Ludwig-Maximilans-
Universität München
Dr. Pascal Reisert

Wintersemester 2018/2019
Übungsblatt 7
28. November 2018

Übungen zur Kryptographie
Lösung

Aufgabe 25

a) Für die Eulersche φ-Funktion erhalten wir z.B. mit

```
function factorlist(x: integer; verbose := 1): array;
var
st, st1: stack;
q, y, bound: integer;
vec: array;
count: integer;
begin
x := abs(x);
 if x < 2 then
return ();
end;
q := 2;
while q := factor16(x,q) do
stack_push(st,q);
x := x div q;
end;
if x < 2**32 then
stack_push(st,x);
else
stack_push(st1,x);
end;
while not stack_empty(st1) do
x := stack_pop(st1);
if rab_primetest(x) then
stack_push(st,x);
else
bound := bit_length(x);
bound := 4*bound**2;
y := rho_factorize(x,bound,verbose);
count := 0;
while (y <= 1 or y >= x) do
if inc(count) > 2 then
writeln("unable to factorize ",x);
return ();
end;
y := qs_factorize(x,verbose);
```

```
end;
stack_push(st1,x div y);
stack_push(st1,y);
end;
end;
vec := stack2array(st);
return sort(vec);
end.

function eulerphi(N:integer):integer;
var
a:array;
k,vp,m:integer;
begin
a:=factorlist(N);
vp:=a[0]-1;
for k:=0 to length(a)-2 do
if a[k+1]=a[k] then vp:=vp*a[k];
else vp:=vp*(a[k+1]-1);
end;
end;
return vp;
end.
```

die Werte

$$\varphi(561) = 2 \cdot 10 \cdot 16 = 320,\ \varphi(1105) = 4 \cdot 12 \cdot 16 = 768,\ \varphi(1729) = 6 \cdot 12 \cdot 18 = 1296,$$

$$\varphi(2000436751) = 486 \cdot 1530 \cdot 2682 = 1994281560.$$

Für eine Carmichaelzahl N gilt $N = p_1 \cdot \cdots \cdot p_r$ und $\alpha_i(p_i - 1) = N - 1$ für gewisse $\alpha_i \in \mathbb{Z}$. In $\mathbb{Z}/p_i$ gibt es genau $\frac{p_i-1}{2}$ Lösungen von $a^{\frac{p_i-1}{2}} = 1 \bmod p_i$ und ebenso $\frac{p_i-1}{2}$ Lösungen von $a^{\frac{p_i-1}{2}} = -1 \bmod p_i$. Sei $M = \{i : \alpha_i = 1 \bmod 2\}$ die Menge der Indizes mit α_i ungerade. Somit gibt es in $\mathbb{Z}/N$ genau $2^{-|M|} \prod_{i=1}^{r}(p_i - 1)$ Lösungen von $a^{\frac{N-1}{2}} = 1 \bmod N \equiv (1, \ldots, 1) \in \prod_{i=1}^{r} \mathbb{Z}/p_i$. Für $|M| = r$ gibt es weitere $2^{-r} \prod_{i=0}^{r}(p_i - 1)$ Lösungen von $a^{\frac{N-1}{2}} = -1 \bmod N \equiv (-1, \ldots, -1) \in \prod_{i=1}^{r} \mathbb{Z}/p_i$.

Da p_i prim ist, folgt $a^{\frac{p_i-1}{2}} = \left(\frac{a}{p_i}\right)$. Weiter gilt $\left(\frac{a}{N}\right) = \prod_{i=1}^{r} \left(\frac{a}{p_i}\right)$. Sei zunächst $|M| \neq r$. Uns interessieren nur die Stellen, an denen $\left(\frac{a}{p_i}\right) \neq 1$ sein kann, also die Stellen mit α_i gerade. Damit unser Produkt wieder 1 ist, muss die Anzahl der -1 an diesen Stellen gerade sein. Damit erfüllen $\sum_{k=0}^{\left\lfloor \frac{r-|M|}{2} \right\rfloor} \binom{r-|M|}{2k} = 2^{r-|M|-1}$ der $2^{r-|M|}$ Kombinationen auch die Jacobi-Bedingung. Somit erhalten wir $FW_{SS}(N) = 2^{-|M|-1} \prod_{i=1}^{r}(p_i - 1)$.

