Technical Manual

Harry Dunne
C00220865@itcarlow.ie
Supervisor
Patrick Tobin

Abstract

Digital Inheritance refers to the transfer of assets which exist on electronic systems. These assets, making up the owner's digital estate, can include passwords, usernames, online accounts, contracts, receipts, financial transactions, and medical information. The transfer of a digital estate should occur when there is a prolonged, or permanent absence of the original data owner.

Complications arise in digital inheritance because of two reasons – The information being bestowed is confidential in nature and needs to be stored securely, the date at which the information should be transferred is unknown. In many cases, increasing security of stored data decreases future accessibility for beneficiaries, while increasing the future accessibility harms overall security of the data.

Currently, individuals looking for a digital inheritance solution, need to either intrust their data with 3rd party cloud services, or store their information physically so that beneficiaries can find it in the case of an unforeseen event. Both solutions create single points of compromise, that can be exploited by criminals, digitally and physically.

This project aims to address this problem by providing a decentralised solution to the storage and transfer of digital assets. The solution provided has been designed to give users full autonomy of their data and protect them against both digital and physical attacks. Using this technology, an individual can create a highly secure digital inheritance solution within minutes.

Contents

Introduction

This document will contain the code for the application. The application can be run in any web browser on any device or operating system. To access the application simply go to:

https://brave-swirles-556603.netlify.app/

The code in this document will be broken down into the code for each page and have subsections for other code in the application that it uses. Code libraries external to the application, such as Bootstrap, popper.js, or jQuery will not be put within this document.

Code

Home Page

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Global site tag (gtag.js) - Google Analytics -->
    <script async src="https://www.googletagmanager.com/gtag/js?id=UA-157005377-1"></script>
    <script>
        window.dataLayer = window.dataLayer || [];

        function gtag() {
            dataLayer.push(arguments);
        }
        gtag('js', new Date());

        gtag('config', 'UA-157005377-1');
    </script>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Use project to secure your digital assets from loss and create digital inheritance systems.">
    <title>Project</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

```
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons"
/>
    <link rel="stylesheet" href="https://unpkg.com/aos@next/dist/aos.css" />
    <link href="./styles/style.css" rel="stylesheet">
    <link href="./styles/style.css" rel="stylesheet">
    <script src="./scripts/scripts.js"></script>
    <link rel="apple-touch-icon" sizes="180x180" href="img/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="img/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="img/favicon-16x16.png">
</head>

<body>

    <!-- Navigation -->

    <nav class="navbar navbar-expand-md navbar-light bg-light sticky-top">
        <div class="container-fluid">

            <h1>Project</h1>

            <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarResponsive">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse" id="navbarResponsive">
                <ul class="navbar-nav ml-auto">
                    <li class="nav-item active">
```

```
                    <a class="nav-link" href="index.html">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="form.html">Encrypt</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="form2.html">Decrypt</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="Sign_In.html">Sign In</a>
                </li>
            </ul>
        </div>
    </div>
</nav>

<!--- Jumbotron -->

<div class="container-fluid padding">

    <div class="row jumbotron padding" style="position:relative;">
                        <!-- Credit to www.webredone.com for the svg !-->
        <svg class="icon-support" viewBox="0 0 297.8 338.3"
style="display:none;width:500px;">
            <g id="XMLID_2_">
                <linearGradient id="XMLID_152_ccc" gradientUnits="userSpaceOnUse"
x1="306.8558" y1="121.1685" x2="473.9469" y2="288.2596"
gradientTransform="matrix(1.2556 -8.690000e-002 -8.690000e-002 1.3971 -298.7608 -
66.8389)">
                    <stop offset="0.1902" style="stop-color:#EFDFED"></stop>
                    <stop offset="1" style="stop-color:#FDF0F6"></stop>
                </linearGradient>
```

```
            <path id="XMLID_103_" fill="#E9ECEF" d="M101.1,289.7c-34.4-24.1-
93.1-10-98.9-63.8C-3.6,171.5,43.5,110,79,72.7
                c82.9-87,156.4-99.9,206.4,1.3c9.7,19.6,18,53.7,1.4,82.8c-9.7,16.9-
36.5,21.1-40.2,41.2c-6.8,36.3,34.5,60.9,11.2,107.9
                C223.2,375.4,129.7,309.7,101.1,289.7z"></path>
            <linearGradient id="XMLID_153_cc" gradientUnits="userSpaceOnUse"
x1="49.8059" y1="126.9058" x2="168.7503" y2="126.9058">
              <stop offset="6.016400e-002" style="stop-color:#DB6B86"></stop>
              <stop offset="8.596202e-002" style="stop-color:#DC6D87"></stop>
              <stop offset="1" style="stop-color:#F7A6A5"></stop>
            </linearGradient>
            <path id="XMLID_100_" class="gear-m hover origin-center"
fill="url(#XMLID_153_cc)" d="M168.7,119.5l-1-5.4c-0.1-0.4-0.4-0.6-0.8-0.6l-4.4,0.8c-
0.9-4-2.3-7.8-4.1-11.4l4-2.3
                c0.3-0.2,0.4-0.6,0.3-0.9l-2.8-4.8c-0.2-0.3-0.6-0.4-0.9-0.3l-3.9,2.3c-
2.2-3.4-4.8-6.5-7.7-9.3l2.9-3.5c0.2-0.3,0.2-0.7-0.1-1
                l-4.3-3.5c-0.3-0.2-0.7-0.2-1,0.1l-2.9,3.5c-3.2-2.4-6.7-4.5-10.5-
6.1l1.5-4.3c0.1-0.4-0.1-0.8-0.4-0.9l-5.2-1.9
                c-0.4-0.1-0.8,0.1-0.9,0.4l-1.5,4.2c-3.8-1.1-7.8-1.8-11.9-2.1l0-4.6c0-
0.4-0.3-0.7-0.7-0.7l-5.5,0c-0.4,0-0.7,0.3-0.7,0.7l0,4.6
                c-4.1,0.2-8.1,0.9-12,2L92.6,70c-0.1-0.4-0.5-0.5-0.9-0.4l-5.2,1.9c-
0.4,0.1-0.5,0.5-0.4,0.9l1.6,4.2c-3.8,1.6-7.2,3.7-10.5,6.1
                l-3-3.5c-0.2-0.3-0.7-0.3-1-0.1l-4.2,3.6c-0.3,0.2-0.3,0.7-
0.1,1l2.9,3.5c-2.9,2.8-5.6,5.8-7.9,9.2l-4-2.3
                c-0.3-0.2-0.8-0.1-0.9,0.3l-2.7,4.8c-0.2,0.3-0.1,0.8,0.3,0.9l3.9,2.2c-
1.8,3.6-3.2,7.4-4.2,11.4l-4.6-0.8
                c-0.4-0.1-0.7,0.2-0.8,0.6L50,119c-0.1,0.4,0.2,0.7,0.6,0.8l4.5,0.8c-
0.3,2.2-0.4,4.5-0.4,6.8c0,1.8,0.1,3.6,0.3,5.3l-4.5,0.8
                c-0.4,0.1-0.6,0.4-0.6,0.8l1,5.4c0.1,0.4,0.4,0.6,0.8,0.6l4.4-
0.8c0.9,4,2.3,7.8,4.1,11.4l-4,2.3c-0.3,0.2-0.4,0.6-0.2,1l2.8,4.8
                c0.2,0.3,0.6,0.4,0.9,0.2l3.9-2.3c2.2,3.4,4.8,6.5,7.7,9.3l-2.9,3.5c-
0.2,0.3-0.2,0.7,0.1,1l4.3,3.5c0.3,0.2,0.7,0.2,1-0.1l2.9-3.5
```

c3.2,2.4,6.7,4.5,10.4,6.1l-1.5,4.3c-
0.1,0.4,0.1,0.8,0.4,0.9l5.2,1.9c0.4,0.1,0.8-0.1,0.9-0.4l1.5-4.2c3.8,1.1,7.8,1.8,11.9,2.1
l0,4.6c0,0.4,0.3,0.7,0.7,0.7l5.5-0.1c0.4,0,0.7-0.3,0.7-0.7l0-4.6c4.1-
0.2,8.1-0.9,11.9-2l1.6,4.3c0.1,0.4,0.5,0.5,0.9,0.4
l5.2-1.9c0.4-0.1,0.5-0.5,0.4-0.9l-1.6-4.2c3.7-1.6,7.3-3.7,10.5-
6.1l3,3.5c0.2,0.3,0.7,0.3,1,0.1l4.2-3.6c0.3-0.3,0.3-0.7,0.1-1
l-2.9-3.5c2.9-2.8,5.6-5.8,7.9-9.2l4,2.3c0.3,0.2,0.8,0.1,1-0.3l2.7-
4.8c0.2-0.3,0.1-0.8-0.3-0.9l-3.9-2.2
c1.8-3.6,3.2-7.4,4.2-11.4l4.6,0.8c0.4,0.1,0.7-0.2,0.8-0.6l0.9-5.5c0.1-
0.4-0.2-0.7-0.6-0.8l-4.5-0.8c0.3-2.2,0.4-4.5,0.4-6.8
c0-1.8-0.1-3.6-0.3-5.3l4.5-
0.8C168.6,120.3,168.8,119.9,168.7,119.5z M109.4,140.5c-7.5,0.1-13.6-6-13.7-13.5
c-0.1-7.5,6-13.6,13.5-13.7c7.5-
0.1,13.6,6,13.7,13.5C122.9,134.3,116.9,140.4,109.4,140.5z" style="transform-origin:
109.313px 126.9px;"></path>
<g id="XMLID_96_">
<path id="XMLID_97_" fill="#7D2360" d="M109.3,157.5c-14.2,0-26.4-
9.7-29.7-23.5c-1.9-7.9-0.6-16.1,3.7-23.1
c4.3-6.9,11-11.8,18.9-13.7c2.3-0.6,4.7-0.8,7.1-
0.8c14.2,0,26.4,9.7,29.7,23.5c1.9,7.9,0.6,16.1-3.7,23.1
c-4.3,7-11,11.8-
18.9,13.7C114,157.2,111.6,157.5,109.3,157.5L109.3,157.5z M109.3,98.1c-2.2,0-4.5,0.3-
6.7,0.8
c-7.5,1.8-13.8,6.4-17.8,12.9c-4,6.5-5.3,14.3-
3.5,21.7c3.1,13,14.6,22.1,28,22.1c2.2,0,4.5-0.3,6.7-0.8
c7.5-1.8,13.8-6.4,17.8-12.9c4-6.5,5.3-14.3,3.5-
21.7C134.2,107.2,122.7,98.1,109.3,98.1L109.3,98.1z"></path>
</g>
<g id="XMLID_92_">
<path id="XMLID_93_" fill="#7D2360" d="M109.3,164.3c-17.4,0-32.3-
11.8-36.3-28.7c-2.3-9.7-0.7-19.7,4.5-28.2
c5.2-8.5,13.5-14.5,23.2-16.8c2.9-0.7,5.8-1,8.7-
1c17.4,0,32.3,11.8,36.3,28.7c4.8,20-7.6,40.2-27.7,45

```
                                C115.1,164,112.2,164.3,109.3,164.3L109.3,164.3z
M109.3,93c-2.6,0-5.3,0.3-7.9,0.9c-8.8,2.1-16.3,7.5-21,15.2
                                c-4.7,7.7-6.2,16.8-
4.1,25.6c3.6,15.3,17.2,26,32.9,26c2.6,0,5.3-0.3,7.9-0.9c18.2-4.3,29.4-22.6,25.1-40.8
                                C138.6,103.7,125,93,109.3,93L109.3,93z"></path>
            </g>
            <path id="XMLID_89_" class="gear-s hover origin-center" fill="#5F2566"
d="M257,35.1l-3.9,0.9c-0.6-1.3-1.3-2.5-2.1-3.7l2.8-2.6c0.7-0.6,0.7-1.7,0.1-2.3l-4.6-4.8
                    c-0.6-0.7-1.7-0.7-2.3-0.1l-2.9,2.7c-1.1-0.8-2.4-1.5-3.6-2.2l1.1-
3.7c0.3-0.9-0.2-1.8-1.1-2.1l-6.4-1.9c-0.9-0.3-1.8,0.2-2.1,1.1
                    l-1.1,3.8c-1.4-0.1-2.8-0.2-4.2-0.1l-0.9-3.7c-0.2-0.9-1.1-1.4-2-1.2l-
6.5,1.5c-0.9,0.2-1.4,1.1-1.2,2l0.9,3.9
                    c-1.3,0.6-2.5,1.3-3.7,2.1l-2.6-2.8c-0.6-0.7-1.7-0.7-2.4-0.1l-4.8,4.6c-
0.7,0.6-0.7,1.7-0.1,2.3l2.7,2.9c-0.8,1.1-1.5,2.4-2.2,3.6
                    l-3.7-1.1c-0.9-0.3-1.8,0.2-2.1,1.1l-1.9,6.4c-
0.3,0.9,0.2,1.8,1.1,2.1l3.8,1.1c-0.1,1.4-0.1,2.8-0.1,4.2l-3.7,0.9
                    c-0.9,0.2-1.5,1.1-1.2,2l1.5,6.5c0.2,0.9,1.1,1.5,2,1.2l3.9-
0.9c0.6,1.3,1.3,2.5,2.1,3.7l-2.8,2.6c-0.7,0.6-0.7,1.7-0.1,2.3
                    l4.6,4.8c0.6,0.7,1.7,0.7,2.3,0.1l2.9-2.7c1.2,0.8,2.4,1.5,3.6,2.2l-
1.1,3.7c-0.3,0.9,0.2,1.8,1.1,2.1l6.4,1.9
                    c0.9,0.3,1.8-0.2,2.1-1.1l1.1-
3.8c1.4,0.1,2.8,0.2,4.2,0.1l0.9,3.7c0.2,0.9,1.1,1.4,2,1.2l6.5-1.5c0.9-0.2,1.4-1.1,1.2-2l-0.9-
3.9
                    c1.3-0.6,2.5-1.3,3.7-2l2.6,2.8c0.6,0.7,1.7,0.7,2.3,0.1l4.8-4.6c0.7-
0.6,0.7-1.7,0.1-2.3l-2.7-2.9c0.8-1.1,1.5-2.4,2.2-3.6
                    l3.7,1.1c0.9,0.3,1.8-0.2,2.1-1.1l1.9-6.4c0.3-0.9-0.2-1.8-1.1-2.1l-3.8-
1.1c0.1-1.4,0.1-2.8,0.1-4.2l3.7-0.9
                    c0.9-0.2,1.4-1.1,1.2-2l-1.5-6.5C258.8,35.4,257.9,34.9,257,35.1z
M231.1,58.8c-6.3,1.5-12.6-2.4-14.1-8.7
                    c-1.5-6.3,2.4-12.6,8.7-14.1c6.3-
1.5,12.6,2.4,14.1,8.7C241.3,51,237.4,57.3,231.1,58.8z" style="transform-origin:
228.233px 47.35px;"></path>
            <g id="XMLID_82_">
```

```
                <linearGradient id="XMLID_154_c" gradientUnits="userSpaceOnUse"
x1="121.1049" y1="291.4615" x2="185.4546" y2="179.1774">
                  <stop offset="9.016400e-002" style="stop-color:#A62257"></stop>
                  <stop offset="0.4541" style="stop-color:#86245E"></stop>
                  <stop offset="1" style="stop-color:#582768"></stop>
                </linearGradient>
                <path id="XMLID_18_" class="gear-b hover origin-center"
fill="url(#XMLID_154_c)" d="M227.6,235.6L219,233c0.3-3.1,0.3-6.3,0.1-9.5l8.4-2c2-
0.5,3.2-2.5,2.8-4.5l-3.5-14.6
                  c-0.5-2-2.5-3.3-4.5-2.8l-8.7,2.1c-1.4-2.9-2.9-5.7-4.6-8.3l6.3-
5.9c1.5-1.4,1.6-3.8,0.1-5.3l-10.3-10.9c-1.4-1.5-3.8-1.6-5.3-0.1
                  l-6.5,6.2c-2.6-1.8-5.3-3.5-8.2-4.9l2.5-8.3c0.6-2-0.5-4.1-2.5-
4.7l-14.4-4.3c-2-0.6-4.1,0.5-4.7,2.5l-2.6,8.6
                  c-3.1-0.3-6.3-0.3-9.5-0.1l-2-8.4c-0.5-2-2.5-3.3-4.5-2.8l-
7.8,1.9l0,0l-6.8,1.6c-2,0.5-3.3,2.5-2.8,4.5l2.1,8.7
                  c-2.9,1.3-5.7,2.9-8.3,4.6l-5.9-6.3c-1.4-1.5-3.8-1.6-5.3-0.1l-
10.9,10.3c-1.5,1.4-1.6,3.8-0.1,5.3l6.2,6.5
                  c-1.8,2.6-3.5,5.3-4.9,8.2l-8.3-2.5c-2-0.6-4.1,0.6-4.7,2.5l-
4.3,14.4c-0.6,2,0.5,4.1,2.5,4.7l8.6,2.6c-0.3,3.1-0.3,6.3-0.1,9.5
                  l-8.4,2c-2,0.5-3.3,2.5-
2.8,4.5l3.5,14.6c0.5,2,2.5,3.3,4.5,2.8l8.7-2.1c1.3,2.9,2.9,5.7,4.6,8.3l-6.3,5.9
                  c-1.5,1.4-1.6,3.8-
0.1,5.3l10.3,10.9c1.4,1.5,3.8,1.6,5.3,0.1l6.5-6.2c2.6,1.8,5.3,3.5,8.2,4.9l-2.5,8.3c-
0.6,2,0.5,4.1,2.5,4.7
                  l14.4,4.3c2,0.6,4.1-0.5,4.7-2.5l2.6-
8.6c3.1,0.3,6.3,0.3,9.5,0.1l2,8.4c0.5,2,2.5,3.3,4.5,2.8l5.2-1.2v0l9.4-2.2
                  c2-0.5,3.3-2.5,2.8-4.5l-2.1-8.7c2.9-1.3,5.7-2.9,8.3-
4.6l5.9,6.3c1.4,1.5,3.8,1.6,5.3,0.1l10.9-10.3c1.5-1.4,1.6-3.8,0.1-5.3
                  l-6.2-6.5c1.8-2.6,3.5-5.3,4.9-8.2l8.3,2.5c2,0.6,4.1-0.5,4.7-
2.5l4.3-14.4C230.7,238.3,229.6,236.2,227.6,235.6z M163.9,253.1
                  c-14.2,3.4-28.4-5.4-31.8-19.6c-3.4-14.2,5.4-28.4,19.6-
31.8c14.2-3.4,28.4,5.4,31.8,19.6C186.9,235.5,178.1,249.7,163.9,253.1z" style="transform-
origin: 157.835px 227.45px;"></path>
```

```
            </g>
            <g id="XMLID_78_">
               <path id="XMLID_79_" fill="#A52257" d="M72.2,192.8l-1.8,0.4c-0.4,0.1-
0.9-0.1-1.2-0.5l-0.5-0.7c-0.3-0.3-0.3-0.9-0.1-1.3l1-1.6
                  c0.2-0.4,0.1-0.8-0.2-1.1l-2.4-1.5c-0.4-0.2-0.9-0.1-1.1,0.3l-
1,1.6c-0.2,0.4-0.7,0.6-1.2,0.5l-0.9-0.1c-0.4,0-0.9-0.4-1-0.8
                  l-0.4-1.8c-0.1-0.4-0.5-0.7-0.9-0.6l-2.8,0.7c-0.4,0.1-0.7,0.5-
0.6,0.9l0.4,1.8c0.1,0.4-0.1,0.9-0.5,1.2l-0.7,0.5
                  c-0.3,0.3-0.9,0.3-1.3,0.1l-1.6-1c-0.4-0.2-0.8-0.1-1.1,0.3l-
1.5,2.4c-0.2,0.4-0.1,0.8,0.3,1.1l1.6,1c0.4,0.2,0.6,0.8,0.5,1.2
                  l-0.1,0.9c0,0.4-0.4,0.9-0.8,1l-1.8,0.4c-0.4,0.1-0.7,0.5-
0.6,0.9l0.7,2.8c0.1,0.4,0.5,0.7,0.9,0.6l1.8-0.4
                  c0.4-0.1,0.9,0.1,1.2,0.5l0.5,0.7c0.3,0.3,0.3,0.9,0.1,1.3l-
1,1.6c-0.2,0.4-0.1,0.8,0.3,1.1l2.4,1.5c0.4,0.2,0.8,0.1,1.1-0.3
                  l1-1.6c0.2-0.4,0.8-0.6,1.2-
0.5l0.9,0.1c0.4,0,0.9,0.4,1,0.8l0.4,1.8c0.1,0.4,0.5,0.7,0.9,0.6l2.8-0.7c0.4-0.1,0.7-0.5,0.6-
0.9
                  l-0.4-1.8c-0.1-0.4,0.1-0.9,0.5-1.2l0.7-0.5c0.3-0.3,0.9-0.3,1.3-
0.1l1.6,1c0.4,0.2,0.8,0.1,1.1-0.3l1.5-2.4
                  c0.2-0.4,0.1-0.8-0.3-1.1l-1.6-1c-0.4-0.2-0.6-0.8-0.5-1.2l0.1-
0.9c0-0.4,0.4-0.9,0.8-1l1.8-0.4c0.4-0.1,0.7-0.5,0.6-0.9l-0.7-2.8
                  C73,193,72.6,192.7,72.2,192.8z M62.7,201.2c-2,0.5-4.1-0.8-
4.5-2.8c-0.5-2,0.8-4.1,2.8-4.5c2-0.5,4.1,0.8,4.5,2.8
                  C65.9,198.7,64.7,200.7,62.7,201.2z"></path>
            </g>
            <g id="XMLID_74_">
               <path id="XMLID_75_" fill="#FFFFFF" d="M157.8,276.2c-22.7,0-42.1-
15.4-47.4-37.5c-6.2-26.2,10-52.5,36.1-58.8
                  c3.7-0.9,7.5-1.3,11.3-
1.3c22.7,0,42.1,15.4,47.4,37.5c6.2,26.2-10,52.5-
36.1,58.7C165.4,275.7,161.6,276.2,157.8,276.2
                  L157.8,276.2z M157.8,180.4c-3.7,0-7.3,0.4-10.9,1.3c-
25.2,6-40.8,31.4-34.8,56.6c5.1,21.3,23.8,36.1,45.7,36.1
```

```
                                c3.7,0,7.3-0.4,10.9-1.3c25.2-6,40.8-31.4,34.8-
56.6C198.4,195.3,179.7,180.4,157.8,180.4L157.8,180.4z"></path>
            </g>
            <g id="XMLID_72_">
               <path id="XMLID_73_" fill="#FFFFFF" d="M169.1,274.8l-0.4-1.7c25.2-
6,40.8-31.4,34.8-56.6c-6-25.2-31.4-40.8-56.6-34.8l-0.4-1.7
                                c26.2-
6.2,52.5,10,58.8,36.1C211.5,242.3,195.3,268.6,169.1,274.8L169.1,274.8z"></path>
            </g>
            <path id="XMLID_71_" fill="#5E2667" d="M111.5,136c-5,1.2-10.1-1.9-
11.3-7c-1.2-5,1.9-10.1,7-11.3c5-1.2,10.1,1.9,11.3,7
                     C119.6,129.8,116.5,134.8,111.5,136z"></path>
            <path id="XMLID_70_" fill="#582768" d="M149,190.4l2.7,11.3c14.2-
3.4,28.4,5.4,31.8,19.6c3.4,14.2-5.4,28.4-19.6,31.8l2.7,11.3
                     c20.4-4.9,33.1-25.4,28.2-
45.8C189.9,198.2,169.4,185.5,149,190.4z"></path>
            <path id="XMLID_69_" fill="#582768" d="M149.5,190.3l2.5,11.3c-14,4-
22.7,18.9-18.7,32.9c4,14,18.6,22.2,32.7,18.2l3.2,11.2
                     c-20.2,5.8-42-6.6-47.8-
26.8C115.7,216.8,129.3,196.1,149.5,190.3z"></path>
            <g id="XMLID_65_">
               <path id="XMLID_66_" fill="#FFFFFF" d="M228.4,64.5c-7.9,0-14.7-5.4-
16.6-13.1c-2.2-9.2,3.5-18.4,12.6-20.5c1.3-0.3,2.6-0.5,4-0.5
                     c7.9,0,14.7,5.4,16.6,13.1c2.2,9.1-3.5,18.4-
12.6,20.5C231.1,64.3,229.8,64.5,228.4,64.5L228.4,64.5z M228.4,32.1
                     c-1.2,0-2.4,0.1-3.5,0.4c-8.2,2-13.3,10.2-
11.3,18.4c1.6,6.9,7.7,11.7,14.8,11.7c1.2,0,2.4-0.1,3.6-0.4c8.2-2,13.3-10.2,11.3-18.4
                     C241.6,37,235.5,32.1,228.4,32.1L228.4,32.1z"></path>
            </g>
            <g id="XMLID_63_">
               <path id="XMLID_64_" fill="#FFFFFF" d="M97.6,109c-0.3,0-0.6-0.2-0.8-
0.4c-0.3-0.4-0.1-1,0.3-1.2l121.8-74.1c0.4-0.3,1-0.1,1.2,0.3
```

```
                                c0.3,0.4,0.1,1-
0.3,1.2L98.1,108.9C97.9,108.9,97.7,109,97.6,109L97.6,109z"></path>
            </g>
            <g id="XMLID_61_">
               <path id="XMLID_62_" fill="#FFFFFF" d="M120.1,146.8c-0.3,0-0.5-0.1-
0.7-0.4c-0.3-0.4-0.2-1,0.2-1.2l120-87c0.4-0.3,1-0.2,1.3,0.2
                                c0.3,0.4,0.2,1-0.2,1.2l-
120,87C120.5,146.8,120.3,146.8,120.1,146.8L120.1,146.8z"></path>
            </g>
            <linearGradient id="XMLID_155_cc" gradientUnits="userSpaceOnUse"
x1="99.2893" y1="101.8832" x2="174.2109" y2="262.5436">
               <stop offset="6.016400e-002" style="stop-color:#DB6B86"></stop>
               <stop offset="8.596202e-002" style="stop-color:#DC6D87"></stop>
               <stop offset="1" style="stop-color:#F7A6A5"></stop>
            </linearGradient>
            <path id="XMLID_56_" fill="url(#XMLID_155_cc)" d="M195.7,218.4c-1.4-
5.8-4-11-7.5-15.4l-57.6-84.8c-3.5-8.6-11.8-14.3-21.3-14.3
                c-1.8,0-3.6,0.2-5.4,0.6c-6,1.4-11.1,5.1-14.3,10.3c-3.2,5.2-4.2,11.4-
2.8,17.4c0.1,0.4,0.2,0.7,0.3,1.1c0,0.1-0.1,0.2,0,0.3

      l33.4,105.2c0.1,0.4,0.3,0.9,0.4,1.3l0.3,0.8c0,0.1,0.1,0.1,0.1,0.1c5.6,15,20,25.2,36.4,
25.2c3,0,6.1-0.4,9.1-1.1
                C187.7,260.3,200.7,239.3,195.7,218.4z M91.6,116c2.9-4.7,7.5-
8,12.9-9.3c1.6-0.4,3.2-0.6,4.8-0.6c9.7,0,18,6.6,20.2,16
                c1.3,5.4,0.4,11-2.5,15.7c-2.9,4.7-7.5,8-12.9,9.3c-1.6,0.4-3.2,0.6-
4.8,0.6c-9.7,0-18-6.6-20.2-16C87.8,126.3,88.7,120.7,91.6,116
                z M109.3,149.9c1.8,0,3.6-0.2,5.4-0.6c6-1.4,11.1-5.1,14.3-10.3c2.9-
4.7,4-10,3.2-15.4l49.6,73c-6.6-5.1-14.9-8.1-23.8-8.1
                c-3,0-6.1,0.4-9.1,1.1c-17.9,4.3-29.9,20.4-29.9,38l-27.3-
85.9C95.9,146.8,102.2,149.9,109.3,149.9z M166.4,263.5
                c-2.8,0.7-5.8,1-8.6,1c-16,0-29.9-10.1-35.1-25l-0.4-1.2c-0.2-0.8-0.5-
1.5-0.7-2.3c-4.7-19.9,7.6-40,27.5-44.8c2.8-0.7,5.7-1,8.6-1
```

```
        c11.6,0,22,5.3,28.9,13.8l2.3,3.4c0.1,0.1,0.2,0.2,0.3,0.2c2.1,3.3,3.7,7,4.6,11.1C198.
7,238.7,186.3,258.8,166.4,263.5z"></path>
            <linearGradient id="XMLID_156_c" gradientUnits="userSpaceOnUse"
x1="193.4497" y1="132.6568" x2="254.0015" y2="132.6568">
                <stop offset="6.016400e-002" style="stop-color:#DB6B86"></stop>
                <stop offset="8.596202e-002" style="stop-color:#DC6D87"></stop>
                <stop offset="1" style="stop-color:#F7A6A5"></stop>
            </linearGradient>
            <path id="XMLID_55_" fill="url(#XMLID_156_c)" d="M253.4,138.3l-3.5-
1.1c0.2-1.5,0.4-3,0.4-4.5c0-1.6-0.2-3.2-0.4-4.7l3.5-1.1
                c0.4-0.1,0.7-0.6,0.5-1.1l-1.8-5.5c-0.1-0.4-0.6-0.7-1.1-0.5l-3.5,1.1c-
1.4-2.8-3.2-5.3-5.4-7.4l2.2-3c0.3-0.4,0.2-0.9-0.2-1.2
                l-4.6-3.4c-0.4-0.3-0.9-0.2-1.2,0.2l-2.2,3c-2.7-1.4-5.6-2.4-8.7-2.8v-
3.7c0-0.5-0.4-0.8-0.8-0.8h-5.7c-0.5,0-0.8,0.4-0.8,0.8v3.7
                c-3.1,0.5-6.1,1.4-8.7,2.9l-2.1-3c-0.3-0.4-0.8-0.5-1.2-0.2l-4.6,3.4c-
0.4,0.3-0.5,0.8-0.2,1.2l2.2,3c-2.2,2.1-4,4.7-5.4,7.4
                l-3.5-1.1c-0.4-0.1-0.9,0.1-1.1,0.5l-1.8,5.5c-
0.1,0.4,0.1,0.9,0.5,1.1l3.5,1.1c-0.2,1.5-0.4,3-0.4,4.5c0,1.6,0.2,3.2,0.4,4.7
                l-3.5,1.1c-0.4,0.1-0.7,0.6-0.5,1.1l1.8,5.5c0.1,0.4,0.6,0.7,1.1,0.5l3.5-
1.1c1.4,2.8,3.2,5.3,5.4,7.4l-2.2,3
                c-0.3,0.4-0.2,0.9,0.2,1.2l4.6,3.4c0.4,0.3,0.9,0.2,1.2-0.2l2.2-
3c2.7,1.4,5.6,2.4,8.7,2.8v3.7c0,0.5,0.4,0.8,0.8,0.8h5.7
                c0.5,0,0.8-0.4,0.8-0.8v-3.7c3.1-0.5,6.1-1.4,8.7-
2.9l2.2,3c0.3,0.4,0.8,0.5,1.2,0.2l4.6-3.4c0.4-0.3,0.5-0.8,0.2-1.2l-2.2-3
                c2.2-2.1,4-4.7,5.4-7.4l3.5,1.1c0.4,0.1,0.9-0.1,1.1-0.5l1.8-
5.5C254.1,138.9,253.9,138.4,253.4,138.3z"></path>
            <circle id="XMLID_54_" fill="#FFFFFF" cx="223.7" cy="132.7"
r="19.4"></circle>
            <path id="XMLID_53_" opacity="0.4" fill="#582768" d="M206.4,134c0-
10.7,8.7-19.4,19.4-19.4c3.5,0,6.8,0.9,9.7,2.6c-3.3-2.5-7.3-4-11.8-4
                c-10.7,0-19.4,8.7-
19.4,19.4c0,7.2,3.9,13.4,9.7,16.8C209.4,145.9,206.4,140.3,206.4,134z"></path>
```

```
            <circle id="XMLID_52_" fill="#8C235D" cx="223.7" cy="132.7"
r="3.5"></circle>
            <g id="XMLID_50_">
               <path id="XMLID_51_" fill="#8C235D" d="M223.7,133.6c-0.5,0-0.9-0.4-
0.9-0.9v-13.7c0-0.5,0.4-0.9,0.9-0.9c0.5,0,0.9,0.4,0.9,0.9


      v13.7C224.7,133.2,224.2,133.6,223.7,133.6L223.7,133.6z"></path>
            </g>
            <g id="XMLID_22_">
               <g id="XMLID_41_">
                  <g id="XMLID_48_">
                     <path id="XMLID_49_" fill="#582768" d="M232,118.9c-0.1,0-0.2,0-
0.2-0.1c-0.2-0.1-0.3-0.4-0.2-0.6l0.5-0.9c0.1-0.2,0.4-0.3,0.6-0.2
                                                c0.2,0.1,0.3,0.4,0.2,0.6l-
0.5,0.9C232.3,118.8,232.1,118.9,232,118.9L232,118.9z"></path>
                  </g>
                  <g id="XMLID_46_">
                     <path id="XMLID_47_" fill="#582768" d="M215,148.3c-0.1,0-0.2,0-
0.2-0.1c-0.2-0.1-0.3-0.4-0.2-0.6l0.5-0.9c0.1-0.2,0.4-0.3,0.6-0.2
                                                c0.2,0.1,0.3,0.4,0.2,0.6l-
0.5,0.9C215.3,148.2,215.1,148.3,215,148.3L215,148.3z"></path>
                  </g>
                  <g id="XMLID_44_">
                     <path id="XMLID_45_" fill="#582768" d="M238.9,141.9c-0.1,0-
0.2,0-0.2-0.1l-0.9-0.5c-0.2-0.1-0.3-0.4-0.2-0.6
                                                c0.1-0.2,0.4-0.3,0.6-
0.2l0.9,0.5c0.2,0.1,0.3,0.4,0.2,0.6C239.2,141.8,239.1,141.9,238.9,141.9L238.9,141.9z"></
path>
                  </g>
                  <g id="XMLID_42_">
                     <path id="XMLID_43_" fill="#582768" d="M209.5,124.9c-0.1,0-
0.2,0-0.2-0.1l-0.9-0.5c-0.2-0.1-0.3-0.4-0.2-0.6
```

```
                                                c0.1-0.2,0.4-0.3,0.6-
0.2l0.9,0.5c0.2,0.1,0.3,0.4,0.2,0.6C209.8,124.8,209.6,124.9,209.5,124.9L209.5,124.9z"></
path>
                    </g>
                  </g>
                  <g id="XMLID_32_">
                    <g id="XMLID_39_">
                      <path id="XMLID_40_" fill="#582768" d="M215.5,118.9c-0.2,0-0.3-
0.1-0.4-0.2l-0.5-0.9c-0.1-0.2-0.1-0.5,0.2-0.6
                                                c0.2-0.1,0.5,0,0.6,0.2l0.5,0.9c0.1,0.2,0,0.5-
0.2,0.6C215.6,118.8,215.6,118.9,215.5,118.9L215.5,118.9z"></path>
                    </g>
                    <g id="XMLID_37_">
                      <path id="XMLID_38_" fill="#582768" d="M232.5,148.3c-0.2,0-0.3-
0.1-0.4-0.2l-0.5-0.9c-0.1-0.2,0-0.5,0.2-0.6
                                                c0.2-0.1,0.5-0.1,0.6,0.2l0.5,0.9c0.1,0.2,0,0.5-
0.2,0.6C232.7,148.3,232.6,148.3,232.5,148.3L232.5,148.3z"></path>
                    </g>
                    <g id="XMLID_35_">
                      <path id="XMLID_36_" fill="#582768" d="M208.5,141.9c-0.2,0-0.3-
0.1-0.4-0.2c-0.1-0.2-0.1-0.5,0.2-0.6l0.9-0.5
                                                c0.2-0.1,0.5-0.1,0.6,0.2c0.1,0.2,0.1,0.5-
0.2,0.6l-0.9,0.5C208.7,141.9,208.6,141.9,208.5,141.9L208.5,141.9z"></path>
                    </g>
                    <g id="XMLID_33_">
                      <path id="XMLID_34_" fill="#582768" d="M238,124.9c-0.2,0-0.3-
0.1-0.4-0.2c-0.1-0.2-0.1-0.5,0.2-0.6l0.9-0.5
                                                c0.2-0.1,0.5-0.1,0.6,0.2c0.1,0.2,0,0.5-0.2,0.6l-
0.9,0.5C238.2,124.9,238.1,124.9,238,124.9L238,124.9z"></path>
                    </g>
                  </g>
                  <g id="XMLID_23_">
                    <g id="XMLID_30_">
```

```
                        <path id="XMLID_31_" fill="#582768"
d="M207.3,133.1L207.3,133.1h-1.1c-0.3,0-0.5-0.2-0.5-0.5c0-0.3,0.2-0.5,0.5-0.5l0,0l1.1,0

        c0.3,0,0.5,0.2,0.5,0.5S207.5,133.1,207.3,133.1L207.3,133.1z"></path>
                    </g>
                    <g id="XMLID_28_">
                      <path id="XMLID_29_" fill="#582768" d="M240.2,133.1c-0.3,0-0.5-
0.2-0.5-0.5c0-0.3,0.2-0.5,0.5-0.5l1.1,0l0,0
                                                c0.3,0,0.5,0.2,0.5,0.5c0,0.3-0.2,0.5-
0.5,0.5L240.2,133.1L240.2,133.1L240.2,133.1z"></path>
                    </g>
                    <g id="XMLID_26_">
                      <path id="XMLID_27_" fill="#582768"
d="M223.7,150.7C223.7,150.7,223.7,150.7,223.7,150.7c-0.3,0-0.5-0.2-0.5-0.5l0-1.1
                                                c0-0.3,0.2-0.5,0.5-
0.5c0,0,0,0,0,0c0.3,0,0.5,0.2,0.5,0.5v1.1C224.2,150.4,224,150.7,223.7,150.7L223.7,150.7z
"></path>
                    </g>
                    <g id="XMLID_24_">
                      <path id="XMLID_25_" fill="#582768" d="M223.7,116.6c-0.3,0-0.5-
0.2-0.5-0.5v-1.1c0-0.3,0.2-0.5,0.5-0.5c0.3,0,0.5,0.2,0.5,0.5v1.1

        C224.2,116.4,224,116.6,223.7,116.6L223.7,116.6z"></path>
                    </g>
                  </g>
                </g>
                <g id="XMLID_17_">
                  <path id="XMLID_21_" fill="#8C235D" d="M217.7,139.6c-0.2,0-0.5-0.1-
0.7-0.3c-0.4-0.4-0.4-0.9,0-1.3l5.8-5.8c0.4-0.4,0.9-0.4,1.3,0
                                c0.4,0.4,0.4,0.9,0,1.3l-
5.8,5.8C218.2,139.5,218,139.6,217.7,139.6L217.7,139.6z"></path>
                </g>
```

```
            <circle id="XMLID_16_" fill="#FFFFFF" cx="223.7" cy="132.7"
r="2.1"></circle>
            <path id="XMLID_15_" fill="#602666" d="M224.8,132.7c0,0.6-0.5,1.1-
1.1,1.1c-0.6,0-1.1-0.5-1.1-1.1c0-0.6,0.5-1.1,1.1-1.1
                  C224.3,131.6,224.8,132.1,224.8,132.7z"></path>
            <g id="XMLID_12_">
               <path id="XMLID_13_" class="clock" fill="#F1939B" d="M223.7,133.1c-
0.2,0-0.4-0.1-0.4-0.3c-0.1-0.2,0.1-0.5,0.3-0.6l12-3.6
                        c0.2-0.1,0.5,0.1,0.6,0.3c0.1,0.2-0.1,0.5-0.3,0.6l-
12,3.6C223.8,133.1,223.8,133.1,223.7,133.1L223.7,133.1z"></path>
            </g>
            <g id="XMLID_6_">
               <defs>
                  <rect id="XMLID_7_" x="62.2" y="165.5" transform="matrix(0.9653 -
0.261 0.261 0.9653 -48.2874 26.8127)" width="29.2" height="59.4"></rect>
               </defs>
               <clipPath id="XMLID_157_c">
                  <use xlink:href="#XMLID_7_" style="overflow:visible;"></use>
               </clipPath>
               <g id="XMLID_8_" clip-path="url(#XMLID_157_c)">
                  <path id="XMLID_9_" fill="#7F245F" d="M72.2,192.8l-1.8,0.4c-
0.4,0.1-0.9-0.1-1.2-0.5l-0.5-0.7c-0.3-0.3-0.3-0.9-0.1-1.3l1-1.6
                        c0.2-0.4,0.1-0.8-0.2-1.1l-2.4-1.5c-0.4-0.2-0.9-0.1-
1.1,0.3l-1,1.6c-0.2,0.4-0.7,0.6-1.2,0.5l-0.9-0.1c-0.4,0-0.9-0.4-1-0.8
                        l-0.4-1.8c-0.1-0.4-0.5-0.7-0.9-0.6l-2.8,0.7c-0.4,0.1-
0.7,0.5-0.6,0.9l0.4,1.8c0.1,0.4-0.1,0.9-0.5,1.2l-0.7,0.5
                        c-0.3,0.3-0.9,0.3-1.3,0.1l-1.6-1c-0.4-0.2-0.8-0.1-
1.1,0.3l-1.5,2.4c-0.2,0.4-0.1,0.8,0.3,1.1l1.6,1c0.4,0.2,0.6,0.8,0.5,1.2
                        l-0.1,0.9c0,0.4-0.4,0.9-0.8,1l-1.8,0.4c-0.4,0.1-0.7,0.5-
0.6,0.9l0.7,2.8c0.1,0.4,0.5,0.7,0.9,0.6l1.8-0.4
                        c0.4-
0.1,0.9,0.1,1.2,0.5l0.5,0.7c0.3,0.3,0.3,0.9,0.1,1.3l-1,1.6c-0.2,0.4-
0.1,0.8,0.3,1.1l2.4,1.5c0.4,0.2,0.8,0.1,1.1-0.3
```

```
                                    l1-1.6c0.2-0.4,0.8-0.6,1.2-
0.5l0.9,0.1c0.4,0,0.9,0.4,1,0.8l0.4,1.8c0.1,0.4,0.5,0.7,0.9,0.6l2.8-0.7c0.4-0.1,0.7-0.5,0.6-
0.9
                                    l-0.4-1.8c-0.1-0.4,0.1-0.9,0.5-1.2l0.7-0.5c0.3-0.3,0.9-
0.3,1.3-0.1l1.6,1c0.4,0.2,0.8,0.1,1.1-0.3l1.5-2.4
                                    c0.2-0.4,0.1-0.8-0.3-1.1l-1.6-1c-0.4-0.2-0.6-0.8-0.5-
1.2l0.1-0.9c0-0.4,0.4-0.9,0.8-1l1.8-0.4c0.4-0.1,0.7-0.5,0.6-0.9
                                    l-0.7-2.8C73,193,72.6,192.7,72.2,192.8z
M62.7,201.2c-2,0.5-4.1-0.8-4.5-2.8c-0.5-2,0.8-4.1,2.8-4.5c2-0.5,4.1,0.8,4.5,2.8
                                    C65.9,198.7,64.7,200.7,62.7,201.2z"></path>
                </g>
            </g>
            <path id="XMLID_5_" class="star star-2 origin-center" fill="#FFFFFF"
d="M219.8,296.8v2.7h-2.7c-0.9,0-1.7,0.8-1.7,1.7v0.1c0,0.9,0.8,1.7,1.7,1.7h2.7v2.7
                    c0,0.9,0.8,1.7,1.7,1.7h0.1c0.9,0,1.7-0.8,1.7-1.7v-2.7h2.7c0.9,0,1.7-
0.8,1.7-1.7v-0.1c0-0.9-0.8-1.7-1.7-1.7h-2.7v-2.7
                    c0-0.9-0.8-1.7-1.7-1.7h-0.1C220.6,295.1,219.8,295.9,219.8,296.8z"
style="transform-origin: 221.55px 301.25px;"></path>
            <path id="XMLID_4_" class="star star-1 origin-center" fill="#FFFFFF"
d="M160.2,32.1v3h-3c-1,0-1.8,0.8-
1.8,1.8V37c0,1,0.8,1.8,1.8,1.8h3v3c0,1,0.8,1.8,1.8,1.8h0.1
                    c1,0,1.8-0.8,1.8-1.8v-3h3c1,0,1.8-0.8,1.8-1.8v-0.1c0-1-0.8-1.8-1.8-
1.8h-3v-3c0-1-0.8-1.8-1.8-1.8H162
                    C161,30.2,160.2,31.1,160.2,32.1z" style="transform-origin:
162.05px 36.9462px;"></path>
            <circle id="XMLID_3_" fill="#FFFFFF" cx="39.7" cy="231.5"
r="3.4"></circle>
        </g>
    </svg>

    <div class="Value-prop-heading">

        <h1>Your Digital Data Is Important.</h1>
```

```
            <div class="mobiledivider" style="display:none">
               <hr>
            </div>

            <div id="typerwriter-effect-line">
               <h2>Secure it </h2>
               <h2><p class="typewrite" data-period="2000" data-type='[ "from loss, ",
"from hackers, ", "for over a lifetime, "]'>
   <span class="wrap"></span></p></h2>
               <div class="headingmovemobile">
                  <h2>with <b>Project</b></h2></div>

            </div>
            <br>

            <div class="buttonmobiledisapear">
               <button type="button" class="btn btn-outline-primary btn-lg"
onclick="window.location.href = 'form.html';">Encrypt now</button>
            </div>

         </div>

      </div>

   </div>

   <!--- Features Section -->

   <div class="container-fluid padding">
      <div class="row welcome text-center">

         <div class="col-12">
```

```
            <h1 class="display-4">Project's features</h1>
        </div>
        <hr>

    </div>
</div>

<!--- Three Column Section -->
<div class="container-fluid padding">
    <div class="row text-center padding">

        <div class="col-xs-12 col-sm-6 col-md-4">
            <img src="img/user-friends-solid.svg" style="height:20%;width:20%">
            <h3>Inheritable</h3>
            <p>Project can be used to create Digital Inheritance systems.</p>
        </div>

        <div class="col-xs-12 col-sm-6 col-md-4">
            <img src="img/shield-alt-solid.svg" style="height:20%;width:20%">
            <h3>Quantum Resistant</h3>
            <p>Project's multi-layer encryption stack utilises quantum resistant
cryptography.</p>

        </div>

        <div class="col-xs-12 col-sm-6 col-md-4">
            <img src="img/user-secret-solid.svg" style="height:20%;width:20%">
            <h3>Attack Resistant</h3>
            <p>Project storage is highly secure against both digital and physical attacks.
            </p>

        </div>
```

```
    </div>

    <div class="row text-center padding">
        <div class="col-xs-12 col-sm-6 col-md-4">
            <img src="img/infinity-solid.svg" style="height:20%;width:20%">
            <h3>Passive</h3>
            <p>Secure and forget, no dead man's switch or smart contract needed.</p>

        </div>

        <div class="col-xs-12 col-sm-6 col-md-4">
            <img src="img/eraser-solid.svg" style="height:20%;width:20%">
            <h3>Deletable</h3>
            <p>The encrypted data can be deleted at anytime.</p>

        </div>

        <div class="col-xs-12 col-sm-6 col-md-4">
            <img src="img/mask-solid.svg" style="height:20%;width:20%">
            <h3>Private</h3>
            <p>None of your personal information needs to be shared with Project.</p>

        </div>

    </div>

    <hr class="my-4">
</div>

<!--- How it works Section -->
<div class="container-fluid padding">
    <div class="row padding" style="background-color:#232322; color: white;">
```

```
        <div class="col-10 col-lg-12">
          <div data-aos="zoom-in">
            <h1 class="display-4" style="text-align: center;">How it works</h1>
          </div>
        </div>

        <div class="row" style="width:100%;">

          <div class="col-10 col-lg-12">

            <div class="container">
              <ol class="step-list">
                <li class="step-list__item">
                  <div class="step-list__item__inner">
                    <div data-aos="fade-left">
                      <div class="content">

                        <div class="body">

                          <h2>Use Project's application</h2>
                          <p>To ensure your data’s safety, the Project application is
open source. You access it <a href="form.html">here</a></p>
                        </div>

                        <div class="icon">
                          <img src="img/download-solid.svg" alt="Check" />
                        </div>

                      </div>

                    </div>
                </li>
                <li class="step-list__item">
```

```
                        <div class="step-list__item__inner">
                            <div data-aos="fade-left">
                                <div class="content">
                                    <div class="body">
                                        <h2>Input Data To Be Saved</h2>
                                        <p>Username, passwords, private keys, seed words, receipts, messages…. If the data can be written in text, Project can secure it. </p>
                                    </div>

                                    <div class="icon">
                                        <img src="img/file-contract-solid.svg" alt="Check" />
                                    </div>
                                </div>
                            </div>
                        </div>
                    </li>
                    <li class="step-list__item">
                        <div class="step-list__item__inner">
                            <div data-aos="fade-left">
                                <div class="content">
                                    <div class="body">
                                        <h2>Specify Beneficaries</h2>
                                        <p>To tweak Project’s encryption stack to your needs, the application will need to know how many beneficiaries you want involved in the decryption process.
                                            <br>
                                            <br>Each of these beneficiaries will receive a piece of the decryption process which they can combine with yours, after you are gone, to decrypt your data.
                                            <br>
                                            <br>If you do not have any specific beneficiaries, or just want a personal data storage method for yourself, you can simply choose 0 beneficiaries. </p>
```

```
                        </div>

                        <div class="icon">
                            <img src="img/users-solid.svg" alt="Check" />
                        </div>
                    </div>
                </div>
            </div>
        </li>
        <li class="step-list__item">
            <div class="step-list__item__inner">
                <div data-aos="fade-left">
                    <div class="content">
                        <div class="body">
                            <h2>Encryption Process</h2>
                            <p>The data now goes through Project’s custom encryption process, fine-tuned to your previous beneficiary specifications. For technical details on Project’s encryption stack click here.</p>
                        </div>

                        <div class="icon">
                            <img src="img/cogs-solid.svg" alt="Check" />
                        </div>
                    </div>
                </div>
            </div>
        </li>
        <li class="step-list__item">
            <div class="step-list__item__inner">
                <div data-aos="fade-left">
                    <div class="content">
                        <div class="body">
                            <h2>Recieve Your Encrypted Data</h2>
```

```
                                <p>Your data has just been nullified, it technically doesn't exist anymore, but it will exist again when all the pieces of the decryption process are put back together.
                                    <br>
                                    <br>The pieces of the decryption process will be returned and the Project application will inform the data owner as to how the information should be stored based on their previous beneficary specifications.</p>
                                </div>

                                <div class="icon">
                                    <img src="img/lock-solid.svg" alt="Check" />
                                </div>
                            </div>
                        </div>
                    </div>
                </li>
                <li class="step-list__item">
                    <div class="step-list__item__inner">
                        <div data-aos="fade-left">
                            <div class="content">
                                <div class="body">
                                    <h2>Optional: Backup Decrytion Pieces With Project Cloud</h2>
                                    <p>Add full data redundancy for all your benefecaries by using Project cloud + blockchain backup.</p>
                                </div>

                                <div class="icon">
                                    <img src="img/cloud-upload-alt-solid.svg" alt="Check" />
                                </div>
                            </div>
                        </div>
                    </div>
```

```
                    </li>
                </ol>
                </div>

            </div>

        </div>

        <div class="col-2">

        </div>

    </div>
</div>

<!--- Pricing Section -->
<div class="container-fluid bg-light padding">
    <div class="row welcome text-center">
        <div class="col-12">
            <h1 class="display-4">Pricing</h1>
        </div>
        <hr>

    </div>

</div>

<!--- Cards -->
<div class="container-fluid bg-light padding">
    <div class="row text-center padding">

        <div class="col-lg-6">
```

```
            <div class="tablemove">
                <div class="pricing-table">
                    <div class="pricing-option" style="max-width: 550px; width: auto;">
                        <i class="material-icons">desktop_windows</i>
                        <h1>Project</h1>
                        <hr/>

                        <div class="pricedescription">
                            <p>Create your own custom data storage and inheritance systems with the Project application.</p>
                            <p><i class="fas fa-check-circle" style="color: #c1cd23;"></i> Data Storage</p>
                            <p><i class="fas fa-check-circle" style="color: #c1cd23;"></i> Inheritance System</p>
                            <p><i class="fas fa-times-circle" style="color: #cd2923;"></i> Cloud-based Data Redundancy</p>
                        </div>
                        <hr/>
                        <div class="price">
                            <div class="front">
                                <span class="price">Free <b></b></span>
                            </div>
                            <div class="back">
                                <a href="#" class="button"><i class="fa fa-download"></i> Use</a>
                            </div>
                        </div>
                    </div>
                </div>

            </div>
```

```
            <div class="col-lg-6">
                <div class="tablemove1">
                    <div class="pricing-table">
                        <div class="pricing-option" style="max-width: 550px; width: auto;">
                            <i class="material-icons">wb_cloudy</i>
                            <h1>Project cloud</h1>
                            <hr/>
                            <p>Make 100% sure your data is never lost with Project's backup
cloud.</p>
                            <p><i class="fas fa-check-circle" style="color: #c1cd23;"></i> Data
Storage</p>
                            <p><i class="fas fa-check-circle" style="color: #c1cd23;"></i>
Inheritance System</p>
                            <p><i class="fas fa-check-circle" style="color: #c1cd23;"></i> Cloud-
based Data Redundancy</p>

                            <hr/>
                            <div class="price">
                                <div class="front">
                                    <span class="price">FREE<b></b></span>
                                </div>
                                <div class="back">
                                    <a href="#" class="button">Use</a>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>

            </div>
```

```
</div>

<!--- Footer -->
<footer>

   <div class="container-fluid padding">
      <div class="row text-center">
         <div class="col-md-4">
            <hr class="light">
            <h5>Pages</h5>
            <hr class="light">
            <div class="footertext">
               <a href="index.html">
                  <p>Home</p>
               </a>
               <a href="form.html">
                  <p>Encrypt</p>
               </a>
               <a href="Sign_In.html">
                  <p>Sign In</p>
               </a>
               <a href="#">
                  <p>Get Cloud</p>
               </a>
            </div>
         </div>
         <div class="col-md-4">
            <hr class="light">
            <h5>Resources</h5>
            <hr class="light">
            <div class="footeritems" style="text-align:center;">
               <div class="footertext">
                  <a href="#">
```

```
                    <p>FAQ</p>
                </a>
                <a href="#">
                    <p>Blog</p>
                </a>
                <a href="#">
                    <p>Technical Documents</p>
                </a>

            </div>
        </div>
    </div>

    <div class="col-md-4">
        <hr class="light">
        <h5>Connect</h5>
        <hr class="light">
        <div class="row">

            <div class="col-md-3 social">
                <a href="#"><i class="fab fa-medium"></i></a>
            </div>

            <div class="col-md-3 social">
                <a href="#"><i class="fab fa-youtube"></i></a>
            </div>
            <div class="col-md-3 social">
                <a href="#"><i class="fab fa-twitter"></i></a>
            </div>

            <div class="col-md-3 social">
                <a href="#"><i class="fab fa-reddit-alien"></i></a>
            </div>
```

```
                </div>

                <div class="row">
                    <div class="col-12">
                        <div class="emailsection">
                            <svg style="vertical-align: middle;" width="18" height="18"
viewBox="0 0 24 24">
                                <path fill="#232322" d="M20 4H4C3 4 2 5 2 6v12c0 1 1 2 2
2h16c1 0 2-1 2-2V6c0-1-1-2-2-2zm0 4l-8 5-8-5V6l8 5 8-5v2z"></path>
                            </svg>
                            <a href="#">info@projectexample32432.com</a>
                        </div>
                    </div>
                </div>

            </div>

            <div class="col-12">
                <hr class="light">

                <div class="termsfooter">

                    <a href="#">
                        <p><u>Terms of Service</u></p>
                    </a>
                    <p> |</p>
                    <a href="#">
                        <p><u>Privacy Policy</u></p>
                    </a>
                    <p> |</p>
                    <a href="#">
                        <p><u>Cookie Policy</u></p>
```

```
                    </a>


                </div>
            </div>

        </div>
    </div>
</footer>

<script src="https://unpkg.com/aos@next/dist/aos.js"></script>
<script>
    AOS.init();

    AOS.init({
        disable: 'phone',
        disable: 'mobile',
    })
</script>

</body>

</html>
```

