

How to not protect a webpage

A simpel collision attack

Skrupellos

JID: skrupellos@swissjabber.de

μ CCC, mehrvortragewagen

19. August 2010

- 1 Überblick
- 2 Hash
- 3 Entschlüsselung
- 4 Ergebnisse

- 1 Überblick
 - Eigenschaften
 - Code
- 2 Hash
- 3 Entschlüsselung
- 4 Ergebnisse

- <http://javascript.internet.com/passwords/login-coder.html>
- JS “Authentifizierung”/“Authorisierung”
- Kein `if(password == "geheim") ;-`
- Multiuser fähig
- Startseite pro user

Zitat von internet.com

Honestly, we don't even totally understand this JavaScript!

Login page generator

John Smith Peter Jones Sue Brown Sally West	User: <input type="text" value="John Smith"/>
	Password: <input type="text" value="password"/> <-- Must be exactly 8 characters
	Page Name: <input type="text" value="mainpage"/> .html <-- Must be exactly 8 characters
	<input type="button" value="New User"/> <input type="button" value="Delete User"/> <input type="button" value="Update/Show Coding"/>
[Note: Password/Page Name must be exactly 8 letters! (a-z)]	

Create your users, their passwords, and their destination pages using the form above.

Then, click 'Show Coding' to see the login HTML and JavaScript that you need to put on your page to password-protect your site with that info.

Questions about this script or how to put it on your site should be directed to it's author via email: dion@mailhub.omen.com.au

Honestly, we don't even totally understand this JavaScript!

Free JavaScripts provided
by [The JavaScript Source](#)

Abbildung: Login page generator

- Genau 8 Zeichen
- Case insensitive
- $26^8 \approx 2,09 \cdot 10^{11}$ kombinationen

$$P = \{ 'A', 'B', 'C', \dots, 'Z' \}$$

$$p \in P^8$$

Members-Only Area!	
Username:	<input type="text" value="John Smith"/> ▼
Password:	<input type="password" value="●●●●●●"/>
<input type="button" value="Login"/>	

Abbildung: “Members only!” O RLY?

Security by ... poor formatted code

```

1 var params=new Array(4);
2 var alpha="ABCDEFGHijklmnopqrstuvwxyzABCDEFGHI*";
3 function check(form) {
4   which=form.memList.selectedIndex;
5   choice = form.memList.options[which].value+"!";
6   if (choice=="x!") {
7     alert("Please Select Your Name From The List*");
8     return;
9   }
10 p=0;
11 for (i=0;i<3;i++) {
12   a=choice.indexOf("!",p);
13   params[i]=choice.substring(a,p);
14   p=a+1;
15 }
16 h1=makehash(form.pass.value,3);
17 h2=makehash(form.pass.value,10)+" ";
18 if (h1!=params[1]) {
19   alert("Incorrect Password!"); return; };
20 var page="";
21 for (var i=0;i<8;i++) {
22   letter=params[2].substring(i,i+1)
23   ul=letter.toUpperCase();
24   a=alpha.indexOf(ul,0);
25   a=(h2.substring(i,i+1)*1);
26   if (a<0) a+=26;
27   page+=alpha.substring(a,a+1); };
28   top.location=page.toLowerCase()+"*.html";
29 }
30 function makehash(pw,mult) {
31   pass=pw.toUpperCase();
32   hash=0;
33   for (i=0;i<8;i++) {
34     letter=pass.substring(i,i+1);
35     c=alpha.indexOf(letter,0)+1;
36     hash=hash*mult+c;
37   }
38   return(hash);
39 }

```

Abbildung: Vorher [sic]

```

1 var params = new Array(4);
2 var alpha = "ABCDEFGHijklmnopqrstuvwxyzABCDEFGHI*";
3
4 function check(form) {
5     //!!! PARSE INPUT
6     which = form.memList.selectedIndex;
7     choice = form.memList.options[which].value + "!";
8
9
10    // Check for invalid selection
11    if(choice == "x!") {
12        alert("Please Select Your Name From The List*");
13        return;
14    }
15
16    // Get parameters from selection
17    p = 0;
18    for(i = 0; i < 3; i++) {
19        a = choice.indexOf("!", p);
20        params[i] = choice.substring(a, p);
21        p = a + 1;
22    }
23
24    //!!! CREATE HASH 1
25    h1 = makehash(form.pass.value, 3);
26
27    //!!! CREATE HASH 2
28    h2 = makehash(form.pass.value, 10) + " ";
29
30    //!!! COMPARE
31    if(h1 != params[1]) {
32        alert("Incorrect Password!");
33        return;
34    }
35
36    //!!! DECRYPT
37    var page = "";
38    for(var i = 0; i < 8; i++) {
39        letter = params[2].substring(i, i + 1);
40        ul = letter.toUpperCase();
41        a = alpha.indexOf(ul, 0);
42
43        a = (h2.substring(i, i + 1) * 1);
44        if(a < 0) {
45            a += 26;
46        }
47        page += alpha.substring(a, a + 1);
48    }
49
50    //!!! GO TO PAGE
51    top.location = page.toLowerCase() + "*.html";
52 }
53
54 function makehash(pw, mult) {
55     pass = pw.toUpperCase();
56     hash = 0;
57
58     for(i = 0; i < 8; i++) {
59         letter = pass.substring(i, i + 1);
60         c = alpha.indexOf(letter, 0) + 1;
61         hash = hash * mult + c;
62     }
63
64     return(hash);
65 }