Ist $|M| = r$ ungerade, so erhalten wir für alle Lösungen $a^{\frac{N-1}{2}} = \pm 1 \bmod N$ bereits $a^{\frac{N-1}{2}} = \left(\frac{a}{N}\right)$ und damit $FW_{SS}(N) = 2^{-r+1} \prod_{i=1}^{r}(p_i - 1)$. Ist r gerade so ist $a^{\frac{N-1}{2}} = -1 \bmod N \neq 1 = \left(\frac{a}{N}\right)$ und wir erhalten nur $FW_{SS}(N) = 2^{-r} \prod_{i=1}^{r}(p_i - 1)$.

In `aribas` erhalten wir

```
function fwss(N:integer);
var
a,b:array;
k,vp,m:integer;
begin
a:=factorlist(N);
b:=a;
if (N-1) mod (a[length(a)-1]-1) <>0 then
return "keine Carmichaelzahl";
end;
for k:=0 to length(a)-2 do;
if a[k]=a[k+1] then
return "keine Carmichaelzahl";
end;
if (N-1) mod (a[k]-1) <>0 then
return "keine Carmichaelzahl";
else b[k]:=(N-1) div (a[k]-1);
end;
end;
vp:=1;
for k:=0 to length(a)-1 do
vp:=(a[k]-1)*vp;
end;
b[length(a)-1]:=(N-1) div (a[length(a)-1]-1);
for k:=0 to length(a)-1 do;
if b[k] mod 2 = 1 then
m:=m+1;
end;
end;
if m<length(a) then
return vp div 2**(m+1);
elsif m = 0 mod 2 then
return vp div 2**(length(a));
else
return vp div 2**(length(a)-1);
end;
end.
```

Es folgt

$$\#FW_{SS}(561) = 80,\ \#FW_{SS}(1105) = 192,\ \#FW_{SS}(1729) = 684,$$

$$\#FW_{SS}(2000436751) = 498570390.$$

Kommen wir schließlich zum Miller-Rabin-Test: Wir verwenden die Notation von zuvor. Aus Aufgabe 18 wissen wir bereits, dass in $\mathbb{Z}/p_i$ die Gleichung $a^{2^t u} \bmod p_i = 1$ genau $\gcd(2^t u, p_i - 1)$ Lösungen hat. Gibt es ein $a^{2^t u} \bmod p_i = -1$, dann gibt es bereits $\gcd(2^t u, p_i - 1)$ Lösungen dieser Gleichung (die Produkte mit Lösungen von $b^{2^t u} \bmod p_i =$

1). Weiterhin hat $a^{2^t u} \bmod p_i = -1$ immer eine Lösung solange $\gcd(2^t u, p_i - 1) \neq p_i - 1$.[1] Wir erhalten $FW_{MR}(N) = \prod_{i=1}^{r} \gcd(u, p_i - 1) + \sum_{k=0}^{t} \prod_{i=1}^{r} \gcd(2^k u, p_i - 1) \cdot 1_{\gcd(2^k u, p_i - 1) \neq 1}$. Mittels der folgenden Funktion erhalten wir in `aribas`

```
function fwmr(N:integer);
var
a:array;
n,t,m,k,vp,su,l:integer;
begin
a:=factorlist(N);
n:=N-1;
while n mod 2 = 0 do;
n:=n div 2;
t:=t+1;
end;
vp:=1;
for k:=0 to length(a)-1 do;
vp:=vp*gcd(n,a[k]-1);
end;
su:=vp;
for l:=0 to t-1 do;
vp:=1;
for k:=0 to length(a)-1 do;
if gcd(n*2**l,a[k]-1)<> a[k]-1 then
vp:=vp*gcd(n*2**l,a[k]-1);
else
vp:=0;
end;
end;
su:=vp+su;
end;
return su;
end.
```