Style.css

```
/*---Media Queries --*/
@media (max-width: 992px) {

  }
@media (max-width: 768px) {

}
@media (max-width: 576px) {

}

._container {
    max-width: 1175px;
}
/*---Firefox Bug Fix --*/
.carousel-item {
  transition: -webkit-transform 0.5s ease;
  transition: transform 0.5s ease;
  transition: transform 0.5s ease, -webkit-transform 0.5s ease;
  -webkit-backface-visibility: visible;
  backface-visibility: visible;
}
/*--- Fixed Background Image --*/
figure {
  position: relative;
  width: 100%;
  height: 60%;
  margin: 0!important;
}
.fixed-wrap {
  clip: rect(0, auto, auto, 0);
  position: absolute;
```

```
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
}
#fixed {
  background-image: url('img/mac.png');
  position: fixed;
  display: block;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-size: cover;
  background-position: center center;
  -webkit-transform: translateZ(0);
          transform: translateZ(0);
  will-change: transform;
}
/*--- Bootstrap Padding Fix --*/
[class*="col-"] {
    padding: 1rem;
}



.icon-support .gear-s, .icon-support .gear-m {
            animation: rotate;
            animation-duration: 6s;
}
 .icon-support .gear-b {
            animation: rotate-reverse 6s;
}
 .icon-support .clock {
```

```
            transform-origin: 223.6px 132.8px;
            animation: rotate 12s linear infinite;
}
 .icon-support .hover {
            animation-play-state: paused;
            animation-iteration-count: infinite;
            animation-timing-function: linear;
            animation-play-state: running;
}

/*
 .icon-support:hover .hover {
            animation-play-state: running;
}

*/

 @keyframes rotate {
            to {
                        transform: rotate(1turn);
            }
}
 @keyframes rotate-reverse {
            to {
                        transform: rotate(-1turn);
            }
}



@media (min-width: 1490px) {
.icon-support{
           display: block !important;
```

```
}
}
```

```
.navbar-toggler:focus {
	outline: none!important;
	box-shadow: 0 0 5px rgba(193, 205, 35, 1) !important;

}

@media (max-width: 420px) {
.navbar-toggler{
	width: 100%;
	text-align: center;

}
```

```
.navbar-collapse {

        text-align: center;

}

}

.nav-item a {
        font-size: 18px;

}

.Value-prop-heading p {
  color: rgb(193,205,35);
}


.welcome hr {
        border-top: 2px solid #b4b4b4;
        width: 95%;

}

.Value-prop-heading{

        margin-top: 180px;

}

@media (max-width: 1490px) {
```

```
.Value-prop-heading {
        margin-top:0px;
        margin-top:70px;
}
}


@media (max-width: 576px) {
.Value-prop-heading{

        margin-top: 0px;

}

}




.Value-prop-heading h1 {
        font-size: 62px;


}
.Value-prop-heading h2 {
        font-size: 48px;


}
```

```
.headingmovemobile {
        display: inline;
}

.jumbotron {

        min-height: 550px;



}



#typerwriter-effect-line h2,
#typerwriter-effect-line p {
   display: inline;

}

@media (max-width: 576px) {
.headingmovemobile {
        display: block;

}
.mobiledivider {
        display: block !important;
}

.jumbotron {

        min-height: 600px;
        max-height: 600px;
}
.buttonmobiledisapear{
```

```
        display: none;

}
}


.btn-outline-primary, .btn-outline-primary:hover, .btn-outline-primary:visited {
   border-color: #c1cd23 !important;
        color: #232322 !important;



}


.btn-outline-primary:hover {
        background-color: #232322 !important;
        color: #ffffff !important;
}



.btn-outline-primary:active,.btn-outline-primary:focus {
        box-shadow: 0 0 5px rgba(193, 205, 35, 1) !important;

}




.column {
  float: left;
  height:100%;
  width: 100%;
```

```
  padding: 10px;

}

/* Clear floats after the columns */

/*
.row:after {
  content: "";
  display: table;
  clear: both;
}
*/









.column {



border: 1px solid;
}



/*
Extra small (xs) devices (portrait phones, less than 576px)
No media query since this is the default in Bootstrap

Small (sm) devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }
```

```
Medium (md) devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

Large (lg) devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

Extra (xl) large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }
*/



.tablemove {
        width: auto;
        float: right;

}

.tablemove1 {
        width: auto;
        float: left;
}

@media (max-width: 992px) {

.tablemove {
        float: none;
         text-align: center;
         display: inline-block;
}
.tablemove1 {
        float: none;
```

```
            text-align: center;
            display: inline-block;
}

}



.pricing-table {
            display: table;
            width: 100%;

}
 .pricing-table .pricing-option {
            width: auto;
            background: white;
            float: left;
            padding: 2%;
            text-align: center;
            transition: all 0.3s ease-in-out;
}
 .pricing-table .pricing-option:nth-child(even) {
            margin: 0 2%;
}
 .pricing-table .pricing-option:hover {
            cursor: pointer;
            box-shadow: 0px 2px 30px rgba(0, 0, 0, 0.3);
            transform: scale(1.04);
}
 .pricing-table .pricing-option:hover i, .pricing-table .pricing-option:hover h1, .pricing-table
.pricing-option:hover span, .pricing-table .pricing-option:hover b {
            color: #232322;
}
 .pricing-table .pricing-option:hover .front {
```

```
        opacity: 0;
        visibility: hidden;
}
 .pricing-table .pricing-option:hover .back {
        opacity: 1 !important;
        visibility: visible !important;
}
 .pricing-table .pricing-option:hover .back a.button {
        transform: translateY(0px) !important;
}
 .pricing-table .pricing-option hr {
        border: none;
        border-bottom: 1px solid #f0f0f0;
}
 .pricing-table .pricing-option i {
        font-size: 3rem;
        color: #d8d8d8;
        transition: all 0.3s ease-in-out;
}
 .pricing-table .pricing-option h1 {
        margin: 10px 0;
        color: #212121;
        transition: all 0.3s ease-in-out;
}
 .pricing-table .pricing-option p {
        color: #999;
        padding: 0 10px;
        line-height: 1.3;
}
 .pricing-table .pricing-option .price {
        position: relative;
}
 .pricing-table .pricing-option .price .front span.price {
```

```
        font-size: 2rem;
        text-transform: uppercase;
        margin-top: 20px;
        display: block;
        font-weight: 700;
        position: relative;
}
 .pricing-table .pricing-option .price .front span.price b {
        position: absolute;
        font-size: 1rem;
        margin-left: 2px;
        font-weight: 600;
}
 .pricing-table .pricing-option .price .back {
        opacity: 0;
        visibility: hidden;
        transition: all 0.3s ease-in-out;
}
 .pricing-table .pricing-option .price .back a.button {
        background: #c1cd23;
        padding: 15px 20px;
        display: inline-block;
        text-decoration: none;
        color: white;
        position: absolute;
        font-size: 13px;
        top: -5px;
        left: 0;
        right: 0;
        width: 150px;
        margin: auto;
        text-transform: uppercase;
        transform: translateY(20px);
```

```
        transition: all 0.3s ease-in-out;
}
 .pricing-table .pricing-option .price .back a.button:hover {
        background: #090909;
}
 @media screen and (max-width: 600px) {
        .pricing-table .pricing-option {
                padding: 5%;
                width: 90%;
        }
        .pricing-table .pricing-option:nth-child(even) {
                margin: 30px 0 !important;
        }
}
```

```
 .step-list {
        margin: 0;
        padding: 0;
        list-style-type: none;
}
 .step-list__item {
        counter-increment: step-counter;
        position: relative;
}
 .step-list__item:before {
        content: counter(step-counter);
        font-weight: 300;
```

```
        color: #d8d8d8;
}
 .step-list__item h2 {
        font-size: 24px;
        line-height: 30px;
        font-weight: 300;
}
 .step-list__item .icon {
        border-radius: 50%;
        display: block;
        flex: 0 0 auto;
}
 .step-list__item .icon img {
        width: 100%;
        height: 100%;
}
 .step-list__item .body {
        /*font-size: 12px;*/
        line-height: 18px;
        font-weight: 300;
}
 .step-list__item .content {
        display: flex;
        flex-direction: row;
        padding-bottom: 18px;
}
 .step-list__item:first-child .icon {
        background: #e9ecef;
}
 .step-list__item:nth-child(2) .icon {
        background: #e9ecef;
}
 .step-list__item:nth-child(3) .icon {
```

```
            background: #e9ecef;
}
 .step-list__item:nth-child(4) .icon {
            background: #e9ecef;
}
 .step-list__item:nth-child(5) .icon {
            background: #e9ecef;
}

 .step-list__item:nth-child(6) .icon {
            background: #e9ecef;
}
 .step-list__item:nth-child(odd) .content {
            justify-content: flex-start;
}
 .step-list__item:nth-child(even) .content {
            justify-content: flex-end;
}
 .step-list__item:first-child > .step-list__item__inner:before {
            content: none;
}
 .step-list__item:last-child > .step-list__item__inner:after {
            content: none;
}
 .step-list__item + li {
            margin-top: 84px;
}
 .step-list__item + li > div {
            margin-top: -1px;
}
 @media screen and (max-width: 600px) {
            .step-list__item {
                        display: flex;
```

```
        }
        .step-list__item:before {
                content: counter(step-counter);
                position: relative;
                font-size: 36px;
                line-height: 42px;
                font-weight: 300;
                color: #c1cd22; /*This changes color of numbers when screen is mobile
size*/
                margin-right: 12px;
        }
        .step-list__item .icon {
                order: 0;
                width: 36px;
                height: 36px;
                padding: 9px;
                position: absolute;
                left: -9px;
                top: 42px;
        }
        .step-list__item .body {
                order: 1;
                margin-top: 8px;
                margin-left: 12px;
        }
        .step-list__item .content {
                align-items: flex-start;
        }
        .step-list__item .content:before {
                content: '';
                position: absolute;
                border-left: #d8d8d8;
                height: 100%;
```

```
                left: 9px;
                bottom: 0;
                display: block;
                width: 1px;
                background: #d8d8d8;
                top: 84px;
        }
        .step-list__item:last-child .content:before {
                content: none;
        }
}
@media screen and (min-width: 601px) {
        .step-list__item:before {
                content: counter(step-counter);
                position: absolute;
                font-size: 90px;
                line-height: 1;
                font-weight: 300;
                color: #c1cd22; /* Change color of numbers here */
        }
        .step-list__item .icon {
                width: 174px;
                height: 174px;
                padding: 48px;
        }
        .step-list__item .body {
                flex: 0 1 33.3333%;
                margin-top: 84px;
        }
        .step-list__item .content {
                align-items: flex-end;
        }
        .step-list__item > .step-list__item__inner {
```

```
                position: relative;
        }
        .step-list__item > .step-list__item__inner:before, .step-list__item > .step-list__item__inner:after {
                border-width: 0px;
                border-style: solid;
                border-color: #d8d8d8;
                display: block;
                content: '';
                position: absolute;
                height: 42px;
                width: calc(33.3333% - 42px);
        }
        .step-list__item:nth-child(odd) > .step-list__item__inner {
                text-align: right;
        }
        .step-list__item:nth-child(odd) > .step-list__item__inner:before, .step-list__item:nth-child(odd) > .step-list__item__inner:after {
                border-left-width: 1px;
                left: calc(33.3333% - 21px);
        }
        .step-list__item:nth-child(odd) > .step-list__item__inner:before {
                border-top-width: 1px;
                border-top-left-radius: 42px;
                margin-top: -42px;
        }
        .step-list__item:nth-child(odd) > .step-list__item__inner:after {
                border-bottom-width: 1px;
                border-bottom-left-radius: 42px;
                margin-bottom: -42px;
        }
        .step-list__item:nth-child(odd):before {
                left: 0;
```

```
                margin-left: 33.3333%;
                transform: translateX(-100%);
        }
        .step-list__item:nth-child(odd) .icon {
                margin-left: 72px;
        }
        .step-list__item:nth-child(even) > .step-list__item__inner:before, .step-
list__item:nth-child(even) > .step-list__item__inner:after {
                border-right-width: 1px;
                right: calc(33.3333% - 21px);
        }
        .step-list__item:nth-child(even) > .step-list__item__inner:before {
                border-top-width: 1px;
                border-top-right-radius: 42px;
                margin-top: -42px;
        }
        .step-list__item:nth-child(even) > .step-list__item__inner:after {
                border-bottom-width: 1px;
                border-bottom-right-radius: 42px;
                margin-bottom: -42px;
        }
        .step-list__item:nth-child(even):before {
                right: 0;
                margin-right: 33.3333%;
                transform: translateX(100%);
        }
        .step-list__item:nth-child(even) .icon {
                margin-right: 72px;
        }
        .step-list__item:nth-child(even) .body {
                order: 1;
        }
}
```

```
.social a{
	font-size: 2.5em;


}

.fa-medium {
	color: #11100E;

}

.fa-youtube {
	color: #FF0000;
}


.fa-twitter {
	color: #1DA1F2;
}

.fa-reddit-alien {
	color: #FF4500;
}

.footertext p {
	color: #232322;

}
```

```
.footertext a:hover {
    color: #c1cd23;
}

.termsfooter p {
        display: inline;
        font-size:12px;
        color: #232322;
}

.emailsection a{
        color: #232322;

}


.emailsection a:hover {
        color: #c1cd23;
}
```

