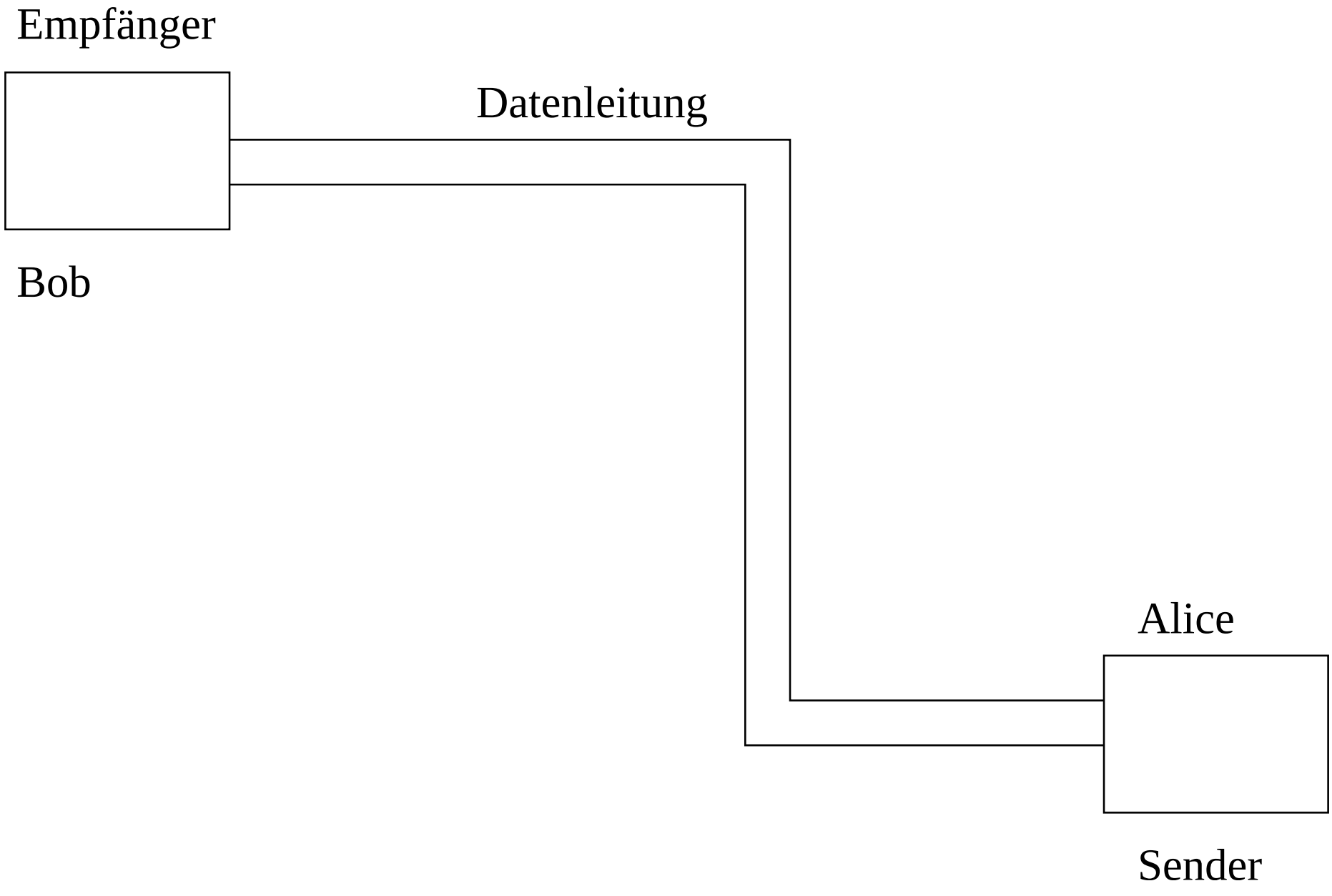
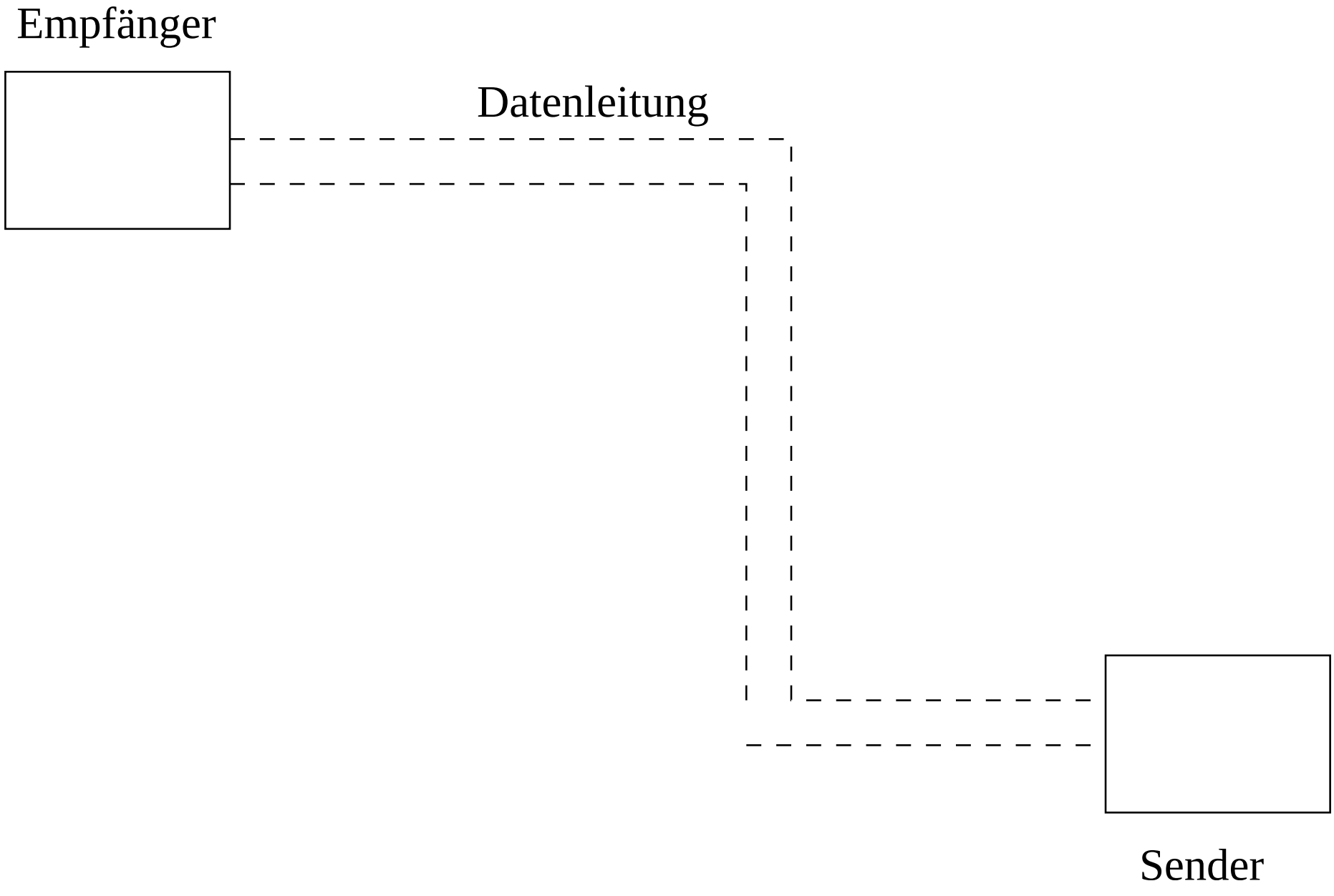
