How to not protect a webpage

A simpel collision attack

Skrupellos
JID: skrupellos@swissjabber.de

μCCC, mehrvorträgewagen

19. August 2010

Kapitel Übersicht

Eigenschaften

- 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

Abbildung: Login page generator

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

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

$$p \in P^8$$

Login page

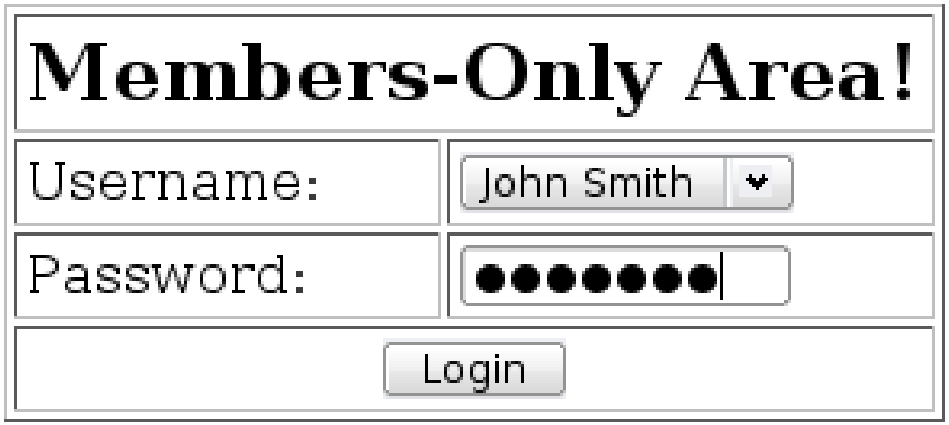


Abbildung: “Members only!” O RLY?

Security by ... poor formated code

```
var params=new Array(4);
var alpha="ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHI";
function check(form) {
which=form.memlist.selectedIndex;
choice = form.memlist.options[which].value+"|";
if (choice=="x|") {
alert("Please Select Your Name From The List");
return;
}
p=0;
for (i=0;i<3;i++) {
a=choice.indexOf("|",p);
params[i]=choice.substring(a,p);
p=a+1;
}
h1=makehash(form.pass.value,3);
h2=makehash(form.pass.value,10)+" ";
if (h1!=params[1]) {
alert("Incorrect Password!"); return; };
var page="";
for (var i=0;i<8;i++) {
letter=params[2].substring(i,i+1)
ul=letter.toUpperCase();
a=alpha.indexOf(ul,0);
a-=(h2.substring(i,i+1)*1);
if (a<0) a+=26;
page+=alpha.substring(a,a+1); };
top.location=page.toLowerCase()+".html";
}
function makehash(pw,mult) {
pass=pw.toUpperCase();
hash=0;
for (i=0;i<8;i++) {
letter=pass.substring(i,i+1);
c=alpha.indexOf(letter,0)+1;
hash=hash*mult+c;
}
return(hash);
}
```

Abbildung: Vorher [sic]

```
var params = new Array(4);
var alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHI";

function check(form) {
    ///// PARSE INPUT
    which = form.memlist.selectedIndex;
    choice = form.memlist.options[which].value + "|";

    // Check for invalid selection
    if(choice == "x|") {
        alert("Please Select Your Name From The List");
        return;
    }

    // Get parameters from selection
    p = 0;
    for(i = 0; i < 3; i++) {
        a = choice.indexOf("|", p);
        params[i] = choice.substring(a, p);
        p = a + 1;
    }

    ///// CREATE HASH 1
    h1 = makehash(form.pass.value, 3);

    ///// CREATE HASH 2
    h2 = makehash(form.pass.value, 10) + " ";

    ///// COMPARE
    if(h1 != params[1]) {
        alert("Incorrect Password!");
        return;
    }

    ///// DECRYPT
    var page = "";
    for(var i = 0; i < 8; i++) {
        letter = params[2].substring(i, i + 1);
        ul = letter.toUpperCase();
        a = alpha.indexOf(ul, 0);

        a -= (h2.substring(i, i + 1) * 1);
        if(a < 0) {
            a += 26;
        }

        page += alpha.substring(a, a + 1);
    }

    ///// GO TO PAGE
    top.location = page.toLowerCase() + ".html";
}

function makehash(pw, mult) {
    pass = pw.toUpperCase();
    hash = 0;

    for(i = 0; i < 8; i++) {
        letter = pass.substring(i, i + 1);
        c = alpha.indexOf(letter, 0) + 1;
        hash = hash * mult + c;
    }

    return(hash);
}
```

