```
<!-- Lizenz: https://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups -->
<script src="./system/js/aes.js"></script>
<script src="./system/js/pbkdf2.js"></script>

<script>

    // Datei einlesen
    function file_read( start ){

        // Erste Datei laden [ Dateiname: file.name ; Dateigröße: file.size ]
        file = document.getElementById('input').files[0];

        // Prüfen ob File vorhanden
        if( file ){

            // Größe der einzelnen Segmente festlegen
            var add_size = 1048576 / 4; // 256KB ( --> 1048576 == 1MB )
            var end   = start + add_size;
            var fertig = "FALSE";
            var debug  = "FALSE";

            // Prüfen ob Ende der Datei erreicht (Revisit: Prüfen ob hier die Angaben korrekt sind)
            if( start < file.size && end > file.size ){
                end = file.size;
                fertig = "TRUE";
            }

            // Prozent-Wert über aktuellen Fortschritt
            if( fertig != "TRUE" )
                var status = Math.round( 100 - ( ( ( file.size - end ) / file.size ) * 100 ) );
            else
                var status = 100;

            // Funktion zum Einlesen der Datei
            var reader = new FileReader();
            reader.onload = function (event) {

                // Die Datei Stückweise einlesen und Sonderzeichen ersetzen
                var sliced_text = encodeURIComponent( event.target.result.slice(start, end) );

                // Verschlüsselung vorbereiten
                var salt = CryptoJS.lib.WordArray.random(128/8);
                var key256Bits500Iterations = CryptoJS.PBKDF2("123456789", salt, { keySize: 256/32, iterations: 500 }); // REVISIT: Bei Gelegenheit durch eine Zufallszahl ersetzen
                var iv  = CryptoJS.enc.Hex.parse('101112131415161718191a1b1c1d1e1f'); // REVISIT: Bei Gelegenheit durch eine Zufallszahl ersetzen

                // Verschlüsseln und nach Base64 kodieren
                var encrypted = CryptoJS.AES.encrypt(sliced_text, key256Bits500Iterations, { iv: iv });
                var data_base64 = encrypted.ciphertext.toString(CryptoJS.enc.Base64);
                var iv_base64   = encrypted.iv.toString(CryptoJS.enc.Base64);
                var key_base64  = encrypted.key.toString(CryptoJS.enc.Base64);

                // HTTPs Request versenden (REVISIT: Große Dateien kommen in der falschen Reihenfolge an!)
                var xhr = new XMLHttpRequest();
                xhr.onreadystatechange = function(){
                    if(xhr.readyState==4 && xhr.status==200){
                        document.getElementById("status").innerHTML = xhr.responseText;
                    }
                }
                xhr.open("POST", "./index_filegetter.php");
                xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
                xhr.send("filename=" + file.name + "&start=" + start + "&ende=" + end + "&status=" + status + "&iv=" + iv_base64 + "&key=" + key_base64 + "&data=" + data_base64);

                // Prozentausgabe
                show_status( fertig, debug, sliced_text, status );

            };
            // Datei zu Einlesen aufrufen -> Hierdurch wird Funktion ausgelöst
            reader.readAsBinaryString(file);

            // Solange noch nicht fertig, "readfile"-Funktion mit neuem Startwert aufrufen
            if( fertig == "FALSE" )
                setTimeout( "file_read(" + ( end ) + ")", 250 ); // 1MB pro Sekunde


        } else {

            alert("Keine Datei ausgewaehlt!");

        }

    }


    // Prozentausgabe
    function show_status( fertig, debug, sliced_text, status ){

        if( debug == "TRUE" )
            document.getElementById('byte_range').innerHTML = 'Status:' + status + "%<br /><br />" + sliced_text;
        else{
            if( fertig == "FALSE" )
                document.getElementById('byte_range').innerHTML = 'Status:' + status + "%";
            else{

                document.getElementById('byte_range').innerHTML = "<strong>Status:</strong> Fertig! <i>(<!--Die Seite l&auml;d in k&uuml;rze neu! - -->Neuladen wurde deaktiviert um
                //setTimeout(function () { location.reload(true); }, 5000);

            }
        }
    }


