分离编解码 javascript 算法使用手册

上海泥娃通信科技有限公司

目录

第一章 分离码算法介绍

利用数学不同进制之间的转换结合变换的码表，实现信息编解码，包括：信息的码表单元；信息的编码单元；信息的解码单元。

实现文档分解成码表、变换序列和位数系列三个部分，或者采用默认码表的变换序列和位数序列两部分；本发明还实现通过码表，变换序列和位数序列还原文档的方法。

基于上述目的不同进制之间转换形成码和位分离编解码的方法包括：

制定码表：确定处理信息的单元位数，确定转换的进制，定义码表；

编码：根据要求读取 64 位（或者 128 位，或者其它）赋值给整数，然后根据要求转换成相应的进制（对应的数字用码表表示的字符表示），转换结果记录到变换序列，转换后的位数记录到位数序列，一直持续到转换完毕，最后形成两个部分。变换序列的字符一定是码表的字符，位数序列主要记载转换单元对应在变换记录中的长度。

解码：读取位数信息，按位数读取相关的字符，查找码表变换成相应的数字，结合原有的进制定义，转换为整数，存入到文件中，一直到转换完毕，得到相关的文件。

有益效果在于：

用于信息的多路存储和传输，不再借助于密钥来保证信息的安全，信息的存储分为三个部分：进制定义和码表、十进制到给定进制之间的转换结果、转换后结果的位数。并且可以针对不同的要求，实现给定进制和码表的信息存储和传输，用于网络之间的多路通信和多路存储；实现自定义进制和码表的信息存储，用于特定场合的传输和保存。

功能描述：

1、信息安全算法，码表对应加密的密钥，利用数学运算的特性实现信息的加密；

2、密文全文搜索：利用加密信息和原文信息的前缀一致性，结合语义树全文索引算法实现密文全文检索。

本算法获得国家发明专利：**《一种分离编解码的方法》,专利号:ZL 2016 1 0238974.2**

第二章 分离码的实现和应用

算法采用 javascript 语言实现，主要分为初始化，编码和解码，以及分离码表示组成。

1.1. 代码说明

/*

模块：分离码

功能：

分离码是一种信息编解码技术，主要利用数学的不同进制转换来形成，结合码表和数学的进制转换，提出码位分离的编解码方法。

版权：

上海泥娃通信科技有限公司

email: zhangliuxue@126.com

*/

1.1.1. 变量 biaodianfuhao ：判断是否标点符号的正则表达式

var biaodianfuhao = /[[\u0021-\u002f]|[\u003a-\u0040]|[\u005b-\u0060]|[\u007b-\u007e]|[\u00a0-\u00bf]|[\u2000-\u206f]|[\u2e00-\u2e7f]|[\u3000-\u303f]|[\uff01-\uff0f]|[\uff1a-\uff20]|[\uff3b-\uff40]|[\uff5b-\uff5e]]/;

1.1.2. 变量 ffff：是否按字词编解码：是，按单个字符进行编解码；否，则按每 64bit 进行编解码

var ffff = false;

1.1.3. 函数 unique：去掉字符串 arr 的重复出现的字符，主要用于用户码表的生成；

输入：arr，需要处理的字符串

输出：arr 中去掉重复字符后的字符数组

function unique(arr) {

```
    var result = [], hash = {};
    for (var i = 0, elem; (elem = arr[i]) != null; i++) {
        if (!hash[elem]) {
            result.push(elem);
            hash[elem] = true;
        }
    }
    return result;
}
```

1.1.4.filterRepeatStr：去掉字符串 str 的重复出现的字符，主要用于用户码表的生成

输入：str，需要处理的字符串

输出：str 中去掉重复字符后的字符串

```
function filterRepeatStr(str)
{
    if (!ffff)
    {
        var ar2 = str.split("");
        var msg = unique(ar2);
        var str = "";
        for (var x = 0; x < msg.length; x++) {
            if (msg[x] != " ")
                str += msg[x];
        }
        return str;

    } else
    {
        var ar2 = str.split("");
        var msg = unique(ar2);
        var str = "";
        for (var x = 0; x < msg.length; x++) {
```

```
                if (msg[x] != " " && !biaodianfuhao.test(msg[x]))
                    str += msg[x];
            }
            return str;
        }
    }
```

1.1.5.InitJz：初始化

输入：st，初始化码表

```
    function InitJz(st) {
        st.iCodeTable = new Object();
        if (st.letters == undefined || st.letters == "")
            //st.letters = "0123456789abcdefghijklmnopqrstuv";
            st.letters = "+-0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
        st.jz = st.letters.length;
        for (var i = 0; i < st.jz; i++) {
            st.iCodeTable[st.letters[i]] = i;
        }
    }
```