Scripts.js

```
// https://www.webredone.com/
let elementsCC = document.querySelectorAll('.origin-center');

elementsCC.forEach(element => {
  let bbox = element.getBBox(),
    x = bbox.x,
    y = bbox.y,
    w = bbox.width,
    h = bbox.height;

  //center center
  let resultCC = (x + (w / 2)) + 'px ' + (y + (h / 2)) + 'px';

  element.style.setProperty("transform-origin", resultCC)
}); // forEach



let elementsTL = document.querySelectorAll('.origin-left');

elementsTL.forEach(element => {
  let bbox = element.getBBox(),
    x = bbox.x,
    y = bbox.y,
    w = bbox.width,
    h = bbox.height;

  //top left
  let resultTL = x + 'px ' + y + 'px';

  element.style.setProperty("transform-origin", resultTL)
}); // forEach
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////
var TxtType = function(el, toRotate, period) {
        this.toRotate = toRotate;
        this.el = el;
        this.loopNum = 0;
        this.period = parseInt(period, 10) || 2000;
        this.txt = '';
        this.tick();
        this.isDeleting = false;
    };

    TxtType.prototype.tick = function() {
        var i = this.loopNum % this.toRotate.length;
        var fullTxt = this.toRotate[i];

        if (this.isDeleting) {
        this.txt = fullTxt.substring(0, this.txt.length - 1);
        } else {
        this.txt = fullTxt.substring(0, this.txt.length + 1);
        }

        this.el.innerHTML = '<span class="wrap">'+this.txt+'</span>';

        var that = this;
        var delta = 200 - Math.random() * 100;

        if (this.isDeleting) { delta /= 2; }

        if (!this.isDeleting && this.txt === fullTxt) {
        delta = this.period;
        this.isDeleting = true;
```

```
    } else if (this.isDeleting && this.txt === '') {
    this.isDeleting = false;
    this.loopNum++;
    delta = 500;
    }

    setTimeout(function() {
    that.tick();
    }, delta);
};

window.onload = function() {
    var elements = document.getElementsByClassName('typewrite');
    for (var i=0; i<elements.length; i++) {
        var toRotate = elements[i].getAttribute('data-type');
        var period = elements[i].getAttribute('data-period');
        if (toRotate) {
          new TxtType(elements[i], JSON.parse(toRotate), period);
        }
    }
    // INJECT CSS
    var css = document.createElement("style");
    css.type = "text/css";
    css.innerHTML = ".typewrite > .wrap { border-right: 0.08em solid #fff}";
    document.body.appendChild(css);
};




        $().ready(function(){
    $('[rel="tooltip"]').tooltip();
```

```
    $('a.scroll-down').click(function(e){
        e.preventDefault();
        scroll_target = $(this).data('href');
         $('html, body').animate({
            scrollTop: $(scroll_target).offset().top - 60
         }, 1000);
    });

});

function rotateCard(btn){
    var $card = $(btn).closest('.card-container');
    console.log($card);
    if($card.hasClass('hover')){
        $card.removeClass('hover');
    } else {
        $card.addClass('hover');
    }
}
```

Sign in Page

Sign_In.html

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="apple-touch-icon" sizes="180x180" href="img/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="img/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="img/favicon-16x16.png">

    <title>Project</title>

    <!-- Bootstrap core CSS -->
    <link href="styles/bootstrap.min.css" rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="styles/signin.css" rel="stylesheet">
</head>

<body class="text-center">
    <form class="form-signin">
        <h1>Project</h1>
        <h1 class="h3 mb-3 font-weight-normal">Sign in</h1>
        <label for="inputEmail" class="sr-only">Email address</label>
        <input type="email" id="inputEmail" class="form-control" placeholder="Email address" required="" autofocus="">
        <label for="inputPassword" class="sr-only">Password</label>
```

```
        <input type="password" id="inputPassword" class="form-control"
placeholder="Password" required="">
        <div class="checkbox mb-3">
          <label>
            <input type="checkbox" value="remember-me"> Remember me
          </label>
        </div>
        <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
        <p class="mt-5 mb-3 text-muted">© 2019-2020</p>
    </form>

</body>

</html>
```

Signin.css

```
html,
body {
  height: 100%;
}

body {
  display: -ms-flexbox;
  display: -webkit-box;
  display: flex;
  -ms-flex-align: center;
  -ms-flex-pack: center;
  -webkit-box-align: center;
  align-items: center;
  -webkit-box-pack: center;
  justify-content: center;
  padding-top: 40px;
  padding-bottom: 40px;
  background-color: #f5f5f5;
}

.form-signin {
  width: 100%;
  max-width: 330px;
  padding: 15px;
  margin: 0 auto;
}
.form-signin .checkbox {
  font-weight: 400;
}
.form-signin .form-control {
  position: relative;
  box-sizing: border-box;
```

```
  height: auto;
  padding: 10px;
  font-size: 16px;
}
.form-signin .form-control:focus {
  z-index: 2;
}
.form-signin input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
}
.form-signin input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;
}
```

Encrypt Page

Form.html

```
<html>

<head>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
    <link href="./styles/style123.css" rel="stylesheet">

    <style>
        .logoSection {
            text-align: center;
            margin-top: 50px;
            margin-bottom: 10px;
        }

        .contentOfTab {
            text-align: center;
        }

        #beneficary_num {
            font-size: 23px;
            margin-bottom: 15px;
```

```
}

#userinput {
    margin-bottom: 15px;
}

#password_input {
    margin-bottom: 15px;
}

.btn-success {
    background-color: rgb(193, 205, 35) !important;
    border-color: rgb(193, 205, 35) !important;
    outline: none !important;
    box-shadow: none !important;
}

.btn-success:hover,
.btn-success:active,
.btn-success:visited,
.btn-success:focus {
    background-color: #232322 !important;
    border-color: #232322 !important;
    outline: none !important;
    box-shadow: none !important;
}

.btn-dark {
    background-color: #232322 !important;
    border-color: #232322 !important;
    outline: none !important;
    box-shadow: none !important;
}
```

```
        .btn-dark:hover,
        .btn-dark:active,
        .btn-dark:visited,
        .btn-dark:focus {
            background-color: rgb(193, 205, 35) !important;
            border-color: rgb(193, 205, 35) !important;
            outline: none !important;
            box-shadow: none !important;
        }
    </style>

</head>

<body>

    <div class="tabs">
        <input type="radio" name="tabs" id="tabone" checked="checked">

        <label class="main-label">Step 1</label>
        <div class="tab">
            <div class="contentOfTab">
                <h2>Enter the information you want to make recoverable</h2>

                <form action="javascript:userInput()" method="get">
                    <textarea id="userinput" rows="4" cols="50" required></textarea>
                    <br>
                    <button type="submit" id="nextbutton" class="btn btn-
success">Next</button>
                </form>

            </div>
        </div>
```

```
    <input type="radio" name="tabs" id="tabtwo">

    <label class="main-label">Step 2</label>
    <div class="tab">
      <div class="contentOfTab">
        <h2>Number of beneficiaries</h2>

        <select id="beneficary_num">
          <option value="3">3</option>
          <option value="4">4</option>
          <option value="5">5</option>
          <option value="6">6</option>
          <option value="7">7</option>
          <option value="8">8</option>
          <option value="9">9</option>
          <option value="10">10</option>
          <option value="11">11</option>
          <option value="12">12</option>
          <option value="13">13</option>
          <option value="14">14</option>
          <option value="15">15</option>
          <option value="16">16</option>
          <option value="17">17</option>
          <option value="18">18</option>
          <option value="19">19</option>
          <option value="20">20</option>
        </select>
        <br>
        <button type="button" id="backbutton" class="back1 btn btn-
dark">Back</button>
        <button type="button" id="nextbutton" class="next2 btn btn-
success">Next</button>
```

```
        </div>
    </div>
    <input type="radio" name="tabs" id="tabthree">

    <label class="main-label">Step 3</label>
    <div class="tab">
        <div class="contentOfTab">
            <h2>Enter the password of the inheritance system</h2>

            <form action="javascript:Password()" method="get">
                <input type="text" maxlength="32" id="password_input"
name="password_input" required>
                <br>
                <div id="status"></div>
                <button type="button" id="backbutton" class="back2 btn btn-
dark">Back</button>
                <button type="submit" id="nextbutton" class="btn btn-
success">Finish</button>
            </form>

        </div>
    </div>
    <input type="radio" name="tabs" id="tabfour">
    <label class="main-label">Finished</label>
    <div class="tab">
        <div class="contentOfTab">
            <h2>Decryption Pieces</h2>
            <h3>Data for each of your beneficiaries</h3>
            <div class="aaa" style="word-wrap: break-word;">
                <div id="buttondiv"></div>
                <p id="demo"></p>
            </div>
```

```
            <h3>Store the below data in a legal document, such as a will, or in a safe. So that your beneficiaries can access it after you are gone.</h3>
            <p id="password-reveal"></p>
            <p>If you would like to backup your beneficiary's data incase they lose it, then try our paid bitherit data backup solution.</p>
            <button class="back3" onClick="window.location.reload();">Restart</button>
          </div>
        </div>
    </div>
    <div class="logoSection" style="height:105px;">
        <h1>Project</h1>
        <p class="mt-5 mb-3 text-muted">© 2019-2020</p>
    </div>

    <div class="FieldToHelpEasyCopy">
        <input type="input" id="keyValue" style="display:none">
    </div>
    <script src="./scripts/secrets.js"></script>
    <script src="./scripts/aes.js"></script>

    <script src="./scripts/SHA512.js"></script>
    <script src="./scripts/PBKDF2.js"></script>

    <script>
        var pbkdfPassword = "";
        var userInputToBeEncrypted;
        var encryptedHex;

        var elements = document.getElementsByTagName("textarea");
        for (var i = 0; i < elements.length; i++) {
            elements[i].oninvalid = function(e) {
                e.target.setCustomValidity("");
                if (!e.target.validity.valid) {
```

```
            e.target.setCustomValidity("Enter the information you want encrypted here");
        }
    };
    elements[i].oninput = function(e) {
        e.target.setCustomValidity("");
    };
}

elements = document.getElementsByTagName("input");
for (var i = 0; i < elements.length; i++) {
    elements[i].oninvalid = function(e) {
        e.target.setCustomValidity("");
        if (!e.target.validity.valid) {
            e.target.setCustomValidity("Enter a secure password");
        }
    };
    elements[i].oninput = function(e) {
        e.target.setCustomValidity("");
    };
}

function userInput() {

    userInputToBeEncrypted = document.getElementById("userinput").value;
    console.log(userInputToBeEncrypted);
    $("#tabtwo").prop("checked", true);
}

function Password() {

    var password = document.getElementById("password_input").value;

    var mypbkdf2 = new PBKDF2(password, "B17H357", 10000, 16);
```

```
        var status_callback = function(percent_done) {
            document.getElementById("status").innerHTML = "Securing " +
percent_done.toFixed(2) + "%";
        };
        var result_callback = function(key) {

            pbkdfPassword = key;
            aesEncrypt();

        };

        mypbkdf2.deriveKey(status_callback, result_callback);
        document.getElementById("password-reveal").innerHTML +=
"<strong>Password:</strong> " + password;

    }

    function aesEncrypt() {
        console.log(pbkdfPassword);

        var pbkdfPasswordArray = pbkdfPassword.split("");

        console.log(pbkdfPasswordArray);

        //Convert character to unit8 num without affecting existing num
        for (var i = 0, len = pbkdfPasswordArray.length; i < len; i++) {

            if (!(pbkdfPasswordArray[i] in ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"])) {
                var currentPasswordArrayLetter = pbkdfPasswordArray[i];
                currentPasswordArrayLetter.charCodeAt(0);
                var letterToNum = currentPasswordArrayLetter.charCodeAt(0);;
                pbkdfPasswordArray[i] = letterToNum;
```

```
    }

}
console.log(pbkdfPasswordArray);
var key = new Uint8Array(pbkdfPasswordArray);
console.log(key);
////////////////////////////////////////////////////////////////

// Convert text to bytes
var text = userInputToBeEncrypted;
var textBytes = aesjs.utils.utf8.toBytes(text);

// The counter is optional, and if omitted will begin at 1
var aesCtr = new aesjs.ModeOfOperation.ctr(key, new aesjs.Counter(5));
var encryptedBytes = aesCtr.encrypt(textBytes);

// To print or store the binary data, you may convert it to hex
encryptedHex = aesjs.utils.hex.fromBytes(encryptedBytes);
console.log(encryptedHex);
// "a338eda3874ed884b6199150d36f49988c90f5c47fe7792b0cf8c7f77eeffd87
//  ea145b73e82aefcf2076f881c88879e4e25b1d7b24ba2788"

// When ready to decrypt the hex string, convert it back to bytes
var encryptedBytes = aesjs.utils.hex.toBytes(encryptedHex);

// The counter mode of operation maintains internal state, so to
// decrypt a new instance must be instantiated.
var aesCtr = new aesjs.ModeOfOperation.ctr(key, new aesjs.Counter(5));
var decryptedBytes = aesCtr.decrypt(encryptedBytes);

// Convert our bytes back into text
var decryptedText = aesjs.utils.utf8.fromBytes(decryptedBytes);
console.log(decryptedText);
```

```
        // "Text may be any length you wish, no padding is required."

        sssEncrypt();
    }

    function sssEncrypt() {

        console.log(encryptedHex);

        var shareAmount = document.getElementById("beneficary_num").value;

        console.log(shareAmount);
        shareAmount = parseInt(shareAmount);

        var thresholdNumber = shareAmount;
        // convert the text into a hex string
        var pwHex = secrets.str2hex(encryptedHex) // => hex string

        // split into 5 shares, with a threshold of 3
        var shares = secrets.share(pwHex, shareAmount, thresholdNumber)

        console.log(shares);
        let multiCollapse;
        for (i = 0; i < shareAmount; i++) {
            var j = i + 1;
            multiCollapse += "multiCollapseExample" + j;
            document.getElementById("demo").innerHTML += "<strong>Key for
beneficiary " + j + "</strong> " + "<i id='buttonNumber" + i + "' style='cursor:
pointer;font-size:14px;color:grey;'>Copy</i><br>" + "<div class='collapse multi-collapse'
id='multiCollapseExample" + j + "'><div id='shareNumber" + i + "'>" + shares[i] +
"</div></div>";
            if (j === shareAmount) {
                console.log("reached");
```

```
                document.getElementById("buttondiv").innerHTML += "<button class='btn
btn-success' type='button' data-toggle='collapse' data-target='.multi-collapse' aria-
expanded='false' aria-controls='" + multiCollapse + "'>Show Decryption Keys</button>";
            }
        }

        ////////////////////////////////////////////////////////////////////////////////
        for (let i = 0; i < shareAmount; i++) {
            var currentButton = "buttonNumber" + i
            var currentShare = "shareNumber" + i
            keyDiv = document.getElementById(currentButton)

            keyDiv.addEventListener('click', shareClickHandler);
            keyDiv.keyval = document.getElementById(currentShare).innerText

        }
        ////////////////////////////////////////////////////////

        finished()
    }

    function finished() {
        $("#tabfour").prop("checked", true);

    }

    function getIdNameToCopy() {

        let varforid =

            copyText(varforid)
        return copyText()
    }
```

```
function shareClickHandler(evt) {
    var divKeyValue = evt.currentTarget.keyval
    console.log(divKeyValue)
        //test = document.getElementById(id).innerText
    let hiddenInput = document.getElementById("keyValue");
    hiddenInput.value = divKeyValue;
    //console.log(test)
    copyText();
}

function copyText() {
    /* Get the text field */
    var copyText = document.getElementById("keyValue");
    copyText.style.display = "block";

    /* Select the text field */
    copyText.select();
    copyText.setSelectionRange(0, 99999); /*For mobile devices*/

    /* Copy the text inside the text field */
    document.execCommand("copy");
    copyText.style.display = "none";
    /* Alert the copied text */
    // alert("Copied the text: " + copyText.value);
}

//Backwards button on page 2 and 3, forward button on page 2
$(document).ready(function() {
    $(".back1").click(function() {
        $("#tabone").prop("checked", true);
    });
    $(".next2").click(function() {
```

```
            $("#tabthree").prop("checked", true);
        });
        $(".back2").click(function() {
            $("#tabtwo").prop("checked", true);
        });
        $(".back3").click(function() {
            $("#tabone").prop("checked", true);
        });
    });
  </script>

</body>

</html>
```

Style123.css

```
/**
 * Tabs
 */
.tabs {
        display: flex;
        flex-wrap: wrap; // make sure it wraps
}
.main-label {
        order: 1; // Put the labels first
        display: block;
        padding: 1rem 2rem;
        margin-right: 0.2rem;
  background: #232322;
  font-weight: bold;
  transition: background ease 0.2s;
  border-left: 10px solid rgb(193,205,35);
  color: white;
  margin-bottom: 0px;
}
.tabs .tab {
  order: 99; // Put the tabs last
  flex-grow: 1;
        width: 100%;
        display: none;
 padding: 1rem;
  background: #fff;
  min-height:260px;



}
.tabs input[type="radio"] {
        display: none;
```

```
}
.tabs input[type="radio"]:checked + label {
	background: #fff;
	background-color: white;
	color: black;
}
.tabs input[type="radio"]:checked + label + .tab {
	display: block;
}

@media (max-width: 45em) {
  .tabs .tab,
  .tabs label {
    order: initial;
  }
  .tabs label {
    width: 100%;
    margin-right: 0;
    margin-top: 0.2rem;
  }
}

/**
 * Generic Styling
*/
body {
  background: #eee;
  min-height: 100vh;
	box-sizing: border-box;
	padding-top: 10vh;
  font-family: "HelveticaNeue-Light", "Helvetica Neue Light", "Helvetica Neue",
Helvetica, Arial, "Lucida Grande", sans-serif;
  font-weight: 300;
```

```
  line-height: 1.5;
  max-width: 60rem;
  margin: 0 auto;
  font-size: 112%;
}
```

Secrets.js

```
// Alexander Stetsyuk
// Glenn Rempe <glenn@rempe.us>
// @license MIT

;(function(root, factory) {
    "use strict"

    if (typeof define === "function" && define.amd) {
        // AMD. Register as an anonymous module.
        define([], function() {
            /*eslint-disable no-return-assign */
            return (root.secrets = factory())
            /*eslint-enable no-return-assign */
        })
    } else if (typeof exports === "object") {
        // Node. Does not work with strict CommonJS, but
        // only CommonJS-like environments that support module.exports,
        // like Node.
        module.exports = factory(require("crypto"))
    } else {
        // Browser globals (root is window)
        root.secrets = factory(root.crypto)
    }
})(this, function(crypto) {
    "use strict"

    var defaults, config, preGenPadding, runCSPRNGTest, CSPRNGTypes

    function reset() {
        defaults = {
            bits: 8, // default number of bits
            radix: 16, // work with HEX by default
```

```
        minBits: 3,
        maxBits: 20, // this permits 1,048,575 shares, though going this high is NOT
recommended in JS!
        bytesPerChar: 2,
        maxBytesPerChar: 6, // Math.pow(256,7) > Math.pow(2,53)

        // Primitive polynomials (in decimal form) for Galois Fields GF(2^n), for 2 <= n <=
30
        // The index of each term in the array corresponds to the n for that polynomial
        // i.e. to get the polynomial for n=16, use primitivePolynomials[16]
        primitivePolynomials: [
          null,
          null,
          1,
          3,
          3,
          5,
          3,
          3,
          29,
          17,
          9,
          5,
          83,
          27,
          43,
          3,
          45,
          9,
          39,
          39,
          9,
          5,
```

```
            3,
            33,
            27,
            9,
            71,
            39,
            9,
            5,
            83
          ]
      }
      config = {}
      preGenPadding = new Array(1024).join("0") // Pre-generate a string of 1024 0's for
use by padLeft().
      runCSPRNGTest = true

      // WARNING : Never use 'testRandom' except for testing.
      CSPRNGTypes = [
        "nodeCryptoRandomBytes",
        "browserCryptoGetRandomValues",
        "testRandom"
      ]
   }

   function isSetRNG() {
      if (config && config.rng && typeof config.rng === "function") {
        return true
      }

      return false
   }

   // Pads a string `str` with zeros on the left so that its length is a multiple of `bits`
```

```
function padLeft(str, multipleOfBits) {
    var missing

    if (multipleOfBits === 0 || multipleOfBits === 1) {
        return str
    }

    if (multipleOfBits && multipleOfBits > 1024) {
        throw new Error(
            "Padding must be multiples of no larger than 1024 bits."
        )
    }

    multipleOfBits = multipleOfBits || config.bits

    if (str) {
        missing = str.length % multipleOfBits
    }

    if (missing) {
        return (preGenPadding + str).slice(
            -(multipleOfBits - missing + str.length)
        )
    }

    return str
}

function hex2bin(str) {
    var bin = "",
        num,
        i
```

```
    for (i = str.length - 1; i >= 0; i--) {
        num = parseInt(str[i], 16)

        if (isNaN(num)) {
            throw new Error("Invalid hex character.")
        }

        bin = padLeft(num.toString(2), 4) + bin
    }
    return bin
}

function bin2hex(str) {
    var hex = "",
        num,
        i

    str = padLeft(str, 4)

    for (i = str.length; i >= 4; i -= 4) {
        num = parseInt(str.slice(i - 4, i), 2)
        if (isNaN(num)) {
            throw new Error("Invalid binary character.")
        }
        hex = num.toString(16) + hex
    }

    return hex
}

// Browser supports crypto.getRandomValues()
function hasCryptoGetRandomValues() {
    if (
```

```
        crypto &&
        typeof crypto === "object" &&
        (typeof crypto.getRandomValues === "function" ||
            typeof crypto.getRandomValues === "object") &&
        (typeof Uint32Array === "function" ||
            typeof Uint32Array === "object")
    ) {
        return true
    }

    return false
}

// Node.js support for crypto.randomBytes()
function hasCryptoRandomBytes() {
    if (
        typeof crypto === "object" &&
        typeof crypto.randomBytes === "function"
    ) {
        return true
    }

    return false
}

// Returns a pseudo-random number generator of the form function(bits){}
// which should output a random string of 1's and 0's of length `bits`.
// `type` (Optional) : A string representing the CSPRNG that you want to
// force to be loaded, overriding feature detection. Can be one of:
//    "nodeCryptoRandomBytes"
//    "browserCryptoGetRandomValues"
//
function getRNG(type) {
```

```
        function construct(bits, arr, radix, size) {
            var i = 0,
                len,
                str = "",
                parsedInt

            if (arr) {
                len = arr.length - 1
            }

            while (i < len || str.length < bits) {
                // convert any negative nums to positive with Math.abs()
                parsedInt = Math.abs(parseInt(arr[i], radix))
                str = str + padLeft(parsedInt.toString(2), size)
                i++
            }

            str = str.substr(-bits)

            // return null so this result can be re-processed if the result is all 0's.
            if ((str.match(/0/g) || []).length === str.length) {
                return null
            }

            return str
        }

        // Node.js : crypto.randomBytes()
        // Note : Node.js and crypto.randomBytes() uses the OpenSSL RAND_bytes()
function for its CSPRNG.
        //       Node.js will need to have been compiled with OpenSSL for this to work.
```

```
        // See :
https://github.com/joyent/node/blob/d8baf8a2a4481940bfed0196308ae6189ca18eee/src/no
de_crypto.cc#L4696
        // See : https://www.openssl.org/docs/crypto/rand.html
        function nodeCryptoRandomBytes(bits) {
            var buf,
                bytes,
                radix,
                size,
                str = null

            radix = 16
            size = 4
            bytes = Math.ceil(bits / 8)

            while (str === null) {
                buf = crypto.randomBytes(bytes)
                str = construct(bits, buf.toString("hex"), radix, size)
            }

            return str
        }

        // Browser : crypto.getRandomValues()
        // See : https://dvcs.w3.org/hg/webcrypto-api/raw-file/tip/spec/Overview.html#dfn-
Crypto
        // See : https://developer.mozilla.org/en-
US/docs/Web/API/RandomSource/getRandomValues
        // Supported Browsers : http://caniuse.com/#search=crypto.getRandomValues
        function browserCryptoGetRandomValues(bits) {
            var elems,
                radix,
                size,
```

```
        str = null

    radix = 10
    size = 32
    elems = Math.ceil(bits / 32)
    while (str === null) {
        str = construct(
            bits,
            crypto.getRandomValues(new Uint32Array(elems)),
            radix,
            size
        )
    }

    return str
}

// ////////////////////////////////////////////////////////////
// WARNING : DO NOT USE. For testing purposes only.
// ////////////////////////////////////////////////////////////
// This function will return repeatable non-random test bits. Can be used
// for testing only. Node.js does not return proper random bytes
// when run within a PhantomJS container.
function testRandom(bits) {
    var arr,
        elems,
        int,
        radix,
        size,
        str = null

    radix = 10
    size = 32
```

```
            elems = Math.ceil(bits / 32)
            int = 123456789
            arr = new Uint32Array(elems)

            // Fill every element of the Uint32Array with the same int.
            for (var i = 0; i < arr.length; i++) {
                arr[i] = int
            }

            while (str === null) {
                str = construct(bits, arr, radix, size)
            }

            return str
        }

        // Return a random generator function for browsers that support
        // crypto.getRandomValues() or Node.js compiled with OpenSSL support.
        // WARNING : NEVER use testRandom outside of a testing context. Totally non-
random!
        if (type && type === "testRandom") {
            config.typeCSPRNG = type
            return testRandom
        } else if (type && type === "nodeCryptoRandomBytes") {
            config.typeCSPRNG = type
            return nodeCryptoRandomBytes
        } else if (type && type === "browserCryptoGetRandomValues") {
            config.typeCSPRNG = type
            return browserCryptoGetRandomValues
        } else if (hasCryptoRandomBytes()) {
            config.typeCSPRNG = "nodeCryptoRandomBytes"
            return nodeCryptoRandomBytes
        } else if (hasCryptoGetRandomValues()) {
```

```
        config.typeCSPRNG = "browserCryptoGetRandomValues"
        return browserCryptoGetRandomValues
    }
}

// Splits a number string `bits`-length segments, after first
// optionally zero-padding it to a length that is a multiple of `padLength.
// Returns array of integers (each less than 2^bits-1), with each element
// representing a `bits`-length segment of the input string from right to left,
// i.e. parts[0] represents the right-most `bits`-length segment of the input string.
function splitNumStringToIntArray(str, padLength) {
    var parts = [],
        i

    if (padLength) {
        str = padLeft(str, padLength)
    }

    for (i = str.length; i > config.bits; i -= config.bits) {
        parts.push(parseInt(str.slice(i - config.bits, i), 2))
    }

    parts.push(parseInt(str.slice(0, i), 2))

    return parts
}

// Polynomial evaluation at `x` using Horner's Method
// NOTE: fx=fx * x + coeff[i] ->  exp(log(fx) + log(x)) + coeff[i],
//       so if fx===0, just set fx to coeff[i] because
//       using the exp/log form will result in incorrect value
function horner(x, coeffs) {
    var logx = config.logs[x],
```

```
        fx = 0,
        i

    for (i = coeffs.length - 1; i >= 0; i--) {
        if (fx !== 0) {
            fx =
                config.exps[(logx + config.logs[fx]) % config.maxShares] ^
                coeffs[i]
        } else {
            fx = coeffs[i]
        }
    }

    return fx
}

// Evaluate the Lagrange interpolation polynomial at x = `at`
// using x and y Arrays that are of the same length, with
// corresponding elements constituting points on the polynomial.
function lagrange(at, x, y) {
    var sum = 0,
        len,
        product,
        i,
        j

    for (i = 0, len = x.length; i < len; i++) {
        if (y[i]) {
            product = config.logs[y[i]]

            for (j = 0; j < len; j++) {
                if (i !== j) {
                    if (at === x[j]) {
```

```
                    // happens when computing a share that is in the list of shares used to
compute it
                    product = -1 // fix for a zero product term, after which the sum should be
sum^0 = sum, not sum^1
                    break
                }
                product =
                    (product +
                        config.logs[at ^ x[j]] -
                        config.logs[x[i] ^ x[j]] +
                        config.maxShares) %
                    config.maxShares // to make sure it's not negative
            }
        }

