```
//zy6 js encode;
//初始变换序列，可以更改;
var zy6_BLOCK_SIZE =  (384) / 8;
var m_zy64 = new Array(63, 0, 34, 17, 58, 23, 16, 2, 59, 55, 56, 7, 47, 61, 8, 4,
38, 25, 48, 3, 37, 41, 1, 32, 39, 57, 33, 30, 18, 45, 14, 28, 15, 60, 5, 40, 43,
53, 10, 20, 49, 35, 62, 31, 6, 24, 22, 26, 46, 29, 36, 9, 11, 52, 44, 13, 27, 54,
19, 50, 51, 12, 42, 21);

//初始化;
function init_sy(zy_st) {
    zy_st = new Object();
    zy_st.m_tot_len = 0;
    zy_st.m_len = 0;
    zy_st.m_zy_c_64 = new Array(64);
    zy_st.m_block = new Array(2 * zy6_BLOCK_SIZE);
    zy_st.m_zy_p = new Array(64);
    for (var i = 0; i < 64; i++) {
        zy_st.m_zy_c_64[i] = m_zy64[i];
    }
    return zy_st;
}
//复制字符串;
function memcpy(s, k, d, m, i) {
    for (var j = 0; j < i; j++) {
        s[j + k] = d[j + m];
    }

//实现 char 字符到 6 bit 的转换;
function To64(c, d) {
    for (var i = 0; i < zy6_BLOCK_SIZE; i++) {
        if (typeof c[i] == 'string') {
            c[i] = c[i].charCodeAt(0);
        }

    for (var i = 0; i < zy6_BLOCK_SIZE/3; i++) {
        var t = i * 4;
        var m = i * 3;
        d[t] = c[m] >> 2;
        d[t + 1] = ((c[m] & 0x3) << 4) + (c[m + 1] >> 4);
        d[t + 2] = ((c[m + 1] & 0xf) << 2) + (c[m + 2] >> 6);
        d[t + 3] = c[m + 2] & 0x3f;
    }
    return d;
    }
function update( z_st,message, len);
{
```

```
    var block_nb
    var new_len, rem_len, tmp_len
    tmp_len = zy6_BLOCK_SIZE - z_st.m_len
    rem_len = len < tmp_len ? len : tmp_len

    memcpy(z_st.m_block, z_st.m_len, message, 0, rem_len)
    if (z_st.m_len + len < zy6_BLOCK_SIZE) {
        z_st.m_len += len;
        return;
    }
    new_len = len - rem_len
    block_nb = parseInt(new_len / zy6_BLOCK_SIZE)
    var shifted_message = new Array(len - rem_len)
    memcpy(shifted_message, 0, message, rem_len,len-rem_len)
    transform(z_st,z_st.m_block, 1)
    transform(z_st,shifted_message, block_nb)
    rem_len = new_len % zy6_BLOCK_SIZE

    memcpy(z_st.m_block, 0, shifted_message, block_nb * zy6_BLOCK_SIZE, rem_len)
    z_st.m_len = rem_len
    z_st.m_tot_len += (block_nb + 1) * zy6_BLOCK_SIZE
};
function final( z_st);
{
    var block_nb
    var pm_len
    block_nb = (1 + ((zy6_BLOCK_SIZE - 9) < (z_st.m_len % zy6_BLOCK_SIZE)))
    pm_len = block_nb * zy6_BLOCK_SIZE

    for (var i = z_st.m_len; i < zy6_BLOCK_SIZE * 2; i++)
    {
            z_st.m_block[i] = 0
    }
    z_st.m_block[z_st.m_len] = 0x80
    transform(z_st, z_st.m_block, block_nb)
    var pt = 0

    for (var i = 0; i < 64; i++) {
            z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[i] + z_st.m_zy_c_64[pt]) % 64;
            pt = m_zy64[pt];
            z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[i] + z_st.m_zy_c_64[pt]) % 64;
            pt = m_zy64[pt];
            z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[i] + z_st.m_zy_c_64[pt]) % 64;
            pt = m_zy64[pt];
            z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[i] + z_st.m_zy_c_64[pt]) % 64;
            pt = m_zy64[pt];
```

```
    }
};
function transform(z_st,  message,  block_nb);
{
    var sub_block = new Array(96)
    var pt = 0
    var i
    for (i = 0; i < block_nb; i++) {
        memcpy(sub_block, 0, message, i * zy6_BLOCK_SIZE, zy6_BLOCK_SIZE);
        var d = new Array(4*zy6_BLOCK_SIZE/3);
        //组成64卦;
        To64(sub_block, d);
        for (var ti = 0; ti < (zy6_BLOCK_SIZE * 4) / 3; ti++) {
            if (d[ti] > 0) {

                pt = ti;
                {
                    z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[pt] + d[ti]) % 64
                    pt = m_zy64[pt]
                    z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[pt] + d[ti]) % 64
                    pt = m_zy64[pt]
                    z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[pt] + d[ti]) % 64
                    pt = m_zy64[pt]
                    z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[pt] + d[ti]) % 64
                };
                pt = d[ti];
                {
                    z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[pt] + m_zy64[pt]) % 64
                    pt = m_zy64[pt]
                    z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[pt] + m_zy64[pt]) % 64
                    pt = m_zy64[pt]
                    z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[pt] + m_zy64[pt]) % 64
