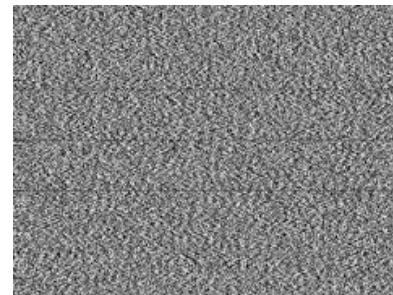


# Bildverschlüsselung mit Matlab

Andy Pollari & Marco Berger



```
03003802 996CB7BA 0EG0161B G0021C06  
BA7CE203 G0030200 01208600 37D14D00  
1B7125G0 024FG002 53D03C00 AD722500  
1BD03C00 887525C1 01A07700 37D14D00  
B7125G0 024FG002 53D03C00 AD722500  
BD03C00 887525C1 4F553E00 53414241  
F4F3D41 4242434E 3D4A6000 64692041  
16C2F4F 553D4553 414A6000 4F3D414  
425604 00312E30 00424101 0003424  
003042 4C000000 024E4E4F 00B1D31  
1254F1 21000009 8833B0CC 2957EE  
13ECAA CB3EE8EF DF038D7F A14217  
2AA4D 04143B75 4F571C83 535C00  
7DED9 B57C659E C820EE07 FA49F  
196DB 7D7F743D 9A36DD29 454E0  
014D 410800C8 9A54E072 5A14C
```



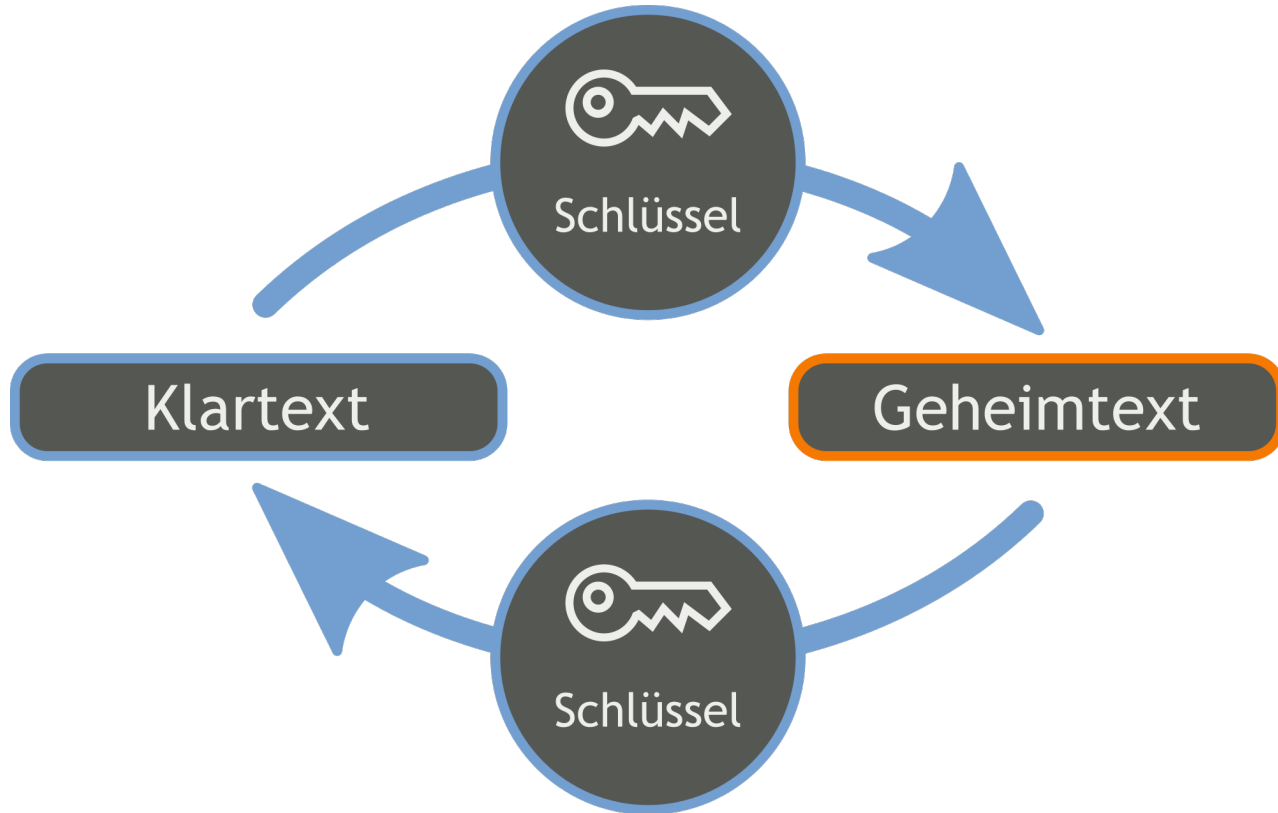
# Fragestellung

- Unterschied Ausgangsbild, Verschlüsseltes Bild?
- Interpretation des Verschlüsselten Bildes?
- Gängige Verschlüsselungsalgorithmen mit Matlab?