        // though exps[-1] === undefined and undefined ^ anything = anything in
        // chrome, this behavior may not hold everywhere, so do the check
        sum = product === -1 ? sum : sum ^ config.exps[product]
    }
  }

  return sum
}

// This is the basic polynomial generation and evaluation function
// for a `config.bits`-length secret (NOT an arbitrary length)
// Note: no error-checking at this stage! If `secret` is NOT
// a NUMBER less than 2^bits-1, the output will be incorrect!
function getShares(secret, numShares, threshold) {
  var shares = [],
    coeffs = [secret],
    i,
    len
```

```
    for (i = 1; i < threshold; i++) {
        coeffs[i] = parseInt(config.rng(config.bits), 2)
    }

    for (i = 1, len = numShares + 1; i < len; i++) {
        shares[i - 1] = {
            x: i,
            y: horner(i, coeffs)
        }
    }

    return shares
}

function constructPublicShareString(bits, id, data) {
    var bitsBase36, idHex, idMax, idPaddingLen, newShareString

    id = parseInt(id, config.radix)
    bits = parseInt(bits, 10) || config.bits
    bitsBase36 = bits.toString(36).toUpperCase()
    idMax = Math.pow(2, bits) - 1
    idPaddingLen = idMax.toString(config.radix).length
    idHex = padLeft(id.toString(config.radix), idPaddingLen)

    if (typeof id !== "number" || id % 1 !== 0 || id < 1 || id > idMax) {
        throw new Error(
            "Share id must be an integer between 1 and " +
                idMax +
                ", inclusive."
        )
    }
```

```
    newShareString = bitsBase36 + idHex + data

    return newShareString
}

// EXPORTED FUNCTIONS
// ////////////////

var secrets = {
    init: function(bits, rngType) {
        var logs = [],
            exps = [],
            x = 1,
            primitive,
            i

        // reset all config back to initial state
        reset()

        if (
            bits &&
            (typeof bits !== "number" ||
                bits % 1 !== 0 ||
                bits < defaults.minBits ||
                bits > defaults.maxBits)
        ) {
            throw new Error(
                "Number of bits must be an integer between " +
                    defaults.minBits +
                    " and " +
                    defaults.maxBits +
                    ", inclusive."
            )
```

```
    }

    if (rngType && CSPRNGTypes.indexOf(rngType) === -1) {
        throw new Error("Invalid RNG type argument : '" + rngType + "'")
    }

    config.radix = defaults.radix
    config.bits = bits || defaults.bits
    config.size = Math.pow(2, config.bits)
    config.maxShares = config.size - 1

    // Construct the exp and log tables for multiplication.
    primitive = defaults.primitivePolynomials[config.bits]

    for (i = 0; i < config.size; i++) {
        exps[i] = x
        logs[x] = i
        x = x << 1 // Left shift assignment
        if (x >= config.size) {
            x = x ^ primitive // Bitwise XOR assignment
            x = x & config.maxShares // Bitwise AND assignment
        }
    }

    config.logs = logs
    config.exps = exps

    if (rngType) {
        this.setRNG(rngType)
    }

    if (!isSetRNG()) {
        this.setRNG()
```

```
    }

    if (
        !isSetRNG() ||
        !config.bits ||
        !config.size ||
        !config.maxShares ||
        !config.logs ||
        !config.exps ||
        config.logs.length !== config.size ||
        config.exps.length !== config.size
    ) {
        throw new Error("Initialization failed.")
    }
},

// Evaluates the Lagrange interpolation polynomial at x=`at` for
// individual config.bits-length segments of each share in the `shares`
// Array. Each share is expressed in base `inputRadix`. The output
// is expressed in base `outputRadix'.
combine: function(shares, at) {
    var i,
        j,
        len,
        len2,
        result = "",
        setBits,
        share,
        splitShare,
        x = [],
        y = []

    at = at || 0
```

```
        for (i = 0, len = shares.length; i < len; i++) {
            share = this.extractShareComponents(shares[i])

            // All shares must have the same bits settings.
            if (setBits === undefined) {
                setBits = share.bits
            } else if (share.bits !== setBits) {
                throw new Error(
                    "Mismatched shares: Different bit settings."
                )
            }

            // Reset everything to the bit settings of the shares.
            if (config.bits !== setBits) {
                this.init(setBits)
            }

            // Proceed if this share.id is not already in the Array 'x' and
            // then split each share's hex data into an Array of Integers,
            // then 'rotate' those arrays where the first element of each row is converted to
            // its own array, the second element of each to its own Array, and so on for all of
the rest.
            // Essentially zipping all of the shares together.
            //
            // e.g.
            //   [ 193, 186, 29, 150, 5, 120, 44, 46, 49, 59, 6, 1, 102, 98, 177, 196 ]
            //   [ 53, 105, 139, 49, 187, 240, 91, 92, 98, 118, 12, 2, 204, 196, 127, 149 ]
            //   [ 146, 211, 249, 167, 209, 136, 118, 114, 83, 77, 10, 3, 170, 166, 206, 81 ]
            //
            // becomes:
            //
            // [ [ 193, 53, 146 ],
```

```
        //   [ 186, 105, 211 ],
        //   [ 29, 139, 249 ],
        //   [ 150, 49, 167 ],
        //   [ 5, 187, 209 ],
        //   [ 120, 240, 136 ],
        //   [ 44, 91, 118 ],
        //   [ 46, 92, 114 ],
        //   [ 49, 98, 83 ],
        //   [ 59, 118, 77 ],
        //   [ 6, 12, 10 ],
        //   [ 1, 2, 3 ],
        //   [ 102, 204, 170 ],
        //   [ 98, 196, 166 ],
        //   [ 177, 127, 206 ],
        //   [ 196, 149, 81 ] ]
        //
        if (x.indexOf(share.id) === -1) {
            x.push(share.id)
            splitShare = splitNumStringToIntArray(hex2bin(share.data))
            for (j = 0, len2 = splitShare.length; j < len2; j++) {
                y[j] = y[j] || []
                y[j][x.length - 1] = splitShare[j]
            }
        }
    }

    // Extract the secret from the 'rotated' share data and return a
    // string of Binary digits which represent the secret directly. or in the
    // case of a newShare() return the binary string representing just that
    // new share.
    for (i = 0, len = y.length; i < len; i++) {
        result = padLeft(lagrange(at, x, y[i]).toString(2)) + result
    }
```

```
            // If 'at' is non-zero combine() was called from newShare(). In this
            // case return the result (the new share data) directly.
            //
            // Otherwise find the first '1' which was added in the share() function as a padding
marker
            // and return only the data after the padding and the marker. Convert this Binary
string
            // to hex, which represents the final secret result (which can be converted from hex
back
            // to the original string in user space using `hex2str()`).
            return bin2hex(
                at >= 1 ? result : result.slice(result.indexOf("1") + 1)
            )
        },

        getConfig: function() {
            var obj = {}
            obj.radix = config.radix
            obj.bits = config.bits
            obj.maxShares = config.maxShares
            obj.hasCSPRNG = isSetRNG()
            obj.typeCSPRNG = config.typeCSPRNG
            return obj
        },

        // Given a public share, extract the bits (Integer), share ID (Integer), and share data
(Hex)
        // and return an Object containing those components.
        extractShareComponents: function(share) {
            var bits,
                id,
                idLen,
```

```
    max,
    obj = {},
    regexStr,
    shareComponents

// Extract the first char which represents the bits in Base 36
bits = parseInt(share.substr(0, 1), 36)

if (
    bits &&
    (typeof bits !== "number" ||
        bits % 1 !== 0 ||
        bits < defaults.minBits ||
        bits > defaults.maxBits)
) {
    throw new Error(
        "Invalid share : Number of bits must be an integer between " +
            defaults.minBits +
            " and " +
            defaults.maxBits +
            ", inclusive."
    )
}

// calc the max shares allowed for given bits
max = Math.pow(2, bits) - 1

// Determine the ID length which is variable and based on the bit count.
idLen = (Math.pow(2, bits) - 1).toString(config.radix).length

// Extract all the parts now that the segment sizes are known.
regexStr =
    "^([a-kA-K3-9]{1})([a-fA-F0-9]{" + idLen + "})([a-fA-F0-9]+)$"
```

```
        shareComponents = new RegExp(regexStr).exec(share)

        // The ID is a Hex number and needs to be converted to an Integer
        if (shareComponents) {
            id = parseInt(shareComponents[2], config.radix)
        }

        if (typeof id !== "number" || id % 1 !== 0 || id < 1 || id > max) {
            throw new Error(
                "Invalid share : Share id must be an integer between 1 and " +
                    config.maxShares +
                    ", inclusive."
            )
        }

        if (shareComponents && shareComponents[3]) {
            obj.bits = bits
            obj.id = id
            obj.data = shareComponents[3]
            return obj
        }

        throw new Error("The share data provided is invalid : " + share)
    },

    // Set the PRNG to use. If no RNG function is supplied, pick a default using getRNG()
    setRNG: function(rng) {
        var errPrefix = "Random number generator is invalid ",
            errSuffix =
                " Supply an CSPRNG of the form function(bits){} that returns a string
containing 'bits' number of random 1's and 0's."

        if (
```

```
    rng &&
    typeof rng === "string" &&
    CSPRNGTypes.indexOf(rng) === -1
) {
    throw new Error("Invalid RNG type argument : '" + rng + "'")
}

// If RNG was not specified at all,
// try to pick one appropriate for this env.
if (!rng) {
    rng = getRNG()
}

// If `rng` is a string, try to forcibly
// set the RNG to the type specified.
if (rng && typeof rng === "string") {
    rng = getRNG(rng)
}

if (runCSPRNGTest) {
    if (rng && typeof rng !== "function") {
        throw new Error(errPrefix + "(Not a function)." + errSuffix)
    }

    if (rng && typeof rng(config.bits) !== "string") {
        throw new Error(
            errPrefix + "(Output is not a string)." + errSuffix
        )
    }

    if (rng && !parseInt(rng(config.bits), 2)) {
        throw new Error(
            errPrefix +
```

```
                "(Binary string output not parseable to an Integer)." +
                errSuffix
            )
        }

        if (rng && rng(config.bits).length > config.bits) {
            throw new Error(
                errPrefix +
                "(Output length is greater than config.bits)." +
                errSuffix
            )
        }

        if (rng && rng(config.bits).length < config.bits) {
            throw new Error(
                errPrefix +
                "(Output length is less than config.bits)." +
                errSuffix
            )
        }
    }

    config.rng = rng

    return true
},

// Converts a given UTF16 character string to the HEX representation.
// Each character of the input string is represented by
// `bytesPerChar` bytes in the output string which defaults to 2.
str2hex: function(str, bytesPerChar) {
    var hexChars,
        max,
```

```
    out = "",
    neededBytes,
    num,
    i,
    len

if (typeof str !== "string") {
    throw new Error("Input must be a character string.")
}

if (!bytesPerChar) {
    bytesPerChar = defaults.bytesPerChar
}

if (
    typeof bytesPerChar !== "number" ||
    bytesPerChar < 1 ||
    bytesPerChar > defaults.maxBytesPerChar ||
    bytesPerChar % 1 !== 0
) {
    throw new Error(
        "Bytes per character must be an integer between 1 and " +
            defaults.maxBytesPerChar +
            ", inclusive."
    )
}

hexChars = 2 * bytesPerChar
max = Math.pow(16, hexChars) - 1

for (i = 0, len = str.length; i < len; i++) {
    num = str[i].charCodeAt()
```

```
            if (isNaN(num)) {
                throw new Error("Invalid character: " + str[i])
            }

            if (num > max) {
                neededBytes = Math.ceil(Math.log(num + 1) / Math.log(256))
                throw new Error(
                    "Invalid character code (" +
                        num +
                        "). Maximum allowable is 256^bytes-1 (" +
                        max +
                        "). To convert this character, use at least " +
                        neededBytes +
                        " bytes."
                )
            }

            out = padLeft(num.toString(16), hexChars) + out
        }
        return out
    },

    // Converts a given HEX number string to a UTF16 character string.
    hex2str: function(str, bytesPerChar) {
        var hexChars,
            out = "",
            i,
            len

        if (typeof str !== "string") {
            throw new Error("Input must be a hexadecimal string.")
        }
        bytesPerChar = bytesPerChar || defaults.bytesPerChar
```

```
    if (
      typeof bytesPerChar !== "number" ||
      bytesPerChar % 1 !== 0 ||
      bytesPerChar < 1 ||
      bytesPerChar > defaults.maxBytesPerChar
    ) {
      throw new Error(
        "Bytes per character must be an integer between 1 and " +
          defaults.maxBytesPerChar +
          ", inclusive."
      )
    }

    hexChars = 2 * bytesPerChar

    str = padLeft(str, hexChars)

    for (i = 0, len = str.length; i < len; i += hexChars) {
      out =
        String.fromCharCode(
          parseInt(str.slice(i, i + hexChars), 16)
        ) + out
    }

    return out
  },

  // Generates a random bits-length number string using the PRNG
  random: function(bits) {
    if (
      typeof bits !== "number" ||
      bits % 1 !== 0 ||
```

```
            bits < 2 ||
            bits > 65536
        ) {
            throw new Error(
                "Number of bits must be an Integer between 1 and 65536."
            )
        }

        return bin2hex(config.rng(bits))
    },

    // Divides a `secret` number String str expressed in radix `inputRadix` (optional, default 16)
    // into `numShares` shares, each expressed in radix `outputRadix` (optional, default to `inputRadix`),
    // requiring `threshold` number of shares to reconstruct the secret.
    // Optionally, zero-pads the secret to a length that is a multiple of padLength before sharing.
    share: function(secret, numShares, threshold, padLength) {
        var neededBits,
            subShares,
            x = new Array(numShares),
            y = new Array(numShares),
            i,
            j,
            len

        // Security:
        // For additional security, pad in multiples of 128 bits by default.
        // A small trade-off in larger share size to help prevent leakage of information
        // about small-ish secrets and increase the difficulty of attacking them.
        padLength = padLength || 128
```

```
    if (typeof secret !== "string") {
        throw new Error("Secret must be a string.")
    }

    if (
        typeof numShares !== "number" ||
        numShares % 1 !== 0 ||
        numShares < 2
    ) {
        throw new Error(
            "Number of shares must be an integer between 2 and 2^bits-1 (" +
                config.maxShares +
                "), inclusive."
        )
    }

    if (numShares > config.maxShares) {
        neededBits = Math.ceil(Math.log(numShares + 1) / Math.LN2)
        throw new Error(
            "Number of shares must be an integer between 2 and 2^bits-1 (" +
                config.maxShares +
                "), inclusive. To create " +
                numShares +
                " shares, use at least " +
                neededBits +
                " bits."
        )
    }

    if (
        typeof threshold !== "number" ||
        threshold % 1 !== 0 ||
        threshold < 2
```

```
    ) {
      throw new Error(
        "Threshold number of shares must be an integer between 2 and 2^bits-1 (" +
          config.maxShares +
          "), inclusive."
      )
    }

    if (threshold > config.maxShares) {
      neededBits = Math.ceil(Math.log(threshold + 1) / Math.LN2)
      throw new Error(
        "Threshold number of shares must be an integer between 2 and 2^bits-1 (" +
          config.maxShares +
          "), inclusive.  To use a threshold of " +
          threshold +
          ", use at least " +
          neededBits +
          " bits."
      )
    }

    if (threshold > numShares) {
      throw new Error(
        "Threshold number of shares was " +
          threshold +
          " but must be less than or equal to the " +
          numShares +
          " shares specified as the total to generate."
      )
    }

    if (
      typeof padLength !== "number" ||
```

```
            padLength % 1 !== 0 ||
            padLength < 0 ||
            padLength > 1024
        ) {
            throw new Error(
                "Zero-pad length must be an integer between 0 and 1024 inclusive."
            )
        }

        secret = "1" + hex2bin(secret) // prepend a 1 as a marker so that we can preserve the correct number of leading zeros in our secret
        secret = splitNumStringToIntArray(secret, padLength)

        for (i = 0, len = secret.length; i < len; i++) {
            subShares = getShares(secret[i], numShares, threshold)
            for (j = 0; j < numShares; j++) {
                x[j] = x[j] || subShares[j].x.toString(config.radix)
                y[j] = padLeft(subShares[j].y.toString(2)) + (y[j] || "")
            }
        }

        for (i = 0; i < numShares; i++) {
            x[i] = constructPublicShareString(
                config.bits,
                x[i],
                bin2hex(y[i])
            )
        }

        return x
    },

    // Generate a new share with id `id` (a number between 1 and 2^bits-1)
```

```
    // `id` can be a Number or a String in the default radix (16)
    newShare: function(id, shares) {
        var share, radid

        if (id && typeof id === "string") {
            id = parseInt(id, config.radix)
        }

        radid = id.toString(config.radix)

        if (id && radid && shares && shares[0]) {
            share = this.extractShareComponents(shares[0])
            return constructPublicShareString(
                share.bits,
                radid,
                this.combine(shares, id)
            )
        }

        throw new Error(
            "Invalid 'id' or 'shares' Array argument to newShare()."
        )
    },

    /* test-code */
    // export private functions so they can be unit tested directly.
    _reset: reset,
    _padLeft: padLeft,
    _hex2bin: hex2bin,
    _bin2hex: bin2hex,
    _hasCryptoGetRandomValues: hasCryptoGetRandomValues,
    _hasCryptoRandomBytes: hasCryptoRandomBytes,
    _getRNG: getRNG,
```

```
        _isSetRNG: isSetRNG,
        _splitNumStringToIntArray: splitNumStringToIntArray,
        _horner: horner,
        _lagrange: lagrange,
        _getShares: getShares,
        _constructPublicShareString: constructPublicShareString
        /* end-test-code */
    }

    // Always initialize secrets with default settings.
    secrets.init()

    return secrets
})
```

Secrets.min.js

```
/*! secrets.js-grempe 2019-09-07 */

!function(t,e){"use strict";"function"==typeof
define&&define.amd?define([],function(){return t.secrets=e()}):"object"==typeof
exports?module.exports=e(require("crypto")):t.secrets=e(t.crypto)}(this,function(n){"use
strict";var u,b,i,a,s;function
h(){u={bits:8,radix:16,minBits:3,maxBits:20,bytesPerChar:2,maxBytesPerChar:6,primitiv
ePolynomials:[null,null,1,3,3,5,3,3,29,17,9,5,83,27,43,3,45,9,39,39,9,5,3,33,27,9,71,39,9,5,
83]},b={},i=new
Array(1024).join("0"),a=!0,s=["nodeCryptoRandomBytes","browserCryptoGetRandomVal
ues","testRandom"]}function f(){return!(!b||!b.rng||"function"!=typeof b.rng)}function
g(t,e){var r;if(0===e||1===e)return t;if(e&&1024<e)throw new Error("Padding must be
multiples of no larger than 1024 bits.");return e=e||b.bits,t&&(r=t.length%e),r?(i+t).slice(-
(e-r+t.length)):t}function c(t){var e,r,n="";for(r=t.length-1;0<=r;r--
){if(e=parseInt(t[r],16),isNaN(e))throw new Error("Invalid hex
character.");n=g(e.toString(2),4)+n}return n}function m(t){var
e,r,n="";for(r=(t=g(t,4)).length;4<=r;r-=4){if(e=parseInt(t.slice(r-4,r),2),isNaN(e))throw
new Error("Invalid binary character.");n=e.toString(16)+n}return n}function
o(){return!(!n||"object"!=typeof n||"function"!=typeof
n.getRandomValues&&"object"!=typeof n.getRandomValues||"function"!=typeof
Uint32Array&&"object"!=typeof Uint32Array)}function l(){return"object"==typeof
n&&"function"==typeof n.randomBytes}function p(t){function a(t,e,r,n){var
i,a=0,o="";for(e&&(i=e.length-
1);a<i||o.length<t;)o+=g(Math.abs(parseInt(e[a],r)).toString(2),n),a++;return((o=o.substr(-
t)).match(/0/g)||[]).length===o.length?null:o}function e(t){var
e,r=null;for(16,4,e=Math.ceil(t/8);null===r;)r=a(t,n.randomBytes(e).toString("hex"),16,4);r
eturn r}function r(t){var
e,r=null;for(10,32,e=Math.ceil(t/32);null===r;)r=a(t,n.getRandomValues(new
Uint32Array(e)),10,32);return r}return
t&&"testRandom"===t?(b.typeCSPRNG=t,function(t){var
e,r,n=null;r=Math.ceil(t/32),e=new Uint32Array(r);for(var
i=0;i<e.length;i++)e[i]=123456789;for(;null===n;)n=a(t,e,10,32);return
```

```
n}):t&&"nodeCryptoRandomBytes"===t?(b.typeCSPRNG=t,e):t&&"browserCryptoGetR
andomValues"===t?(b.typeCSPRNG=t,r):l()?(b.typeCSPRNG="nodeCryptoRandomBytes
",e):o()?(b.typeCSPRNG="browserCryptoGetRandomValues",r):void 0}function
d(t,e){var r,n=[];for(e&&(t=g(t,e)),r=t.length;r>b.bits;r-=b.bits)n.push(parseInt(t.slice(r-
b.bits,r),2));return n.push(parseInt(t.slice(0,r),2)),n}function w(t,e){var
r,n=b.logs[t],i=0;for(r=e.length-1;0<=r;r--
)i=0!==i?b.exps[(n+b.logs[i])%b.maxShares]^e[r]:e[r];return i}function y(t,e,r){var
n,i,a,o,s=0;for(a=0,n=e.length;a<n;a++)if(r[a]){for(i=b.logs[r[a]],o=0;o<n;o++)if(a!==o){i
f(t===e[o]){i=-1;break}i=(i+b.logs[t^e[o]]-
b.logs[e[a]^e[o]]+b.maxShares)%b.maxShares}s=-1===i?s:s^b.exps[i]}return s}function
x(t,e,r){var
n,i,a=[],o=[t];for(n=1;n<r;n++)o[n]=parseInt(b.rng(b.bits),2);for(i=e+(n=1);n<i;n++)a[n-
1]={x:n,y:w(n,o)};return a}function v(t,e,r){var
n,i,a,o;if(e=parseInt(e,b.radix),n=(t=parseInt(t,10)||b.bits).toString(36).toUpperCase(),o=(a
=Math.pow(2,t)-1).toString(b.radix).length,i=g(e.toString(b.radix),o),"number"!=typeof
e||e%1!=0||e<1||a<e)throw new Error("Share id must be an integer between 1 and "+a+",
inclusive.");return n+i+r}var t={init:function(t,e){var
r,n,i=[],a=[],o=1;if(h(),t&&("number"!=typeof t||t%1!=0||t<u.minBits||t>u.maxBits))throw
new Error("Number of bits must be an integer between "+u.minBits+" and "+u.maxBits+",
inclusive.");if(e&&-1===s.indexOf(e))throw new Error("Invalid RNG type argument :
'"+e+"'");for(b.radix=u.radix,b.bits=t||u.bits,b.size=Math.pow(2,b.bits),b.maxShares=b.size
-
1,r=u.primitivePolynomials[b.bits],n=0;n<b.size;n++)i[a[n]=o]=n,(o<<=1)>=b.size&&(o^
=r,o&=b.maxShares);if(b.logs=i,b.exps=a,e&&this.setRNG(e),f()||this.setRNG(),!(f()&&b.
bits&&b.size&&b.maxShares&&b.logs&&b.exps&&b.logs.length===b.size&&b.exps.len
gth===b.size))throw new Error("Initialization failed.")},combine:function(t,e){var
r,n,i,a,o,s,h,u="",f=[],l=[];for(e=e||0,r=0,i=t.length;r<i;r++){if(s=this.extractShareCompone
nts(t[r]),void 0===o)o=s.bits;else if(s.bits!==o)throw new Error("Mismatched shares:
Different bit settings.");if(b.bits!==o&&this.init(o),-
1===f.indexOf(s.id))for(f.push(s.id),n=0,a=(h=d(c(s.data))).length;n<a;n++)l[n]=l[n]||[],l[n
][f.length-1]=h[n]}for(r=0,i=l.length;r<i;r++)u=g(y(e,f,l[r]).toString(2))+u;return
m(1<=e?u:u.slice(u.indexOf("1")+1))},getConfig:function(){var t={};return
t.radix=b.radix,t.bits=b.bits,t.maxShares=b.maxShares,t.hasCSPRNG=f(),t.typeCSPRNG=
```

```
b.typeCSPRNG,t},extractShareComponents:function(t){var
e,r,n,i,a,o={};if((e=parseInt(t.substr(0,1),36))&&("number"!=typeof
e||e%1!=0||e<u.minBits||e>u.maxBits))throw new Error("Invalid share : Number of bits
must be an integer between "+u.minBits+" and "+u.maxBits+",
inclusive.");if(i=Math.pow(2,e)-1,n=(Math.pow(2,e)-1).toString(b.radix).length,(a=new
RegExp("^([a-kA-K3-9]{1})([a-fA-F0-9]{"+n+"})([a-fA-F0-
9]+)$").exec(t))&&(r=parseInt(a[2],b.radix)),"number"!=typeof r||r%1!=0||r<1||i<r)throw
new Error("Invalid share : Share id must be an integer between 1 and "+b.maxShares+",
inclusive.");if(a&&a[3])return o.bits=e,o.id=r,o.data=a[3],o;throw new Error("The share
data provided is invalid : "+t)},setRNG:function(t){var e="Random number generator is
invalid ",r=" Supply an CSPRNG of the form function(bits){} that returns a string
containing 'bits' number of random 1's and 0's.";if(t&&"string"==typeof t&&-
1===s.indexOf(t))throw new Error("Invalid RNG type argument :
'"+t+"'");if((t=t||p())&&"string"==typeof t&&(t=p(t)),a){if(t&&"function"!=typeof t)throw
new Error(e+"(Not a function)."+r);if(t&&"string"!=typeof t(b.bits))throw new
Error(e+"(Output is not a string)."+r);if(t&&!parseInt(t(b.bits),2))throw new
Error(e+"(Binary string output not parseable to an
Integer)."+r);if(t&&t(b.bits).length>b.bits)throw new Error(e+"(Output length is greater
than config.bits)."+r);if(t&&t(b.bits).length<b.bits)throw new Error(e+"(Output length is
less than config.bits)."+r)}return b.rng=t,!0},str2hex:function(t,e){var
r,n,i,a,o,s,h="";if("string"!=typeof t)throw new Error("Input must be a character
string.");if("number"!=typeof(e=e||u.bytesPerChar)||e<1||e>u.maxBytesPerChar||e%1!=0)th
row new Error("Bytes per character must be an integer between 1 and
"+u.maxBytesPerChar+", inclusive.");for(r=2*e,n=Math.pow(16,r)-
1,o=0,s=t.length;o<s;o++){if(a=t[o].charCodeAt(),isNaN(a))throw new Error("Invalid
character: "+t[o]);if(n<a)throw i=Math.ceil(Math.log(a+1)/Math.log(256)),new
Error("Invalid character code ("+a+"). Maximum allowable is 256^bytes-1 ("+n+"). To
convert this character, use at least "+i+" bytes.");h=g(a.toString(16),r)+h}return
h},hex2str:function(t,e){var r,n,i,a="";if("string"!=typeof t)throw new Error("Input must be
a hexadecimal
string.");if("number"!=typeof(e=e||u.bytesPerChar)||e%1!=0||e<1||e>u.maxBytesPerChar)th
row new Error("Bytes per character must be an integer between 1 and
"+u.maxBytesPerChar+",
```

```
inclusive.");for(n=0,i=(t=g(t,r=2*e)).length;n<i;n+=r)a=String.fromCharCode(parseInt(t.slice(n,n+r),16))+a;return a},random:function(t){if("number"!=typeof t||t%1!=0||t<2||65536<t)throw new Error("Number of bits must be an Integer between 1 and 65536.");return m(b.rng(t))},share:function(t,e,r,n){var i,a,o,s,h,u=new Array(e),f=new Array(e);if(n=n||128,"string"!=typeof t)throw new Error("Secret must be a string.");if("number"!=typeof e||e%1!=0||e<2)throw new Error("Number of shares must be an integer between 2 and 2^bits-1 ("+b.maxShares+"), inclusive.");if(e>b.maxShares)throw i=Math.ceil(Math.log(e+1)/Math.LN2),new Error("Number of shares must be an integer between 2 and 2^bits-1 ("+b.maxShares+"), inclusive. To create "+e+" shares, use at least "+i+" bits.");if("number"!=typeof r||r%1!=0||r<2)throw new Error("Threshold number of shares must be an integer between 2 and 2^bits-1 ("+b.maxShares+"), inclusive.");if(r>b.maxShares)throw i=Math.ceil(Math.log(r+1)/Math.LN2),new Error("Threshold number of shares must be an integer between 2 and 2^bits-1 ("+b.maxShares+"), inclusive.  To use a threshold of "+r+", use at least "+i+" bits.");if(e<r)throw new Error("Threshold number of shares was "+r+" but must be less than or equal to the "+e+" shares specified as the total to generate.");if("number"!=typeof n||n%1!=0||n<0||1024<n)throw new Error("Zero-pad length must be an integer between 0 and 1024
inclusive.");for(o=0,h=(t=d(t="1"+c(t),n)).length;o<h;o++)for(a=x(t[o],e,r),s=0;s<e;s++)u[s]=u[s]||a[s].x.toString(b.radix),f[s]=g(a[s].y.toString(2))+(f[s]||"");for(o=0;o<e;o++)u[o]=v(b.bits,u[o],m(f[o]));return u},newShare:function(t,e){var r;if(t&&"string"==typeof t&&(t=parseInt(t,b.radix)),r=t.toString(b.radix),t&&r&&e&&e[0])return v(this.extractShareComponents(e[0]).bits,r,this.combine(e,t));throw new Error("Invalid 'id' or 'shares' Array argument to
newShare().")},_reset:h,_padLeft:g,_hex2bin:c,_bin2hex:m,_hasCryptoGetRandomValues:o,_hasCryptoRandomBytes:l,_getRNG:p,_isSetRNG:f,_splitNumStringToIntArray:d,_horner:w,_lagrange:y,_getShares:x,_constructPublicShareString:v};return t.init(),t});
```

SHA512.js

```
var hexcase = 0;  /* hex output format. 0 - lowercase; 1 - uppercase        */
var b64pad  = ""; /* base-64 pad character. "=" for strict RFC compliance   */

/*
 * These take string arguments and return either hex or base-64 encoded strings
 */
function hex_sha512(s)    { return rstr2hex(rstr_sha512(str2rstr_utf8(s))); }
function b64_sha512(s)    { return rstr2b64(rstr_sha512(str2rstr_utf8(s))); }
function any_sha512(s, e) { return rstr2any(rstr_sha512(str2rstr_utf8(s)), e);}
function hex_hmac_sha512(k, d)
  { return rstr2hex(rstr_hmac_sha512(str2rstr_utf8(k), str2rstr_utf8(d))); }
function b64_hmac_sha512(k, d)
  { return rstr2b64(rstr_hmac_sha512(str2rstr_utf8(k), str2rstr_utf8(d))); }
function any_hmac_sha512(k, d, e)
  { return rstr2any(rstr_hmac_sha512(str2rstr_utf8(k), str2rstr_utf8(d)), e);}

/*
 * Perform a simple self-test to see if the VM is working
 */
function sha512_vm_test()
{
  return hex_sha512("abc").toLowerCase() ==
    "ddaf35a193617abacc417349ae20413112e6fa4e89a97ea20a9eeee64b55d39a" +
    "2192992a274fc1a836ba3c23a3feebbd454d4423643ce80e2a9ac94fa54ca49f";
}

/*
 * Calculate the SHA-512 of a raw string
 */
function rstr_sha512(s)
{
```

```
  return binb2rstr(binb_sha512(rstr2binb(s), s.length * 8));
}

/*
 * Calculate the HMAC-SHA-512 of a key and some data (raw strings)
 */
function rstr_hmac_sha512(key, data)
{
  var bkey = rstr2binb(key);
  if(bkey.length > 32) bkey = binb_sha512(bkey, key.length * 8);

  var ipad = Array(32), opad = Array(32);
  for(var i = 0; i < 32; i++)
  {
    ipad[i] = bkey[i] ^ 0x36363636;
    opad[i] = bkey[i] ^ 0x5C5C5C5C;
  }

  var hash = binb_sha512(ipad.concat(rstr2binb(data)), 1024 + data.length * 8);
  return binb2rstr(binb_sha512(opad.concat(hash), 1024 + 512));
}

/*
 * Convert a raw string to a hex string
 */
function rstr2hex(input)
{
  try { hexcase } catch(e) { hexcase=0; }
  var hex_tab = hexcase ? "0123456789ABCDEF" : "0123456789abcdef";
  var output = "";
  var x;
  for(var i = 0; i < input.length; i++)
  {
```

```
    x = input.charCodeAt(i);
    output += hex_tab.charAt((x >>> 4) & 0x0F)
           +  hex_tab.charAt( x        & 0x0F);
  }
  return output;
}

