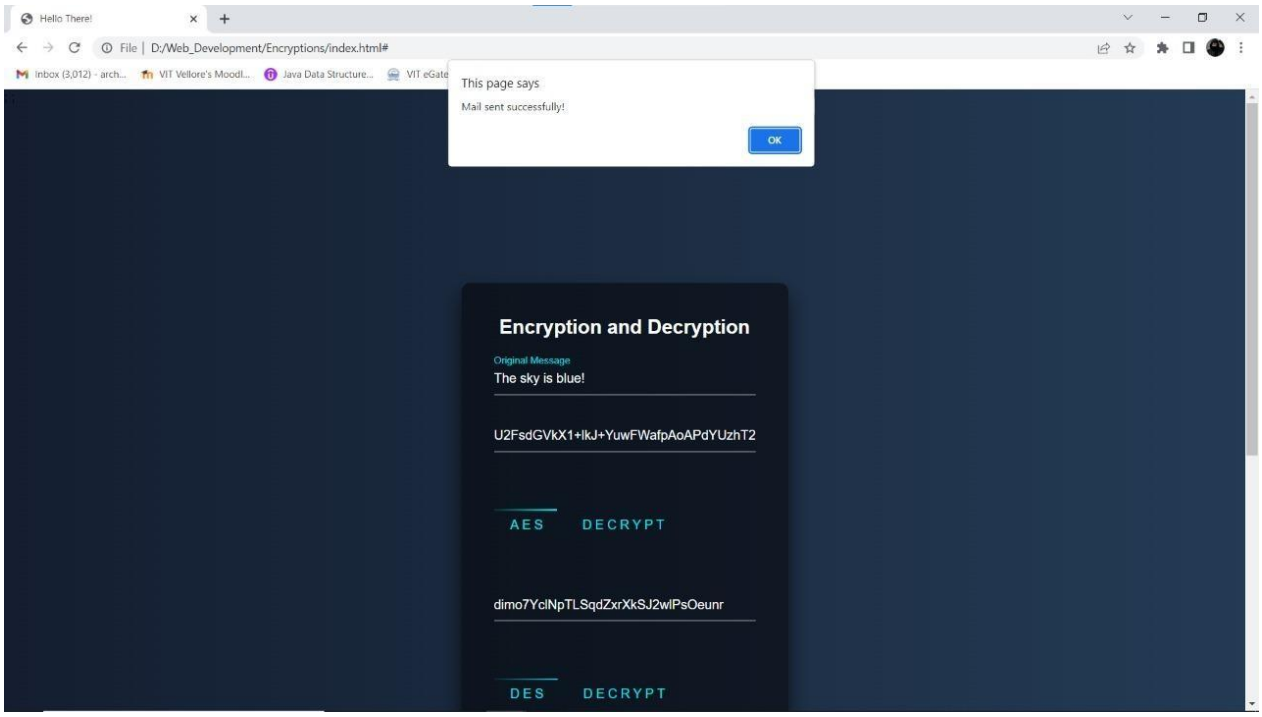
Screenshot of the Website:



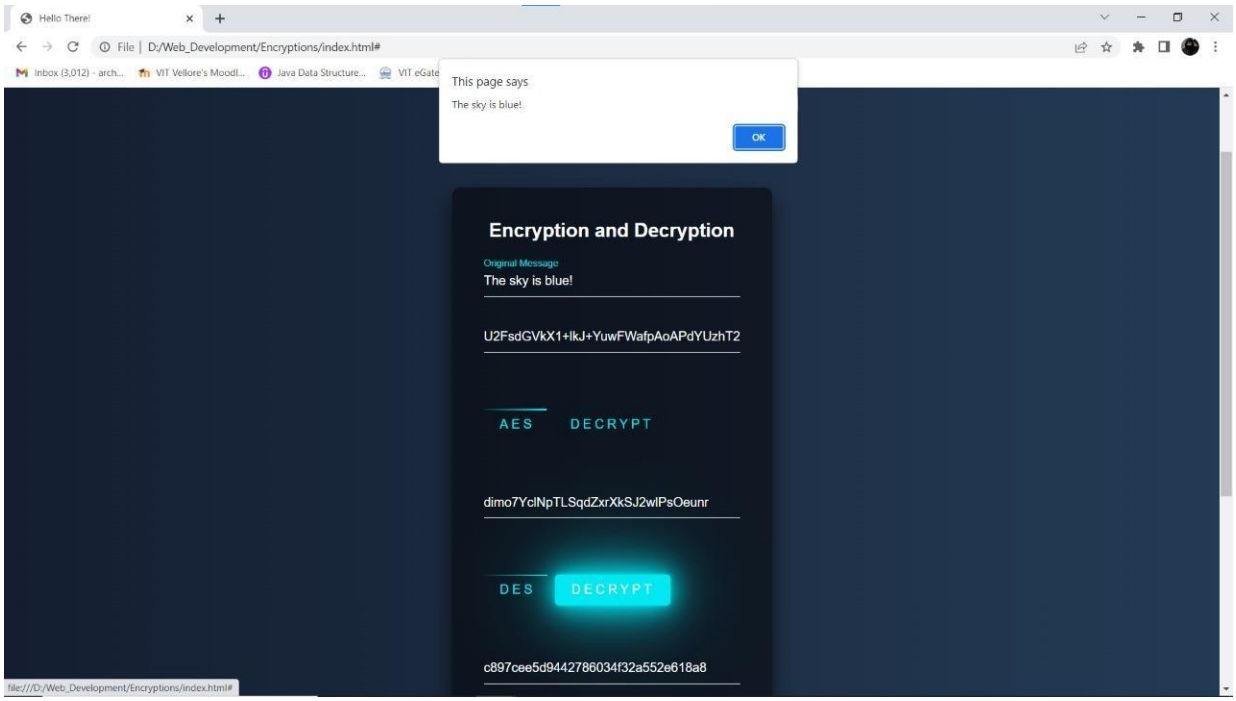
MD5:



KeyLogger:



DES:



CONCLUSION

With this website I have successfully implemented the above-mentioned Data Encryption algorithms and Hashing Algorithms. With this website the user can successfully encrypt ,decrypt, hash the input messages using these algorithms which will be beneficial for understanding of these algorithms .I have also used a keylogger which stores the keystrokes ,stores it in a file and sends the file to the email . It also demonstrates the threat to the privacy of our information and the privacy of internet activity.

REFERENCES

1. J. Xue, C. Xu, and Y. Zhang, “Private blockchain-based secure access control for smart home systems,” KSII Trans. Internet Inf. Syst., vol. 12, no. 12, pp. 6057–6078, 2018.
2. Zhao Yong-Xia, and Zhen Ge, “MD5 Research,” in IEEE 2nd International Conference on Multimedia and Information Technology, pp. 271 - 273, April 2019
3. Priya.P, and B. Gopinathan, “Improving Security Based on Detecting Selfish Nodes using MD5 Encryption Algorithm in Manets,” International Journal of Advanced Technology in Engineering and Sciences, vol. 4, pp. 1311-1317, Feb. 2016.
4. Irfan A. Landge, and B.K.Mishra, “Hardware Based MD5 Implementation Using VHDL for Secured Embedded and VLSI Based Designs,” in International Conference on Communication and Electronics Systems (ICCES), pp. 1 - 6, October 2016
5. R.Rivest, A.Shamir, and L.Adleman,"A Method For Obtaining Digital Signatures and Public Key Cryptosystems", ACM Transactions on Communications, Vol. 21, pp. 120- 126, 2020.
6. A.Alhasib and A.L.Haque, "A Comparative Study of the Performance Issues of the AES and RSA Cryptography",in Proc. 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT), Busan, 2018, pp.505-510.
7. Dan Boneh and Hovav Shacham, "Fast Variants of RSA", CryptoBytes, Vol.1,No.5, pp. 1-9, 2002.