                    pt = m_zy64[pt]
                    z_st.m_zy_c_64[pt] = (z_st.m_zy_c_64[pt] + m_zy64[pt]) % 64
                };
            }
        }


};
//变换;
function Tostring64(m_zy6_st) {

    var letters =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ=-';
    var ss = '';
```

```
    for (var i = 0; i < 64; i++) {
        ss += letters[m_zy6_st.m_zy_c_64[i]];
    }
    return ss;
}
function Tostring64HEX(m_zy6_st) {

    var j = 0;
    for (var i = 0; i < 64 / 4; i = i + 1) {
        m_zy6_st.m_zy_p[j * 3] = (m_zy6_st.m_zy_c_64[i * 4] << 2) +
(m_zy6_st.m_zy_c_64[i * 4 + 1] >> 4);
        m_zy6_st.m_zy_p[j * 3 + 1] = ((m_zy6_st.m_zy_c_64[i * 4 + 1] & 0xf) << 4)
+ (m_zy6_st.m_zy_c_64[i * 4 + 2] >> 2);
        m_zy6_st.m_zy_p[j * 3 + 2] = ((m_zy6_st.m_zy_c_64[i * 4 + 2] & 0x3) << 6)
+ m_zy6_st.m_zy_c_64[i * 4 + 3];
        j = j + 1;
    }

    var letters = '0123456789abcdef';
    var ss = '';
    for (var i = 0; i < 48; i++) {
        var c = m_zy6_st.m_zy_p[i];
        ss += letters[(c >> 4) & 0xf];
        ss += letters[c & 0xf];
    }
    return ss;
}

function Tostring64BG(m_zy6_st) {

    var letters ="䷀䷁䷂䷃䷄䷅䷆䷇䷈䷉䷊䷋䷌䷍䷎䷏䷐䷑䷒䷓䷔䷕䷖䷗䷘䷙䷚䷛䷜䷝䷞䷟䷠䷡䷢䷣䷤䷥䷦䷧䷨䷩䷪䷫䷬䷭䷮䷯䷰䷱䷲䷳䷴䷵䷶䷷䷸䷹䷺䷻䷼䷽䷾䷿";
    var ss = '';
    for (var i = 0; i < 64; i++) {
        ss += letters[m_zy6_st.m_zy_p_64[i]];
    }
    return ss;
}

String.prototype.replaceAll = function (s1, s2) {
    return this.replace(new RegExp(s1, 'gm'), s2);
}
// 将字符串格式化为UTF8编码的字节
var writeUTF = function (String, isGetBytes) {
    var back = [];
    var byteSize = 0;
    for (var i = 0; i < String.length; i++) {
```

```
        var code = String.charCodeAt(i);
        if (0x00 <= code && code <= 0x7f) {
            byteSize += 1;
            back.push(code);
        } else if (0x80 <= code && code <= 0x7ff) {
            byteSize += 2;
            back.push((192 | (31 & (code >> 6))));
            back.push((128 | (63 & code)));
        } else if ((0x800 <= code && code <= 0xd7ff);
                || (0xe000 <= code && code <= 0xffff) {
            byteSize += 3;
            back.push((224 | (15 & (code >> 12))));
            back.push((128 | (63 & (code >> 6))));
            back.push((128 | (63 & code)));
               }

    for (i = 0; i < back.length; i++) {
        back[i] &= 0xff;
        }
    if (isGetBytes) {
        return back;
    }
    if (byteSize <= 0xff) {
        return [0, byteSize].concat(back);
    } else {
        return [byteSize >> 8, byteSize & 0xff].concat(back);
    }

function updatestr(m_zy6_st, String) {
    var qstr = writeUTF(str,1);
    return update(m_zy6_st, qstr, qstr.length);
}
function Zy6String(ss) {
    var zy_st = new Object();
    zy_st = init_sy(zy_st);
    updatestr(zy_st, ss);
    final(zy_st);
    return Tostring64(zy_st);
}
function Zy6StringHEX(ss) {
    var zy_st = new Object();
    zy_st = init_sy(zy_st);
    updatestr(zy_st, ss);
    final(zy_st);
    return Tostring64HEX(zy_st);
}
```

```
function zy6() {
    var ss = document.getElementById('ss').value;
    var dd = document.getElementById('dd');
}
    var zy_st = new Object();
    zy_st = init_sy(zy_st);
    updatestr(zy_st, ss);
    final(zy_st);
    }
    dd.value = Zy6String(ss);
}

function zy6HEX() {
    var ss = document.getElementById('ss').value;
    var dd = document.getElementById('ddHEX');
    var zy_st = new Object();
    zy_st = init_sy(zy_st);
    updatestr(zy_st, ss);
    final(zy_st);
    dd.value = Zy6StringHEX(ss);

}
```

网页 html 源码：

```
<!DOCTYPE html>
<html>
<head>
    <title>泥娃软件--基于路径散列算法</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta name="keywords" content="基于路径散列的消息摘要算法,hash 算法，易经，八卦阵" />
    <meta name="author" content="上海泥娃通信科技有限公司，苏州泥娃软件科技有限公司" />
    <meta name="description" content="基于路径散列的消息摘要方法和系统，消息经过分组，对分组序列进行路径散列计算，循环计算完毕，对消息摘要结果再进行调和路径散列计算，消息摘要计算的结果通过字符串形式输出。分组有利于大容量的信息的处理，消息分组结合路径散列算法，使得运算的结果不可逆，" />
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <link href="style.css" rel="stylesheet" type="text/css" />
    <script type="text/javascript" src="js/cufon-yui.js"></script>
    <meta charset="utf-8" />
    <script src="zy6n.js"></script>
</head>
<body>
    <div class="main">

        <div class="header">
```

```
            <div class="header_resize">
                <div class="logo">
                    <h1><a href="index.html">Newa<span>Soft<label
style="color:greenyellow;vertical-align:super;font-