/*
 * Convert a raw string to a base-64 string
 */
function rstr2b64(input)
{
  try { b64pad } catch(e) { b64pad=''; }
  var tab =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
  var output = "";
  var len = input.length;
  for(var i = 0; i < len; i += 3)
  {
    var triplet = (input.charCodeAt(i) << 16)
                | (i + 1 < len ? input.charCodeAt(i+1) << 8 : 0)
                | (i + 2 < len ? input.charCodeAt(i+2)      : 0);
    for(var j = 0; j < 4; j++)
    {
      if(i * 8 + j * 6 > input.length * 8) output += b64pad;
      else output += tab.charAt((triplet >>> 6*(3-j)) & 0x3F);
    }
  }
  return output;
}

/*
 * Convert a raw string to an arbitrary string encoding
```

```
 */
function rstr2any(input, encoding)
{
  var divisor = encoding.length;
  var i, j, q, x, quotient;

  /* Convert to an array of 16-bit big-endian values, forming the dividend */
  var dividend = Array(Math.ceil(input.length / 2));
  for(i = 0; i < dividend.length; i++)
  {
    dividend[i] = (input.charCodeAt(i * 2) << 8) | input.charCodeAt(i * 2 + 1);
  }

  /*
   * Repeatedly perform a long division. The binary array forms the dividend,
   * the length of the encoding is the divisor. Once computed, the quotient
   * forms the dividend for the next step. All remainders are stored for later
   * use.
   */
  var full_length = Math.ceil(input.length * 8 /
                              (Math.log(encoding.length) / Math.log(2)));
  var remainders = Array(full_length);
  for(j = 0; j < full_length; j++)
  {
    quotient = Array();
    x = 0;
    for(i = 0; i < dividend.length; i++)
    {
      x = (x << 16) + dividend[i];
      q = Math.floor(x / divisor);
      x -= q * divisor;
      if(quotient.length > 0 || q > 0)
        quotient[quotient.length] = q;
```

```
    }
    remainders[j] = x;
    dividend = quotient;
  }

  /* Convert the remainders to the output string */
  var output = "";
  for(i = remainders.length - 1; i >= 0; i--)
    output += encoding.charAt(remainders[i]);

  return output;
}

/*
 * Encode a string as utf-8.
 * For efficiency, this assumes the input is valid utf-16.
 */
function str2rstr_utf8(input)
{
  var output = "";
  var i = -1;
  var x, y;

  while(++i < input.length)
  {
    /* Decode utf-16 surrogate pairs */
    x = input.charCodeAt(i);
    y = i + 1 < input.length ? input.charCodeAt(i + 1) : 0;
    if(0xD800 <= x && x <= 0xDBFF && 0xDC00 <= y && y <= 0xDFFF)
    {
      x = 0x10000 + ((x & 0x03FF) << 10) + (y & 0x03FF);
      i++;
    }
```

```
    /* Encode output as utf-8 */
    if(x <= 0x7F)
      output += String.fromCharCode(x);
    else if(x <= 0x7FF)
      output += String.fromCharCode(0xC0 | ((x >>> 6 ) & 0x1F),
                                    0x80 | ( x         & 0x3F));
    else if(x <= 0xFFFF)
      output += String.fromCharCode(0xE0 | ((x >>> 12) & 0x0F),
                                    0x80 | ((x >>> 6 ) & 0x3F),
                                    0x80 | ( x         & 0x3F));
    else if(x <= 0x1FFFFF)
      output += String.fromCharCode(0xF0 | ((x >>> 18) & 0x07),
                                    0x80 | ((x >>> 12) & 0x3F),
                                    0x80 | ((x >>> 6 ) & 0x3F),
                                    0x80 | ( x         & 0x3F));
  }
  return output;
}

/*
 * Encode a string as utf-16
 */
function str2rstr_utf16le(input)
{
  var output = "";
  for(var i = 0; i < input.length; i++)
    output += String.fromCharCode( input.charCodeAt(i)        & 0xFF,
                                  (input.charCodeAt(i) >>> 8) & 0xFF);
  return output;
}

function str2rstr_utf16be(input)
```

```
{
  var output = "";
  for(var i = 0; i < input.length; i++)
    output += String.fromCharCode((input.charCodeAt(i) >>> 8) & 0xFF,
                                  input.charCodeAt(i)        & 0xFF);
  return output;
}

/*
 * Convert a raw string to an array of big-endian words
 * Characters >255 have their high-byte silently ignored.
 */
function rstr2binb(input)
{
  var output = Array(input.length >> 2);
  for(var i = 0; i < output.length; i++)
    output[i] = 0;
  for(var i = 0; i < input.length * 8; i += 8)
    output[i>>5] |= (input.charCodeAt(i / 8) & 0xFF) << (24 - i % 32);
  return output;
}

/*
 * Convert an array of big-endian words to a string
 */
function binb2rstr(input)
{
  var output = "";
  for(var i = 0; i < input.length * 32; i += 8)
    output += String.fromCharCode((input[i>>5] >>> (24 - i % 32)) & 0xFF);
  return output;
}
```

```
/*
 * Calculate the SHA-512 of an array of big-endian dwords, and a bit length
 */
var sha512_k;
function binb_sha512(x, len)
{
  if(sha512_k == undefined)
  {
    //SHA512 constants
    sha512_k = new Array(
new int64(0x428a2f98, -685199838), new int64(0x71374491, 0x23ef65cd),
new int64(-1245643825, -330482897), new int64(-373957723, -2121671748),
new int64(0x3956c25b, -213338824), new int64(0x59f111f1, -1241133031),
new int64(-1841331548, -1357295717), new int64(-1424204075, -630357736),
new int64(-670586216, -1560083902), new int64(0x12835b01, 0x45706fbe),
new int64(0x243185be, 0x4ee4b28c), new int64(0x550c7dc3, -704662302),
new int64(0x72be5d74, -226784913), new int64(-2132889090, 0x3b1696b1),
new int64(-1680079193, 0x25c71235), new int64(-1046744716, -815192428),
new int64(-459576895, -1628353838), new int64(-272742522, 0x384f25e3),
new int64(0xfc19dc6, -1953704523), new int64(0x240ca1cc, 0x77ac9c65),
new int64(0x2de92c6f, 0x592b0275), new int64(0x4a7484aa, 0x6ea6e483),
new int64(0x5cb0a9dc, -1119749164), new int64(0x76f988da, -2096016459),
new int64(-1740746414, -295247957), new int64(-1473132947, 0x2db43210),
new int64(-1341970488, -1728372417), new int64(-1084653625, -1091629340),
new int64(-958395405, 0x3da88fc2), new int64(-710438585, -1828018395),
new int64(0x6ca6351, -536640913), new int64(0x14292967, 0xa0e6e70),
new int64(0x27b70a85, 0x46d22ffc), new int64(0x2e1b2138, 0x5c26c926),
new int64(0x4d2c6dfc, 0x5ac42aed), new int64(0x53380d13, -1651133473),
new int64(0x650a7354, -1951439906), new int64(0x766a0abb, 0x3c77b2a8),
new int64(-2117940946, 0x47edaee6), new int64(-1838011259, 0x1482353b),
new int64(-1564481375, 0x4cf10364), new int64(-1474664885, -1136513023),
new int64(-1035236496, -789014639), new int64(-949202525, 0x654be30),
new int64(-778901479, -688958952), new int64(-694614492, 0x5565a910),
```

```
new int64(-200395387, 0x5771202a), new int64(0x106aa070, 0x32bbd1b8),
new int64(0x19a4c116, -1194143544), new int64(0x1e376c08, 0x5141ab53),
new int64(0x2748774c, -544281703), new int64(0x34b0bcb5, -509917016),
new int64(0x391c0cb3, -976659869), new int64(0x4ed8aa4a, -482243893),
new int64(0x5b9cca4f, 0x7763e373), new int64(0x682e6ff3, -692930397),
new int64(0x748f82ee, 0x5defb2fc), new int64(0x78a5636f, 0x43172f60),
new int64(-2067236844, -1578062990), new int64(-1933114872, 0x1a6439ec),
new int64(-1866530822, 0x23631e28), new int64(-1538233109, -561857047),
new int64(-1090935817, -1295615723), new int64(-965641998, -479046869),
new int64(-903397682, -366583396), new int64(-779700025, 0x21c0c207),
new int64(-354779690, -840897762), new int64(-176337025, -294727304),
new int64(0x6f067aa, 0x72176fba), new int64(0xa637dc5, -1563912026),
new int64(0x113f9804, -1090974290), new int64(0x1b710b35, 0x131c471b),
new int64(0x28db77f5, 0x23047d84), new int64(0x32caab7b, 0x40c72493),
new int64(0x3c9ebe0a, 0x15c9bebc), new int64(0x431d67c4, -1676669620),
new int64(0x4cc5d4be, -885112138), new int64(0x597f299c, -60457430),
new int64(0x5fcb6fab, 0x3ad6faec), new int64(0x6c44198c, 0x4a475817));
  }

  //Initial hash values
  var H = new Array(
new int64(0x6a09e667, -205731576),
new int64(-1150833019, -2067093701),
new int64(0x3c6ef372, -23791573),
new int64(-1521486534, 0x5f1d36f1),
new int64(0x510e527f, -1377402159),
new int64(-1694144372, 0x2b3e6c1f),
new int64(0x1f83d9ab, -79577749),
new int64(0x5be0cd19, 0x137e2179));

  var T1 = new int64(0, 0),
    T2 = new int64(0, 0),
    a = new int64(0,0),
```

```
  b = new int64(0,0),
  c = new int64(0,0),
  d = new int64(0,0),
  e = new int64(0,0),
  f = new int64(0,0),
  g = new int64(0,0),
  h = new int64(0,0),
  //Temporary variables not specified by the document
  s0 = new int64(0, 0),
  s1 = new int64(0, 0),
  Ch = new int64(0, 0),
  Maj = new int64(0, 0),
  r1 = new int64(0, 0),
  r2 = new int64(0, 0),
  r3 = new int64(0, 0);
var j, i;
var W = new Array(80);
for(i=0; i<80; i++)
  W[i] = new int64(0, 0);

// append padding to the source string. The format is described in the FIPS.
x[len >> 5] |= 0x80 << (24 - (len & 0x1f));
x[((len + 128 >> 10)<< 5) + 31] = len;

for(i = 0; i<x.length; i+=32) //32 dwords is the block size
{
  int64copy(a, H[0]);
  int64copy(b, H[1]);
  int64copy(c, H[2]);
  int64copy(d, H[3]);
  int64copy(e, H[4]);
  int64copy(f, H[5]);
  int64copy(g, H[6]);
```

```
int64copy(h, H[7]);

for(j=0; j<16; j++)
{
   W[j].h = x[i + 2*j];
   W[j].l = x[i + 2*j + 1];
}

for(j=16; j<80; j++)
{
  //sigma1
  int64rrot(r1, W[j-2], 19);
  int64revrrot(r2, W[j-2], 29);
  int64shr(r3, W[j-2], 6);
  s1.l = r1.l ^ r2.l ^ r3.l;
  s1.h = r1.h ^ r2.h ^ r3.h;
  //sigma0
  int64rrot(r1, W[j-15], 1);
  int64rrot(r2, W[j-15], 8);
  int64shr(r3, W[j-15], 7);
  s0.l = r1.l ^ r2.l ^ r3.l;
  s0.h = r1.h ^ r2.h ^ r3.h;

  int64add4(W[j], s1, W[j-7], s0, W[j-16]);
}

for(j = 0; j < 80; j++)
{
  //Ch
  Ch.l = (e.l & f.l) ^ (~e.l & g.l);
  Ch.h = (e.h & f.h) ^ (~e.h & g.h);

  //Sigma1
```

```
  int64rrot(r1, e, 14);
  int64rrot(r2, e, 18);
  int64revrrot(r3, e, 9);
  s1.l = r1.l ^ r2.l ^ r3.l;
  s1.h = r1.h ^ r2.h ^ r3.h;

  //Sigma0
  int64rrot(r1, a, 28);
  int64revrrot(r2, a, 2);
  int64revrrot(r3, a, 7);
  s0.l = r1.l ^ r2.l ^ r3.l;
  s0.h = r1.h ^ r2.h ^ r3.h;

  //Maj
  Maj.l = (a.l & b.l) ^ (a.l & c.l) ^ (b.l & c.l);
  Maj.h = (a.h & b.h) ^ (a.h & c.h) ^ (b.h & c.h);

  int64add5(T1, h, s1, Ch, sha512_k[j], W[j]);
  int64add(T2, s0, Maj);

  int64copy(h, g);
  int64copy(g, f);
  int64copy(f, e);
  int64add(e, d, T1);
  int64copy(d, c);
  int64copy(c, b);
  int64copy(b, a);
  int64add(a, T1, T2);
}
int64add(H[0], H[0], a);
int64add(H[1], H[1], b);
int64add(H[2], H[2], c);
int64add(H[3], H[3], d);
```

```
    int64add(H[4], H[4], e);
    int64add(H[5], H[5], f);
    int64add(H[6], H[6], g);
    int64add(H[7], H[7], h);
  }

  //represent the hash as an array of 32-bit dwords
  var hash = new Array(16);
  for(i=0; i<8; i++)
  {
    hash[2*i] = H[i].h;
    hash[2*i + 1] = H[i].l;
  }
  return hash;
}

//A constructor for 64-bit numbers
function int64(h, l)
{
  this.h = h;
  this.l = l;
  //this.toString = int64toString;
}

//Copies src into dst, assuming both are 64-bit numbers
function int64copy(dst, src)
{
  dst.h = src.h;
  dst.l = src.l;
}

//Right-rotates a 64-bit number by shift
//Won't handle cases of shift>=32
```

```
//The function revrrot() is for that
function int64rrot(dst, x, shift)
{
    dst.l = (x.l >>> shift) | (x.h << (32-shift));
    dst.h = (x.h >>> shift) | (x.l << (32-shift));
}

//Reverses the dwords of the source and then rotates right by shift.
//This is equivalent to rotation by 32+shift
function int64revrrot(dst, x, shift)
{
    dst.l = (x.h >>> shift) | (x.l << (32-shift));
    dst.h = (x.l >>> shift) | (x.h << (32-shift));
}

//Bitwise-shifts right a 64-bit number by shift
//Won't handle shift>=32, but it's never needed in SHA512
function int64shr(dst, x, shift)
{
    dst.l = (x.l >>> shift) | (x.h << (32-shift));
    dst.h = (x.h >>> shift);
}

//Adds two 64-bit numbers
//Like the original implementation, does not rely on 32-bit operations
function int64add(dst, x, y)
{
   var w0 = (x.l & 0xffff) + (y.l & 0xffff);
   var w1 = (x.l >>> 16) + (y.l >>> 16) + (w0 >>> 16);
   var w2 = (x.h & 0xffff) + (y.h & 0xffff) + (w1 >>> 16);
   var w3 = (x.h >>> 16) + (y.h >>> 16) + (w2 >>> 16);
   dst.l = (w0 & 0xffff) | (w1 << 16);
   dst.h = (w2 & 0xffff) | (w3 << 16);
```

```
}

//Same, except with 4 addends. Works faster than adding them one by one.
function int64add4(dst, a, b, c, d)
{
  var w0 = (a.l & 0xffff) + (b.l & 0xffff) + (c.l & 0xffff) + (d.l & 0xffff);
  var w1 = (a.l >>> 16) + (b.l >>> 16) + (c.l >>> 16) + (d.l >>> 16) + (w0 >>> 16);
  var w2 = (a.h & 0xffff) + (b.h & 0xffff) + (c.h & 0xffff) + (d.h & 0xffff) + (w1 >>> 16);
  var w3 = (a.h >>> 16) + (b.h >>> 16) + (c.h >>> 16) + (d.h >>> 16) + (w2 >>> 16);
  dst.l = (w0 & 0xffff) | (w1 << 16);
  dst.h = (w2 & 0xffff) | (w3 << 16);
}

//Same, except with 5 addends
function int64add5(dst, a, b, c, d, e)
{
  var w0 = (a.l & 0xffff) + (b.l & 0xffff) + (c.l & 0xffff) + (d.l & 0xffff) + (e.l & 0xffff);
  var w1 = (a.l >>> 16) + (b.l >>> 16) + (c.l >>> 16) + (d.l >>> 16) + (e.l >>> 16) + (w0
>>> 16);
  var w2 = (a.h & 0xffff) + (b.h & 0xffff) + (c.h & 0xffff) + (d.h & 0xffff) + (e.h & 0xffff)
+ (w1 >>> 16);
  var w3 = (a.h >>> 16) + (b.h >>> 16) + (c.h >>> 16) + (d.h >>> 16) + (e.h >>> 16) +
(w2 >>> 16);
  dst.l = (w0 & 0xffff) | (w1 << 16);
  dst.h = (w2 & 0xffff) | (w3 << 16);
}
```

PBKDF2.js

```
function PBKDF2(password, salt, num_iterations, num_bytes)
{
	// Remember the password and salt
	var m_bpassword = rstr2binb(password);
	var m_salt = salt;

	// Total number of iterations
	var m_total_iterations = num_iterations;

	// Run iterations in chunks instead of all at once, so as to not block.
	// Define size of chunk here; adjust for slower or faster machines if necessary.
	var m_iterations_in_chunk = 10;

	// Iteration counter
	var m_iterations_done = 0;

	// Key length, as number of bytes
	var m_key_length = num_bytes;

	// The hash cache
	var m_hash = null;

	// The length (number of bytes) of the output of the pseudo-random function.
	// Since HMAC-SHA1 is the standard, and what is used here, it's 20 bytes.
	var m_hash_length = 20;

	// Number of hash-sized blocks in the derived key (called 'l' in RFC2898)
	var m_total_blocks = Math.ceil(m_key_length/m_hash_length);

	// Start computation with the first block
```

```
	var m_current_block = 1;

	// Used in the HMAC-SHA1 computations
	var m_ipad = new Array(16);
	var m_opad = new Array(16);

	// This is where the result of the iterations gets sotred
	var m_buffer = new Array(0x0,0x0,0x0,0x0,0x0);

	// The result
	var m_key = "";

	// This object
	var m_this_object = this;

	// The function to call with the result
	var m_result_func;

	// The function to call with status after computing every chunk
	var m_status_func;

	// Set up the HMAC-SHA1 computations
	if (m_bpassword.length > 16) m_bpassword = binb_sha512(m_bpassword,
password.length * chrsz);
	for(var i = 0; i < 16; ++i)
	{
		m_ipad[i] = m_bpassword[i] ^ 0x36363636;
		m_opad[i] = m_bpassword[i] ^ 0x5C5C5C5C;
	}



	// Starts the computation
	this.deriveKey = function(status_callback, result_callback)
```

```
	{
		m_status_func = status_callback;
		m_result_func = result_callback;
		setTimeout(function() { m_this_object.do_PBKDF2_iterations() }, 0);
	}



	// The workhorse
	this.do_PBKDF2_iterations = function()
	{
		var iterations = m_iterations_in_chunk;
		if (m_total_iterations - m_iterations_done < m_iterations_in_chunk)
			iterations = m_total_iterations - m_iterations_done;

		for(var i=0; i<iterations; ++i)
		{
			// compute HMAC-SHA1
			if (m_iterations_done == 0)
			{
				var salt_block = m_salt +
						String.fromCharCode(m_current_block >> 24
& 0xF) +
						String.fromCharCode(m_current_block >> 16
& 0xF) +
						String.fromCharCode(m_current_block >>  8
& 0xF) +
						String.fromCharCode(m_current_block      &
0xF);

				m_hash =
binb_sha512(m_ipad.concat(rstr2binb(salt_block)),
								512 + salt_block.length * 8);
				m_hash = binb_sha512(m_opad.concat(m_hash), 512 + 160);
```

```
                        }
                        else
                        {
                                m_hash = binb_sha512(m_ipad.concat(m_hash),
                                                                512 + m_hash.length * 32);
                                m_hash = binb_sha512(m_opad.concat(m_hash), 512 + 160);
                        }

        for(var j=0; j<m_hash.length; ++j)
                m_buffer[j] ^= m_hash[j];

                        m_iterations_done++;
                }

                // Call the status callback function
                m_status_func( (m_current_block - 1 +
m_iterations_done/m_total_iterations) / m_total_blocks * 100);

                if (m_iterations_done < m_total_iterations)
                {
                        setTimeout(function() { m_this_object.do_PBKDF2_iterations() },
0);
                }
                else
                {
                        if (m_current_block < m_total_blocks)
                        {
                                // Compute the next block (T_i in RFC 2898)

                                m_key += rstr2hex(binb2rstr(m_buffer));

                                m_current_block++;
                                m_buffer = new Array(0x0,0x0,0x0,0x0,0x0);
```

```
                                m_iterations_done = 0;

                                setTimeout(function() {
m_this_object.do_PBKDF2_iterations() }, 0);
                        }
                        else
                        {
                                // We've computed the final block T_l; we're done.

                                var tmp = rstr2hex(binb2rstr(m_buffer));
                                m_key += tmp.substr(0, (m_key_length - (m_total_blocks -
1) * m_hash_length) * 2 );

                                // Call the result callback function
                                m_result_func(m_key);
                        }
                }
        }
}
```

Aes.js

```
(function(root) {
    "use strict";

    function checkInt(value) {
        return (parseInt(value) === value);
    }

    function checkInts(arrayish) {
        if (!checkInt(arrayish.length)) { return false; }

        for (var i = 0; i < arrayish.length; i++) {
            if (!checkInt(arrayish[i]) || arrayish[i] < 0 || arrayish[i] > 255) {
                return false;
            }
        }

        return true;
    }

    function coerceArray(arg, copy) {

        // ArrayBuffer view
        if (arg.buffer && arg.name === 'Uint8Array') {

            if (copy) {
                if (arg.slice) {
                    arg = arg.slice();
                } else {
                    arg = Array.prototype.slice.call(arg);
                }
            }
```

```
        return arg;
    }

    // It's an array; check it is a valid representation of a byte
    if (Array.isArray(arg)) {
        if (!checkInts(arg)) {
            throw new Error('Array contains invalid value: ' + arg);
        }

        return new Uint8Array(arg);
    }

    // Something else, but behaves like an array (maybe a Buffer? Arguments?)
    if (checkInt(arg.length) && checkInts(arg)) {
        return new Uint8Array(arg);
    }

    throw new Error('unsupported array-like object');
}

function createArray(length) {
    return new Uint8Array(length);
}

function copyArray(sourceArray, targetArray, targetStart, sourceStart, sourceEnd) {
    if (sourceStart != null || sourceEnd != null) {
        if (sourceArray.slice) {
            sourceArray = sourceArray.slice(sourceStart, sourceEnd);
        } else {
            sourceArray = Array.prototype.slice.call(sourceArray, sourceStart, sourceEnd);
        }
    }
```

```
    targetArray.set(sourceArray, targetStart);
}



var convertUtf8 = (function() {
    function toBytes(text) {
        var result = [], i = 0;
        text = encodeURI(text);
        while (i < text.length) {
            var c = text.charCodeAt(i++);

            // if it is a % sign, encode the following 2 bytes as a hex value
            if (c === 37) {
                result.push(parseInt(text.substr(i, 2), 16))
                i += 2;

            // otherwise, just the actual byte
            } else {
                result.push(c)
            }
        }

        return coerceArray(result);
    }

    function fromBytes(bytes) {
        var result = [], i = 0;

        while (i < bytes.length) {
            var c = bytes[i];

            if (c < 128) {
```

```
                result.push(String.fromCharCode(c));
                i++;
            } else if (c > 191 && c < 224) {
                result.push(String.fromCharCode(((c & 0x1f) << 6) | (bytes[i + 1] & 0x3f)));
                i += 2;
            } else {
                result.push(String.fromCharCode(((c & 0x0f) << 12) | ((bytes[i + 1] & 0x3f)
<< 6) | (bytes[i + 2] & 0x3f)));
                i += 3;
            }
        }

        return result.join('');
    }

    return {
        toBytes: toBytes,
        fromBytes: fromBytes,
    }
})();

var convertHex = (function() {
    function toBytes(text) {
        var result = [];
        for (var i = 0; i < text.length; i += 2) {
            result.push(parseInt(text.substr(i, 2), 16));
        }

        return result;
    }

    // http://ixti.net/development/javascript/2011/11/11/base64-encodedecode-of-utf8-in-
browser-with-js.html
```

```
        var Hex = '0123456789abcdef';

        function fromBytes(bytes) {
            var result = [];
            for (var i = 0; i < bytes.length; i++) {
                var v = bytes[i];
                result.push(Hex[(v & 0xf0) >> 4] + Hex[v & 0x0f]);
            }
            return result.join('');
        }

        return {
            toBytes: toBytes,
            fromBytes: fromBytes,
        }
    })();


    // Number of rounds by keysize
    var numberOfRounds = {16: 10, 24: 12, 32: 14}

    // Round constant words
    var rcon = [0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x1b, 0x36, 0x6c, 0xd8,
0xab, 0x4d, 0x9a, 0x2f, 0x5e, 0xbc, 0x63, 0xc6, 0x97, 0x35, 0x6a, 0xd4, 0xb3, 0x7d, 0xfa,
0xef, 0xc5, 0x91];

    // S-box and Inverse S-box (S is for Substitution)
    var S = [0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67, 0x2b, 0xfe,
0xd7, 0xab, 0x76, 0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2, 0xaf,
0x9c, 0xa4, 0x72, 0xc0, 0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5,
0xf1, 0x71, 0xd8, 0x31, 0x15, 0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07,
0x12, 0x80, 0xe2, 0xeb, 0x27, 0xb2, 0x75, 0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a,
0xa0, 0x52, 0x3b, 0xd6, 0xb3, 0x29, 0xe3, 0x2f, 0x84, 0x53, 0xd1, 0x00, 0xed, 0x20,
```

```
0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe, 0x39, 0x4a, 0x4c, 0x58, 0xcf, 0xd0, 0xef, 0xaa, 0xfb,
0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02, 0x7f, 0x50, 0x3c, 0x9f, 0xa8, 0x51, 0xa3, 0x40,
0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda, 0x21, 0x10, 0xff, 0xf3, 0xd2, 0xcd, 0x0c,
0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e, 0x3d, 0x64, 0x5d, 0x19, 0x73,
0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8, 0x14, 0xde, 0x5e,
0x0b, 0xdb, 0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac, 0x62,
0x91, 0x95, 0xe4, 0x79, 0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56,
0xf4, 0xea, 0x65, 0x7a, 0xae, 0x08, 0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8,
0xdd, 0x74, 0x1f, 0x4b, 0xbd, 0x8b, 0x8a, 0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6,
0x0e, 0x61, 0x35, 0x57, 0xb9, 0x86, 0xc1, 0x1d, 0x9e, 0xe1, 0xf8, 0x98, 0x11, 0x69,
0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87, 0xe9, 0xce, 0x55, 0x28, 0xdf, 0x8c, 0xa1, 0x89,
0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d, 0x0f, 0xb0, 0x54, 0xbb, 0x16];
   var Si =[0x52, 0x09, 0x6a, 0xd5, 0x30, 0x36, 0xa5, 0x38, 0xbf, 0x40, 0xa3, 0x9e, 0x81,
0xf3, 0xd7, 0xfb, 0x7c, 0xe3, 0x39, 0x82, 0x9b, 0x2f, 0xff, 0x87, 0x34, 0x8e, 0x43, 0x44,
0xc4, 0xde, 0xe9, 0xcb, 0x54, 0x7b, 0x94, 0x32, 0xa6, 0xc2, 0x23, 0x3d, 0xee, 0x4c,
0x95, 0x0b, 0x42, 0xfa, 0xc3, 0x4e, 0x08, 0x2e, 0xa1, 0x66, 0x28, 0xd9, 0x24, 0xb2,
0x76, 0x5b, 0xa2, 0x49, 0x6d, 0x8b, 0xd1, 0x25, 0x72, 0xf8, 0xf6, 0x64, 0x86, 0x68,
0x98, 0x16, 0xd4, 0xa4, 0x5c, 0xcc, 0x5d, 0x65, 0xb6, 0x92, 0x6c, 0x70, 0x48, 0x50,
0xfd, 0xed, 0xb9, 0xda, 0x5e, 0x15, 0x46, 0x57, 0xa7, 0x8d, 0x9d, 0x84, 0x90, 0xd8,
0xab, 0x00, 0x8c, 0xbc, 0xd3, 0x0a, 0xf7, 0xe4, 0x58, 0x05, 0xb8, 0xb3, 0x45, 0x06,
0xd0, 0x2c, 0x1e, 0x8f, 0xca, 0x3f, 0x0f, 0x02, 0xc1, 0xaf, 0xbd, 0x03, 0x01, 0x13, 0x8a,
0x6b, 0x3a, 0x91, 0x11, 0x41, 0x4f, 0x67, 0xdc, 0xea, 0x97, 0xf2, 0xcf, 0xce, 0xf0, 0xb4,
0xe6, 0x73, 0x96, 0xac, 0x74, 0x22, 0xe7, 0xad, 0x35, 0x85, 0xe2, 0xf9, 0x37, 0xe8, 0x1c,
0x75, 0xdf, 0x6e, 0x47, 0xf1, 0x1a, 0x71, 0x1d, 0x29, 0xc5, 0x89, 0x6f, 0xb7, 0x62, 0x0e,
0xaa, 0x18, 0xbe, 0x1b, 0xfc, 0x56, 0x3e, 0x4b, 0xc6, 0xd2, 0x79, 0x20, 0x9a, 0xdb, 0xc0,
0xfe, 0x78, 0xcd, 0x5a, 0xf4, 0x1f, 0xdd, 0xa8, 0x33, 0x88, 0x07, 0xc7, 0x31, 0xb1, 0x12,
0x10, 0x59, 0x27, 0x80, 0xec, 0x5f, 0x60, 0x51, 0x7f, 0xa9, 0x19, 0xb5, 0x4a, 0x0d,
0x2d, 0xe5, 0x7a, 0x9f, 0x93, 0xc9, 0x9c, 0xef, 0xa0, 0xe0, 0x3b, 0x4d, 0xae, 0x2a, 0xf5,
0xb0, 0xc8, 0xeb, 0xbb, 0x3c, 0x83, 0x53, 0x99, 0x61, 0x17, 0x2b, 0x04, 0x7e, 0xba,
0x77, 0xd6, 0x26, 0xe1, 0x69, 0x14, 0x63, 0x55, 0x21, 0x0c, 0x7d];