```

Abbildung: Nacher

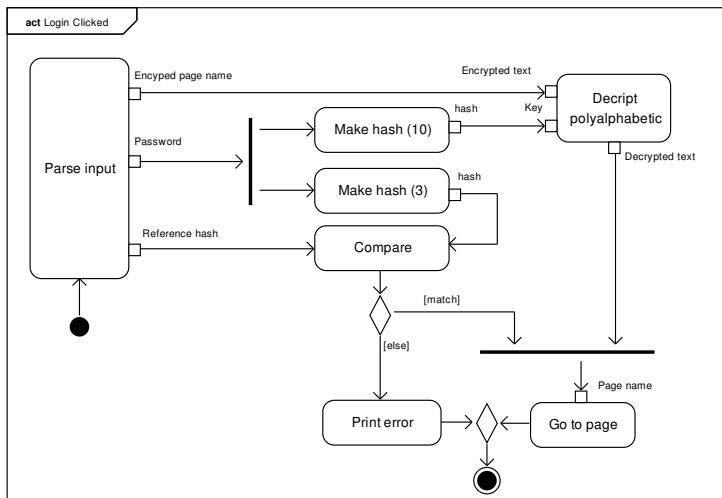


Abbildung: Anmelde Funktion

1 Überblick**2** Hash

- Herleitung
- Umkehrung
- Implementierung

3 Entschlüsselung**4** Ergebnisse

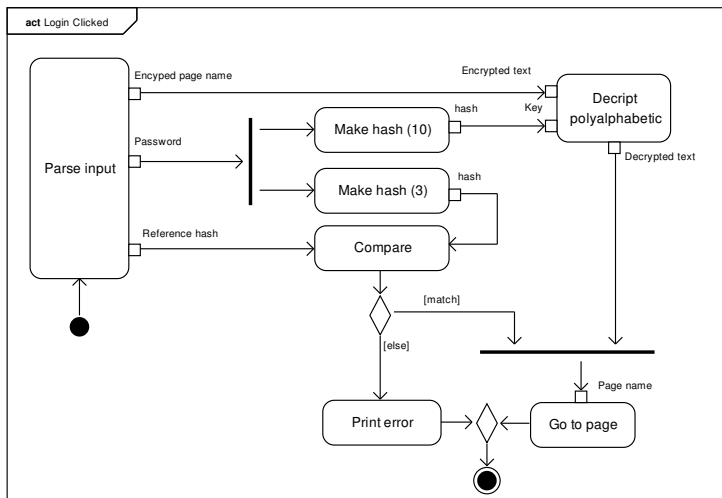


Abbildung: Hashing im System

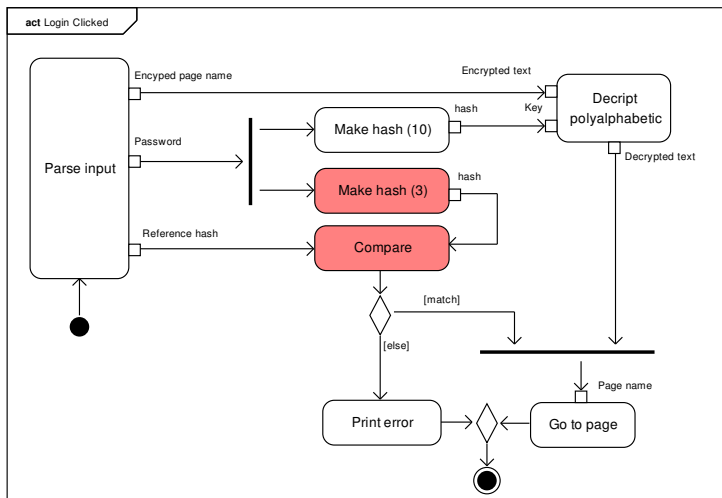


Abbildung: Hashing im System

Zeile 8 - 9

```
1 var alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
2
3 function makehash(pw, mult) {
4     pass = pw.toUpperCase();
5     hash = 0;
6
7     for(i = 0; i < 8; i++) {
8         letter = pass.substring(i, i + 1);
9         c = alpha.indexOf(letter, 0) + 1;
10        hash = hash * mult + c;
11    }
12
13    return(hash);
14 }
```