und damit

$$\#FW_{MR}(561) = 10,\ \#FW_{MR}(1105) = 30,\ \#FW_{SS}(1729) = 162,$$

$$\#FW_{MR}(2000436751) = 498570390.$$

Bemerkung. Es ist nun einfach auch für große Carmichaelzahlen die Anzahl der falschen Zeugen zu bestimmen. Z.B. für

$$t = 10000000000000000000000000000000000000000000091400$$

und $N := (6t + 1)(12t + 1)(18t + 1)$ ist

$$\begin{aligned} FW_{SS}(N) = {} & 32400000000000000000000000000000000000000000088840800000000 \\ & 00000000000000000000000000000000812004912000000000000000000000000 \\ & 00000000000024739082985600000 0 \end{aligned}$$

[1] $2^t uk / \gcd(2^t u, p_i - 1) = j(p-1)/(2\gcd(2^t u, p_i - 1)) \Rightarrow k = j(p-1)/2$. Wähle z.B. $j = 1$.

und

$$FW_{MR}(N) = 92707031250000000000000000000000000000000000254202679687500000$$
$$000000000000000000000000000000232341249234375000000000000000000000000000$$
$$000000070786633933406250$$

während

$$\varphi(N) = 12960000000000000000000000000000000000000000035536320000000$$
$$000000000000000000000000000000032480196480000000000000000000000000000000$$
$$000000000009895633194240000000.$$

Man beachte $\#FW_{SS}(N) = \frac{\varphi(N)}{4}$.

b) Ist $a \in FW_{MR}(N)$ und $N = 3 \bmod 4$, dann ist $(N-1)/2$ ungerade, also impliziert $a^{(N-1)/2} = \pm 1 \bmod N$ bereits

$$\left(\frac{a}{N}\right)^{(N-1)/2} = \left(\frac{a^{(N-1)/2}}{N}\right) = \left(\frac{\pm 1}{N}\right) = \pm 1 = a^{(N-1)/2} \Rightarrow a^{(N-1)/2} = \left(\frac{a}{N}\right)$$

und $a \in FW_{SS}(N)$ folgt. Anderseits ist $2^t u = 2 \bmod 4$ nur für $t = 1$ erfüllt, also $(N-1)/2 = u$ und trivial $FW_{SS}(N) \subset FW_{MR}(N)$.

Aufgabe 26

Man beachte, dass $p-1 = 2(q-1)$ und $pq-1 = 2q^2-q-1 = q(q-1)+q^2-1 = (q-1)(2q+1)$. Sei $a \in \mathbb{Z}/p \times \mathbb{Z}/q$. Wie zuvor gilt $a^{(q-1)/2} \bmod q = \left(\frac{a}{q}\right)$. Weiterhin wissen wir mit $2q+1$ ungerade, dass $a^{(N-1)/2} \bmod q = \left(\frac{a}{q}\right)$. Aus $(q-1)(2q+1) \bmod p-1 = q-1 \bmod p-1 = \frac{p-1}{2} \bmod p-1$ folgt $a^{(N-1)/2} \bmod p = a^{(p-1)/4} \bmod p$. Damit ist $a^{(N-1)/2} \bmod p$ für je $\frac{p-1}{4}$ Zahlen 1 oder -1, das Jacobisymbol aber immer 1 - aus $a^{(p-1)/4} \bmod p = \pm 1$ folgt $\left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p = (\pm 1)^2 = 1$. Wir erhalten die Kombinationen $(1,1)$ und $(-1,-1)$ jeweils in $\frac{p-1}{4}\frac{q-1}{2}$ Fällen. Da dann auch $\left(\frac{a}{N}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right) = \left(\frac{a}{q}\right)$ gilt, sind diese Elemente auch schon in $FW_{SS}(N)$. Es gilt $\#FW_{SS}(N) = \frac{\varphi(N)}{4}$.