   // Transformations for encryption
```

```
var T1 = [0xc66363a5, 0xf87c7c84, 0xee777799, 0xf67b7b8d, 0xfff2f20d, 0xd66b6bbd,
0xde6f6fb1, 0x91c5c554, 0x60303050, 0x02010103, 0xce6767a9, 0x562b2b7d,
0xe7fefe19, 0xb5d7d762, 0x4dababe6, 0xec76769a, 0x8fcaca45, 0x1f82829d,
0x89c9c940, 0xfa7d7d87, 0xeffafa15, 0xb25959eb, 0x8e4747c9, 0xfbf0f00b, 0x41adadec,
0xb3d4d467, 0x5fa2a2fd, 0x45afafea, 0x239c9cbf, 0x53a4a4f7, 0xe4727296, 0x9bc0c05b,
0x75b7b7c2, 0xe1fdfd1c, 0x3d9393ae, 0x4c26266a, 0x6c36365a, 0x7e3f3f41, 0xf5f7f702,
0x83cccc4f, 0x6834345c, 0x51a5a5f4, 0xd1e5e534, 0xf9f1f108, 0xe2717193,
0xabd8d873, 0x62313153, 0x2a15153f, 0x0804040c, 0x95c7c752, 0x46232365,
0x9dc3c35e, 0x30181828, 0x379696a1, 0x0a05050f, 0x2f9a9ab5, 0x0e070709,
0x24121236, 0x1b80809b, 0xdfe2e23d, 0xcdebeb26, 0x4e272769, 0x7fb2b2cd,
0xea75759f, 0x1209091b, 0x1d83839e, 0x582c2c74, 0x341a1a2e, 0x361b1b2d,
0xdc6e6eb2, 0xb45a5aee, 0x5ba0a0fb, 0xa45252f6, 0x763b3b4d, 0xb7d6d661,
0x7db3b3ce, 0x5229297b, 0xdde3e33e, 0x5e2f2f71, 0x13848497, 0xa65353f5,
0xb9d1d168, 0x00000000, 0xc1eded2c, 0x40202060, 0xe3fcfc1f, 0x79b1b1c8,
0xb65b5bed, 0xd46a6abe, 0x8dcbcb46, 0x67bebed9, 0x7239394b, 0x944a4ade,
0x984c4cd4, 0xb05858e8, 0x85cfcf4a, 0xbbd0d06b, 0xc5efef2a, 0x4faaaae5, 0xedfbfb16,
0x864343c5, 0x9a4d4dd7, 0x66333355, 0x11858594, 0x8a4545cf, 0xe9f9f910,
0x04020206, 0xfe7f7f81, 0xa05050f0, 0x783c3c44, 0x259f9fba, 0x4ba8a8e3, 0xa25151f3,
0x5da3a3fe, 0x804040c0, 0x058f8f8a, 0x3f9292ad, 0x219d9dbc, 0x70383848,
0xf1f5f504, 0x63bcbcdf, 0x77b6b6c1, 0xafdada75, 0x42212163, 0x20101030, 0xe5ffff1a,
0xfdf3f30e, 0xbfd2d26d, 0x81cdcd4c, 0x180c0c14, 0x26131335, 0xc3ecec2f, 0xbe5f5fe1,
0x359797a2, 0x884444cc, 0x2e171739, 0x93c4c457, 0x55a7a7f2, 0xfc7e7e82,
0x7a3d3d47, 0xc86464ac, 0xba5d5de7, 0x3219192b, 0xe6737395, 0xc06060a0,
0x19818198, 0x9e4f4fd1, 0xa3dcdc7f, 0x44222266, 0x542a2a7e, 0x3b9090ab,
0x0b888883, 0x8c4646ca, 0xc7eeee29, 0x6bb8b8d3, 0x2814143c, 0xa7dede79,
0xbc5e5ee2, 0x160b0b1d, 0xaddbdb76, 0xdbe0e03b, 0x64323256, 0x743a3a4e,
0x140a0a1e, 0x924949db, 0x0c06060a, 0x4824246c, 0xb85c5ce4, 0x9fc2c25d,
0xbdd3d36e, 0x43acacef, 0xc46262a6, 0x399191a8, 0x319595a4, 0xd3e4e437,
0xf279798b, 0xd5e7e732, 0x8bc8c843, 0x6e373759, 0xda6d6db7, 0x018d8d8c,
0xb1d5d564, 0x9c4e4ed2, 0x49a9a9e0, 0xd86c6cb4, 0xac5656fa, 0xf3f4f407, 0xcfeaea25,
0xca6565af, 0xf47a7a8e, 0x47aeaee9, 0x10080818, 0x6fbabad5, 0xf0787888, 0x4a25256f,
0x5c2e2e72, 0x381c1c24, 0x57a6a6f1, 0x73b4b4c7, 0x97c6c651, 0xcbe8e823,
0xa1dddd7c, 0xe874749c, 0x3e1f1f21, 0x964b4bdd, 0x61bdbddc, 0x0d8b8b86,
```

```
0x0f8a8a85, 0xe0707090, 0x7c3e3e42, 0x71b5b5c4, 0xcc6666aa, 0x904848d8,
0x06030305, 0xf7f6f601, 0x1c0e0e12, 0xc26161a3, 0x6a35355f, 0xae5757f9,
0x69b9b9d0, 0x17868691, 0x99c1c158, 0x3a1d1d27, 0x279e9eb9, 0xd9e1e138,
0xebf8f813, 0x2b9898b3, 0x22111133, 0xd26969bb, 0xa9d9d970, 0x078e8e89,
0x339494a7, 0x2d9b9bb6, 0x3c1e1e22, 0x15878792, 0xc9e9e920, 0x87cece49,
0xaa5555ff, 0x50282878, 0xa5dfdf7a, 0x038c8c8f, 0x59a1a1f8, 0x09898980,
0x1a0d0d17, 0x65bfbfda, 0xd7e6e631, 0x844242c6, 0xd06868b8, 0x824141c3,
0x299999b0, 0x5a2d2d77, 0x1e0f0f11, 0x7bb0b0cb, 0xa85454fc, 0x6dbbbbd6,
0x2c16163a];
    var T2 = [0xa5c66363, 0x84f87c7c, 0x99ee7777, 0x8df67b7b, 0x0dfff2f2, 0xbdd66b6b,
0xb1de6f6f, 0x5491c5c5, 0x50603030, 0x03020101, 0xa9ce6767, 0x7d562b2b,
0x19e7fefe, 0x62b5d7d7, 0xe64dabab, 0x9aec7676, 0x458fcaca, 0x9d1f8282,
0x4089c9c9, 0x87fa7d7d, 0x15effafa, 0xebb25959, 0xc98e4747, 0x0bfbf0f0, 0xec41adad,
0x67b3d4d4, 0xfd5fa2a2, 0xea45afaf, 0xbf239c9c, 0xf753a4a4, 0x96e47272, 0x5b9bc0c0,
0xc275b7b7, 0x1ce1fdfd, 0xae3d9393, 0x6a4c2626, 0x5a6c3636, 0x417e3f3f, 0x02f5f7f7,
0x4f83cccc, 0x5c683434, 0xf451a5a5, 0x34d1e5e5, 0x08f9f1f1, 0x93e27171,
0x73abd8d8, 0x53623131, 0x3f2a1515, 0x0c080404, 0x5295c7c7, 0x65462323,
0x5e9dc3c3, 0x28301818, 0xa1379696, 0x0f0a0505, 0xb52f9a9a, 0x090e0707,
0x36241212, 0x9b1b8080, 0x3ddfe2e2, 0x26cdebeb, 0x694e2727, 0xcd7fb2b2,
0x9fea7575, 0x1b120909, 0x9e1d8383, 0x74582c2c, 0x2e341a1a, 0x2d361b1b,
0xb2dc6e6e, 0xeeb45a5a, 0xfb5ba0a0, 0xf6a45252, 0x4d763b3b, 0x61b7d6d6,
0xce7db3b3, 0x7b522929, 0x3edde3e3, 0x715e2f2f, 0x97138484, 0xf5a65353,
0x68b9d1d1, 0x00000000, 0x2cc1eded, 0x60402020, 0x1fe3fcfc, 0xc879b1b1,
0xedb65b5b, 0xbed46a6a, 0x468dcbcb, 0xd967bebe, 0x4b723939, 0xde944a4a,
0xd4984c4c, 0xe8b05858, 0x4a85cfcf, 0x6bbbd0d0, 0x2ac5efef, 0xe54faaaa, 0x16edfbfb,
0xc5864343, 0xd79a4d4d, 0x55663333, 0x94118585, 0xcf8a4545, 0x10e9f9f9,
0x06040202, 0x81fe7f7f, 0xf0a05050, 0x44783c3c, 0xba259f9f, 0xe34ba8a8, 0xf3a25151,
0xfe5da3a3, 0xc0804040, 0x8a058f8f, 0xad3f9292, 0xbc219d9d, 0x48703838,
0x04f1f5f5, 0xdf63bcbc, 0xc177b6b6, 0x75afdada, 0x63422121, 0x30201010, 0x1ae5ffff,
0x0efdf3f3, 0x6dbfd2d2, 0x4c81cdcd, 0x14180c0c, 0x35261313, 0x2fc3ecec, 0xe1be5f5f,
0xa2359797, 0xcc884444, 0x392e1717, 0x5793c4c4, 0xf255a7a7, 0x82fc7e7e,
0x477a3d3d, 0xacc86464, 0xe7ba5d5d, 0x2b321919, 0x95e67373, 0xa0c06060,
0x98198181, 0xd19e4f4f, 0x7fa3dcdc, 0x66442222, 0x7e542a2a, 0xab3b9090,
```

```
0x830b8888, 0xca8c4646, 0x29c7eeee, 0xd36bb8b8, 0x3c281414, 0x79a7dede,
0xe2bc5e5e, 0x1d160b0b, 0x76addbdb, 0x3bdbe0e0, 0x56643232, 0x4e743a3a,
0x1e140a0a, 0xdb924949, 0x0a0c0606, 0x6c482424, 0xe4b85c5c, 0x5d9fc2c2,
0x6ebdd3d3, 0xef43acac, 0xa6c46262, 0xa8399191, 0xa4319595, 0x37d3e4e4,
0x8bf27979, 0x32d5e7e7, 0x438bc8c8, 0x596e3737, 0xb7da6d6d, 0x8c018d8d,
0x64b1d5d5, 0xd29c4e4e, 0xe049a9a9, 0xb4d86c6c, 0xfaac5656, 0x07f3f4f4, 0x25cfeaea,
0xafca6565, 0x8ef47a7a, 0xe947aeae, 0x18100808, 0xd56fbaba, 0x88f07878, 0x6f4a2525,
0x725c2e2e, 0x24381c1c, 0xf157a6a6, 0xc773b4b4, 0x5197c6c6, 0x23cbe8e8,
0x7ca1dddd, 0x9ce87474, 0x213e1f1f, 0xdd964b4b, 0xdc61bdbd, 0x860d8b8b,
0x850f8a8a, 0x90e07070, 0x427c3e3e, 0xc471b5b5, 0xaacc6666, 0xd8904848,
0x05060303, 0x01f7f6f6, 0x121c0e0e, 0xa3c26161, 0x5f6a3535, 0xf9ae5757,
0xd069b9b9, 0x91178686, 0x5899c1c1, 0x273a1d1d, 0xb9279e9e, 0x38d9e1e1,
0x13ebf8f8, 0xb32b9898, 0x33221111, 0xbbd26969, 0x70a9d9d9, 0x89078e8e,
0xa7339494, 0xb62d9b9b, 0x223c1e1e, 0x92158787, 0x20c9e9e9, 0x4987cece,
0xffaa5555, 0x78502828, 0x7aa5dfdf, 0x8f038c8c, 0xf859a1a1, 0x80098989,
0x171a0d0d, 0xda65bfbf, 0x31d7e6e6, 0xc6844242, 0xb8d06868, 0xc3824141,
0xb0299999, 0x775a2d2d, 0x111e0f0f, 0xcb7bb0b0, 0xfca85454, 0xd66dbbbb,
0x3a2c1616];
    var T3 = [0x63a5c663, 0x7c84f87c, 0x7799ee77, 0x7b8df67b, 0xf20dfff2, 0x6bbdd66b,
0x6fb1de6f, 0xc55491c5, 0x30506030, 0x01030201, 0x67a9ce67, 0x2b7d562b,
0xfe19e7fe, 0xd762b5d7, 0xabe64dab, 0x769aec76, 0xca458fca, 0x829d1f82,
0xc94089c9, 0x7d87fa7d, 0xfa15effa, 0x59ebb259, 0x47c98e47, 0xf00bfbf0, 0xadec41ad,
0xd467b3d4, 0xa2fd5fa2, 0xafea45af, 0x9cbf239c, 0xa4f753a4, 0x7296e472, 0xc05b9bc0,
0xb7c275b7, 0xfd1ce1fd, 0x93ae3d93, 0x266a4c26, 0x365a6c36, 0x3f417e3f, 0xf702f5f7,
0xcc4f83cc, 0x345c6834, 0xa5f451a5, 0xe534d1e5, 0xf108f9f1, 0x7193e271,
0xd873abd8, 0x31536231, 0x153f2a15, 0x040c0804, 0xc75295c7, 0x23654623,
0xc35e9dc3, 0x18283018, 0x96a13796, 0x050f0a05, 0x9ab52f9a, 0x07090e07,
0x12362412, 0x809b1b80, 0xe23ddfe2, 0xeb26cdeb, 0x27694e27, 0xb2cd7fb2,
0x759fea75, 0x091b1209, 0x839e1d83, 0x2c74582c, 0x1a2e341a, 0x1b2d361b,
0x6eb2dc6e, 0x5aeeb45a, 0xa0fb5ba0, 0x52f6a452, 0x3b4d763b, 0xd661b7d6,
0xb3ce7db3, 0x297b5229, 0xe33edde3, 0x2f715e2f, 0x84971384, 0x53f5a653,
0xd168b9d1, 0x00000000, 0xed2cc1ed, 0x20604020, 0xfc1fe3fc, 0xb1c879b1,
0x5bedb65b, 0x6abed46a, 0xcb468dcb, 0xbed967be, 0x394b7239, 0x4ade944a,
```

```
0x4cd4984c, 0x58e8b058, 0xcf4a85cf, 0xd06bbbd0, 0xef2ac5ef, 0xaae54faa, 0xfb16edfb,
0x43c58643, 0x4dd79a4d, 0x33556633, 0x85941185, 0x45cf8a45, 0xf910e9f9,
0x02060402, 0x7f81fe7f, 0x50f0a050, 0x3c44783c, 0x9fba259f, 0xa8e34ba8, 0x51f3a251,
0xa3fe5da3, 0x40c08040, 0x8f8a058f, 0x92ad3f92, 0x9dbc219d, 0x38487038,
0xf504f1f5, 0xbcdf63bc, 0xb6c177b6, 0xda75afda, 0x21634221, 0x10302010, 0xff1ae5ff,
0xf30efdf3, 0xd26dbfd2, 0xcd4c81cd, 0x0c14180c, 0x13352613, 0xec2fc3ec, 0x5fe1be5f,
0x97a23597, 0x44cc8844, 0x17392e17, 0xc45793c4, 0xa7f255a7, 0x7e82fc7e,
0x3d477a3d, 0x64acc864, 0x5de7ba5d, 0x192b3219, 0x7395e673, 0x60a0c060,
0x81981981, 0x4fd19e4f, 0xdc7fa3dc, 0x22664422, 0x2a7e542a, 0x90ab3b90,
0x88830b88, 0x46ca8c46, 0xee29c7ee, 0xb8d36bb8, 0x143c2814, 0xde79a7de,
0x5ee2bc5e, 0x0b1d160b, 0xdb76addb, 0xe03bdbe0, 0x32566432, 0x3a4e743a,
0x0a1e140a, 0x49db9249, 0x060a0c06, 0x246c4824, 0x5ce4b85c, 0xc25d9fc2,
0xd36ebdd3, 0xacef43ac, 0x62a6c462, 0x91a83991, 0x95a43195, 0xe437d3e4,
0x798bf279, 0xe732d5e7, 0xc8438bc8, 0x37596e37, 0x6db7da6d, 0x8d8c018d,
0xd564b1d5, 0x4ed29c4e, 0xa9e049a9, 0x6cb4d86c, 0x56faac56, 0xf407f3f4, 0xea25cfea,
0x65afca65, 0x7a8ef47a, 0xaee947ae, 0x08181008, 0xbad56fba, 0x7888f078, 0x256f4a25,
0x2e725c2e, 0x1c24381c, 0xa6f157a6, 0xb4c773b4, 0xc65197c6, 0xe823cbe8,
0xdd7ca1dd, 0x749ce874, 0x1f213e1f, 0x4bdd964b, 0xbddc61bd, 0x8b860d8b,
0x8a850f8a, 0x7090e070, 0x3e427c3e, 0xb5c471b5, 0x66aacc66, 0x48d89048,
0x03050603, 0xf601f7f6, 0x0e121c0e, 0x61a3c261, 0x355f6a35, 0x57f9ae57,
0xb9d069b9, 0x86911786, 0xc15899c1, 0x1d273a1d, 0x9eb9279e, 0xe138d9e1,
0xf813ebf8, 0x98b32b98, 0x11332211, 0x69bbd269, 0xd970a9d9, 0x8e89078e,
0x94a73394, 0x9bb62d9b, 0x1e223c1e, 0x87921587, 0xe920c9e9, 0xce4987ce,
0x55ffaa55, 0x28785028, 0xdf7aa5df, 0x8c8f038c, 0xa1f859a1, 0x89800989,
0x0d171a0d, 0xbfda65bf, 0xe631d7e6, 0x42c68442, 0x68b8d068, 0x41c38241,
0x99b02999, 0x2d775a2d, 0x0f111e0f, 0xb0cb7bb0, 0x54fca854, 0xbbd66dbb,
0x163a2c16];
   var T4 = [0x6363a5c6, 0x7c7c84f8, 0x777799ee, 0x7b7b8df6, 0xf2f20dff, 0x6b6bbdd6,
0x6f6fb1de, 0xc5c55491, 0x30305060, 0x01010302, 0x6767a9ce, 0x2b2b7d56,
0xfefe19e7, 0xd7d762b5, 0xababe64d, 0x76769aec, 0xcaca458f, 0x82829d1f,
0xc9c94089, 0x7d7d87fa, 0xfafa15ef, 0x5959ebb2, 0x4747c98e, 0xf0f00bfb, 0xadadec41,
0xd4d467b3, 0xa2a2fd5f, 0xafafea45, 0x9c9cbf23, 0xa4a4f753, 0x727296e4, 0xc0c05b9b,
0xb7b7c275, 0xfdfd1ce1, 0x9393ae3d, 0x26266a4c, 0x36365a6c, 0x3f3f417e, 0xf7f702f5,
```

0xcccc4f83, 0x34345c68, 0xa5a5f451, 0xe5e534d1, 0xf1f108f9, 0x717193e2,
0xd8d873ab, 0x31315362, 0x15153f2a, 0x04040c08, 0xc7c75295, 0x23236546,
0xc3c35e9d, 0x18182830, 0x9696a137, 0x05050f0a, 0x9a9ab52f, 0x0707090e,
0x12123624, 0x80809b1b, 0xe2e23ddf, 0xebeb26cd, 0x2727694e, 0xb2b2cd7f,
0x75759fea, 0x09091b12, 0x83839e1d, 0x2c2c7458, 0x1a1a2e34, 0x1b1b2d36,
0x6e6eb2dc, 0x5a5aeeb4, 0xa0a0fb5b, 0x5252f6a4, 0x3b3b4d76, 0xd6d661b7,
0xb3b3ce7d, 0x29297b52, 0xe3e33edd, 0x2f2f715e, 0x84849713, 0x5353f5a6,
0xd1d168b9, 0x00000000, 0xeded2cc1, 0x20206040, 0xfcfc1fe3, 0xb1b1c879,
0x5b5bedb6, 0x6a6abed4, 0xcbcb468d, 0xbebed967, 0x39394b72, 0x4a4ade94,
0x4c4cd498, 0x5858e8b0, 0xcfcf4a85, 0xd0d06bbb, 0xefef2ac5, 0xaaaae54f, 0xfbfb16ed,
0x4343c586, 0x4d4dd79a, 0x33335566, 0x85859411, 0x4545cf8a, 0xf9f910e9,
0x02020604, 0x7f7f81fe, 0x5050f0a0, 0x3c3c4478, 0x9f9fba25, 0xa8a8e34b, 0x5151f3a2,
0xa3a3fe5d, 0x4040c080, 0x8f8f8a05, 0x9292ad3f, 0x9d9dbc21, 0x38384870,
0xf5f504f1, 0xbcbcdf63, 0xb6b6c177, 0xdada75af, 0x21216342, 0x10103020, 0xffff1ae5,
0xf3f30efd, 0xd2d26dbf, 0xcdcd4c81, 0x0c0c1418, 0x13133526, 0xecec2fc3, 0x5f5fe1be,
0x9797a235, 0x4444cc88, 0x1717392e, 0xc4c45793, 0xa7a7f255, 0x7e7e82fc,
0x3d3d477a, 0x6464acc8, 0x5d5de7ba, 0x19192b32, 0x737395e6, 0x6060a0c0,
0x81819819, 0x4f4fd19e, 0xdcdc7fa3, 0x22226644, 0x2a2a7e54, 0x9090ab3b,
0x8888830b, 0x4646ca8c, 0xeeee29c7, 0xb8b8d36b, 0x14143c28, 0xdede79a7,
0x5e5ee2bc, 0x0b0b1d16, 0xdbdb76ad, 0xe0e03bdb, 0x32325664, 0x3a3a4e74,
0x0a0a1e14, 0x4949db92, 0x06060a0c, 0x24246c48, 0x5c5ce4b8, 0xc2c25d9f,
0xd3d36ebd, 0xacacef43, 0x6262a6c4, 0x9191a839, 0x9595a431, 0xe4e437d3,
0x79798bf2, 0xe7e732d5, 0xc8c8438b, 0x3737596e, 0x6d6db7da, 0x8d8d8c01,
0xd5d564b1, 0x4e4ed29c, 0xa9a9e049, 0x6c6cb4d8, 0x5656faac, 0xf4f407f3, 0xeaea25cf,
0x6565afca, 0x7a7a8ef4, 0xaeaee947, 0x08081810, 0xbabad56f, 0x787888f0, 0x25256f4a,
0x2e2e725c, 0x1c1c2438, 0xa6a6f157, 0xb4b4c773, 0xc6c65197, 0xe8e823cb,
0xdddd7ca1, 0x74749ce8, 0x1f1f213e, 0x4b4bdd96, 0xbdbddc61, 0x8b8b860d,
0x8a8a850f, 0x707090e0, 0x3e3e427c, 0xb5b5c471, 0x6666aacc, 0x4848d890,
0x03030506, 0xf6f601f7, 0x0e0e121c, 0x6161a3c2, 0x35355f6a, 0x5757f9ae,
0xb9b9d069, 0x86869117, 0xc1c15899, 0x1d1d273a, 0x9e9eb927, 0xe1e138d9,
0xf8f813eb, 0x9898b32b, 0x11113322, 0x6969bbd2, 0xd9d970a9, 0x8e8e8907,
0x9494a733, 0x9b9bb62d, 0x1e1e223c, 0x87879215, 0xe9e920c9, 0xcece4987,
0x5555ffaa, 0x28287850, 0xdfdf7aa5, 0x8c8c8f03, 0xa1a1f859, 0x89898009,

```
0x0d0d171a, 0xbfbfda65, 0xe6e631d7, 0x4242c684, 0x6868b8d0, 0x4141c382,
0x9999b029, 0x2d2d775a, 0x0f0f111e, 0xb0b0cb7b, 0x5454fca8, 0xbbbbd66d,
0x16163a2c];

    // Transformations for decryption
    var T5 = [0x51f4a750, 0x7e416553, 0x1a17a4c3, 0x3a275e96, 0x3bab6bcb,
0x1f9d45f1, 0xacfa58ab, 0x4be30393, 0x2030fa55, 0xad766df6, 0x88cc7691,
0xf5024c25, 0x4fe5d7fc, 0xc52acbd7, 0x26354480, 0xb562a38f, 0xdeb15a49,
0x25ba1b67, 0x45ea0e98, 0x5dfec0e1, 0xc32f7502, 0x814cf012, 0x8d4697a3,
0x6bd3f9c6, 0x038f5fe7, 0x15929c95, 0xbf6d7aeb, 0x955259da, 0xd4be832d,
0x587421d3, 0x49e06929, 0x8ec9c844, 0x75c2896a, 0xf48e7978, 0x99583e6b,
0x27b971dd, 0xbee14fb6, 0xf088ad17, 0xc920ac66, 0x7dce3ab4, 0x63df4a18,
0xe51a3182, 0x97513360, 0x62537f45, 0xb16477e0, 0xbb6bae84, 0xfe81a01c,
0xf9082b94, 0x70486858, 0x8f45fd19, 0x94de6c87, 0x527bf8b7, 0xab73d323,
0x724b02e2, 0xe31f8f57, 0x6655ab2a, 0xb2eb2807, 0x2fb5c203, 0x86c57b9a,
0xd33708a5, 0x302887f2, 0x23bfa5b2, 0x02036aba, 0xed16825c, 0x8acf1c2b,
0xa779b492, 0xf307f2f0, 0x4e69e2a1, 0x65daf4cd, 0x0605bed5, 0xd134621f, 0xc4a6fe8a,
0x342e539d, 0xa2f355a0, 0x058ae132, 0xa4f6eb75, 0x0b83ec39, 0x4060efaa,
0x5e719f06, 0xbd6e1051, 0x3e218af9, 0x96dd063d, 0xdd3e05ae, 0x4de6bd46,
0x91548db5, 0x71c45d05, 0x0406d46f, 0x605015ff, 0x1998fb24, 0xd6bde997,
0x894043cc, 0x67d99e77, 0xb0e842bd, 0x07898b88, 0xe7195b38, 0x79c8eedb,
0xa17c0a47, 0x7c420fe9, 0xf8841ec9, 0x00000000, 0x09808683, 0x322bed48,
0x1e1170ac, 0x6c5a724e, 0xfd0efffb, 0x0f853856, 0x3daed51e, 0x362d3927, 0x0a0fd964,
0x685ca621, 0x9b5b54d1, 0x24362e3a, 0x0c0a67b1, 0x9357e70f, 0xb4ee96d2,
0x1b9b919e, 0x80c0c54f, 0x61dc20a2, 0x5a774b69, 0x1c121a16, 0xe293ba0a,
0xc0a02ae5, 0x3c22e043, 0x121b171d, 0x0e090d0b, 0xf28bc7ad, 0x2db6a8b9,
0x141ea9c8, 0x57f11985, 0xaf75074c, 0xee99ddbb, 0xa37f60fd, 0xf701269f, 0x5c72f5bc,
0x44663bc5, 0x5bfb7e34, 0x8b432976, 0xcb23c6dc, 0xb6edfc68, 0xb8e4f163,
0xd731dcca, 0x42638510, 0x13972240, 0x84c61120, 0x854a247d, 0xd2bb3df8,
0xaef93211, 0xc729a16d, 0x1d9e2f4b, 0xdcb230f3, 0x0d8652ec, 0x77c1e3d0,
0x2bb3166c, 0xa970b999, 0x119448fa, 0x47e96422, 0xa8fc8cc4, 0xa0f03f1a,
0x567d2cd8, 0x223390ef, 0x87494ec7, 0xd938d1c1, 0x8ccaa2fe, 0x98d40b36,
0xa6f581cf, 0xa57ade28, 0xdab78e26, 0x3fadbfa4, 0x2c3a9de4, 0x5078920d, 0x6a5fcc9b,
```

```
0x547e4662, 0xf68d13c2, 0x90d8b8e8, 0x2e39f75e, 0x82c3aff5, 0x9f5d80be,
0x69d0937c, 0x6fd52da9, 0xcf2512b3, 0xc8ac993b, 0x10187da7, 0xe89c636e,
0xdb3bbb7b, 0xcd267809, 0x6e5918f4, 0xec9ab701, 0x834f9aa8, 0xe6956e65,
0xaaffe67e, 0x21bccf08, 0xef15e8e6, 0xbae79bd9, 0x4a6f36ce, 0xea9f09d4, 0x29b07cd6,
0x31a4b2af, 0x2a3f2331, 0xc6a59430, 0x35a266c0, 0x744ebc37, 0xfc82caa6,
0xe090d0b0, 0x33a7d815, 0xf104984a, 0x41ecdaf7, 0x7fcd500e, 0x1791f62f,
0x764dd68d, 0x43efb04d, 0xccaa4d54, 0xe49604df, 0x9ed1b5e3, 0x4c6a881b,
0xc12c1fb8, 0x4665517f, 0x9d5eea04, 0x018c355d, 0xfa877473, 0xfb0b412e,
0xb3671d5a, 0x92dbd252, 0xe9105633, 0x6dd64713, 0x9ad7618c, 0x37a10c7a,
0x59f8148e, 0xeb133c89, 0xcea927ee, 0xb761c935, 0xe11ce5ed, 0x7a47b13c,
0x9cd2df59, 0x55f2733f, 0x1814ce79, 0x73c737bf, 0x53f7cdea, 0x5ffdaa5b, 0xdf3d6f14,
0x7844db86, 0xcaaff381, 0xb968c43e, 0x3824342c, 0xc2a3405f, 0x161dc372,
0xbce2250c, 0x283c498b, 0xff0d9541, 0x39a80171, 0x080cb3de, 0xd8b4e49c,
0x6456c190, 0x7bcb8461, 0xd532b670, 0x486c5c74, 0xd0b85742];
    var T6 = [0x5051f4a7, 0x537e4165, 0xc31a17a4, 0x963a275e, 0xcb3bab6b,
0xf11f9d45, 0xabacfa58, 0x934be303, 0x552030fa, 0xf6ad766d, 0x9188cc76,
0x25f5024c, 0xfc4fe5d7, 0xd7c52acb, 0x80263544, 0x8fb562a3, 0x49deb15a,
0x6725ba1b, 0x9845ea0e, 0xe15dfec0, 0x02c32f75, 0x12814cf0, 0xa38d4697,
0xc66bd3f9, 0xe7038f5f, 0x9515929c, 0xebbf6d7a, 0xda955259, 0x2dd4be83,
0xd3587421, 0x2949e069, 0x448ec9c8, 0x6a75c289, 0x78f48e79, 0x6b99583e,
0xdd27b971, 0xb6bee14f, 0x17f088ad, 0x66c920ac, 0xb47dce3a, 0x1863df4a,
0x82e51a31, 0x60975133, 0x4562537f, 0xe0b16477, 0x84bb6bae, 0x1cfe81a0,
0x94f9082b, 0x58704868, 0x198f45fd, 0x8794de6c, 0xb7527bf8, 0x23ab73d3,
0xe2724b02, 0x57e31f8f, 0x2a6655ab, 0x07b2eb28, 0x032fb5c2, 0x9a86c57b,
0xa5d33708, 0xf2302887, 0xb223bfa5, 0xba02036a, 0x5ced1682, 0x2b8acf1c,
0x92a779b4, 0xf0f307f2, 0xa14e69e2, 0xcd65daf4, 0xd50605be, 0x1fd13462, 0x8ac4a6fe,
0x9d342e53, 0xa0a2f355, 0x32058ae1, 0x75a4f6eb, 0x390b83ec, 0xaa4060ef,
0x065e719f, 0x51bd6e10, 0xf93e218a, 0x3d96dd06, 0xaedd3e05, 0x464de6bd,
0xb591548d, 0x0571c45d, 0x6f0406d4, 0xff605015, 0x241998fb, 0x97d6bde9,
0xcc894043, 0x7767d99e, 0xbdb0e842, 0x8807898b, 0x38e7195b, 0xdb79c8ee,
0x47a17c0a, 0xe97c420f, 0xc9f8841e, 0x00000000, 0x83098086, 0x48322bed,
0xac1e1170, 0x4e6c5a72, 0xfbfd0eff, 0x560f8538, 0x1e3daed5, 0x27362d39, 0x640a0fd9,
0x21685ca6, 0xd19b5b54, 0x3a24362e, 0xb10c0a67, 0x0f9357e7, 0xd2b4ee96,
```

```
0x9e1b9b91, 0x4f80c0c5, 0xa261dc20, 0x695a774b, 0x161c121a, 0x0ae293ba,
0xe5c0a02a, 0x433c22e0, 0x1d121b17, 0x0b0e090d, 0xadf28bc7, 0xb92db6a8,
0xc8141ea9, 0x8557f119, 0x4caf7507, 0xbbee99dd, 0xfda37f60, 0x9ff70126, 0xbc5c72f5,
0xc544663b, 0x345bfb7e, 0x768b4329, 0xdccb23c6, 0x68b6edfc, 0x63b8e4f1,
0xcad731dc, 0x10426385, 0x40139722, 0x2084c611, 0x7d854a24, 0xf8d2bb3d,
0x11aef932, 0x6dc729a1, 0x4b1d9e2f, 0xf3dcb230, 0xec0d8652, 0xd077c1e3,
0x6c2bb316, 0x99a970b9, 0xfa119448, 0x2247e964, 0xc4a8fc8c, 0x1aa0f03f,
0xd8567d2c, 0xef223390, 0xc787494e, 0xc1d938d1, 0xfe8ccaa2, 0x3698d40b,
0xcfa6f581, 0x28a57ade, 0x26dab78e, 0xa43fadbf, 0xe42c3a9d, 0x0d507892, 0x9b6a5fcc,
0x62547e46, 0xc2f68d13, 0xe890d8b8, 0x5e2e39f7, 0xf582c3af, 0xbe9f5d80,
0x7c69d093, 0xa96fd52d, 0xb3cf2512, 0x3bc8ac99, 0xa710187d, 0x6ee89c63,
0x7bdb3bbb, 0x09cd2678, 0xf46e5918, 0x01ec9ab7, 0xa8834f9a, 0x65e6956e,
0x7eaaffe6, 0x0821bccf, 0xe6ef15e8, 0xd9bae79b, 0xce4a6f36, 0xd4ea9f09, 0xd629b07c,
0xaf31a4b2, 0x312a3f23, 0x30c6a594, 0xc035a266, 0x37744ebc, 0xa6fc82ca,
0xb0e090d0, 0x1533a7d8, 0x4af10498, 0xf741ecda, 0x0e7fcd50, 0x2f1791f6,
0x8d764dd6, 0x4d43efb0, 0x54ccaa4d, 0xdfe49604, 0xe39ed1b5, 0x1b4c6a88,
0xb8c12c1f, 0x7f466551, 0x049d5eea, 0x5d018c35, 0x73fa8774, 0x2efb0b41,
0x5ab3671d, 0x5292dbd2, 0x33e91056, 0x136dd647, 0x8c9ad761, 0x7a37a10c,
0x8e59f814, 0x89eb133c, 0xeecea927, 0x35b761c9, 0xede11ce5, 0x3c7a47b1,
0x599cd2df, 0x3f55f273, 0x791814ce, 0xbf73c737, 0xea53f7cd, 0x5b5ffdaa, 0x14df3d6f,
0x867844db, 0x81caaff3, 0x3eb968c4, 0x2c382434, 0x5fc2a340, 0x72161dc3,
0x0cbce225, 0x8b283c49, 0x41ff0d95, 0x7139a801, 0xde080cb3, 0x9cd8b4e4,
0x906456c1, 0x617bcb84, 0x70d532b6, 0x74486c5c, 0x42d0b857];
    var T7 = [0xa75051f4, 0x65537e41, 0xa4c31a17, 0x5e963a27, 0x6bcb3bab,
0x45f11f9d, 0x58abacfa, 0x03934be3, 0xfa552030, 0x6df6ad76, 0x769188cc,
0x4c25f502, 0xd7fc4fe5, 0xcbd7c52a, 0x44802635, 0xa38fb562, 0x5a49deb1,
0x1b6725ba, 0x0e9845ea, 0xc0e15dfe, 0x7502c32f, 0xf012814c, 0x97a38d46,
0xf9c66bd3, 0x5fe7038f, 0x9c951592, 0x7aebbf6d, 0x59da9552, 0x832dd4be,
0x21d35874, 0x692949e0, 0xc8448ec9, 0x896a75c2, 0x7978f48e, 0x3e6b9958,
0x71dd27b9, 0x4fb6bee1, 0xad17f088, 0xac66c920, 0x3ab47dce, 0x4a1863df,
0x3182e51a, 0x33609751, 0x7f456253, 0x77e0b164, 0xae84bb6b, 0xa01cfe81,
0x2b94f908, 0x68587048, 0xfd198f45, 0x6c8794de, 0xf8b7527b, 0xd323ab73,
0x02e2724b, 0x8f57e31f, 0xab2a6655, 0x2807b2eb, 0xc2032fb5, 0x7b9a86c5,
```