$$A = \{1, 2, 3, \dots, 26\}$$

$$t: P \rightarrow A$$

$$t(p) = \begin{cases} 1 \text{ für } p = 'A' \\ 2 \text{ für } p = 'B' \\ \vdots \\ 26 \text{ für } p = 'Z' \end{cases}$$

Einzelschritt

```
1 var alpha = "ABCDEFGHJKLMNOPQRSTUVWXYZABCDEFGHI";
2
3 function makehash(pw, mult) {
4     pass = pw.toUpperCase();
5     hash = 0;
6
7     for(i = 0; i < 8; i++) {
8         c = t(pass.substring(i, i + 1));
9         hash = hash * mult + c;
10    }
11
12    return(hash);
13 }
```

$$pw = \{a_4, a_3, a_2, a_1, a_0\}$$

$$mul = 3$$

$$a_4$$

$$(a_4) \cdot 3 + a_3$$

$$((a_4) \cdot 3 + a_3) \cdot 3 + a_2$$

$$(((a_4) \cdot 3 + a_3) \cdot 3 + a_2) \cdot 3 + a_1$$

$$((((a_4) \cdot 3 + a_3) \cdot 3 + a_2) \cdot 3 + a_1) \cdot 3 + a_0$$

Vereinfachen

$$\begin{aligned}
& (((((a_4) \cdot 3 + a_3) \cdot 3 + a_2) \cdot 3 + a_1) \cdot 3 + a_0 \\
& (((a_4) \cdot 3 + a_3) \cdot 3 + a_2) \cdot 3 \cdot 3 + a_1 \cdot 3 + a_0 \\
& ((a_4) \cdot 3 + a_3) \cdot 3 \cdot 3 \cdot 3 + a_2 \cdot 3 \cdot 3 + a_1 \cdot 3 + a_0 \\
& (a_4) \cdot 3 \cdot 3 \cdot 3 \cdot 3 + a_3 \cdot 3 \cdot 3 \cdot 3 + a_2 \cdot 3 \cdot 3 + a_1 \cdot 3 + a_0 \\
& a_4 \cdot 3 \cdot 3 \cdot 3 \cdot 3 + a_3 \cdot 3 \cdot 3 \cdot 3 + a_2 \cdot 3 \cdot 3 + a_1 \cdot 3 + a_0 \\
& a_4 \cdot 3^4 + a_3 \cdot 3^3 + a_2 \cdot 3^2 + a_1 \cdot 3^1 + a_0 \\
& a_4 \cdot 3^4 + a_3 \cdot 3^3 + a_2 \cdot 3^2 + a_1 \cdot 3^1 + a_0 \cdot 3^0 \\
& \sum_{i=0}^4 a_i 3^i
\end{aligned}$$

a_i isolieren

$$h(a) = \sum_{n=0}^7 a_n 3^n \quad (1)$$

$$= \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \sum_{n=0}^{i-1} a_n 3^n \quad (4)$$

Einzelnes Element an $i \in \{0, 1, 2, \dots, 7\}$ herauslösen.

Definition

$$\sum_{n=s}^t f(n) = \sum_{n=s}^i f(n) + \sum_{n=i+1}^t f(n) \text{ mit } s \leq i < t$$

Definition

$$\sum_{n=s}^s f(n) = f(n)$$

$$\underbrace{h(a)}_{\text{bekannt}} = \underbrace{\sum_{n=i+1}^7 a_n 3^n}_{\text{bekannt}} + \underbrace{a_i 3^i}_{\text{gesucht}} + \underbrace{\sum_{n=0}^{i-1} a_n 3^n}_{\text{unbekannt}}$$

unbekannt *Minimum* wenn alle $a_n = \min A = 1$ mit $n < i$

gesucht *Maximal* so groß, dass die Unbekannten die Möglichkeit haben $h(a)$ nicht zu *überschreiten*.