size:smaller">®</label></span></a></h1>
                </div>      <div class="nav_menu">
                    <ul>
                        <li><a href="index.html">主页</a></li>
                        <li><a href="product.html">产品</a></li>
                        <li><a href="download.html">下载</a></li>
                        <li><a href="support.html">泥娃软件</a></li>
                        <li><a href="about.html">关于我们</a></li>
                        <li class="active"><a href="flcode.html">算法演示
</a></li>
                        <li><a href="contact.html">联系我们</a></li>
                    </ul>
                </div>
                <div class="clr"></div>
            </div>
        </div>

        <div class="hbg">
            <div class="hbg_resize">
                <img src="images/search.png" width="500" height="275" alt="pix"
class="hbgimg" />
                <h2>路径散列算法演示</h2>
                <h3>自主知识产权算法，获得多项国家发明专利</h3>
                <a href="flcode.html">分离码算法</a><br />
                <a href="imgflcode.html">图片信息分离码算法</a><br />
                <a href="zy6.html">zy6 消息摘要算法</a>
            </div>
        </div>

        <div class="content">
            <div class="content_resize">
                <div class="mainbar">
                    <div class="article">
                        <div class="container">

                            输入文字：<input type="text" style="width:800px"
id="ss" /><br />
                            编码结果：<input type="text" style="width:800px"
id="dd" /><br />
                            编码结果（HEX）：<input type="text"
style="width:800px" id="ddHEX" /><br />
```

```
                            <input type="button" id="ss" value="zy6 散列"
onclick="javascript:zy6();" />
                            <input type="button" id="ssHEX" onclick="javascript:
zy6HEX();" value="zy6 散列 HEX" />
                        </div>
                    </div>

                </div>

            </div>
        </div>
        <div class="clr"></div>

        <div class="fbg">
            <div class="fbg_resize">
                <div class="col c1">

                    <div class="clr"></div>
                    <h2>联系我们</h2>
                    <p>
                        <strong>电话:</strong> （86）021-54261835<br />
                        <strong>地址:</strong>上海市闵行区元江路 5500 号第 1 幢
2597 室<br />
                        <strong>地址:</strong>江苏省苏州太仓市宁波东路 66 号 519 室
<br />
                        <strong>Email:</strong> <a
href="mailto:zhangliuxue@126.com">zhangliuxue@126.com</a><br />
                    </p>
                </div>
                <div class="col c2">
                    <h2>算法简介</h2>
                    路径散列的消息摘要方法和系统，可以扩展和衍生不同的摘要算法。
                    路径散列计算结合 k 分组可以实现 k 乘以 2 的 k 次方 bit 的消息摘
要。
                    本例展示的为 6 位分组乘 64 的摘要，简记为 zy6。
                </div>
                <div class="col c3">
                    <h2>特点</h2>
                    <div id="transText">
                       参照周易 64 卦的原理编制，结合 8 卦阵的思想构建算法。
                       路径由 0 到 63 定义，每个路径的节点数字 0 到 7，符号 64 卦，每
卦 8 爻。
                       散列的结果空间为 2 的 384 次方。
                    </div>
                    <div class="clr"></div>
                </div>
```

```
            </div>

            <div class="footer">
                <div class="footer_resize">
                    <p class="lf">
                        &copy; 上海泥娃通信科技有限公司 <br />
                        <a style="color:green" href="https://beian.miit.gov.cn"
target="_new">
                            沪 ICP 备 13005075 号   沪 ICP 备
13005075 号-2
                        </a>
                        <span><img src="/img/gongan.png" />公安备案号：
31011202011025</span>
                        <ul class="fmenu">
                            <li class="active"><a href="index.html">主页</a></li>
                            <li><a href="product.html">产品</a></li>
                            <li><a href="download.html">下载</a></li>
                            <li><a href="support.html">泥娃软件</a></li>
                            <li><a href="flcode.html">算法演示</a></li>
                            <li><a href="about.html">关于我们</a></li>
                            <li><a href="contact.html">联系我们</a></li>
                        </ul>
                    </p>
                </div>
            </div>
        </div>

</body>
</html>
```