Abbildung: Nacher

Ablauf

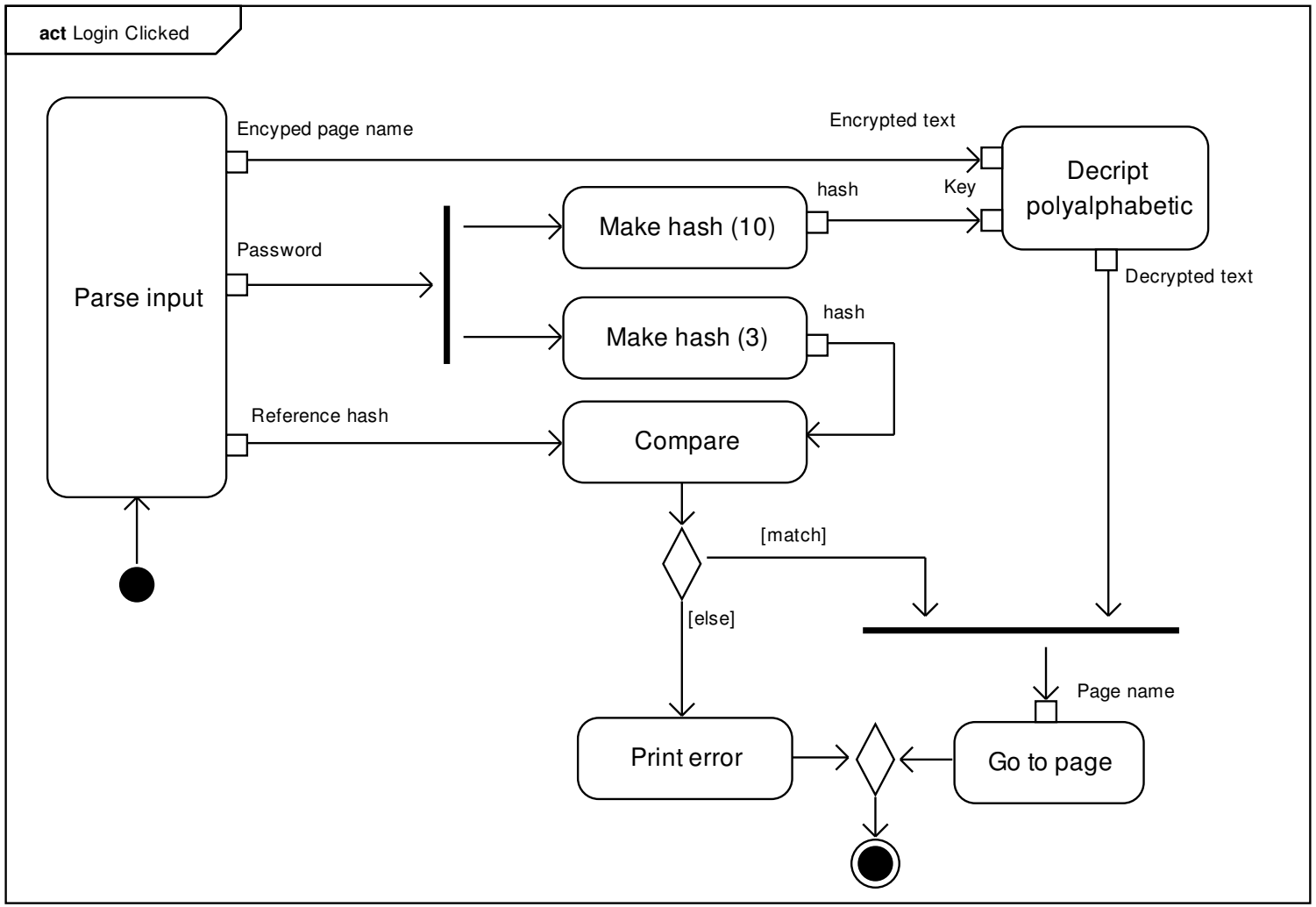


Abbildung: Anmelde Funktion

Kapitel Übersicht

Systemübersicht