Verschlüsselung

(und ihr mathematischer Hintergrund)

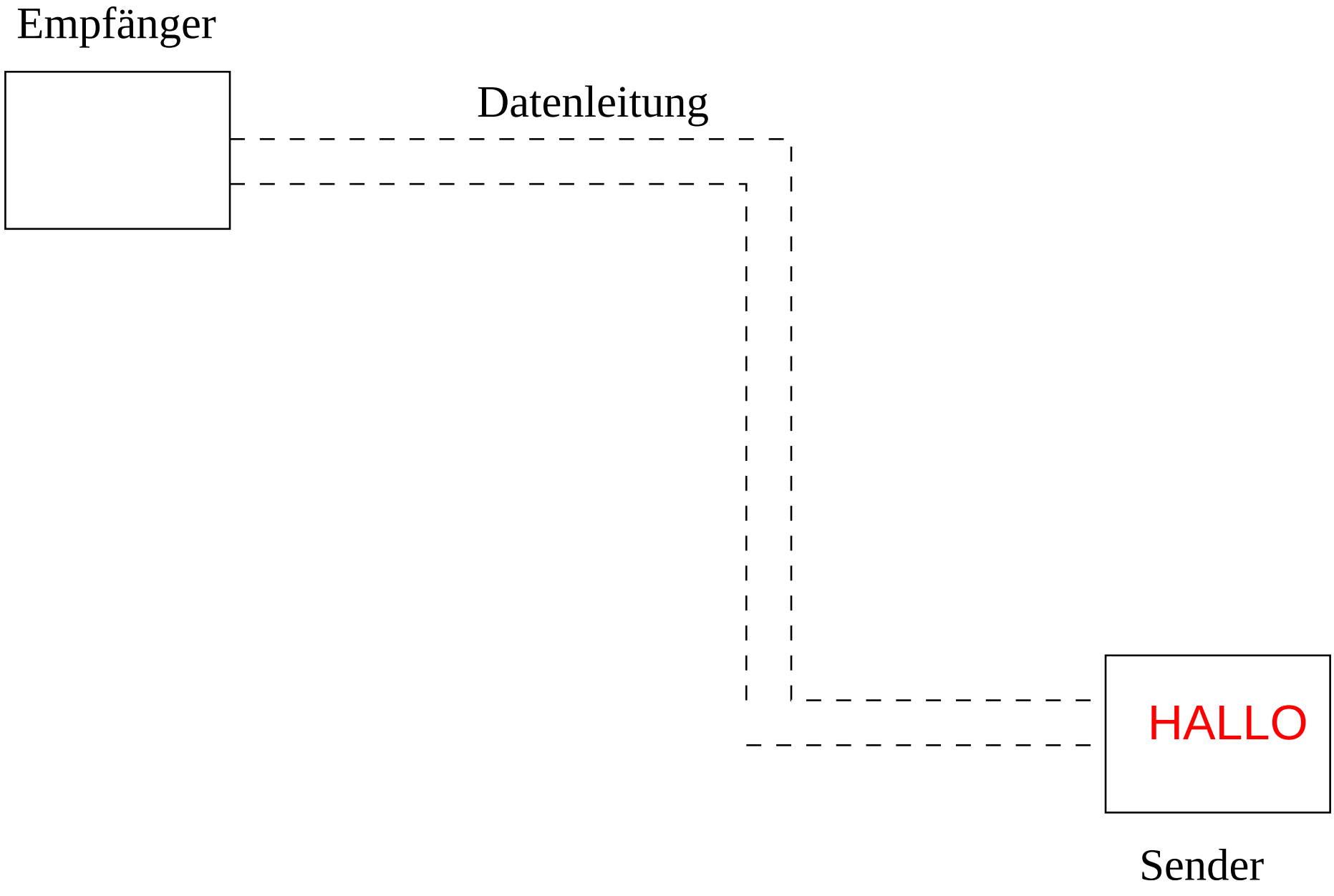
Übertragung von Daten



