Das Paillier Cryptosystem mit Beispielen in Cran-R

Thomas Liebig
thomas.liebig@tu-dortmund.de

15. Dezember 2015

Das Kryptosystem besteht aus drei Komponenten: (1) Schlüsselgenerierung, (2) Verschlüsselung, und (3) Entschlüsselung.

1 Schlüsselgenerierung

Wähle zwei Primzahlen p und q. $n = p \cdot q$, g mit $|g|$ ist n in $\mathbb{Z}^*_{n^2}$. Der öffentliche Schlüssel ist dann das Tupel (n, g).

Carmichael's function $\lambda(n)$ ist das kleinste m zu einer Zahl n, so dass $\forall a, ggT(a, n) = 1$ gilt: $a^m \equiv 1 \bmod n$. Wenn p und q Primfaktoren von n sind, ist $\lambda(n) = kgV(\phi(p), \phi(q))$. Für den Fall, dass p und q Primzahlen sind gilt damit: $\lambda(n) = kgV(p-1, q-1)$. λ ist der private Schlüssel.

2 Eigenschaften

$\forall w \in \mathbb{Z}^*_{n^2}$: $w^\lambda \equiv 1 \bmod n$ und $w^{n\lambda} \equiv 1 \bmod n^2$. Die Menge $\mathcal{S}_n = \{u < n^2 | u = 1 \bmod n\}$ ist eine multiplikative Teilmenge modulo n^2 auf der die Funktion $L(u) = (u-1)/n$ für alle $u \in \mathcal{S}_n$ wohldefiniert ist.

3 Verschlüsselung

$$\begin{aligned} \mathcal{E}_g : \mathbb{Z}_n \times \mathbb{Z}^*_n &\rightarrow \mathbb{Z}^*_{n^2} \\ \mathcal{E}_g(x, y) &\rightarrow g^x \cdot y^n \bmod n^2 \\ \mathcal{E}_g(m, r) &\rightarrow g^m \cdot r^n \bmod n^2 \end{aligned}$$

4 Entschlüsselung

$$m = L(c^\lambda \bmod n^2) * L^{-1}(g^\lambda \bmod n^2)$$

mit $L(x) = (x-1)/n$

5 Beispiel

```
require("pracma")   # provides gcd()
require("numbers")  # provides isPrime(), modpower()

p=3
q=5
```

Schlüsselgenerierung

```
n=p*q
phi=(p-1)*(q-1)
g=n+1

lambda=Lcm((p-1),(q-1)) # = (p-1)*(q-1)/gcd((p-1),(q-1))
```

$n = 15$, $\phi(n) = 8$, $\lambda(n) = 4$, $g = 16$

Berechnung von $A = \mathbb{Z}^*_{n^2}$

```
A=matrix(ncol=n,nrow=phi)
j=0
for (i in 1:phi) {
 j=j+1
 while (gcd(j,n)!=1) {j=j+1}
 for (m in 0:(n-1)) {
  A[i,(m+1)]=(modpower(g,m,(n^2)) * modpower(j,n,(n^2))) %% n^2
 }
}
```

Betrachtung von A

```
>print(A)
```

r\m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	16	31	46	61	76	91	106	121	136	151	166	181	196	211
2	143	38	158	53	173	68	188	83	203	98	218	113	8	128	23
4	199	34	94	154	214	49	109	169	4	64	124	184	19	79	139
7	118	88	58	28	223	193	163	133	103	73	43	13	208	178	148
8	107	137	167	197	2	32	62	92	122	152	182	212	17	47	77
11	26	191	131	71	11	176	116	56	221	161	101	41	206	146	86
13	82	187	67	172	52	157	37	142	22	127	7	112	217	97	202
14	224	209	194	179	164	149	134	119	104	89	74	59	44	29	14

Betrachtung von A^λ

```
>modpower(A,lambda,n^2)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[2,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[3,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[4,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[5,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[6,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[7,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
[8,]    1   61  121  181   16   76  136  196   31    91   151   211    46   106   166
```

Betrachtung von $L(A^\lambda)$ mod n^2

```
>((modpower(A,lambda,n^2)-1)/n)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[2,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[3,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[4,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[5,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[6,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[7,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
[8,]    0    4    8   12    1    5    9   13    2     6    10    14     3     7    11
```

Betrachtung von $L(A^\lambda)/L(g^\lambda) \bmod n$

$L(g^\lambda \bmod n^2) = L(61) = 60/15 = 4$
$L^{-1}(g^\lambda \bmod n^2) = 4$

```
> Lg = (modpower(g,lambda,n^2)-1)/n
> Lg_inv = (extGCD(Lg,n)[2]+n) %% n
>(((modpower(A,lambda,n^2)-1)/n)  * Lg_inv) %% n
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[2,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[3,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[4,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[5,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[6,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[7,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
[8,]    0    1    2    3    4    5    6    7    8     9    10    11    12    13    14
```

Homeomorphie

$$92 = (191 * 37) \bmod 15^2$$

Entschlüsselung

$$(((((77^4) \bmod (15^2)) - 1)/15) * 4) \bmod 15 = 14$$

```
(((((7^4) %% (15^2))-1)/15) * 4) %% 15 = 10
(((((164^4) %% (15^2))-1)/15) * 4) %% 15 = 4
(((((82^4) %% (15^2))-1)/15) * 4) %% 15 = 0
(((((98^4) %% (15^2))-1)/15) * 4) %% 15 = 9
```

6 Quelltext

```
require("pracma")   # provides gcd()
require("numbers")  # provides isPrime(), modpower()

p=3
q=5
n=p*q
phi=(p-1)*(q-1)
g=n+1
lambda=Lcm((p-1),(q-1)) # = (p-1)*(q-1)/gcd((p-1),(q-1))

encrypt <- function(m,n) {
  r=n
  while (gcd(r,n)!=1) {
    r=sample(1:n,1)
  }
  c=(modpower(g,m,n^2)*modpower(r,n,n^2)) %% (n^2)
  return(c)
}

decrypt <- function(c,lambda,n) {
  c=c %% n^2
  Lg = (modpower(g,lambda,n^2)-1)/n
  Lg_inv = (extGCD(Lg,n)[2]+n) %% n

  m = ((modpower(c,lambda,n^2)-1)/n * Lg_inv) %% n
  return(m)
}

# Korrektheit
if (5 == decrypt(encrypt(5,n),lambda,n)) {
   print("m = D(E(m)) seems to be ok")
```

```
}

# Homeomorphism
if (5 == decrypt(encrypt(2,n)*encrypt(3,n),lambda,n)) {
   print("m1+m2 = D(E(m1)*E(m2)) seems to be ok")
}
```

References

[1] Michael O Keeffe. The paillier cryptosystem. *A Look Into The Cryptosystem And Its Potential Application, college of New Jersey*, 2008.

[2] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology EUROCRYPT 99*, pages 223–238. Springer, 1999.

[3] Andreas Steffen. The paillier cryptosystem. http://slideplayer.com/slide/8488065/, 2010. [Online; accessed 14-Dezember-2015].