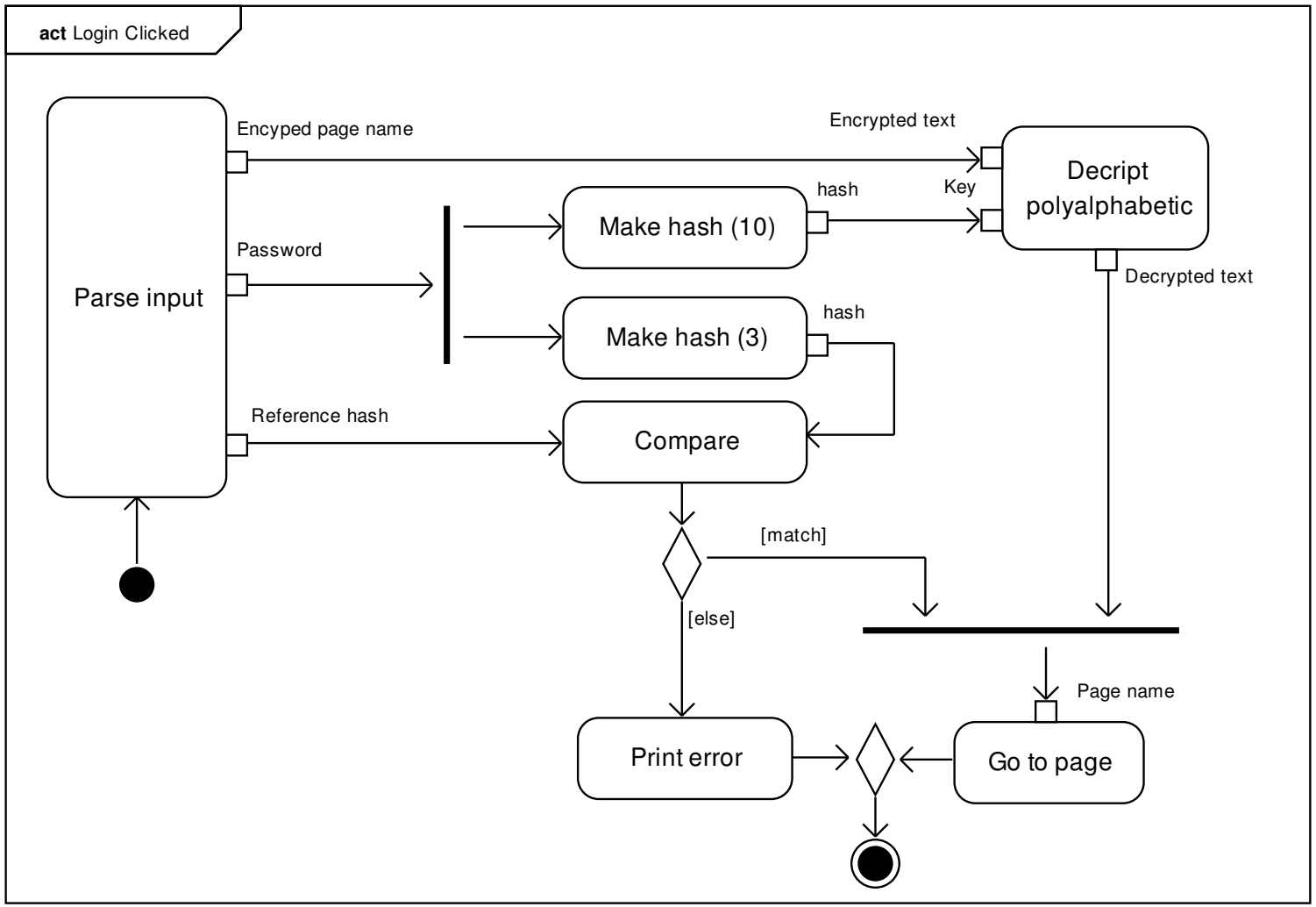


Abbildung: Hashing im System

Systemübersicht

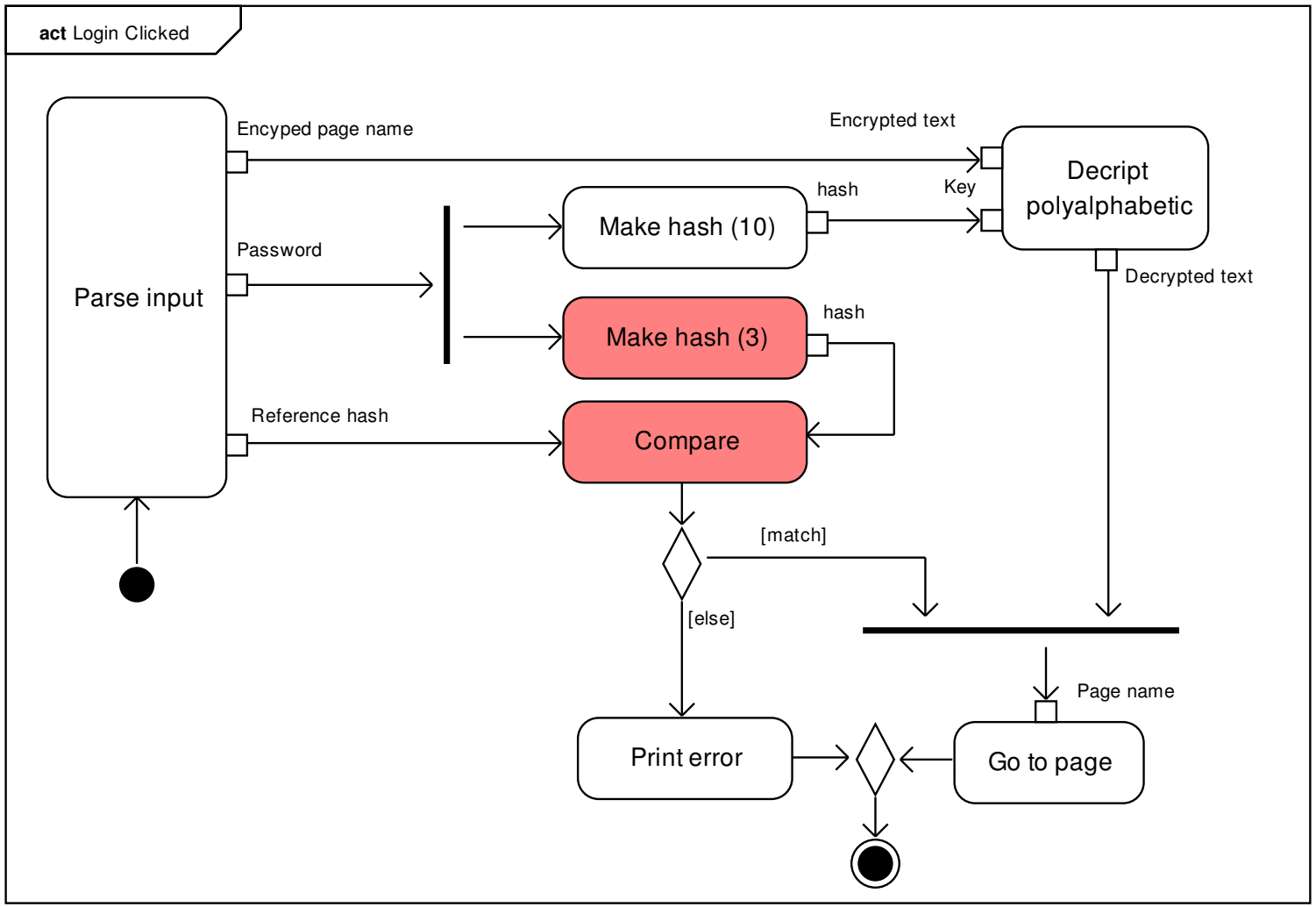


Abbildung: Hashing im System

Einzelschrittvorbereitung

```
1  var alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHI";
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 }
```