Aufgabe 27

a) Für $N_1 = 551$ ist $x_{10} = 24$ und für $k = 0$ ergibt sich $24^2 - 551 = 25 = 5^2$. Also $511 = (24-5)(24+5) = 19 \cdot 29$.
Für $N_2 = 1643$ ist $x_{20} = 41$ und für $k = 0$ ergibt sich $41^2 - 1643 = 38$, was keine Quadratzahl ist. Für $k = 1$ gilt aber $42^2 - 1643 = 121 = 11^2$. Also $1643 = (42-11)(42+11) = 31 \cdot 53$.
Für $N_3 = 2479$ ist $x_{30} = 50$ und für $k = 0, 1$ ergeben sich keine Qudratzahlen. Für $k = 2$ erhalten wir $52^2 - 2479 = 225 = 15^2$ und $2479 = 37 \cdot 67$.

b) Nach Voraussetzung gilt $N = uv$. Setzt man $x = \frac{u+v}{2}$ und $y = \frac{u-v}{2}$ für (o.B.d.A) $v \leq u$, dann folgt $x+y = u$ und $x-y = v$, also $(x+y)(x-y) = N \Leftrightarrow x^2 - N = y^2$ mit $x \geq \sqrt{N}$

c) Aus $|u-v| \leq \alpha\sqrt[4]{N}$ folgt wie zuvor $y \leq \frac{\alpha}{2} \cdot \sqrt[4]{N}$. Sei $x = x_0 + k$ und $x^2 = x_0^2 + 2x_0k + k^2$. Dann gilt $y^2 = x^2 - N \geq 2\sqrt{N}k + k^2$. Wir erhalten

$$2\sqrt{N}k + k^2 \leq \frac{\alpha^2}{4} \cdot \sqrt{N} \Rightarrow 2k + k\frac{k}{\sqrt{N}} \leq \frac{\alpha^2}{4}.$$

Ist $k \leq \sqrt{N}$ so gilt

$$3k \leq \frac{\alpha^2}{4} \Rightarrow k \leq \frac{\alpha^2}{12}.$$

Allgemein erhalten wir

$$k \leq -\sqrt{N} + \sqrt{N + \frac{\alpha^2}{4} \cdot \sqrt{N}} \leq \frac{\alpha}{2}\sqrt[4]{N},$$

wobei wir verwendet haben, dass die Wurzelfunktion konkav ist.
Insbesondere folgt aus

$$-\sqrt{N} + \sqrt{N + \frac{\alpha^2}{4} \cdot \sqrt{N}} \leq \frac{\alpha^2}{12} \Leftrightarrow \frac{\alpha^2}{4}\sqrt{N} \leq \frac{\alpha^4}{12^2} + 2 \cdot \frac{\alpha^2}{12}\sqrt{N} \Leftrightarrow \alpha > 2\sqrt{3}\sqrt[4]{N}.$$

$$k > \sqrt{N} \Rightarrow 3N < k^2 + 2k\sqrt{N} \leq y^2 \leq \frac{\alpha^2}{4}\sqrt{N} \Rightarrow \alpha > 2\sqrt{3}\sqrt[4]{N}$$

bereits, dass $k \leq \frac{\alpha^2}{12}$ zwar immer gilt, aber nicht besonders stark ist.

Aufgabe 28

a) Ist $2y = p - q > 2^{m-7}$ und $p > q$, dann ist $y \geq 2^{m-8}$ und $2x_0k + k^2 \geq 2^{2m-16} \Rightarrow k \geq -x_0 + \sqrt{x_0^2 + 2^{2m-16}} \geq -2^m - 1 + \sqrt{2^{2m} + 2^{2m-16}} = -2^m - 1 + 2^m\sqrt{1 + 2^{-16}} = -1 + \frac{1}{2} \cdot 2^{m-16} + O(2^{m-32})$. D.h. k ist (für große m) sehr groß und wird nur sehr langsam gefunden.

