# M183 Applikationssicherheit Implementieren

## Tutorial zum Lab Encryption Vigenere Cipher

| | | |
|---|---|---|
| Version 2 | 12.12.2017 | Jürg Nietlispach |
| Version 1 | 30.11.2017 | Jürg Nietlispach |

# Contents

## Idee

In diesem Lab soll eine .Javascript Applikation erstell werden, welche Hilft, Geheimtexte, die mit der Vigenere Cipher bzw. Polyalphabetischer Substitution erstellt wurden, zu decodieren.

## Herangehensweise Cryptoanalysis

1. Index.html File erstellen
2. GUI-Elemente erstellen
3. Eventhandling und Routinen für Verschlüsselung und Entschlüsselung erstellen

Leeres index.html File erstellen mit folgendem Header-Informationen

```
<!doctype html>
<html class="no-js" lang="">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />

<body>
    <div class="container">
```

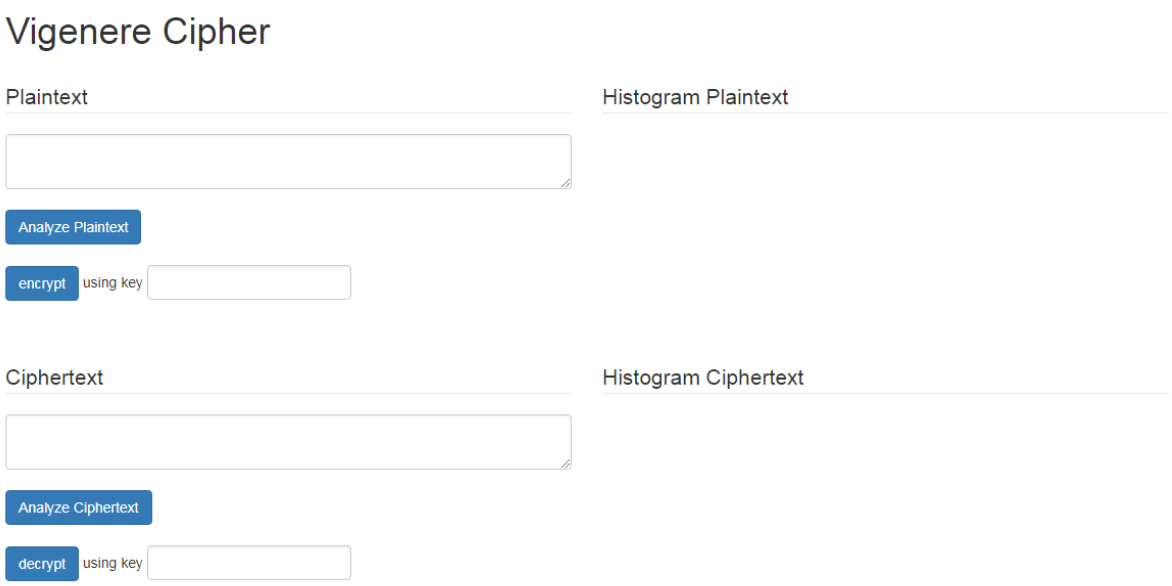
Es müssen nun Elemente für den Verschlüsselungsteil erstellt werden:

```
<div class="row">
    <div class="col-md-12">
        <h1>Vigenere Cipher</h1>
    </div>
</div>

<div class="row">

    <div class="col-md-6">
    <br>
        <legend>Plaintext</legend>
        <textarea id="plaintext" class="form-control"></textarea><br>
        <button class="btn btn-primary" id="analyze_plaintext">Analyze Plaintext</button> <br><br>
        <button class="btn btn-primary" id="encrypt">encrypt</button> using key
        <input name="shift" id="encryptkey" value="" type="text" class="form-control" style="width:200px;display:inline-block;" />
    </div>

    <div class="col-md-6">
    <br>
        <legend>Histogram Plaintext</legend>
        <div id="chart_plaintext"></div>

    </div>

</div>
```

Ebenfalls für den Entschlüsselungsteil

```
<div class="row">

    <div class="col-md-12"><br><br></div>
</div>
<div class="row">

    <div class="col-md-6">
    <br>
    <legend>Ciphertext</legend>
        <textarea id="ciphertext" class="form-control"></textarea><br>
        <button class="btn btn-primary" id="analyze_ciphertext">Analyze Ciphertext</button> <br><br>
        <button class="btn btn-primary" id="decrypt">decrypt</button> using key
        <input name="backshift" id="decryptkey" value="" type="text" class="form-control" style="width:200px;display:inline-block;" />
    </div>

    <div class="col-md-6">
    <br>
        <legend>Histogram Ciphertext</legend>
        <div id="chart_ciphertext"></div>

    </div>
```

Wurden alle GUI Elemente korrekt hinzugefügt, sieht das dann so aus im Browser.