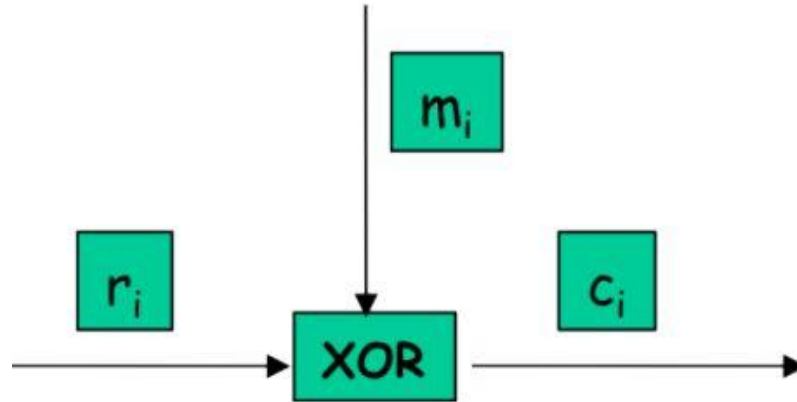
# Verschlüsselungsarten

- Symmetrische Verschlüsselung
- Asymmetrische Verschlüsselung

# Symmetrische Verschlüsselung



# Symmetrisch - One Time Pad

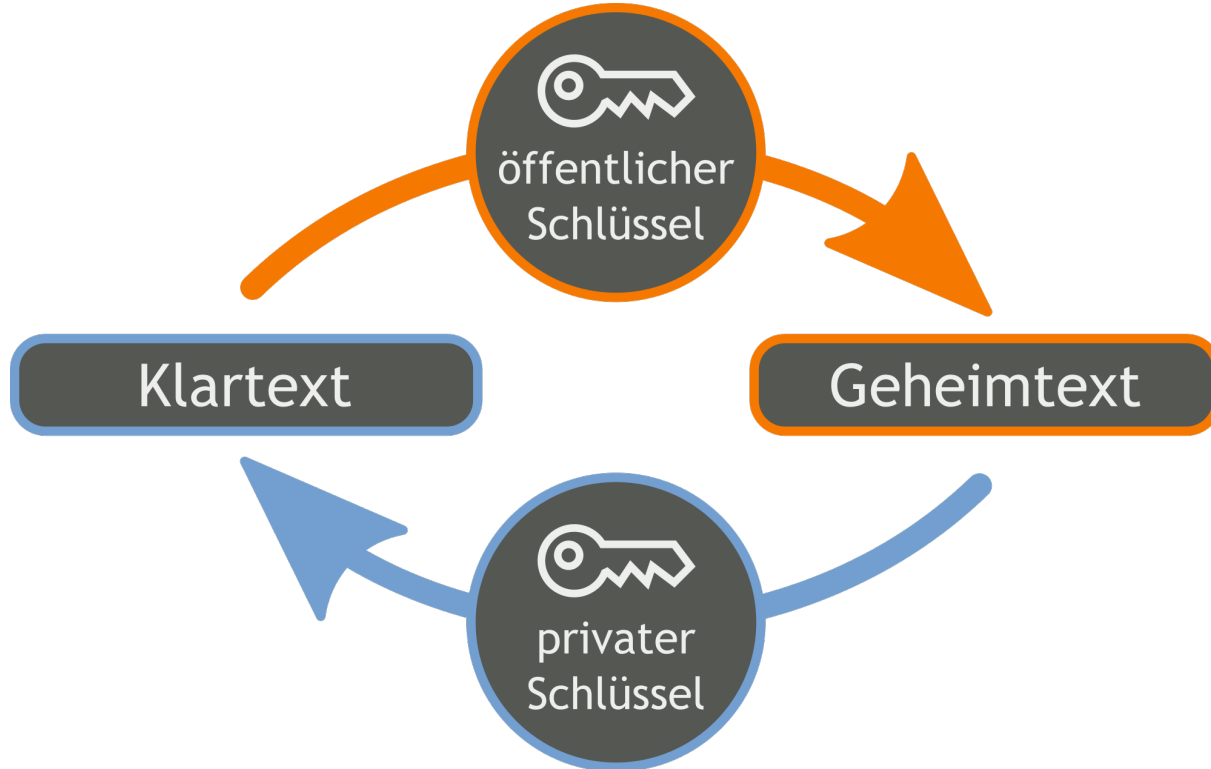


m:	0	1	1	1	0	1	0	0	0	0	1
r:	1	1	1	0	0	0	1	1	1	0	0
c:	1	0	0	1	0	1	1	1	1	0	1

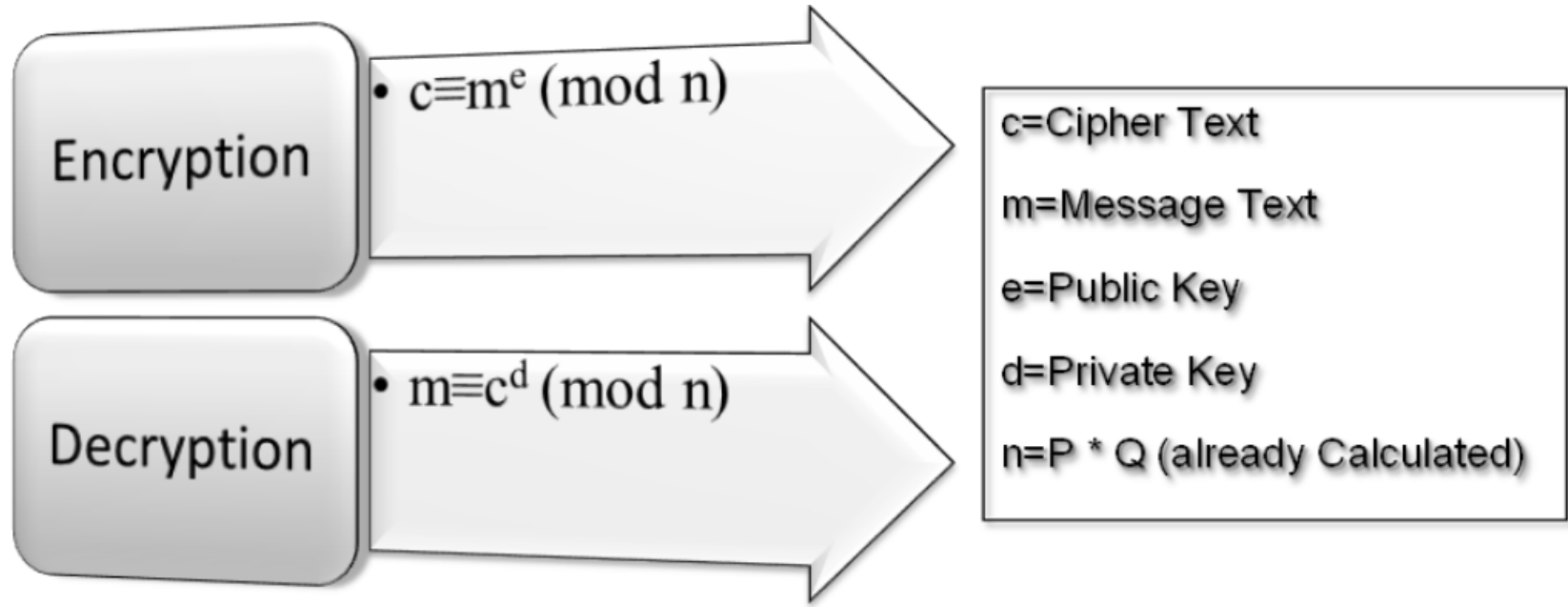
# Beispiel mit Matlab: Symmetrisch



# Asymmetrische Verschlüsselung



# Asymmetrische Verschlüsselung mit RSA





# Beispielrechnung RSA


$$n = 33$$

$$m = 16$$

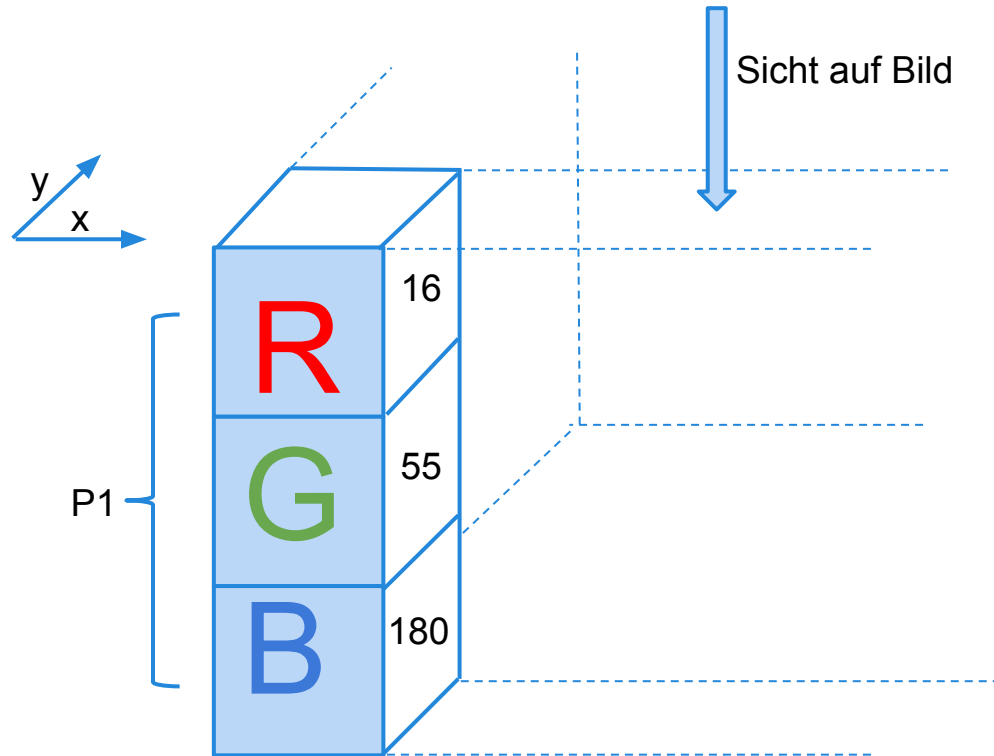
$$e = 3$$

$$d = 7$$

$$\text{Verschlüsselung: } 16^3 \equiv 4 \pmod{33}$$

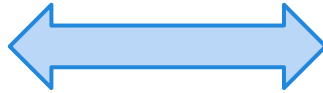
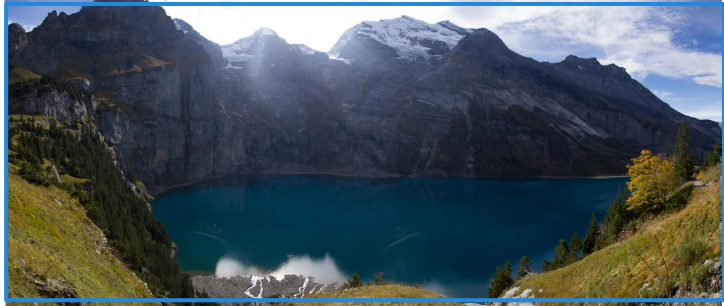
$$\text{Entschlüsselung: } 4^7 \equiv 16 \pmod{33}$$


# Bildverschlüsselung mit RSA



Original:  $P1(R, G, B)$   
Verschlüsselt:  $P1'(enc(R), enc(G), enc(B))$