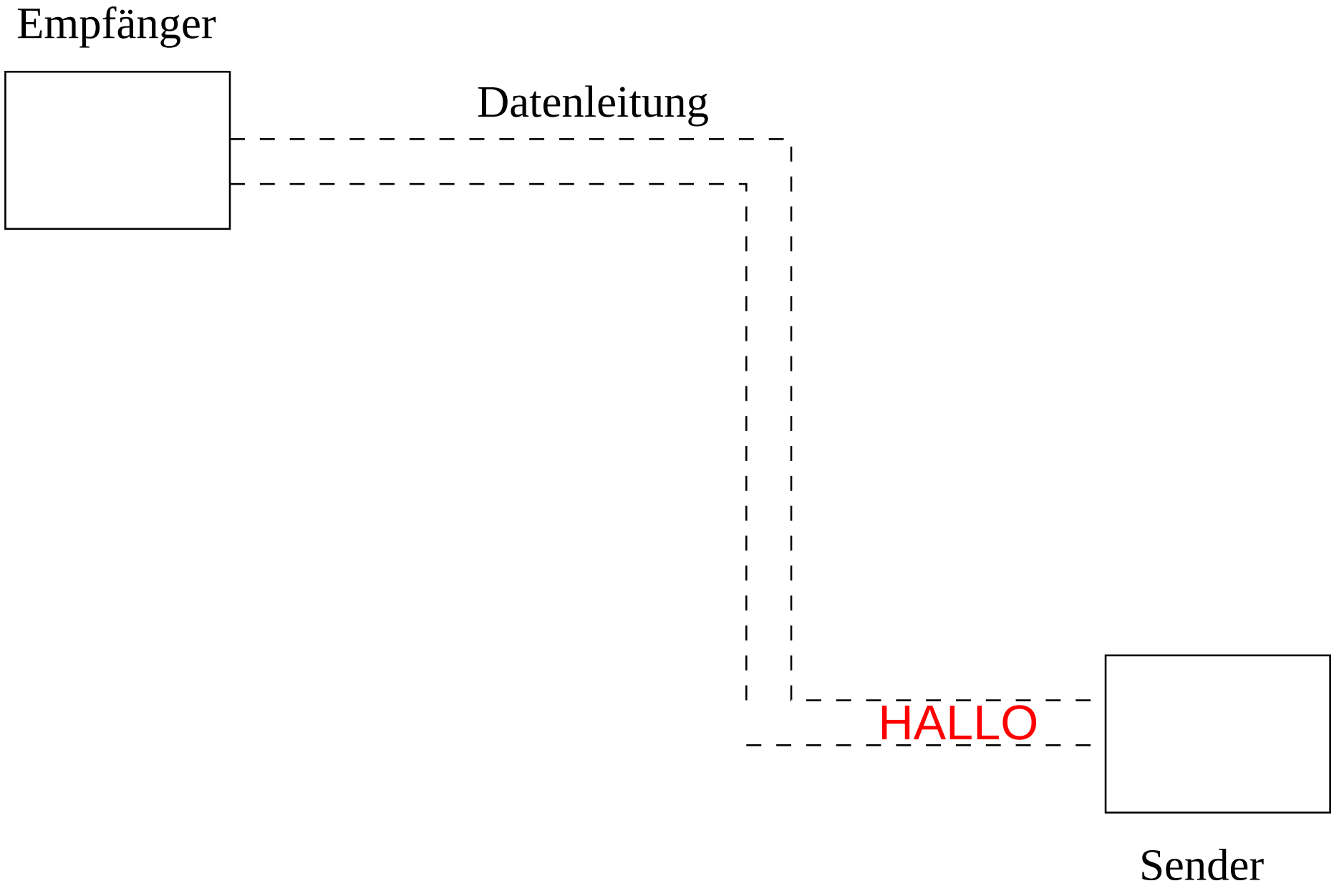
Übertragung von Daten