Nun müssen die Events registriert werden und die Entschlüsselungs- und Verschlüsselungsroutinen erstellt werden:

```
document.getElementById("analyze_plaintext").addEventListener("click", function (e) {

    e.preventDefault();
    e.stopPropagation();

    var text = document.getElementById("plaintext").value;
    var histogram = createHistogramValues(text);

    document.getElementById("chart_plaintext").innerHTML = createHTMLTable(histogram);

});

document.getElementById("encrypt").addEventListener("click", function (e) {

    e.preventDefault();
    e.stopPropagation();

    var plaintext = document.getElementById("plaintext").value;
    var key = document.getElementById("encryptkey").value;
    document.getElementById("ciphertext").innerHTML = createVigenere(plaintext, key);

});
```

Die beiden Histogramm-Funktionen sind wieder dieselben:

```
function createHistogramValues(text)
{
    var histogram_prepare = [];

    for (var i = 0, len = text.length; i < len; i++) {

        var letter = text[i];
        if (letter.match(/[a-z]/i))
        {
            histogram_prepare[letter] = (histogram_prepare[letter] || 0) + 1;
        }
    }

    histogram = histogram_prepare.sort(function(a, b) {
        a = a[1];
        b = b[1];

        return a < b ? -1 : (a > b ? 1 : 0);
    });

    return histogram;
}

function createHTMLTable(histogram)
{
    var html = "<table>";
    for (var key in histogram) {
        html += "<tr>";
        html += "<td>" + key + ": </td><td> </td>";
        html +=  "<td>" + histogram[key] + "</td>";
        html += "</tr>";
    }
    html += "</table>";
    return html;
}
```

Verschlüsselung und Entschlüsselung mit Vigenere:

```
function createVigenere(input, key) {

    var output = "";
    for (var i = 0, j = 0; i < input.length; i++)
    {
        var char = input[i];

        if (char.match(/[a-z]/i))
        {
            var c = input.charCodeAt(i);
            var key_char_shift = key[j % key.length].charCodeAt(0);

            if ((c >= 65) && (c <= 90))
            {
                output += String.fromCharCode((c - 65 + key_char_shift - 65) % 26 + 65);
                j++;
            }
            else if ((c >= 97) && (c <= 122))
            {
                output += String.fromCharCode((c - 97 + key_char_shift - 97) % 26 + 97);
                j++;
            }
        }
    }

    return output;
}
```

```
function decryptVigenere(input, key) {

    var output = "";
    for (var i = 0, j = 0; i < input.length; i++)
    {
        var char = input[i];

        if (char.match(/[a-z]/i))
        {
            var cr = parseInt(input.charCodeAt(i));
            //var key_char_shift = key[j % key.length].charCodeAt(0);

            if ((cr >= 65) && (cr <= 90))
            {
                //output += String.fromCharCode((c - 65 - (key_char_shift - 65)) % 26 + 65);
                var c = parseInt(cr - 65);
                var key_char_shift = parseInt(key[i].charCodeAt(0) - 65);

                var new_char_position = (c - key_char_shift) % 26 ;
                if(new_char_position < 0)
                    new_char_position += 26;

                output += String.fromCharCode(new_char_position + 65);

                j++;
            }
            else if ((cr >= 97) && (cr <= 122))
            {
                //output += String.fromCharCode((c - 97 - (key_char_shift - 97)) % 26 + 97);
                var c = parseInt(cr - 97);
                var key_char_shift = parseInt(key[j % key.length].charCodeAt(0) - 97);

                var new_char_position = (c - key_char_shift) % 26 ;
                if(new_char_position < 0)
                    new_char_position += 26;

                output += String.fromCharCode(new_char_position + 97);
                j++;
            }
        }
    }

    return output;
}
```

Und hier noch die zugehörigen Events, innerhalb welchen die obigen Routinen angestossen werden:

```
document.getElementById("analyze_ciphertext").addEventListener("click", function (e) {

    e.preventDefault();
    e.stopPropagation();

    var text = document.getElementById("ciphertext").value;
    var histogram = createHistogramValues(text);

    document.getElementById("chart_ciphertext").innerHTML = createHTMLTable(histogram);

});

document.getElementById("decrypt").addEventListener("click", function (e) {

    e.preventDefault();
    e.stopPropagation();

    var plaintext = document.getElementById("ciphertext").value;
    var key = document.getElementById("decryptkey").value;
    document.getElementById("ciphertext").innerHTML = decryptVigenere(plaintext, key);

});
```

Für ExpertInnen: GUI Erweitern, dass je Key-Spalte die Frequenzanalyse angezeigt wird.