# Beispiel mit Matlab: Asymmetrisch



# Schwierigkeiten Asym. Verschl.

- Pixelgetrennte (1 Pixel)  
Verschlüsselung  
sehr schwach

```
val(:, :, 1) =
```

Columns 1 through 21

301	296	296	188	88	325	325	159	159
301	296	296	188	88	325	325	159	159
301	296	296	188	88	325	325	159	159
301	296	88	188	88	325	325	159	159
296	296	88	188	88	325	159	198	159
296	88	88	325	325	325	159	198	159
296	88	88	325	325	198	159	198	307
296	88	188	325	159	198	307	198	307
381	133	108	308	88	325	198	307	188
188	325	325	188	188	188	188	188	28
28	242	53	159	88	296	301	308	424
53	307	159	325	296	308	176	108	308

# Performancevergleich Sym. vs. Asym.

	imgName	width	height	megapixel	time	time
2	large-1.jpg	5184	3456	17.915904	0.0394096317168997	20.47877273464
3	large-10.jpg	5184	3456	17.915904	0.0252226261655185	177.995743556211
4	large-2.jpg	5074	3383	17.165342	0.024913859721674	95.2499947612342
5	large-4.jpg	5184	3456	17.915904	0.023964039400651	373.204953776979
6	large-5.jpg	5184	3456	17.915904	0.0244887712934449	374.036259957397
7	large-6.jpg	5184	3456	17.915904	0.0260471437607154	6.49680435283708
8	large-7.jpg	11029	4695	51.781155	0.0743836324704266	336.762452974058
9	large-8.jpg	5530	4142	22.90526	0.0322934632604282	17.1722168361532
10	large-9.jpg	3456	5184	17.915904	0.0238216305006507	53.3108581012916
11	medium-1.jpg	3000	2000	6	0.0081733300204376	9.6781392074795
12	medium-10.jpg	3000	2000	6	0.0089987029243432	7.44447742134688

# Fazit

- Symmetrisch wesentlich effizienter + unkompliziert
- Für direkte Bildverschlüsselung nur symmetrisch sicher (o. Zusatzberechnungen)

# Fragestellung

- Unterschied Ausgangsbild, Verschlüsseltes Bild?
- Interpretation des Verschlüsselten Bildes?
- Gängige Verschlüsselungsalgorithmen mit Matlab?

Fragen?