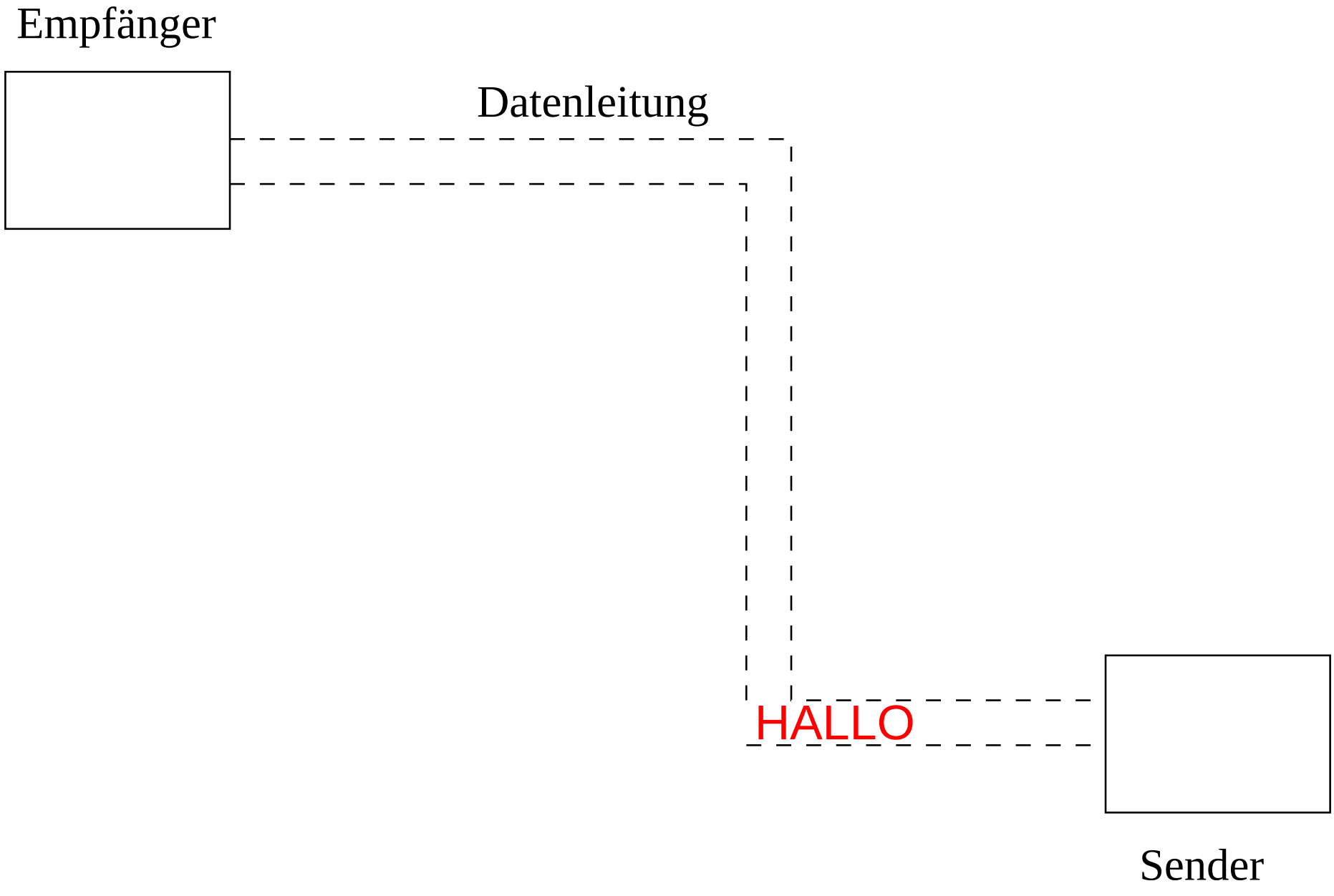
Übertragung von Daten