</script>

<p class='bold'>Bitte w&auml;hlen Sie eine Datei zum verschl&uuml;sselten Upload aus:</p>
<p style="font-size: 11px; text-decoration:line-through;">Die Seite wird im Anschluss neu geladen, um den aktuellen Ordnerinhalt anzuzeigen.</p>
<input type="file" id="input"> <input type="button" value="Senden" onClick="file_read('0')">

<br />

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


<hr />
<p><strong>Ordnerinhalt:</strong> <a style="font-size: 11px;" href="#" onClick="document.getElementById('reload_button').click();">(Ansicht bitte nach Upload neu laden!)</a></p>
<form name="delete" action="./index.php?site=losung_6_7" method="POST" accept-charset="UTF-8">
```

```
            <input type="hidden" name="action" value="delete" />
            <input type="submit" value=" Ordnerinhalt leeren " class="float-right"/>
        </form>
        <form name="reload" action="./index.php?site=losung_6_7" method="POST" accept-charset="UTF-8">
            <input type="hidden" name="action" value="reload" />
            <input type="submit" value=" Ansicht neu laden " class="float-right" id="reload_button"/>
        </form>
        {foreach from=$filelist item=item}
          <p>{$item}</p>
        {/foreach}
```

```php
<?php

    // Vorbelegungen
    $arr_linklist = array();

    // Verzeichniss-Inhalt auslesen
    $dir = scandir('./output/');

    foreach( $dir as $file ){

        if( $file == '.' || $file == '..' || $file == '.DS_Store')
            continue;

        if( $_POST['action'] == 'delete' ){
            unlink('./output/' . $file);
            continue;
        }

        $filesize = filesize( './output/' . $file );

        if( substr( $file, -7 ) == '_upload' )
            $filename = substr( $file, 0, -7 );
        else
            $filename = $file;

        $arr_linklist[] = '<a href="./output/'.$file.'" target="_blank" download="'.$filename.'" style="color:#004ea0;">'.$filename.'</a> ('.$filesize.' Byte)';

    }

    // Prüfen ob Verzeichnis leer
    if( count($arr_linklist) === 0)
        $arr_linklist[] = '-leer-';

    // An Template geben
    $smarty->assign('filelist', $arr_linklist);

?>
```

```php
<?php

    // Funktionen
    //include_once( './system/functions/function.php' );

    if( isset( $_POST ) ){
        // nix
    } else {
        die('Kein POST');
    }

    // POST entgegennehmen
    $filename   = $_POST['filename'];   //unset( $_POST['filename'] );
    $start      = $_POST['start'];      //unset( $_POST['start'] );
    $ende       = $_POST['ende'];       //unset( $_POST['ende'] );
    $status     = $_POST['status'];     //unset( $_POST['status'] );
    $iv         = $_POST['iv'];         //unset( $_POST['iv'] );
    $key        = $_POST['key'];        //unset( $_POST['key'] );
    $data       = $_POST['data'];


    // Debug:
    echo "<hr /><p class='bold'>Debugausgabe des HTTP-POST REQUEST:</p>";
    echo "<pre>";
    print_r($_POST);
    echo "</pre>";
    echo "<hr />";

    // Plus-Zeichen korriegieren, die durch leerzeichen ersetzt wurden
    $iv = str_replace(' ', '+', $iv);
    $key = str_replace(' ', '+', $key);
    $data = str_replace(' ', '+', $data);


    // Debug: print_arr($data);


    // Entschlüsselung durchführen
    $encrypted = base64_decode($data); // Aus JS: "data_base64"-Wert
    $iv        = base64_decode($iv);   // Aus JS: "iv_base64"-Wert
    $key       = base64_decode($key);  // Aus JS: "key_base64"-Wert

    $sliced_text = rtrim( mcrypt_decrypt( MCRYPT_RIJNDAEL_128, $key, $encrypted, MCRYPT_MODE_CBC, $iv ), "\t\0 " );



    // JS-Zeichencodierung rückgängig machen
    $sliced_text = utf8_urldecode( $sliced_text );


    // UTF-8 wiederherstellen
    $sliced_text = utf8_decode( $sliced_text );


    // Datei schreiben (Revisit: Hier erweitern)
```

```php
    if( $start == 0 )
        $fp = fopen("./output/" . $filename . "_upload", "w");
    else
        $fp = fopen("./output/" . $filename . "_upload", "a");
    fwrite($fp,$sliced_text);
    fclose($fp);



    /* JS-ZEICHENCODIERUNG RÜCKGÄNGIG MACHEN */
    function utf8_urldecode($str) {

        $str = preg_replace("/%u([0-9a-f]{3,4})/i","&#x\\1;", urldecode($str));
        return html_entity_decode($str,null,'UTF-8');

    }

?>
```