$$h(a) \geq \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \sum_{n=0}^{i-1} 1 \cdot 3^n$$

$$\underbrace{h(a)}_{\text{bekannt}} = \underbrace{\sum_{n=i+1}^7 a_n 3^n}_{\text{bekannt}} + \underbrace{a_i 3^i}_{\text{gesucht}} + \underbrace{\sum_{n=0}^{i-1} a_n 3^n}_{\text{unbekannt}}$$

unbekannt *Maximum* wenn alle $a_n = \max A = 26$ mit $n < i$

gesucht *Mindestens* so groß, dass die Unbekannten die
 Möglichkeit haben $h(a)$ zu *erreichen*.

$$h(a) \leq \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \sum_{n=0}^{i-1} 26 \cdot 3^n$$

Summe auflösen

$$h(a) \geq \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \sum_{n=0}^{i-1} 1 \cdot 3^n \quad (5)$$

$$\wedge h(a) \leq \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \sum_{n=0}^{i-1} 26 \cdot 3^n \quad (6)$$

$$h(a) \geq \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \frac{3^i - 1}{2} \cdot 1 \quad (15)$$

$$\wedge h(a) \leq \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \frac{3^i - 1}{2} \cdot 26 \quad (16)$$

Definition

$$\sum_{n=0}^t c^n = \frac{c^{t+1} - 1}{c - 1}$$

$$\sum_{n=s}^t f(n) \cdot c = c \cdot \sum_{n=s}^t f(n)$$

Nach a_j auflösen

$$h(a) \geq \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \frac{3^i - 1}{2} \cdot 1 \quad (15)$$

$$\wedge h(a) \leq \sum_{n=i+1}^7 a_n 3^n + a_i 3^i + \frac{3^i - 1}{2} \cdot 26 \quad (16)$$

$$a_i \leq \frac{2 \cdot h(a) + (1 - 3^i) \cdot 1}{2 \cdot 3^i} - \sum_{n=i+1}^7 a_n 3^{n-i} \quad (43)$$

$$\wedge a_i \geq \frac{2 \cdot h(a) + (1 - 3^i) \cdot 26}{2 \cdot 3^i} - \sum_{n=i+1}^7 a_n 3^{n-i} \quad (44)$$

$$a_i \leq \frac{2 \cdot h(a) + (1 - 3^i) \cdot 1}{2 \cdot 3^i} - \underbrace{\sum_{n=i+1}^7 a_n 3^{n-i}}_{0 \text{ für } i=7}$$

$$\wedge a_i \geq \frac{2 \cdot h(a) + (1 - 3^i) \cdot 26}{2 \cdot 3^i} - \sum_{n=i+1}^7 a_n 3^{n-i}$$

Definition

$$\sum_{n=s}^t f(n) = 0 \text{ mit } t < s$$

- a_7 nur von $h(a)$ abhängig $\Rightarrow a_7$ zuerst bestimmen
- Sukzessiv den Rest

$$a_i \leq \underbrace{\frac{2 \cdot h(a) + (1 - 3^i) \cdot 1}{2 \cdot 3^i}}_{\text{Nur von } i \text{ abhängig}} - \sum_{n=i+1}^7 a_n 3^{n-i}$$

$$\wedge a_i \geq \underbrace{\frac{2 \cdot h(a) + (1 - 3^i) \cdot 26}{2 \cdot 3^i}}_{\text{(Cacheable!)}} - \sum_{n=i+1}^7 a_n 3^{n-i}$$

$$\Delta = \text{Obergrenze} - \text{Untergrenze}$$

$$\Delta = \left(1 - \frac{1}{3^i}\right) \cdot \frac{25}{2} = 12,5 - \frac{12,5}{3^i}$$

- Δ nur von i abhängig (Cacheable!)
- Untergrenze komplex ausrechnen
- Obergrenze = Untergrenze + Δ