1.1.6.GetJzInt：获取字符串表示的整数

输入：st，初始化的码表结构；cValue，字符串；offset，获取字符串的起始位置；k 获取字符串的长度

```
    function GetJzInt(st, cValue, offset, iL) {
        var i = 0;
        for (k = 0; k < iL; k++) {
            i = i + st.iCodeTable[cValue[offset + k]] * Math.pow(st.jz, k);
        }
        return i;
    }
```

1.1.7.GetJzChar：获取整数表示的字符串，并给出字符串的长度

输入：st，初始化的码表结果；iValue，整数
输出：整数表示的字符串

```
function GetJzChar(st, iValue) {
    var i = iValue;
    var p = "";
    var k = 0;
    while (i != 0) {
        i = parseInt(iValue / st.jz);
        p += st.letters[parseInt(iValue % st.jz)];

        iValue = i;
        if (i >= st.jz) {
            k++;
        } else {
            p += st.letters[parseInt(iValue % st.jz)];
            k++;
            break;
        }
    }

    k++;
    var e = {};
    e.str = p;
    e.m = k;
    return e;
}
```

1.1.8.Chartoint：字符串转换为整数

输入：c，字符串；st，字符起始位置；len，转换字符串的长度

```
function chartoint(c, st, len) {
    var iv = "";
```

```
    for (var i = st + len - 1; i >= st; i--) {
        iv += c[i];
    }
    return parseInt(iv, 16);
}
```

1.1.9.编码：分离码编码 输入消息，转换为字符串序列和位数序列

输入：st，码表结构；msg，需要编码的消息字符串；iSer 为位序列

输出：编码后的字符串序列

```
function EncodeJz(st, msg, iSer) {
    var len = msg.length;
    var i = Math.floor(len / 4);
    var j = len % 4;
    var outstr = "";
    for (var k = 0; k < i; k++) {
        var p = chartoint(msg, k * 4, 4);
        var m = 0;
        if (p != NaN) {
            var tmp = GetJzChar(st, p);
            outstr += tmp.str;
            iSer.push(tmp.m);
        }
    }
    if (j > 0) {
        var p = chartoint(msg, k * 4, j);
        var m = 0;
        var tmp = GetJzChar(st, p);
        outstr += tmp.str;
        iSer.push(tmp.m);
    }
    return outstr;
}
```

1.1.10. 编码：EncodeJzByspace，空格分割，不再产生位序列，位序列由空格分隔的字符串长度表示

输入：st，码表结构；msg，需要编码的字符串
输出：空格分隔的字符串

```
function EncodeJzByspace(st, msg) {
    var len = msg.length;
    var i = Math.floor(len / 4);
    var j = len % 4;
    var outstr = "";
    for (var k = 0; k < i; k++) {
        var p = chartoint(msg, k * 4, 4);
        var m = 0;
        if (p != NaN) {
            var tmp = GetJzChar(st, p);
            outstr += tmp.str;
            outstr += " ";
        }
    }
    if (j > 0) {
        var p = chartoint(msg, k * 4, j);
        var m = 0;
        var tmp = GetJzChar(st, p);
        outstr += tmp.str;
        outstr += " ";
    }
    return outstr;
}
```

1.1.11. 解码：DecodeJz，按字符串序列和位序列实现解码

输入：st，码表结构体；msg，编码消息的字符串序列；len，保留参数；iSer，位序列
输出：解码后的字符串

```
function DecodeJz(st, msg, len, iSer) {
    var p1 = "";
```

```
    var offset = 0;

    for (x in iSer) {
        var m = GetJzInt(st, msg, offset, iSer[x]);
        var p = m.toString(16);
        if (p.length % 2 != 0) {
            p = "0" + p;
        }
        m = "";
        for (var i = 0; i < p.length / 2; i++) {
            var tmp = p.substr(i * 2, 2);
            m = tmp + m;
        }
        p1 += m.toString(16);
        offset += iSer[x];
    }
    return p1;
}
```