Zeile 8 - 9

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

$$t\colon P \longrightarrow A$$

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

Einzelschritt

```
 1 var alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHI";
 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

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

$$\sum_{i=0}^{4} a_i 3^i$$

a_i isolieren

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

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

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

Definition

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

Definition

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

Obergrenze

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

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*.

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

Untergrenze

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

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

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

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

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

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

Definition

$\sum_{n=0}^{t} c^n = \frac{c^{t+1}-1}{c-1}$ $\qquad$ $\sum_{n=s}^{t} f(n) \cdot c = c \cdot \sum_{n=s}^{t} f(n)$

Nach a_i auflösen

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

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

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

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

Anwendung

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

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

Definition

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

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

Cache 1

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

(Cacheable!)

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

Cache 2

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

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

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

Kapitel Übersicht

Systemübersicht

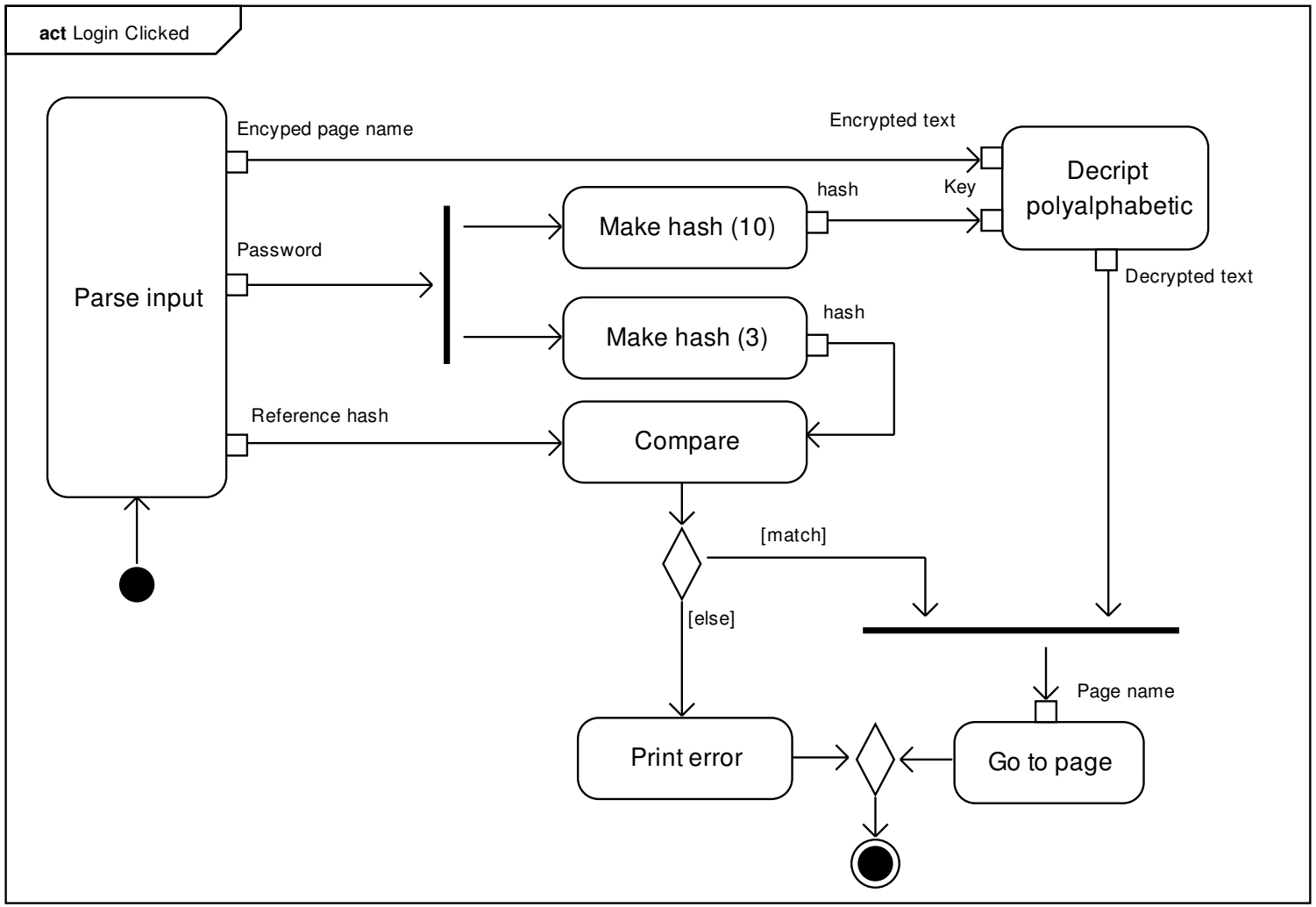


Abbildung: Entschlüsselung im System

Systemübersicht

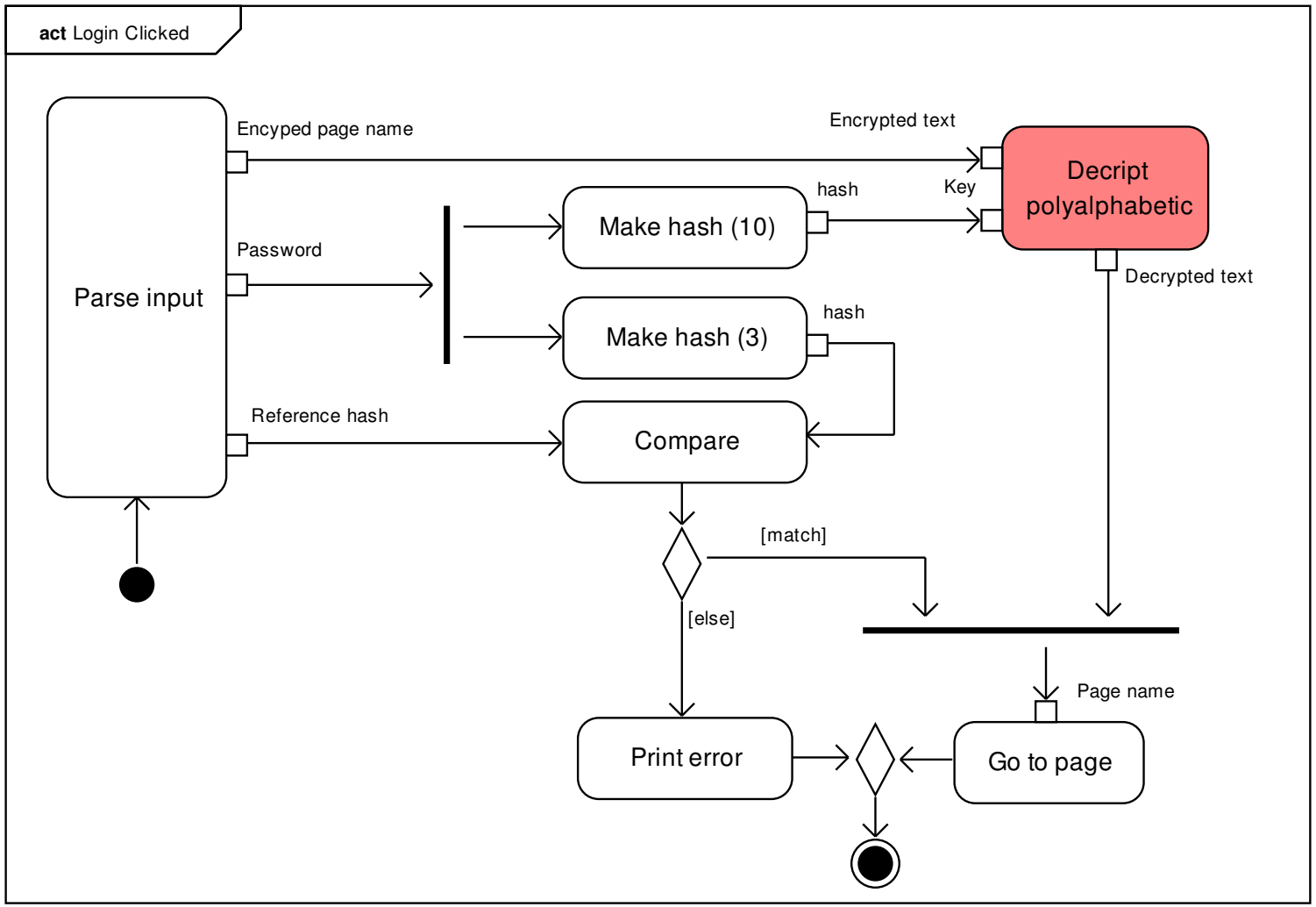


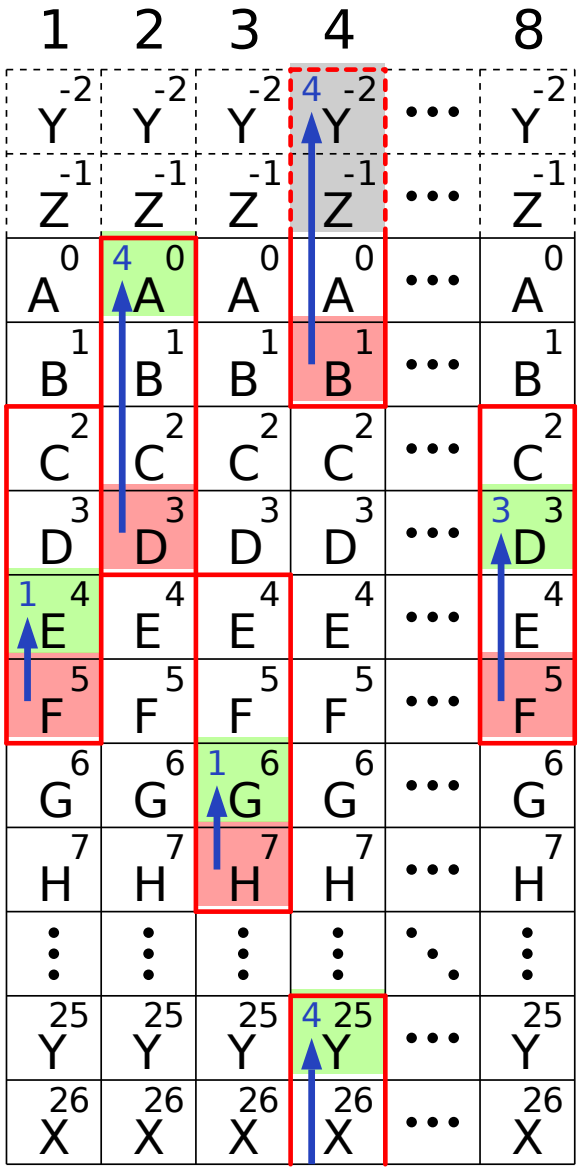
Abbildung: Entschlüsselung im System

Herleitung

```
 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



Encrypted text: FDHB...F

Key: 1414...3

Decrypted text: EAGY...D

Zusammenfassung

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

Definition (Vigenère-Verschlüsselung)

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

Kapitel Übersicht

Passwort eingränzen

- archive.org
  - Alte Passwörter, gleiche Seite
  - Alte User, gleiche Seite
  - $\Rightarrow$ Regelmäsige passwortänderungen sind unsicher xD
- Mehrere User/Passwörter, gleiche Seite
- Dictionary attack

Meine Implementierung

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

How to not protect a webpage

A simpel collision attack

Skrupellos
JID: skrupellos@swissjabber.de

μCCC, mehrvorträgewagen

19. August 2010