b) Hier noch einmal N im Dezimalsystem:

```
1009_77042_42321_01394_03849_48381_68815_42391_80009_88180_16303_45296_
78321_05511_34893_92826_50371_25189_11722_24736_69299_00324_91078_50166_28383_
88874_39546_36853_94159_65198_28475_67185_91258_15631_37965_52614_93050_98708_
15829_37892_89181_84998_51930_79460_70925_41356_23715_34593_60597_51711_17508_
09140_08219_33221_78001_66171_35115_29696_51184_65816_73369_97987
```

Wir sehen, dass $2^{1023} < N < 2^{1024}$, also $m = 512$. Wir wissen jetzt, dass $|p - q| < 2^{m-7}$. Setzt man $p = N/q$, so erhält man eine quadratische Ungleichung $q^2 + 2^{m-7}q - N \geq 0$ und somit

```
q_min := 9996_49561_13697_11184_09221_19787_24191_41099_85835_08020_16031_03983_
10859_05501_00356_96364_30153_19840_89120_14502_66299_03497_31091_17098_11424_
59344_58055_39924_32501_58991_76079
```

Implementiert man den fermatschen Faktorisierungsalgorithmus, z.B. mit `aribas`

```
function fer_factorize(N:integer):array;
var
k,n:integer;
y:real;
v:array[2];
begin
n:=isqrt(N)+1;
for k:=0 to N do;
y:=sqrt((n+k)**2-N);
if floor(y)=y then v[0..1]:=(n+k+floor(y),n+k-floor(y));
return v;
break;
end;
end;
end.
```

so erhält man

```
p = 10048_73337_40730_79634_67536_53643_05329_09112_41447_08262_97113_79013_
01157_48230_22477_43433_47702_64940_88851_94486_11772_27726_60569_85671_01305_
05064_98828_74954_44691_89122_38711
q = 10048_73337_40730_79634_67536_53643_
05329_09112_41447_08262_97113_79013_01157_48230_22477_37418_87478_30649_28665_
86145_22991_63698_10857_82177_64071_25870_03321_34428_19510_07131_00917.
```

Sei $\varphi(N) = (p-1)(q-1)$, dann erhalten wir mit der `aribas`-Funktion `mod_inverse` d mit $de = 1 \bmod \varphi(N)$.

```
==> d:=mod_inverse(e,vp).
-: 437_57694_20051_52503_02414_29025_43515_23626_76232_97745_03914_09530_
93127_82039_49387_17919_23196_72175_57912_50477_16680_52691_26548_81131_61208_
20971_86064_61794_89359_20344_71042_36344_53608_90648_36580_42604_76917_20106_
49434_20082_75988_03391_89570_70829_28953_18641_38689_73302_39806_65137_12365_
52669_22760_68708_55369_41614_34284_75247_84887_57890_34440_25473
```

Nun können wir den Geheimtext y mittel $y^d \bmod N$ in den Klartext

```
40_89616_35403_94925_94836_72057_83924_52018_68163_86110_51782_38337_50685_
28462_19497_04916_18787_94378_64383_06805_84959_30357_58954_03085_99719_52551_
68166_54687_35771_81338_10604_85515_31335_74428_67848_14580_26840_62218_98491_
36984_03243_72524_66457_46929_15997_85500_14308_30204_01828_36893_57304_16996_
28895_13597_14104
```

umwandeln. Umwandlung in `Ascii`-Zeichen liefert

```
x:=y**d mod N;
b:=byte_string(x);
mem_byteswap(b,length(b));
string(b).
```

wobei `mem_byteswap` entsprechend der Vorgabe $x = \sum_{i=1}^{n} a_i 2^{n-i}$ die Reihenfolge der `length(b) = 113` Bytes in b umkehrt. Wie erhalten den Klartext:

```
Michael O. Rabin: Probabilistic Algorithm for Testing Primality,
Journal of Number Theory, Vol. 12 (1980) 128-138
```