1.1.12. 解码：DecodeJzByspace，空格分割字符串解码

输入：st,码表结构体；msg，空格分隔的字符串；len，保留
输出：解码后的字符串

```
function DecodeJzByspace(st, msg, len)
{
    var p1 = "";
    if (ffff)
    {
        var segv;
        var biaodianfuhaog = /[[\u0021-\u002f]|[\u003a-\u0040]|[\u005b-\u0060]|[\u007b-\u007e]|[\u00a0-\u00bf]|[\u2000-\u206f]|[\u2e00-\u2e7f]|[\u3000-\u303f]|[\uff01-\uff0f]|[\uff1a-\uff20]|[\uff3b-\uff40]|[\uff5b-\uff5e]]/g;
        var pdArr = msg.match(biaodianfuhaog)

        segv = msg.split(biaodianfuhao);
        for (n = 0; n < segv.length; n++) {
```

```
            var offset = 0;
            var sarray = segv[n].split(" ");
            for (var x = 0; x < sarray.length; x++) {
                if (sarray[x].length > 0)
                {
                    var m = GetJzInt(st, sarray[x], 0, sarray[x].length);
                    var p = m.toString(16);
                    if (p.length % 2 != 0) {
                        p = "0" + p;
                    }
                    m = "";
                    for (var i = 0; i < p.length / 2; i++) {
                        var tmp = p.substr(i * 2, 2);
                        m = tmp + m;
                    }
                    p1 += m.toString(16);
                }

            }
            try {
                var code = encodeURI(pdArr[n]);
                var s_str = Encode16(code);
                if (s_str != " " && pdArr[n] != undefined)
                    p1 += s_str;
            } catch (e) {

            }

        }
    } else
    {
        var sarray = msg.split(" ");
        for (var x = 0; x < sarray.length; x++) {

            var m = GetJzInt(st, sarray[x], 0, sarray[x].length);
```

```
                var p = m.toString(16);
                if (p.length % 2 != 0) {
                    p = "0" + p;
                }
                m = "";
                for (var i = 0; i < p.length / 2; i++) {
                    var tmp = p.substr(i * 2, 2);
                    m = tmp + m;
                }
                p1 += m.toString(16);
            }
        }
        return p1;
    }
```

1.1.13. Encode16：字符串的 hex 表示

输入：str，字符串
输出：hex 表示的字符串

```
    function Encode16(str) {
        var rstr = ""
        for (i = 0; i < str.length; i++) {
            if (str.charAt(i) == '%') {
                rstr += str.charAt(i + 1);
                rstr += str.charAt(i + 2);
                i = i + 2;
            } else {
                rstr += str.charCodeAt(i).toString(16)
            }
        }
        return rstr;
    }
```

1.1.14. Encode 网页编码示例

```
    function Encode() {
```

```
    var st = new Object();
    var mb = document.getElementById("mb").value;
    if (mb != undefined && mb != "") {
        st.letters = filterRepeatStr(mb);
    }
    InitJz(st);
    var zlx = document.getElementById("ss").value;
    var code = encodeURI(zlx);
    var s_str = Encode16(code);

    var co = new Array();
    for (var i = 0; i < s_str.length / 2; i++) {
        co.push(s_str.substr(i * 2, 2));
    }
    var iSer = new Array();
    var zxj = EncodeJz(st, co, iSer);
    document.getElementById("dd").value = zxj;
    var pp = DecodeJz(st, zxj, zxj.length, iSer);
    var msg = "";
    for (var i = 0; i < pp.length / 2; i++) {
        msg += "%" + pp.substr(i * 2, 2);
    }
    document.getElementById("dl").value = iSer.toString();
    document.getElementById("d1").value = msg;
    document.getElementById("de").value = unescape(decodeURI(msg));
}
```

1.1.15. EncodeByspace:网页按空格表示的编码示例

```
function EncodeByspace() {

    var st = new Object();
    var flcodestyle = document.getElementById("flcodestyle");
    ffff = false;
    if (flcodestyle != undefined) {
        if (flcodestyle.checked) {
```

```
            ffff = true;
        } else
            ffff = false;
}
var mb = document.getElementById("mb").value;
if (mb != undefined && mb != "") {
        st.letters = filterRepeatStr(mb);
}

if (!ffff)
{
        InitJz(st);
        var zlx = document.getElementById("ss").value;
        var code = encodeURI(zlx);
        var s_str = Encode16(code);

        var co = new Array();
        for (var i = 0; i < s_str.length / 2; i++) {
            co.push(s_str.substr(i * 2, 2));
        }

        var zxj = EncodeJzByspace(st, co);
        zxj = zxj.substr(0, zxj.length - 1);

        document.getElementById("dd").value = zxj;
} else
{
        var zlx = document.getElementById("ss").value;
        var zxj = "";
        for (var word = 0; word < zlx.length; word++)
        {
            if (biaodianfuhao.test(zlx[word]))
            {
                zxj = zxj.substr(0, zxj.length - 1);
                zxj += zlx[word];
```

```
                } else
                {
                    InitJz(st);

                    var code = encodeURI(zlx.substr(word, 1));
                    var s_str = Encode16(code);

                    var co = new Array();
                    for (var i = 0; i < s_str.length / 2; i++) {
                        co.push(s_str.substr(i * 2, 2));
                    }
                    zxj += EncodeJzByspace(st, co);
                }
            }

            document.getElementById("dd").value = zxj;
        }

        var pp = DecodeJzByspace(st, zxj, zxj.length);
        var msg = "";
        for (var i = 0; i < pp.length / 2; i++) {
            msg += "%" + pp.substr(i * 2, 2);
        }
        document.getElementById("d1").value = msg;
        document.getElementById("de").value = unescape(decodeURI(msg));
    }
```