- 1 Überblick
- 2 Hash
- 3 Entschlüsselung**
- 4 Ergebnisse

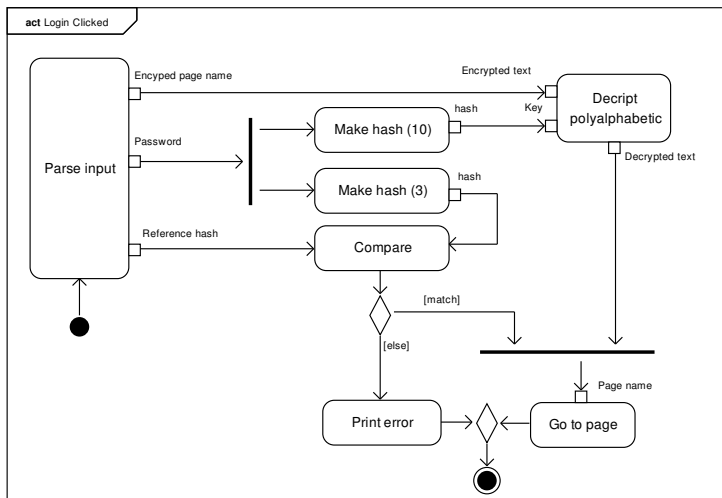


Abbildung: Entschlüsselung im System

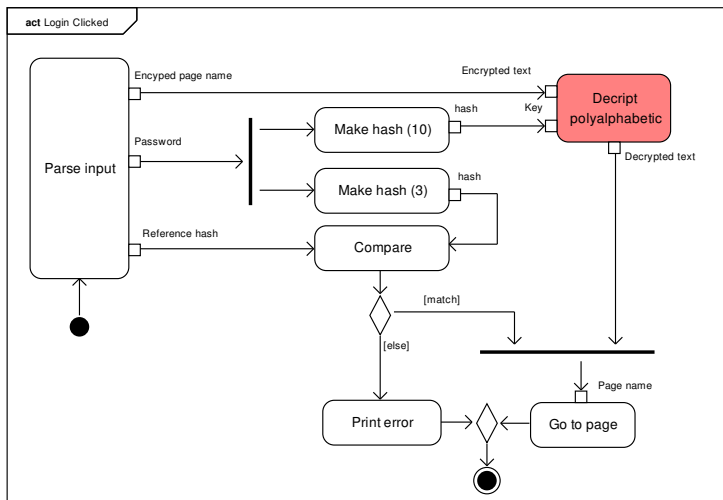


Abbildung: Entschlüsselung im System

```
1 var page = "";  
2 for(var i = 0; i < 8; i++) {  
3     letter = params[2].substring(i, i + 1);  
4     ul = letter.toUpperCase();  
5     a = alpha.indexOf(ul, 0);  
6  
7     a -= (h2.substring(i, i + 1) * 1);  
8     if(a < 0) {  
9         a += 26;  
10    }  
11  
12    page += alpha.substring(a, a + 1);  
13 }
```

params[2] Verschlüsselte User-Startseite

h2 10er Hash (mul = 10) des Passworts

Zeile 3 - 5 Encrypted-Char zu Zahl (links → rechts)

Zeile 7 - 10 Zahl um Hash Ziffer (MSD → LSD) verringern

Zeile 12 Zahl zu Decryped-Char

Graphische Herleitung

1	2	3	4	...	8
Y^{-2}	Y^{-2}	Y^{-2}	Y^{-2}	...	Y^{-2}
Z^{-1}	Z^{-1}	Z^{-1}	Z^{-1}	...	Z^{-1}
A^0	A^0	A^0	A^0	...	A^0
B^1	B^1	B^1	B^1	...	B^1
C^2	C^2	C^2	C^2	...	C^2
D^3	D^3	D^3	D^3	...	D^3
E^4	E^4	E^4	E^4	...	E^4
F^5	F^5	F^5	F^5	...	F^5
G^6	G^6	G^6	G^6	...	G^6
H^7	H^7	H^7	H^7	...	H^7
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
Y^{25}	Y^{25}	Y^{25}	Y^{25}	...	Y^{25}
X^{26}	X^{26}	X^{26}	X^{26}	...	X^{26}

■ Encrypted text: FDHB...F

■ Key: 1414...3

■ Decrypted text: EAGY...D

- Polyalphabetische Substitution
- Vigenère ähnlich
- Dezimalstellen des 10er Hash sind Schlüssel!!

Definition (Vigenère-Verschlüsselung)

Mehrfache Caesar-Substitution (Buchstaben k_i im Klartext um s_i aus Schlüssel Verschieben)

- 1 Überblick
- 2 Hash
- 3 Entschlüsselung
- 4 Ergebnisse**

- archive.org
 - Alte Passwörter, gleiche Seite
 - Alte User, gleiche Seite
 - ⇒ Regelmäßige passwortänderungen sind unsicher xD
- Mehrere User/Passwörter, gleiche Seite
- Dictionary attack

- Rekursiv, 4s (verbesserbar: Iterativ!)
- ~ 30 000 mögliche Passwörter (min 1)
Von Ursprünglich ~ $2,09 \cdot 10^{11}$

How to not protect a webpage

A simpel collision attack

Skrupellos

JID: skrupellos@swissjabber.de

μ CCC, mehrvortragewagen

19. August 2010