Cryptography texts

```
        public string encrypt(string key, string plaintxt)
        {
            using (DESCryptoServiceProvider des = new
DESCryptoServiceProvider())
            {
                byte[] keys = Encoding.UTF8.GetBytes(key);
                ICryptoTransform encryptor = des.CreateEncryptor(keys,
keys);
                var ms =  new MemoryStream();
                var cs = new
CryptoStream(ms,encryptor,CryptoStreamMode.Write);
                byte[] input = Encoding.UTF8.GetBytes(plaintxt);
                cs.Write(input, 0, input.Length);
                cs.FlushFinalBlock();
                return Convert.ToBase64String(ms.ToArray());
            }
        }

        public string decrypt(string key, string ciphertxt)
        {
            byte[] buffer = Convert.FromBase64String(ciphertxt);
            using (DESCryptoServiceProvider des = new
DESCryptoServiceProvider())
            {
                byte[] keys = Encoding.UTF8.GetBytes(key);
                ICryptoTransform encryptor = des.CreateDecryptor(keys,
keys);
                var ms = new MemoryStream();
                var cs = new CryptoStream(ms, encryptor,
CryptoStreamMode.Write);
                cs.Write(buffer, 0, buffer.Length);
                cs.FlushFinalBlock();
                return Encoding.UTF8.GetString(ms.ToArray());
            }
        }
```

FILE CRYPTOGRAPHY

```
 class encr
```

```
    {
        private DESCryptoServiceProvider des = new
DESCryptoServiceProvider();
        public encr (string key)
        {
            des.Key = UTF8Encoding.UTF8.GetBytes(key);
            des.Mode = CipherMode.ECB;
        }
        public void encrypt(string filpath)
        {
            byte[] Bytes = File.ReadAllBytes(filpath);
            byte[] eBytes =
des.CreateEncryptor().TransformFinalBlock(Bytes, 0, Bytes.Length);
            File.WriteAllBytes(filpath, eBytes);
        }

        public void decrypt(string filpath)
        {
            byte[] Bytes = File.ReadAllBytes(filpath);
            byte[] dBytes =
des.CreateDecryptor().TransformFinalBlock(Bytes, 0, Bytes.Length);
            File.WriteAllBytes(filpath, dBytes);
        }
    }
//////

 //DES
            //try
            //{
            //    encr tr = new encr(textBox2.Text);
            //    tr.encrypt(textBox1.Text);
            //    GC.Collect();
            //    MessageBox.Show("OK");
            //}
            //catch(Exception ex)
            //{
            //    MessageBox.Show(ex.Message);
            //}

 //DES
            //try
            //{
            //    encr tr = new encr(textBox2.Text);
            //    tr.decrypt(textBox1.Text);
            //    GC.Collect();
            //    MessageBox.Show("OK");
            //}
            //catch (Exception ex)
            //{
```

```
            //    MessageBox.Show(ex.Message);
            //}
```

BUTTON ONE SHOW FILE

```
 OpenFileDialog ofile = new OpenFileDialog();
            if (ofile.ShowDialog() == DialogResult.OK)
                textBox1.Text = ofile.FileName;
```

TRIPLE

```
 class triple
    {
        private TripleDESCryptoServiceProvider tdes = new
TripleDESCryptoServiceProvider();
        public triple(string key)
        {
            tdes.Key = UTF8Encoding.UTF8.GetBytes(key);
            tdes.Mode = CipherMode.ECB;
            tdes.Padding = PaddingMode.PKCS7;
        }
        public void encrypt(string filpath)
        {
            byte[] Bytes = File.ReadAllBytes(filpath);
            byte[] eBytes =
tdes.CreateEncryptor().TransformFinalBlock(Bytes, 0, Bytes.Length);
            File.WriteAllBytes(filpath, eBytes);
        }

        public void decrypt(string filpath)
        {
            byte[] Bytes = File.ReadAllBytes(filpath);
            byte[] dBytes =
tdes.CreateDecryptor().TransformFinalBlock(Bytes, 0, Bytes.Length);
            File.WriteAllBytes(filpath, dBytes);
        }
    }


 try
            {
                triple tr = new triple(textBox2.Text);
                tr.encrypt(textBox1.Text);
                GC.Collect();
                MessageBox.Show("OK");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }



 try
            {
                triple tr = new triple(textBox2.Text);
                tr.decrypt(textBox1.Text);
                GC.Collect();
                MessageBox.Show("OK");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
```