1.1.16. Decodeimg：网页图形解码实例

```
    function Decodeimg() {
        var st = new Object();
        var mb = document.getElementById("mb").value;
        if (mb != undefined && mb != "") {
            st.letters = filterRepeatStr(mb);
        }
        InitJz(st);
```

```
    var zxj = document.getElementById("dd").value;

    var pp = DecodeJzByspace(st, zxj, zxj.length);
    var msg = "";
    for (var i = 0; i < pp.length / 2; i++) {
        msg += "%" + pp.substr(i * 2, 2);
    }
    document.getElementById("d1").value = msg;
    try {
        document.getElementById("de").value = unescape(decodeURI(msg));
    } catch (e) {
        document.getElementById("de").value = e.message;
    }
    txshow1.src = document.getElementById("de").value;
}
```

1.1.17. Decode：网页解码实例

```
function Decode() {
    var st = new Object();
    var mb = document.getElementById("mb").value;
    if (mb != undefined && mb != "") {
        st.letters = filterRepeatStr(mb);
    }
    InitJz(st);
    var zxj = document.getElementById("dd").value;
    var iSer = document.getElementById("dl").value.split(",");
    for (x in iSer) {
        iSer[x] = parseInt(iSer[x]);
    }
    var pp = DecodeJz(st, zxj, zxj.length, iSer);
    var msg = "";
    for (var i = 0; i < pp.length / 2; i++) {
        msg += "%" + pp.substr(i * 2, 2);
    }
    document.getElementById("d1").value = msg;
```

```
        try {
            document.getElementById("de").value = unescape(decodeURI(msg));
        } catch (e) {
            document.getElementById("de").value = e.message;
        }

    }
```

1.1.18. DecodeByspace：网页按空格解码示例

```
    function DecodeByspace() {
        var st = new Object();
        var mb = document.getElementById("mb").value;
        if (mb != undefined && mb != "") {
            st.letters = filterRepeatStr(mb);
        }
        InitJz(st);
        var zxj = document.getElementById("dd").value;

        var pp = DecodeJzByspace(st, zxj, zxj.length);
        var msg = "";
        for (var i = 0; i < pp.length / 2; i++) {
            msg += "%" + pp.substr(i * 2, 2);
        }
        document.getElementById("d1").value = msg;
        try {
            document.getElementById("de").value = unescape(decodeURI(msg));
        } catch (e) {
            document.getElementById("de").value = e.message;
        }

    }
```

1.1.19. StartEncode：网页启用 js 线程编码示例

```
    function StartEncode() {
```

```
    tishi.innerHTML = "编码开始！";
    if (typeof (Worker) !== "undefined") {
        var w = new Worker("./js/Encode.js");
        var postdata = [];

        postdata.pwd = filterRepeatStr(document.getElementById("mb").value);
        postdata.value = document.getElementById("ss").value;
        delete postdata;
        w.postMessage(postdata);
        w.onmessage = function (event) {
            // document.getElementById("dl").value = event.data.ival.toString();
             document.getElementById("dd").innerHTML = event.data.value;
             tishi.innerHTML = "编码完毕！";
        };
    }
    else {
        tishi.innerHTML = "Sorry, your browser does not support Web Workers...";
    }
}
```