```
0x08a5d337, 0x87f23028, 0xa5b223bf, 0x6aba0203, 0x825ced16, 0x1c2b8acf,
0xb492a779, 0xf2f0f307, 0xe2a14e69, 0xf4cd65da, 0xbed50605, 0x621fd134, 0xfe8ac4a6,
0x539d342e, 0x55a0a2f3, 0xe132058a, 0xeb75a4f6, 0xec390b83, 0xefaa4060,
0x9f065e71, 0x1051bd6e, 0x8af93e21, 0x063d96dd, 0x05aedd3e, 0xbd464de6,
0x8db59154, 0x5d0571c4, 0xd46f0406, 0x15ff6050, 0xfb241998, 0xe997d6bd,
0x43cc8940, 0x9e7767d9, 0x42bdb0e8, 0x8b880789, 0x5b38e719, 0xeedb79c8,
0x0a47a17c, 0x0fe97c42, 0x1ec9f884, 0x00000000, 0x86830980, 0xed48322b,
0x70ac1e11, 0x724e6c5a, 0xfffbfd0e, 0x38560f85, 0xd51e3dae, 0x3927362d, 0xd9640a0f,
0xa621685c, 0x54d19b5b, 0x2e3a2436, 0x67b10c0a, 0xe70f9357, 0x96d2b4ee,
0x919e1b9b, 0xc54f80c0, 0x20a261dc, 0x4b695a77, 0x1a161c12, 0xba0ae293,
0x2ae5c0a0, 0xe0433c22, 0x171d121b, 0x0d0b0e09, 0xc7adf28b, 0xa8b92db6,
0xa9c8141e, 0x198557f1, 0x074caf75, 0xddbbee99, 0x60fda37f, 0x269ff701, 0xf5bc5c72,
0x3bc54466, 0x7e345bfb, 0x29768b43, 0xc6dccb23, 0xfc68b6ed, 0xf163b8e4,
0xdccad731, 0x85104263, 0x22401397, 0x112084c6, 0x247d854a, 0x3df8d2bb,
0x3211aef9, 0xa16dc729, 0x2f4b1d9e, 0x30f3dcb2, 0x52ec0d86, 0xe3d077c1,
0x166c2bb3, 0xb999a970, 0x48fa1194, 0x642247e9, 0x8cc4a8fc, 0x3f1aa0f0,
0x2cd8567d, 0x90ef2233, 0x4ec78749, 0xd1c1d938, 0xa2fe8cca, 0x0b3698d4,
0x81cfa6f5, 0xde28a57a, 0x8e26dab7, 0xbfa43fad, 0x9de42c3a, 0x920d5078, 0xcc9b6a5f,
0x4662547e, 0x13c2f68d, 0xb8e890d8, 0xf75e2e39, 0xaff582c3, 0x80be9f5d,
0x937c69d0, 0x2da96fd5, 0x12b3cf25, 0x993bc8ac, 0x7da71018, 0x636ee89c,
0xbb7bdb3b, 0x7809cd26, 0x18f46e59, 0xb701ec9a, 0x9aa8834f, 0x6e65e695,
0xe67eaaff, 0xcf0821bc, 0xe8e6ef15, 0x9bd9bae7, 0x36ce4a6f, 0x09d4ea9f, 0x7cd629b0,
0xb2af31a4, 0x23312a3f, 0x9430c6a5, 0x66c035a2, 0xbc37744e, 0xcaa6fc82,
0xd0b0e090, 0xd81533a7, 0x984af104, 0xdaf741ec, 0x500e7fcd, 0xf62f1791,
0xd68d764d, 0xb04d43ef, 0x4d54ccaa, 0x04dfe496, 0xb5e39ed1, 0x881b4c6a,
0x1fb8c12c, 0x517f4665, 0xea049d5e, 0x355d018c, 0x7473fa87, 0x412efb0b,
0x1d5ab367, 0xd25292db, 0x5633e910, 0x47136dd6, 0x618c9ad7, 0x0c7a37a1,
0x148e59f8, 0x3c89eb13, 0x27eecea9, 0xc935b761, 0xe5ede11c, 0xb13c7a47,
0xdf599cd2, 0x733f55f2, 0xce791814, 0x37bf73c7, 0xcdea53f7, 0xaa5b5ffd, 0x6f14df3d,
0xdb867844, 0xf381caaf, 0xc43eb968, 0x342c3824, 0x405fc2a3, 0xc372161d,
0x250cbce2, 0x498b283c, 0x9541ff0d, 0x017139a8, 0xb3de080c, 0xe49cd8b4,
0xc1906456, 0x84617bcb, 0xb670d532, 0x5c74486c, 0x5742d0b8];
```

```
    var T8 = [0xf4a75051, 0x4165537e, 0x17a4c31a, 0x275e963a, 0xab6bcb3b,
0x9d45f11f, 0xfa58abac, 0xe303934b, 0x30fa5520, 0x766df6ad, 0xcc769188,
0x024c25f5, 0xe5d7fc4f, 0x2acbd7c5, 0x35448026, 0x62a38fb5, 0xb15a49de,
0xba1b6725, 0xea0e9845, 0xfec0e15d, 0x2f7502c3, 0x4cf01281, 0x4697a38d,
0xd3f9c66b, 0x8f5fe703, 0x929c9515, 0x6d7aebbf, 0x5259da95, 0xbe832dd4,
0x7421d358, 0xe0692949, 0xc9c8448e, 0xc2896a75, 0x8e7978f4, 0x583e6b99,
0xb971dd27, 0xe14fb6be, 0x88ad17f0, 0x20ac66c9, 0xce3ab47d, 0xdf4a1863,
0x1a3182e5, 0x51336097, 0x537f4562, 0x6477e0b1, 0x6bae84bb, 0x81a01cfe,
0x082b94f9, 0x48685870, 0x45fd198f, 0xde6c8794, 0x7bf8b752, 0x73d323ab,
0x4b02e272, 0x1f8f57e3, 0x55ab2a66, 0xeb2807b2, 0xb5c2032f, 0xc57b9a86,
0x3708a5d3, 0x2887f230, 0xbfa5b223, 0x036aba02, 0x16825ced, 0xcf1c2b8a,
0x79b492a7, 0x07f2f0f3, 0x69e2a14e, 0xdaf4cd65, 0x05bed506, 0x34621fd1, 0xa6fe8ac4,
0x2e539d34, 0xf355a0a2, 0x8ae13205, 0xf6eb75a4, 0x83ec390b, 0x60efaa40,
0x719f065e, 0x6e1051bd, 0x218af93e, 0xdd063d96, 0x3e05aedd, 0xe6bd464d,
0x548db591, 0xc45d0571, 0x06d46f04, 0x5015ff60, 0x98fb2419, 0xbde997d6,
0x4043cc89, 0xd99e7767, 0xe842bdb0, 0x898b8807, 0x195b38e7, 0xc8eedb79,
0x7c0a47a1, 0x420fe97c, 0x841ec9f8, 0x00000000, 0x80868309, 0x2bed4832,
0x1170ac1e, 0x5a724e6c, 0x0efffbfd, 0x8538560f, 0xaed51e3d, 0x2d392736, 0x0fd9640a,
0x5ca62168, 0x5b54d19b, 0x362e3a24, 0x0a67b10c, 0x57e70f93, 0xee96d2b4,
0x9b919e1b, 0xc0c54f80, 0xdc20a261, 0x774b695a, 0x121a161c, 0x93ba0ae2,
0xa02ae5c0, 0x22e0433c, 0x1b171d12, 0x090d0b0e, 0x8bc7adf2, 0xb6a8b92d,
0x1ea9c814, 0xf1198557, 0x75074caf, 0x99ddbbee, 0x7f60fda3, 0x01269ff7, 0x72f5bc5c,
0x663bc544, 0xfb7e345b, 0x4329768b, 0x23c6dccb, 0xedfc68b6, 0xe4f163b8,
0x31dccad7, 0x63851042, 0x97224013, 0xc6112084, 0x4a247d85, 0xbb3df8d2,
0xf93211ae, 0x29a16dc7, 0x9e2f4b1d, 0xb230f3dc, 0x8652ec0d, 0xc1e3d077,
0xb3166c2b, 0x70b999a9, 0x9448fa11, 0xe9642247, 0xfc8cc4a8, 0xf03f1aa0,
0x7d2cd856, 0x3390ef22, 0x494ec787, 0x38d1c1d9, 0xcaa2fe8c, 0xd40b3698,
0xf581cfa6, 0x7ade28a5, 0xb78e26da, 0xadbfa43f, 0x3a9de42c, 0x78920d50, 0x5fcc9b6a,
0x7e466254, 0x8d13c2f6, 0xd8b8e890, 0x39f75e2e, 0xc3aff582, 0x5d80be9f,
0xd0937c69, 0xd52da96f, 0x2512b3cf, 0xac993bc8, 0x187da710, 0x9c636ee8,
0x3bbb7bdb, 0x267809cd, 0x5918f46e, 0x9ab701ec, 0x4f9aa883, 0x956e65e6,
0xffe67eaa, 0xbccf0821, 0x15e8e6ef, 0xe79bd9ba, 0x6f36ce4a, 0x9f09d4ea, 0xb07cd629,
0xa4b2af31, 0x3f23312a, 0xa59430c6, 0xa266c035, 0x4ebc3774, 0x82caa6fc,
```

```
0x90d0b0e0, 0xa7d81533, 0x04984af1, 0xecdaf741, 0xcd500e7f, 0x91f62f17,
0x4dd68d76, 0xefb04d43, 0xaa4d54cc, 0x9604dfe4, 0xd1b5e39e, 0x6a881b4c,
0x2c1fb8c1, 0x65517f46, 0x5eea049d, 0x8c355d01, 0x877473fa, 0x0b412efb,
0x671d5ab3, 0xdbd25292, 0x105633e9, 0xd647136d, 0xd7618c9a, 0xa10c7a37,
0xf8148e59, 0x133c89eb, 0xa927eece, 0x61c935b7, 0x1ce5ede1, 0x47b13c7a,
0xd2df599c, 0xf2733f55, 0x14ce7918, 0xc737bf73, 0xf7cdea53, 0xfdaa5b5f, 0x3d6f14df,
0x44db8678, 0xaff381ca, 0x68c43eb9, 0x24342c38, 0xa3405fc2, 0x1dc37216,
0xe2250cbc, 0x3c498b28, 0x0d9541ff, 0xa8017139, 0x0cb3de08, 0xb4e49cd8,
0x56c19064, 0xcb84617b, 0x32b670d5, 0x6c5c7448, 0xb85742d0];

    // Transformations for decryption key expansion
    var U1 = [0x00000000, 0x0e090d0b, 0x1c121a16, 0x121b171d, 0x3824342c,
0x362d3927, 0x24362e3a, 0x2a3f2331, 0x70486858, 0x7e416553, 0x6c5a724e,
0x62537f45, 0x486c5c74, 0x4665517f, 0x547e4662, 0x5a774b69, 0xe090d0b0,
0xee99ddbb, 0xfc82caa6, 0xf28bc7ad, 0xd8b4e49c, 0xd6bde997, 0xc4a6fe8a, 0xcaaff381,
0x90d8b8e8, 0x9ed1b5e3, 0x8ccaa2fe, 0x82c3aff5, 0xa8fc8cc4, 0xa6f581cf, 0xb4ee96d2,
0xbae79bd9, 0xdb3bbb7b, 0xd532b670, 0xc729a16d, 0xc920ac66, 0xe31f8f57,
0xed16825c, 0xff0d9541, 0xf104984a, 0xab73d323, 0xa57ade28, 0xb761c935,
0xb968c43e, 0x9357e70f, 0x9d5eea04, 0x8f45fd19, 0x814cf012, 0x3bab6bcb,
0x35a266c0, 0x27b971dd, 0x29b07cd6, 0x038f5fe7, 0x0d8652ec, 0x1f9d45f1,
0x119448fa, 0x4be30393, 0x45ea0e98, 0x57f11985, 0x59f8148e, 0x73c737bf,
0x7dce3ab4, 0x6fd52da9, 0x61dc20a2, 0xad766df6, 0xa37f60fd, 0xb16477e0,
0xbf6d7aeb, 0x955259da, 0x9b5b54d1, 0x894043cc, 0x87494ec7, 0xdd3e05ae,
0xd33708a5, 0xc12c1fb8, 0xcf2512b3, 0xe51a3182, 0xeb133c89, 0xf9082b94,
0xf701269f, 0x4de6bd46, 0x43efb04d, 0x51f4a750, 0x5ffdaa5b, 0x75c2896a,
0x7bcb8461, 0x69d0937c, 0x67d99e77, 0x3daed51e, 0x33a7d815, 0x21bccf08,
0x2fb5c203, 0x058ae132, 0x0b83ec39, 0x1998fb24, 0x1791f62f, 0x764dd68d,
0x7844db86, 0x6a5fcc9b, 0x6456c190, 0x4e69e2a1, 0x4060efaa, 0x527bf8b7,
0x5c72f5bc, 0x0605bed5, 0x080cb3de, 0x1a17a4c3, 0x141ea9c8, 0x3e218af9,
0x302887f2, 0x223390ef, 0x2c3a9de4, 0x96dd063d, 0x98d40b36, 0x8acf1c2b,
0x84c61120, 0xaef93211, 0xa0f03f1a, 0xb2eb2807, 0xbce2250c, 0xe6956e65,
0xe89c636e, 0xfa877473, 0xf48e7978, 0xdeb15a49, 0xd0b85742, 0xc2a3405f,
0xccaa4d54, 0x41ecdaf7, 0x4fe5d7fc, 0x5dfec0e1, 0x53f7cdea, 0x79c8eedb, 0x77c1e3d0,
```

```
0x65daf4cd, 0x6bd3f9c6, 0x31a4b2af, 0x3fadbfa4, 0x2db6a8b9, 0x23bfa5b2,
0x09808683, 0x07898b88, 0x15929c95, 0x1b9b919e, 0xa17c0a47, 0xaf75074c,
0xbd6e1051, 0xb3671d5a, 0x99583e6b, 0x97513360, 0x854a247d, 0x8b432976,
0xd134621f, 0xdf3d6f14, 0xcd267809, 0xc32f7502, 0xe9105633, 0xe7195b38,
0xf5024c25, 0xfb0b412e, 0x9ad7618c, 0x94de6c87, 0x86c57b9a, 0x88cc7691,
0xa2f355a0, 0xacfa58ab, 0xbee14fb6, 0xb0e842bd, 0xea9f09d4, 0xe49604df, 0xf68d13c2,
0xf8841ec9, 0xd2bb3df8, 0xdcb230f3, 0xcea927ee, 0xc0a02ae5, 0x7a47b13c,
0x744ebc37, 0x6655ab2a, 0x685ca621, 0x42638510, 0x4c6a881b, 0x5e719f06,
0x5078920d, 0x0a0fd964, 0x0406d46f, 0x161dc372, 0x1814ce79, 0x322bed48,
0x3c22e043, 0x2e39f75e, 0x2030fa55, 0xec9ab701, 0xe293ba0a, 0xf088ad17,
0xfe81a01c, 0xd4be832d, 0xdab78e26, 0xc8ac993b, 0xc6a59430, 0x9cd2df59,
0x92dbd252, 0x80c0c54f, 0x8ec9c844, 0xa4f6eb75, 0xaaffe67e, 0xb8e4f163, 0xb6edfc68,
0x0c0a67b1, 0x02036aba, 0x10187da7, 0x1e1170ac, 0x342e539d, 0x3a275e96,
0x283c498b, 0x26354480, 0x7c420fe9, 0x724b02e2, 0x605015ff, 0x6e5918f4,
0x44663bc5, 0x4a6f36ce, 0x587421d3, 0x567d2cd8, 0x37a10c7a, 0x39a80171,
0x2bb3166c, 0x25ba1b67, 0x0f853856, 0x018c355d, 0x13972240, 0x1d9e2f4b,
0x47e96422, 0x49e06929, 0x5bfb7e34, 0x55f2733f, 0x7fcd500e, 0x71c45d05,
0x63df4a18, 0x6dd64713, 0xd731dcca, 0xd938d1c1, 0xcb23c6dc, 0xc52acbd7,
0xef15e8e6, 0xe11ce5ed, 0xf307f2f0, 0xfd0efffb, 0xa779b492, 0xa970b999, 0xbb6bae84,
0xb562a38f, 0x9f5d80be, 0x91548db5, 0x834f9aa8, 0x8d4697a3];
   var U2 = [0x00000000, 0x0b0e090d, 0x161c121a, 0x1d121b17, 0x2c382434,
0x27362d39, 0x3a24362e, 0x312a3f23, 0x58704868, 0x537e4165, 0x4e6c5a72,
0x4562537f, 0x74486c5c, 0x7f466551, 0x62547e46, 0x695a774b, 0xb0e090d0,
0xbbee99dd, 0xa6fc82ca, 0xadf28bc7, 0x9cd8b4e4, 0x97d6bde9, 0x8ac4a6fe, 0x81caaff3,
0xe890d8b8, 0xe39ed1b5, 0xfe8ccaa2, 0xf582c3af, 0xc4a8fc8c, 0xcfa6f581, 0xd2b4ee96,
0xd9bae79b, 0x7bdb3bbb, 0x70d532b6, 0x6dc729a1, 0x66c920ac, 0x57e31f8f,
0x5ced1682, 0x41ff0d95, 0x4af10498, 0x23ab73d3, 0x28a57ade, 0x35b761c9,
0x3eb968c4, 0x0f9357e7, 0x049d5eea, 0x198f45fd, 0x12814cf0, 0xcb3bab6b,
0xc035a266, 0xdd27b971, 0xd629b07c, 0xe7038f5f, 0xec0d8652, 0xf11f9d45,
0xfa119448, 0x934be303, 0x9845ea0e, 0x8557f119, 0x8e59f814, 0xbf73c737,
0xb47dce3a, 0xa96fd52d, 0xa261dc20, 0xf6ad766d, 0xfda37f60, 0xe0b16477,
0xebbf6d7a, 0xda955259, 0xd19b5b54, 0xcc894043, 0xc787494e, 0xaedd3e05,
0xa5d33708, 0xb8c12c1f, 0xb3cf2512, 0x82e51a31, 0x89eb133c, 0x94f9082b,
```

```
0x9ff70126, 0x464de6bd, 0x4d43efb0, 0x5051f4a7, 0x5b5ffdaa, 0x6a75c289,
0x617bcb84, 0x7c69d093, 0x7767d99e, 0x1e3daed5, 0x1533a7d8, 0x0821bccf,
0x032fb5c2, 0x32058ae1, 0x390b83ec, 0x241998fb, 0x2f1791f6, 0x8d764dd6,
0x867844db, 0x9b6a5fcc, 0x906456c1, 0xa14e69e2, 0xaa4060ef, 0xb7527bf8,
0xbc5c72f5, 0xd50605be, 0xde080cb3, 0xc31a17a4, 0xc8141ea9, 0xf93e218a,
0xf2302887, 0xef223390, 0xe42c3a9d, 0x3d96dd06, 0x3698d40b, 0x2b8acf1c,
0x2084c611, 0x11aef932, 0x1aa0f03f, 0x07b2eb28, 0x0cbce225, 0x65e6956e,
0x6ee89c63, 0x73fa8774, 0x78f48e79, 0x49deb15a, 0x42d0b857, 0x5fc2a340,
0x54ccaa4d, 0xf741ecda, 0xfc4fe5d7, 0xe15dfec0, 0xea53f7cd, 0xdb79c8ee, 0xd077c1e3,
0xcd65daf4, 0xc66bd3f9, 0xaf31a4b2, 0xa43fadbf, 0xb92db6a8, 0xb223bfa5,
0x83098086, 0x8807898b, 0x9515929c, 0x9e1b9b91, 0x47a17c0a, 0x4caf7507,
0x51bd6e10, 0x5ab3671d, 0x6b99583e, 0x60975133, 0x7d854a24, 0x768b4329,
0x1fd13462, 0x14df3d6f, 0x09cd2678, 0x02c32f75, 0x33e91056, 0x38e7195b,
0x25f5024c, 0x2efb0b41, 0x8c9ad761, 0x8794de6c, 0x9a86c57b, 0x9188cc76,
0xa0a2f355, 0xabacfa58, 0xb6bee14f, 0xbdb0e842, 0xd4ea9f09, 0xdfe49604, 0xc2f68d13,
0xc9f8841e, 0xf8d2bb3d, 0xf3dcb230, 0xeecea927, 0xe5c0a02a, 0x3c7a47b1,
0x37744ebc, 0x2a6655ab, 0x21685ca6, 0x10426385, 0x1b4c6a88, 0x065e719f,
0x0d507892, 0x640a0fd9, 0x6f0406d4, 0x72161dc3, 0x791814ce, 0x48322bed,
0x433c22e0, 0x5e2e39f7, 0x552030fa, 0x01ec9ab7, 0x0ae293ba, 0x17f088ad,
0x1cfe81a0, 0x2dd4be83, 0x26dab78e, 0x3bc8ac99, 0x30c6a594, 0x599cd2df,
0x5292dbd2, 0x4f80c0c5, 0x448ec9c8, 0x75a4f6eb, 0x7eaaffe6, 0x63b8e4f1, 0x68b6edfc,
0xb10c0a67, 0xba02036a, 0xa710187d, 0xac1e1170, 0x9d342e53, 0x963a275e,
0x8b283c49, 0x80263544, 0xe97c420f, 0xe2724b02, 0xff605015, 0xf46e5918,
0xc544663b, 0xce4a6f36, 0xd3587421, 0xd8567d2c, 0x7a37a10c, 0x7139a801,
0x6c2bb316, 0x6725ba1b, 0x560f8538, 0x5d018c35, 0x40139722, 0x4b1d9e2f,
0x2247e964, 0x2949e069, 0x345bfb7e, 0x3f55f273, 0x0e7fcd50, 0x0571c45d,
0x1863df4a, 0x136dd647, 0xcad731dc, 0xc1d938d1, 0xdccb23c6, 0xd7c52acb,
0xe6ef15e8, 0xede11ce5, 0xf0f307f2, 0xfbfd0eff, 0x92a779b4, 0x99a970b9, 0x84bb6bae,
0x8fb562a3, 0xbe9f5d80, 0xb591548d, 0xa8834f9a, 0xa38d4697];
    var U3 = [0x00000000, 0x0d0b0e09, 0x1a161c12, 0x171d121b, 0x342c3824,
0x3927362d, 0x2e3a2436, 0x23312a3f, 0x68587048, 0x65537e41, 0x724e6c5a,
0x7f456253, 0x5c74486c, 0x517f4665, 0x4662547e, 0x4b695a77, 0xd0b0e090,
0xddbbee99, 0xcaa6fc82, 0xc7adf28b, 0xe49cd8b4, 0xe997d6bd, 0xfe8ac4a6, 0xf381caaf,
```

0xb8e890d8, 0xb5e39ed1, 0xa2fe8cca, 0xaff582c3, 0x8cc4a8fc, 0x81cfa6f5, 0x96d2b4ee, 0x9bd9bae7, 0xbb7bdb3b, 0xb670d532, 0xa16dc729, 0xac66c920, 0x8f57e31f, 0x825ced16, 0x9541ff0d, 0x984af104, 0xd323ab73, 0xde28a57a, 0xc935b761, 0xc43eb968, 0xe70f9357, 0xea049d5e, 0xfd198f45, 0xf012814c, 0x6bcb3bab, 0x66c035a2, 0x71dd27b9, 0x7cd629b0, 0x5fe7038f, 0x52ec0d86, 0x45f11f9d, 0x48fa1194, 0x03934be3, 0x0e9845ea, 0x198557f1, 0x148e59f8, 0x37bf73c7, 0x3ab47dce, 0x2da96fd5, 0x20a261dc, 0x6df6ad76, 0x60fda37f, 0x77e0b164, 0x7aebbf6d, 0x59da9552, 0x54d19b5b, 0x43cc8940, 0x4ec78749, 0x05aedd3e, 0x08a5d337, 0x1fb8c12c, 0x12b3cf25, 0x3182e51a, 0x3c89eb13, 0x2b94f908, 0x269ff701, 0xbd464de6, 0xb04d43ef, 0xa75051f4, 0xaa5b5ffd, 0x896a75c2, 0x84617bcb, 0x937c69d0, 0x9e7767d9, 0xd51e3dae, 0xd81533a7, 0xcf0821bc, 0xc2032fb5, 0xe132058a, 0xec390b83, 0xfb241998, 0xf62f1791, 0xd68d764d, 0xdb867844, 0xcc9b6a5f, 0xc1906456, 0xe2a14e69, 0xefaa4060, 0xf8b7527b, 0xf5bc5c72, 0xbed50605, 0xb3de080c, 0xa4c31a17, 0xa9c8141e, 0x8af93e21, 0x87f23028, 0x90ef2233, 0x9de42c3a, 0x063d96dd, 0x0b3698d4, 0x1c2b8acf, 0x112084c6, 0x3211aef9, 0x3f1aa0f0, 0x2807b2eb, 0x250cbce2, 0x6e65e695, 0x636ee89c, 0x7473fa87, 0x7978f48e, 0x5a49deb1, 0x5742d0b8, 0x405fc2a3, 0x4d54ccaa, 0xdaf741ec, 0xd7fc4fe5, 0xc0e15dfe, 0xcdea53f7, 0xeedb79c8, 0xe3d077c1, 0xf4cd65da, 0xf9c66bd3, 0xb2af31a4, 0xbfa43fad, 0xa8b92db6, 0xa5b223bf, 0x86830980, 0x8b880789, 0x9c951592, 0x919e1b9b, 0x0a47a17c, 0x074caf75, 0x1051bd6e, 0x1d5ab367, 0x3e6b9958, 0x33609751, 0x247d854a, 0x29768b43, 0x621fd134, 0x6f14df3d, 0x7809cd26, 0x7502c32f, 0x5633e910, 0x5b38e719, 0x4c25f502, 0x412efb0b, 0x618c9ad7, 0x6c8794de, 0x7b9a86c5, 0x769188cc, 0x55a0a2f3, 0x58abacfa, 0x4fb6bee1, 0x42bdb0e8, 0x09d4ea9f, 0x04dfe496, 0x13c2f68d, 0x1ec9f884, 0x3df8d2bb, 0x30f3dcb2, 0x27eecea9, 0x2ae5c0a0, 0xb13c7a47, 0xbc37744e, 0xab2a6655, 0xa621685c, 0x85104263, 0x881b4c6a, 0x9f065e71, 0x920d5078, 0xd9640a0f, 0xd46f0406, 0xc372161d, 0xce791814, 0xed48322b, 0xe0433c22, 0xf75e2e39, 0xfa552030, 0xb701ec9a, 0xba0ae293, 0xad17f088, 0xa01cfe81, 0x832dd4be, 0x8e26dab7, 0x993bc8ac, 0x9430c6a5, 0xdf599cd2, 0xd25292db, 0xc54f80c0, 0xc8448ec9, 0xeb75a4f6, 0xe67eaaff, 0xf163b8e4, 0xfc68b6ed, 0x67b10c0a, 0x6aba0203, 0x7da71018, 0x70ac1e11, 0x539d342e, 0x5e963a27, 0x498b283c, 0x44802635, 0x0fe97c42, 0x02e2724b, 0x15ff6050, 0x18f46e59, 0x3bc54466, 0x36ce4a6f, 0x21d35874, 0x2cd8567d, 0x0c7a37a1, 0x017139a8,

```
0x166c2bb3, 0x1b6725ba, 0x38560f85, 0x355d018c, 0x22401397, 0x2f4b1d9e,
0x642247e9, 0x692949e0, 0x7e345bfb, 0x733f55f2, 0x500e7fcd, 0x5d0571c4,
0x4a1863df, 0x47136dd6, 0xdccad731, 0xd1c1d938, 0xc6dccb23, 0xcbd7c52a,
0xe8e6ef15, 0xe5ede11c, 0xf2f0f307, 0xfffbfd0e, 0xb492a779, 0xb999a970, 0xae84bb6b,
0xa38fb562, 0x80be9f5d, 0x8db59154, 0x9aa8834f, 0x97a38d46];
    var U4 = [0x00000000, 0x090d0b0e, 0x121a161c, 0x1b171d12, 0x24342c38,
0x2d392736, 0x362e3a24, 0x3f23312a, 0x48685870, 0x4165537e, 0x5a724e6c,
0x537f4562, 0x6c5c7448, 0x65517f46, 0x7e466254, 0x774b695a, 0x90d0b0e0,
0x99ddbbee, 0x82caa6fc, 0x8bc7adf2, 0xb4e49cd8, 0xbde997d6, 0xa6fe8ac4, 0xaff381ca,
0xd8b8e890, 0xd1b5e39e, 0xcaa2fe8c, 0xc3aff582, 0xfc8cc4a8, 0xf581cfa6, 0xee96d2b4,
0xe79bd9ba, 0x3bbb7bdb, 0x32b670d5, 0x29a16dc7, 0x20ac66c9, 0x1f8f57e3,
0x16825ced, 0x0d9541ff, 0x04984af1, 0x73d323ab, 0x7ade28a5, 0x61c935b7,
0x68c43eb9, 0x57e70f93, 0x5eea049d, 0x45fd198f, 0x4cf01281, 0xab6bcb3b,
0xa266c035, 0xb971dd27, 0xb07cd629, 0x8f5fe703, 0x8652ec0d, 0x9d45f11f,
0x9448fa11, 0xe303934b, 0xea0e9845, 0xf1198557, 0xf8148e59, 0xc737bf73,
0xce3ab47d, 0xd52da96f, 0xdc20a261, 0x766df6ad, 0x7f60fda3, 0x6477e0b1,
0x6d7aebbf, 0x5259da95, 0x5b54d19b, 0x4043cc89, 0x494ec787, 0x3e05aedd,
0x3708a5d3, 0x2c1fb8c1, 0x2512b3cf, 0x1a3182e5, 0x133c89eb, 0x082b94f9,
0x01269ff7, 0xe6bd464d, 0xefb04d43, 0xf4a75051, 0xfdaa5b5f, 0xc2896a75,
0xcb84617b, 0xd0937c69, 0xd99e7767, 0xaed51e3d, 0xa7d81533, 0xbccf0821,
0xb5c2032f, 0x8ae13205, 0x83ec390b, 0x98fb2419, 0x91f62f17, 0x4dd68d76,
0x44db8678, 0x5fcc9b6a, 0x56c19064, 0x69e2a14e, 0x60efaa40, 0x7bf8b752,
0x72f5bc5c, 0x05bed506, 0x0cb3de08, 0x17a4c31a, 0x1ea9c814, 0x218af93e,
0x2887f230, 0x3390ef22, 0x3a9de42c, 0xdd063d96, 0xd40b3698, 0xcf1c2b8a,
0xc6112084, 0xf93211ae, 0xf03f1aa0, 0xeb2807b2, 0xe2250cbc, 0x956e65e6,
0x9c636ee8, 0x877473fa, 0x8e7978f4, 0xb15a49de, 0xb85742d0, 0xa3405fc2,
0xaa4d54cc, 0xecdaf741, 0xe5d7fc4f, 0xfec0e15d, 0xf7cdea53, 0xc8eedb79, 0xc1e3d077,
0xdaf4cd65, 0xd3f9c66b, 0xa4b2af31, 0xadbfa43f, 0xb6a8b92d, 0xbfa5b223,
0x80868309, 0x898b8807, 0x929c9515, 0x9b919e1b, 0x7c0a47a1, 0x75074caf,
0x6e1051bd, 0x671d5ab3, 0x583e6b99, 0x51336097, 0x4a247d85, 0x4329768b,
0x34621fd1, 0x3d6f14df, 0x267809cd, 0x2f7502c3, 0x105633e9, 0x195b38e7,
0x024c25f5, 0x0b412efb, 0xd7618c9a, 0xde6c8794, 0xc57b9a86, 0xcc769188,
0xf355a0a2, 0xfa58abac, 0xe14fb6be, 0xe842bdb0, 0x9f09d4ea, 0x9604dfe4, 0x8d13c2f6,
```