1.1.20. StartDecode：网页启用 js 线程解码示例

```
function StartDecode() {
    tishi.innerHTML = "解码开始！";
    if (typeof (Worker) !== "undefined") {
        var w = new Worker("./js/Decode.js");
        var postdata = [];

        postdata.pwd = filterRepeatStr(document.getElementById("mb").value);
        postdata.value = document.getElementById("dd").value;



        w.postMessage(postdata);
        delete postdata;
        w.onmessage = function (event) {
            tishi.innerHTML = "解码完毕！";
```

```
                document.getElementById("d1").value = event.data;
                try {
                    document.getElementById("de").value = unescape(decodeURI(event.data));
                } catch (e) {
                    document.getElementById("de").value = e.message;
                }
                txshow1.src = document.getElementById("de").value;
            };
        }
        else {
            tishi.innerHTML = "Sorry, your browser does not support Web Workers...";
        }
    }
```

1.1.21. EncodeFlcode：分离码编码

输入：mb，码表；value，需要编码的信息

```
    function EncodeFlcode(mb, value) {
        var result = [];
        var st = new Object();
        if (mb != undefined && mb != "") {
            st.letters = filterRepeatStr(mb);
        }
        InitJz(st);
        var code = encodeURI(value);
        var s_str = Encode16(code);

        var co = new Array();
        var m_str = "";
        for (var i = 0; i < s_str.length / 2; i++) {
            co.push(s_str.substr(i * 2, 2));
        }

        var zxj = EncodeJzByspace(st, co);
        result.value = zxj;
```

```
        self.postMessage(result);
        delete st;
        delete co;
        delete zxj;
        delete result;

        self.close();
    }
```

1.1.22. EncodeFlcode_miwen：用于密文全文搜索的前端编码算法

输入：mb，码表；value，需要编码的信息
输出：生成空格分隔的编码字符串

```
    function EncodeFlcode_miwen(mb, value) {
        var st = new Object();
        if (mb != undefined && mb != "") {
            st.letters = filterRepeatStr(mb);
        }
        InitJz(st);
        value = value.replace(/(^\s*)|(\s*$)/g, "");
        var code = encodeURI(value);
        var s_str = Encode16(code);

        var co = new Array();
        var m_str = "";
        for (var i = 0; i < s_str.length / 2; i++) {
            co.push(s_str.substr(i * 2, 2));
        }

        var zxj = EncodeJzByspace(st, co);

        delete st;
        delete co;
        delete zxj;
```

```
        return zxj;
    }
```

1.1.23. DecodeFlcode：分离码解码

输入：mb，码表；value，编码的字符串序列；ival，保留
输出：解码后的信息

```
    function DecodeFlcode(mb, value, ival) {
        var st = new Object();
        if (mb != undefined && mb != "") {
            st.letters = filterRepeatStr(mb);
        }
        InitJz(st);

        var pp = DecodeJzByspace(st, value, value.length);
        var msg = "";
        for (var i = 0; i < pp.length / 2; i++) {
            msg += "%" + pp.substr(i * 2, 2);
        }

        self.postMessage(msg);
        delete st;
        delete pp;
        delete msg;

        self.close();
    }
```

1.1.24. DecodeFlcode_miwen：用于密文全文搜索的前端解码

输入：mb，码表；value，需要解码的信息

```
    function DecodeFlcode_miwen(mb, value) {
        var st = new Object();
        var rstr = "";
        if (mb != undefined && mb != "") {
            st.letters = filterRepeatStr(mb);
```

```
}
InitJz(st);
loop:
try {
    var pp = DecodeJzByspace(st, value, value.length);
    var msg = "";
    for (var i = 0; i < pp.length / 2; i++)
    {
        msg += "%" + pp.substr(i * 2, 2);
    }

    delete st;
    delete pp;

    rstr = unescape(decodeURI(msg));

    //for (var y = rstr.length-1; y < rstr.length; y++)
    //{
    //    var m = rstr.charCodeAt(y);
    //    if(m<10 || (m>=32 && m<=47) || (m>=58 && m<=64)|| (m>=91 && m<=96)||
(m>=123 && m<=126))
    //    {
    //        rstr = rstr.substr(0, y);
    //    }
    //}
    return rstr;
} catch (err)
{
    value = value.substr(0, value.length - 1);
    return DecodeFlcode_miwen(mb, value);

}
rstr = rstr.toString();

return rstr;
```

```
}
```

1.1.25. GetStrDif ：从 str2 中去掉 str1 中包含的部分

输入：str1，需要从 str2 中去掉的字符换；str2，字符串

输出：从 str2 中去掉 str1 中包含的部分

```
function GetStrDif(str1,str2)
{
    var r = "";
    for(var i=0;i<str2.length;i++)
    {
        if(str1.charAt(i)==str2.charAt(i))
        {
            continue;
        } else {
            r = str2.substr(i);
            return r;
        }
    }
    return "";
}
```