```
0x841ec9f8, 0xbb3df8d2, 0xb230f3dc, 0xa927eece, 0xa02ae5c0, 0x47b13c7a,
0x4ebc3774, 0x55ab2a66, 0x5ca62168, 0x63851042, 0x6a881b4c, 0x719f065e,
0x78920d50, 0x0fd9640a, 0x06d46f04, 0x1dc37216, 0x14ce7918, 0x2bed4832,
0x22e0433c, 0x39f75e2e, 0x30fa5520, 0x9ab701ec, 0x93ba0ae2, 0x88ad17f0,
0x81a01cfe, 0xbe832dd4, 0xb78e26da, 0xac993bc8, 0xa59430c6, 0xd2df599c,
0xdbd25292, 0xc0c54f80, 0xc9c8448e, 0xf6eb75a4, 0xffe67eaa, 0xe4f163b8, 0xedfc68b6,
0x0a67b10c, 0x036aba02, 0x187da710, 0x1170ac1e, 0x2e539d34, 0x275e963a,
0x3c498b28, 0x35448026, 0x420fe97c, 0x4b02e272, 0x5015ff60, 0x5918f46e,
0x663bc544, 0x6f36ce4a, 0x7421d358, 0x7d2cd856, 0xa10c7a37, 0xa8017139,
0xb3166c2b, 0xba1b6725, 0x8538560f, 0x8c355d01, 0x97224013, 0x9e2f4b1d,
0xe9642247, 0xe0692949, 0xfb7e345b, 0xf2733f55, 0xcd500e7f, 0xc45d0571,
0xdf4a1863, 0xd647136d, 0x31dccad7, 0x38d1c1d9, 0x23c6dccb, 0x2acbd7c5,
0x15e8e6ef, 0x1ce5ede1, 0x07f2f0f3, 0x0efffbfd, 0x79b492a7, 0x70b999a9, 0x6bae84bb,
0x62a38fb5, 0x5d80be9f, 0x548db591, 0x4f9aa883, 0x4697a38d];

  function convertToInt32(bytes) {
    var result = [];
    for (var i = 0; i < bytes.length; i += 4) {
      result.push(
        (bytes[i    ] << 24) |
        (bytes[i + 1] << 16) |
        (bytes[i + 2] <<  8) |
         bytes[i + 3]
      );
    }
    return result;
  }

  var AES = function(key) {
    if (!(this instanceof AES)) {
      throw Error('AES must be instanitated with `new`');
    }
```

```
    Object.defineProperty(this, 'key', {
        value: coerceArray(key, true)
    });

    this._prepare();
}


AES.prototype._prepare = function() {

    var rounds = numberOfRounds[this.key.length];
    if (rounds == null) {
        throw new Error('invalid key size (must be 16, 24 or 32 bytes)');
    }

    // encryption round keys
    this._Ke = [];

    // decryption round keys
    this._Kd = [];

    for (var i = 0; i <= rounds; i++) {
        this._Ke.push([0, 0, 0, 0]);
        this._Kd.push([0, 0, 0, 0]);
    }

    var roundKeyCount = (rounds + 1) * 4;
    var KC = this.key.length / 4;

    // convert the key into ints
    var tk = convertToInt32(this.key);

    // copy values into round key arrays
```

```
    var index;
    for (var i = 0; i < KC; i++) {
        index = i >> 2;
        this._Ke[index][i % 4] = tk[i];
        this._Kd[rounds - index][i % 4] = tk[i];
    }

    // key expansion (fips-197 section 5.2)
    var rconpointer = 0;
    var t = KC, tt;
    while (t < roundKeyCount) {
        tt = tk[KC - 1];
        tk[0] ^= ((S[(tt >> 16) & 0xFF] << 24) ^
                  (S[(tt >>  8) & 0xFF] << 16) ^
                  (S[ tt        & 0xFF] <<  8) ^
                   S[(tt >> 24) & 0xFF]       ^
                  (rcon[rconpointer] << 24));
        rconpointer += 1;

        // key expansion (for non-256 bit)
        if (KC != 8) {
            for (var i = 1; i < KC; i++) {
                tk[i] ^= tk[i - 1];
            }

        // key expansion for 256-bit keys is "slightly different" (fips-197)
        } else {
            for (var i = 1; i < (KC / 2); i++) {
                tk[i] ^= tk[i - 1];
            }
            tt = tk[(KC / 2) - 1];

            tk[KC / 2] ^= (S[ tt        & 0xFF]       ^
```

```
                    (S[(tt >>  8) & 0xFF] <<  8) ^
                    (S[(tt >> 16) & 0xFF] << 16) ^
                    (S[(tt >> 24) & 0xFF] << 24));

        for (var i = (KC / 2) + 1; i < KC; i++) {
            tk[i] ^= tk[i - 1];
        }
    }

    // copy values into round key arrays
    var i = 0, r, c;
    while (i < KC && t < roundKeyCount) {
        r = t >> 2;
        c = t % 4;
        this._Ke[r][c] = tk[i];
        this._Kd[rounds - r][c] = tk[i++];
        t++;
    }
}

// inverse-cipher-ify the decryption round key (fips-197 section 5.3)
for (var r = 1; r < rounds; r++) {
    for (var c = 0; c < 4; c++) {
        tt = this._Kd[r][c];
        this._Kd[r][c] = (U1[(tt >> 24) & 0xFF] ^
                          U2[(tt >> 16) & 0xFF] ^
                          U3[(tt >>  8) & 0xFF] ^
                          U4[ tt        & 0xFF]);
    }
}
}

AES.prototype.encrypt = function(plaintext) {
```

```
if (plaintext.length != 16) {
    throw new Error('invalid plaintext size (must be 16 bytes)');
}

var rounds = this._Ke.length - 1;
var a = [0, 0, 0, 0];

// convert plaintext to (ints ^ key)
var t = convertToInt32(plaintext);
for (var i = 0; i < 4; i++) {
    t[i] ^= this._Ke[0][i];
}

// apply round transforms
for (var r = 1; r < rounds; r++) {
    for (var i = 0; i < 4; i++) {
        a[i] = (T1[(t[ i         ] >> 24) & 0xff] ^
                T2[(t[(i + 1) % 4] >> 16) & 0xff] ^
                T3[(t[(i + 2) % 4] >>  8) & 0xff] ^
                T4[ t[(i + 3) % 4]        & 0xff] ^
                this._Ke[r][i]);
    }
    t = a.slice();
}

// the last round is special
var result = createArray(16), tt;
for (var i = 0; i < 4; i++) {
    tt = this._Ke[rounds][i];
    result[4 * i    ] = (S[(t[ i         ] >> 24) & 0xff] ^ (tt >> 24)) & 0xff;
    result[4 * i + 1] = (S[(t[(i + 1) % 4] >> 16) & 0xff] ^ (tt >> 16)) & 0xff;
    result[4 * i + 2] = (S[(t[(i + 2) % 4] >>  8) & 0xff] ^ (tt >>  8)) & 0xff;
    result[4 * i + 3] = (S[ t[(i + 3) % 4]        & 0xff] ^  tt       ) & 0xff;
```

```
    }

    return result;
}

AES.prototype.decrypt = function(ciphertext) {
    if (ciphertext.length != 16) {
        throw new Error('invalid ciphertext size (must be 16 bytes)');
    }

    var rounds = this._Kd.length - 1;
    var a = [0, 0, 0, 0];

    // convert plaintext to (ints ^ key)
    var t = convertToInt32(ciphertext);
    for (var i = 0; i < 4; i++) {
        t[i] ^= this._Kd[0][i];
    }

    // apply round transforms
    for (var r = 1; r < rounds; r++) {
        for (var i = 0; i < 4; i++) {
            a[i] = (T5[(t[ i          ] >> 24) & 0xff] ^
                    T6[(t[(i + 3) % 4] >> 16) & 0xff] ^
                    T7[(t[(i + 2) % 4] >>  8) & 0xff] ^
                    T8[ t[(i + 1) % 4]        & 0xff] ^
                    this._Kd[r][i]);
        }
        t = a.slice();
    }

    // the last round is special
    var result = createArray(16), tt;
```

```
    for (var i = 0; i < 4; i++) {
        tt = this._Kd[rounds][i];
        result[4 * i    ] = (Si[(t[ i         ] >> 24) & 0xff] ^ (tt >> 24)) & 0xff;
        result[4 * i + 1] = (Si[(t[(i + 3) % 4] >> 16) & 0xff] ^ (tt >> 16)) & 0xff;
        result[4 * i + 2] = (Si[(t[(i + 2) % 4] >>  8) & 0xff] ^ (tt >>  8)) & 0xff;
        result[4 * i + 3] = (Si[ t[(i + 1) % 4]        & 0xff] ^  tt       ) & 0xff;
    }

    return result;
}



/**
 *  Mode Of Operation - Electonic Codebook (ECB)
 */
var ModeOfOperationECB = function(key) {
    if (!(this instanceof ModeOfOperationECB)) {
        throw Error('AES must be instanitated with `new`');
    }

    this.description = "Electronic Code Block";
    this.name = "ecb";

    this._aes = new AES(key);
}

ModeOfOperationECB.prototype.encrypt = function(plaintext) {
    plaintext = coerceArray(plaintext);

    if ((plaintext.length % 16) !== 0) {
        throw new Error('invalid plaintext size (must be multiple of 16 bytes)');
    }
```

```
    var ciphertext = createArray(plaintext.length);
    var block = createArray(16);

    for (var i = 0; i < plaintext.length; i += 16) {
        copyArray(plaintext, block, 0, i, i + 16);
        block = this._aes.encrypt(block);
        copyArray(block, ciphertext, i);
    }

    return ciphertext;
}

ModeOfOperationECB.prototype.decrypt = function(ciphertext) {
    ciphertext = coerceArray(ciphertext);

    if ((ciphertext.length % 16) !== 0) {
        throw new Error('invalid ciphertext size (must be multiple of 16 bytes)');
    }

    var plaintext = createArray(ciphertext.length);
    var block = createArray(16);

    for (var i = 0; i < ciphertext.length; i += 16) {
        copyArray(ciphertext, block, 0, i, i + 16);
        block = this._aes.decrypt(block);
        copyArray(block, plaintext, i);
    }

    return plaintext;
}


/**
```

```
 *  Mode Of Operation - Cipher Block Chaining (CBC)
 */
var ModeOfOperationCBC = function(key, iv) {
    if (!(this instanceof ModeOfOperationCBC)) {
        throw Error('AES must be instanitated with `new`');
    }

    this.description = "Cipher Block Chaining";
    this.name = "cbc";

    if (!iv) {
        iv = createArray(16);

    } else if (iv.length != 16) {
        throw new Error('invalid initialation vector size (must be 16 bytes)');
    }

    this._lastCipherblock = coerceArray(iv, true);

    this._aes = new AES(key);
}

ModeOfOperationCBC.prototype.encrypt = function(plaintext) {
    plaintext = coerceArray(plaintext);

    if ((plaintext.length % 16) !== 0) {
        throw new Error('invalid plaintext size (must be multiple of 16 bytes)');
    }

    var ciphertext = createArray(plaintext.length);
    var block = createArray(16);

    for (var i = 0; i < plaintext.length; i += 16) {
```

```
        copyArray(plaintext, block, 0, i, i + 16);

        for (var j = 0; j < 16; j++) {
            block[j] ^= this._lastCipherblock[j];
        }

        this._lastCipherblock = this._aes.encrypt(block);
        copyArray(this._lastCipherblock, ciphertext, i);
    }

    return ciphertext;
}

ModeOfOperationCBC.prototype.decrypt = function(ciphertext) {
    ciphertext = coerceArray(ciphertext);

    if ((ciphertext.length % 16) !== 0) {
        throw new Error('invalid ciphertext size (must be multiple of 16 bytes)');
    }

    var plaintext = createArray(ciphertext.length);
    var block = createArray(16);

    for (var i = 0; i < ciphertext.length; i += 16) {
        copyArray(ciphertext, block, 0, i, i + 16);
        block = this._aes.decrypt(block);

        for (var j = 0; j < 16; j++) {
            plaintext[i + j] = block[j] ^ this._lastCipherblock[j];
        }

        copyArray(ciphertext, this._lastCipherblock, 0, i, i + 16);
    }
```

```
    return plaintext;
}



/**
 *  Mode Of Operation - Cipher Feedback (CFB)
 */
var ModeOfOperationCFB = function(key, iv, segmentSize) {
    if (!(this instanceof ModeOfOperationCFB)) {
        throw Error('AES must be instanitated with `new`');
    }

    this.description = "Cipher Feedback";
    this.name = "cfb";

    if (!iv) {
        iv = createArray(16);

    } else if (iv.length != 16) {
        throw new Error('invalid initialation vector size (must be 16 size)');
    }

    if (!segmentSize) { segmentSize = 1; }

    this.segmentSize = segmentSize;

    this._shiftRegister = coerceArray(iv, true);

    this._aes = new AES(key);
}

ModeOfOperationCFB.prototype.encrypt = function(plaintext) {
```

```
        if ((plaintext.length % this.segmentSize) != 0) {
            throw new Error('invalid plaintext size (must be segmentSize bytes)');
        }

        var encrypted = coerceArray(plaintext, true);

        var xorSegment;
        for (var i = 0; i < encrypted.length; i += this.segmentSize) {
            xorSegment = this._aes.encrypt(this._shiftRegister);
            for (var j = 0; j < this.segmentSize; j++) {
                encrypted[i + j] ^= xorSegment[j];
            }

            // Shift the register
            copyArray(this._shiftRegister, this._shiftRegister, 0, this.segmentSize);
            copyArray(encrypted, this._shiftRegister, 16 - this.segmentSize, i, i +
this.segmentSize);
        }

        return encrypted;
    }

    ModeOfOperationCFB.prototype.decrypt = function(ciphertext) {
        if ((ciphertext.length % this.segmentSize) != 0) {
            throw new Error('invalid ciphertext size (must be segmentSize bytes)');
        }

        var plaintext = coerceArray(ciphertext, true);

        var xorSegment;
        for (var i = 0; i < plaintext.length; i += this.segmentSize) {
            xorSegment = this._aes.encrypt(this._shiftRegister);
```

```
            for (var j = 0; j < this.segmentSize; j++) {
                plaintext[i + j] ^= xorSegment[j];
            }

            // Shift the register
            copyArray(this._shiftRegister, this._shiftRegister, 0, this.segmentSize);
            copyArray(ciphertext, this._shiftRegister, 16 - this.segmentSize, i, i +
this.segmentSize);
        }

        return plaintext;
    }

    /**
     *  Mode Of Operation - Output Feedback (OFB)
     */
    var ModeOfOperationOFB = function(key, iv) {
        if (!(this instanceof ModeOfOperationOFB)) {
            throw Error('AES must be instanitated with `new`');
        }

        this.description = "Output Feedback";
        this.name = "ofb";

        if (!iv) {
            iv = createArray(16);

        } else if (iv.length != 16) {
            throw new Error('invalid initialation vector size (must be 16 bytes)');
        }

        this._lastPrecipher = coerceArray(iv, true);
        this._lastPrecipherIndex = 16;
```

```
    this._aes = new AES(key);
}

ModeOfOperationOFB.prototype.encrypt = function(plaintext) {
    var encrypted = coerceArray(plaintext, true);

    for (var i = 0; i < encrypted.length; i++) {
        if (this._lastPrecipherIndex === 16) {
            this._lastPrecipher = this._aes.encrypt(this._lastPrecipher);
            this._lastPrecipherIndex = 0;
        }
        encrypted[i] ^= this._lastPrecipher[this._lastPrecipherIndex++];
    }

    return encrypted;
}

// Decryption is symetric
ModeOfOperationOFB.prototype.decrypt = ModeOfOperationOFB.prototype.encrypt;



/**
 *  Counter object for CTR common mode of operation
 */
var Counter = function(initialValue) {
    if (!(this instanceof Counter)) {
        throw Error('Counter must be instanitated with `new`');
    }

    // We allow 0, but anything false-ish uses the default 1
    if (initialValue !== 0 && !initialValue) { initialValue = 1; }
```

```
    if (typeof(initialValue) === 'number') {
        this._counter = createArray(16);
        this.setValue(initialValue);

    } else {
        this.setBytes(initialValue);
    }
}

Counter.prototype.setValue = function(value) {
    if (typeof(value) !== 'number' || parseInt(value) != value) {
        throw new Error('invalid counter value (must be an integer)');
    }

    // We cannot safely handle numbers beyond the safe range for integers
    if (value > Number.MAX_SAFE_INTEGER) {
        throw new Error('integer value out of safe range');
    }

    for (var index = 15; index >= 0; --index) {
        this._counter[index] = value % 256;
        value = parseInt(value / 256);
    }
}

Counter.prototype.setBytes = function(bytes) {
    bytes = coerceArray(bytes, true);

    if (bytes.length != 16) {
        throw new Error('invalid counter bytes size (must be 16 bytes)');
    }

    this._counter = bytes;
```

```
};

Counter.prototype.increment = function() {
    for (var i = 15; i >= 0; i--) {
        if (this._counter[i] === 255) {
            this._counter[i] = 0;
        } else {
            this._counter[i]++;
            break;
        }
    }
}



/**
 *  Mode Of Operation - Counter (CTR)
 */
var ModeOfOperationCTR = function(key, counter) {
    if (!(this instanceof ModeOfOperationCTR)) {
        throw Error('AES must be instanitated with `new`');
    }

    this.description = "Counter";
    this.name = "ctr";

    if (!(counter instanceof Counter)) {
        counter = new Counter(counter)
    }

    this._counter = counter;

    this._remainingCounter = null;
    this._remainingCounterIndex = 16;
```

```
    this._aes = new AES(key);
}

ModeOfOperationCTR.prototype.encrypt = function(plaintext) {
    var encrypted = coerceArray(plaintext, true);

    for (var i = 0; i < encrypted.length; i++) {
        if (this._remainingCounterIndex === 16) {
            this._remainingCounter = this._aes.encrypt(this._counter._counter);
            this._remainingCounterIndex = 0;
            this._counter.increment();
        }
        encrypted[i] ^= this._remainingCounter[this._remainingCounterIndex++];
    }

    return encrypted;
}

// Decryption is symetric
ModeOfOperationCTR.prototype.decrypt = ModeOfOperationCTR.prototype.encrypt;



//////////////////////
// Padding

// See:https://tools.ietf.org/html/rfc2315
function pkcs7pad(data) {
    data = coerceArray(data, true);
    var padder = 16 - (data.length % 16);
    var result = createArray(data.length + padder);
    copyArray(data, result);
    for (var i = data.length; i < result.length; i++) {
```

```
        result[i] = padder;
    }
    return result;
}

function pkcs7strip(data) {
    data = coerceArray(data, true);
    if (data.length < 16) { throw new Error('PKCS#7 invalid length'); }

    var padder = data[data.length - 1];
    if (padder > 16) { throw new Error('PKCS#7 padding byte out of range'); }

    var length = data.length - padder;
    for (var i = 0; i < padder; i++) {
        if (data[length + i] !== padder) {
            throw new Error('PKCS#7 invalid padding byte');
        }
    }

    var result = createArray(length);
    copyArray(data, result, 0, 0, length);
    return result;
}

///////////////////////
// Exporting


// The block cipher
var aesjs = {
    AES: AES,
    Counter: Counter,
```

```
    ModeOfOperation: {
        ecb: ModeOfOperationECB,
        cbc: ModeOfOperationCBC,
        cfb: ModeOfOperationCFB,
        ofb: ModeOfOperationOFB,
        ctr: ModeOfOperationCTR
    },

    utils: {
        hex: convertHex,
        utf8: convertUtf8
    },

    padding: {
        pkcs7: {
            pad: pkcs7pad,
            strip: pkcs7strip
        }
    },

    _arrayTest: {
        coerceArray: coerceArray,
        createArray: createArray,
        copyArray: copyArray,
    }
};


// node.js
if (typeof exports !== 'undefined') {
    module.exports = aesjs

// RequireJS/AMD
```

```
    // http://www.requirejs.org/docs/api.html
    // https://github.com/amdjs/amdjs-api/wiki/AMD
    } else if (typeof(define) === 'function' && define.amd) {
        define([], function() { return aesjs; });

    // Web Browsers
    } else {

        // If there was an existing library at "aesjs" make sure it's still available
        if (root.aesjs) {
            aesjs._aesjs = root.aesjs;
        }

        root.aesjs = aesjs;
    }


})(this);
```

Decryption Page

From2.html

```
<html>

<head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <script src="https://use.fontawesome.com/releases/v5.0.8/js/all.js"></script>
    <link href="./styles/style123.css" rel="stylesheet">
    <style>
        .logoSection {
            text-align: center;
            margin-top: 50px;
            margin-bottom: 10px;
        }

        .contentOfTab {
            text-align: center;
        }

        #decrypt_share_num {
            font-size: 23px;
            margin-bottom: 15px;
        }
```

```
#userinput {
    margin-bottom: 15px;
}

#password_input {
    margin-bottom: 15px;
}

.btn-success {
    background-color: rgb(193, 205, 35) !important;
    border-color: rgb(193, 205, 35) !important;
    outline: none !important;
    box-shadow: none !important;
}

.btn-success:hover,
.btn-success:active,
.btn-success:visited,
.btn-success:focus {
    background-color: #232322 !important;
    border-color: #232322 !important;
    outline: none !important;
    box-shadow: none !important;
}

.btn-dark {
    background-color: #232322 !important;
    border-color: #232322 !important;
    outline: none !important;
    box-shadow: none !important;
}

.btn-dark:hover,
```

```
        .btn-dark:active,
        .btn-dark:visited,
        .btn-dark:focus {
            background-color: rgb(193, 205, 35) !important;
            border-color: rgb(193, 205, 35) !important;
            outline: none !important;
            box-shadow: none !important;
        }
    </style>
</head>

<body>

    <div class="tabs">
        <input type="radio" name="tabs" id="tabone" checked="checked">

        <label class="main-label">Step 1</label>
        <div class="tab">
            <div class="contentOfTab">

                <h2>Amount of beneficiary keys</h2>
                <select id="decrypt_share_num">
                    <option value="2">2</option>
                    <option value="3">3</option>
                    <option value="4">4</option>
                    <option value="5">5</option>
                    <option value="6">6</option>
                    <option value="7">7</option>
                    <option value="8">8</option>
                    <option value="9">9</option>
                    <option value="10">10</option>
                    <option value="11">11</option>
                    <option value="12">12</option>
```

```
            <option value="13">13</option>
            <option value="14">14</option>
            <option value="15">15</option>
            <option value="16">16</option>
            <option value="17">17</option>
            <option value="18">18</option>
            <option value="19">19</option>
            <option value="20">20</option>
        </select>
        <br>
        <button class="btn btn-success" onclick="decryptAmount()">Next</button>

    </div>
</div>
<input type="radio" name="tabs" id="tabtwo">

<label class="main-label">Step 2</label>
<div class="tab">
    <div class="contentOfTab">
        <h2>Enter Beneficiary Keys</h2>
        <form action="javascript:decrypt()" id="demo2">

        </form>

    </div>
</div>
<input type="radio" name="tabs" id="tabthree">

<label class="main-label">Step 3</label>
<div class="tab">
    <div class="contentOfTab">
        <h2>Enter the password of the inheritance system</h2>
```

```
            <form action="javascript:Password2()" method="get">
                <input type="text" maxlength="32" id="password_input2"
name="password_input2">
                <br>
                <br>

                <button type="button" class="back2 btn btn-dark">Back</button>
                <input type="submit" class="btn btn-success" value="Finish">
            </form>

            <div id="status"></div>

        </div>
    </div>
    <input type="radio" name="tabs" id="tabfour">
    <label class="main-label">Finished</label>
    <div class="tab">
        <div class="contentOfTab">
            <h2>Decrypted Data</h2>

            <p id="informationDecrypted"></p>

            <button class="btn btn-dark"
onClick="window.location.reload();">Restart</button>
        </div>
    </div>
  </div>

  <div class="logoSection" style="height:105px;">
    <h1>Project</h1>
    <p class="mt-5 mb-3 text-muted">© 2019-2020</p>
  </div>
```

```
<script src="./scripts/secrets.js"></script>
<script src="./scripts/aes.js"></script>

<script src="./scripts/SHA512.js"></script>
<script src="./scripts/PBKDF2.js"></script>

<script>
    var decryptionShares = 0;
    var pbkdfPassword = "";
    var userInputToBeEncrypted;
    var encryptedHex;

    function removeElement(elementId) {
        // Removes an element from the document
        console.log(elementId);
        var element = document.getElementById(elementId);
        element.parentNode.removeChild(element);
    }

    var decryptionShares = 0;

    function decryptAmount() {

        for (i = 1; i <= decryptionShares; i++) {

            var elementNameToDelete = "decrypt_key" + i;
            removeElement(elementNameToDelete);

            if (i === decryptionShares) {
                elementNameToDelete = "keyInputSubmit";
                removeElement(elementNameToDelete);
                elementNameToDelete = "buttonBack2";
                removeElement(elementNameToDelete);
```

```
                elementNameToDelete = "removeBr";
                removeElement(elementNameToDelete);
            } else if (i < decryptionShares) {
                elementNameToDelete = "removeBr";
                removeElement(elementNameToDelete);
                elementNameToDelete = "removeBr";
                removeElement(elementNameToDelete);
            }
        }

        decryptionShares = document.getElementById("decrypt_share_num").value;

        decryptionShares = parseInt(decryptionShares);

        for (i = 1; i <= decryptionShares; i++) {

            if (i < decryptionShares) {
                console.log("works");
                document.getElementById("demo2").innerHTML += "<input type='text'
id='decrypt_key" + i + "' name='fname' placeholder='Key " + i + "' required><br
id='removeBr'><br id='removeBr'>";
            } else {
                console.log("works");
                document.getElementById("demo2").innerHTML += "<input type='text'
id='decrypt_key" + i + "' name='fname' placeholder='Key " + i + "' required><br
id='removeBr'><div class=btn-toolbar'><button type='button' id='buttonBack2'
class='back1 btn btn-dark mr-1' style='margin-top:15px;'>Back</button><button
type='submit' id='keyInputSubmit' class='btn btn-success' style='margin-
top:15px;'>Next</button>";
            }
        }

        $("#tabtwo").prop("checked", true);
```

```
    }

    function decrypt() {

        var allDecryptKeysArray = [];

        for (i = 1; i <= decryptionShares; i++) {

            var getKey = document.getElementById("decrypt_key" + i).value;

            allDecryptKeysArray.push(getKey);

        }

        switch (decryptionShares) {
            case 2:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1]]);
                break;
            case 3:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2]]);
                break;
            case 4:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3]]);
                break;
            case 5:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4]]);
                break;
            case 6:
```

```
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5]]);
                break;
            case 7:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6]]);
                break;
            case 8:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7]]);
                break;
            case 9:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8]]);
                break;
            case 10:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9]]);
                break;
            case 11:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10]]);
                break;
            case 12:
```

```
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
allDecryptKeysArray[11]]);
                break;
            case 13:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
allDecryptKeysArray[11], allDecryptKeysArray[12]]);
                break;
            case 14:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
allDecryptKeysArray[11], allDecryptKeysArray[12], allDecryptKeysArray[13]]);
                break;
            case 15:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
allDecryptKeysArray[11], allDecryptKeysArray[12], allDecryptKeysArray[13],
allDecryptKeysArray[14]]);
                break;
            case 16:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
```

```
allDecryptKeysArray[11], allDecryptKeysArray[12], allDecryptKeysArray[13],
allDecryptKeysArray[14], allDecryptKeysArray[15]]);
                break;
            case 17:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
allDecryptKeysArray[11], allDecryptKeysArray[12], allDecryptKeysArray[13],
allDecryptKeysArray[14], allDecryptKeysArray[15], allDecryptKeysArray[16]]);
                break;
            case 18:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
allDecryptKeysArray[11], allDecryptKeysArray[12], allDecryptKeysArray[13],
allDecryptKeysArray[14], allDecryptKeysArray[15], allDecryptKeysArray[16],
allDecryptKeysArray[17]]);
                break;
            case 19:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
allDecryptKeysArray[11], allDecryptKeysArray[12], allDecryptKeysArray[13],
allDecryptKeysArray[14], allDecryptKeysArray[15], allDecryptKeysArray[16],
allDecryptKeysArray[17], allDecryptKeysArray[18]]);
                break;
            case 20:
                comb = secrets.combine([allDecryptKeysArray[0], allDecryptKeysArray[1],
allDecryptKeysArray[2], allDecryptKeysArray[3], allDecryptKeysArray[4],
allDecryptKeysArray[5], allDecryptKeysArray[6], allDecryptKeysArray[7],
```

```
allDecryptKeysArray[8], allDecryptKeysArray[9], allDecryptKeysArray[10],
allDecryptKeysArray[11], allDecryptKeysArray[12], allDecryptKeysArray[13],
allDecryptKeysArray[14], allDecryptKeysArray[15], allDecryptKeysArray[16],
allDecryptKeysArray[17], allDecryptKeysArray[18], allDecryptKeysArray[19]]);
                break;
            default:
                console.log("Out of bounds");
        }

        //convert back to UTF string:
        comb = secrets.hex2str(comb)

        console.log(comb);

        encryptedHex = comb;

        ////////////////////////////////////////////////////////////////////////

        $("#tabthree").prop("checked", true);

    }

    function Password2() {
        var passwordnew = document.getElementById("password_input2").value;

        var mypbkdf2 = new PBKDF2(passwordnew, "B17H357", 10000, 16);

        var status_callback = function(percent_done) {
            document.getElementById("status").innerHTML = "Computed " +
percent_done.toFixed(2) + "%"
        };
        var result_callback = function(key) {
```

```
        //Add button in here to move to next section
        document.getElementById("status").innerHTML = "The derived key is: " + key;
        pbkdfPassword = key;
        aesDecrypt();
    };

    mypbkdf2.deriveKey(status_callback, result_callback);

}

function aesDecrypt() {

    var pbkdfPasswordArray = pbkdfPassword.split("");

    console.log(pbkdfPasswordArray);

    //Convert character to unit8 num without affecting existing num
    for (var i = 0, len = pbkdfPasswordArray.length; i < len; i++) {

        if (!(pbkdfPasswordArray[i] in ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"])) {
            var currentPasswordArrayLetter = pbkdfPasswordArray[i];
            currentPasswordArrayLetter.charCodeAt(0);
            var letterToNum = currentPasswordArrayLetter.charCodeAt(0);;
            pbkdfPasswordArray[i] = letterToNum;
        }

    }
    console.log(pbkdfPasswordArray);
    var key = new Uint8Array(pbkdfPasswordArray);

    var encryptedBytes = aesjs.utils.hex.toBytes(encryptedHex);

    // The counter mode of operation maintains internal state, so to
```

```
        // decrypt a new instance must be instantiated.
        var aesCtr = new aesjs.ModeOfOperation.ctr(key, new aesjs.Counter(5));
        var decryptedBytes = aesCtr.decrypt(encryptedBytes);

        // Convert our bytes back into text
        var decryptedText = aesjs.utils.utf8.fromBytes(decryptedBytes);
        console.log(decryptedText);

        document.getElementById("informationDecrypted").innerHTML = decryptedText;

        finished()
    }

    function finished() {
        $("#tabfour").prop("checked", true);

    }

    ////////////////////////////////////////////////////////////////////////////
    $(document).ready(function() {
        $(".back1").click(function() {
            $("#tabone").prop("checked", true);
        });
        $(demo2).on("click", ".back1", function() {
            $("#tabone").prop("checked", true);
        });
        $(".back2").click(function() {
            $("#tabtwo").prop("checked", true);
        });
    });
  </script>
</body>
</html>
```

Style123.css, Secrets.js, Secrets.min.js, SHA512.js, PBKDF2.js, Aes.js

Code for these modules can already be found in the encryption section, however, they are also used in the decryption section as well.