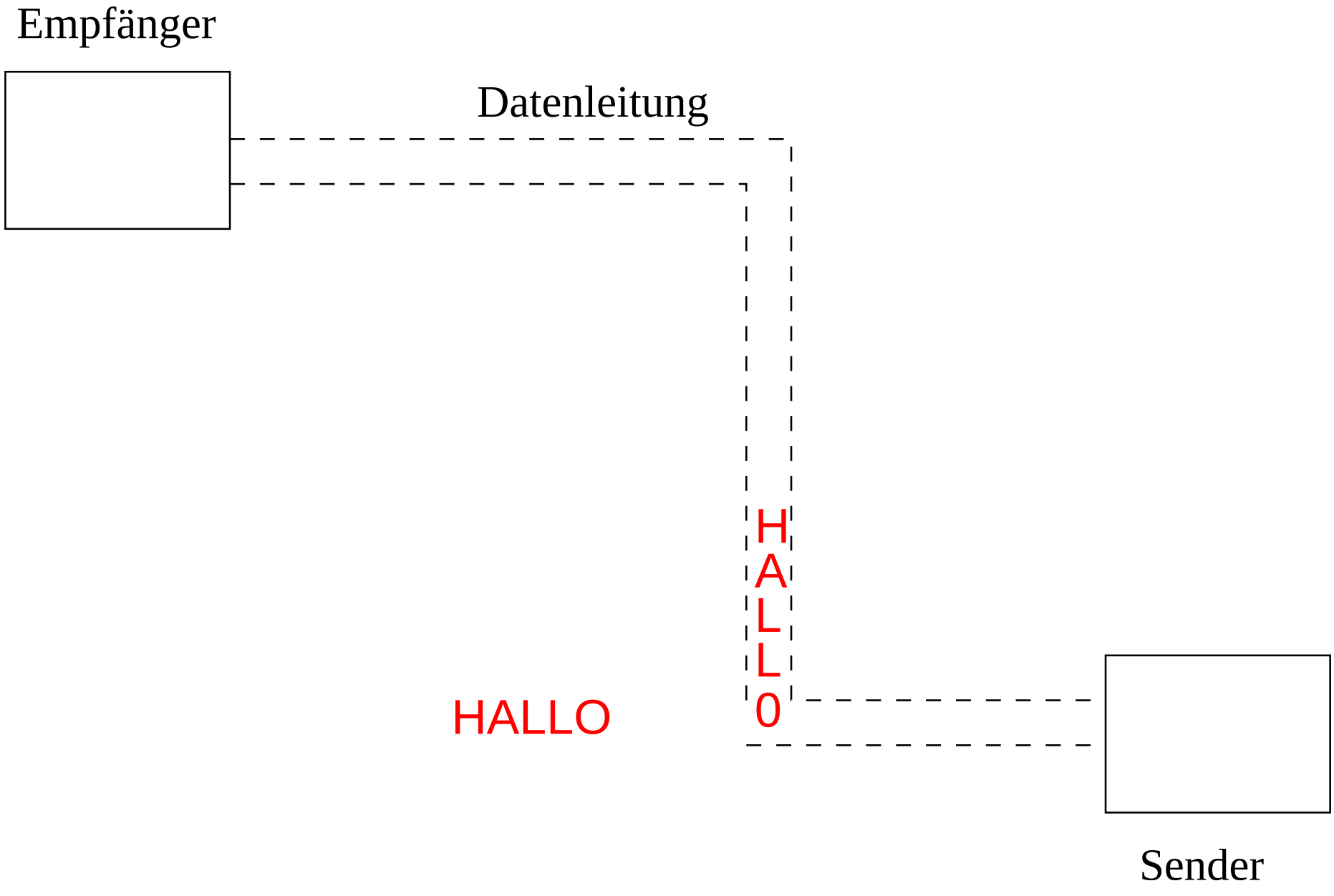
Übertragung von Daten



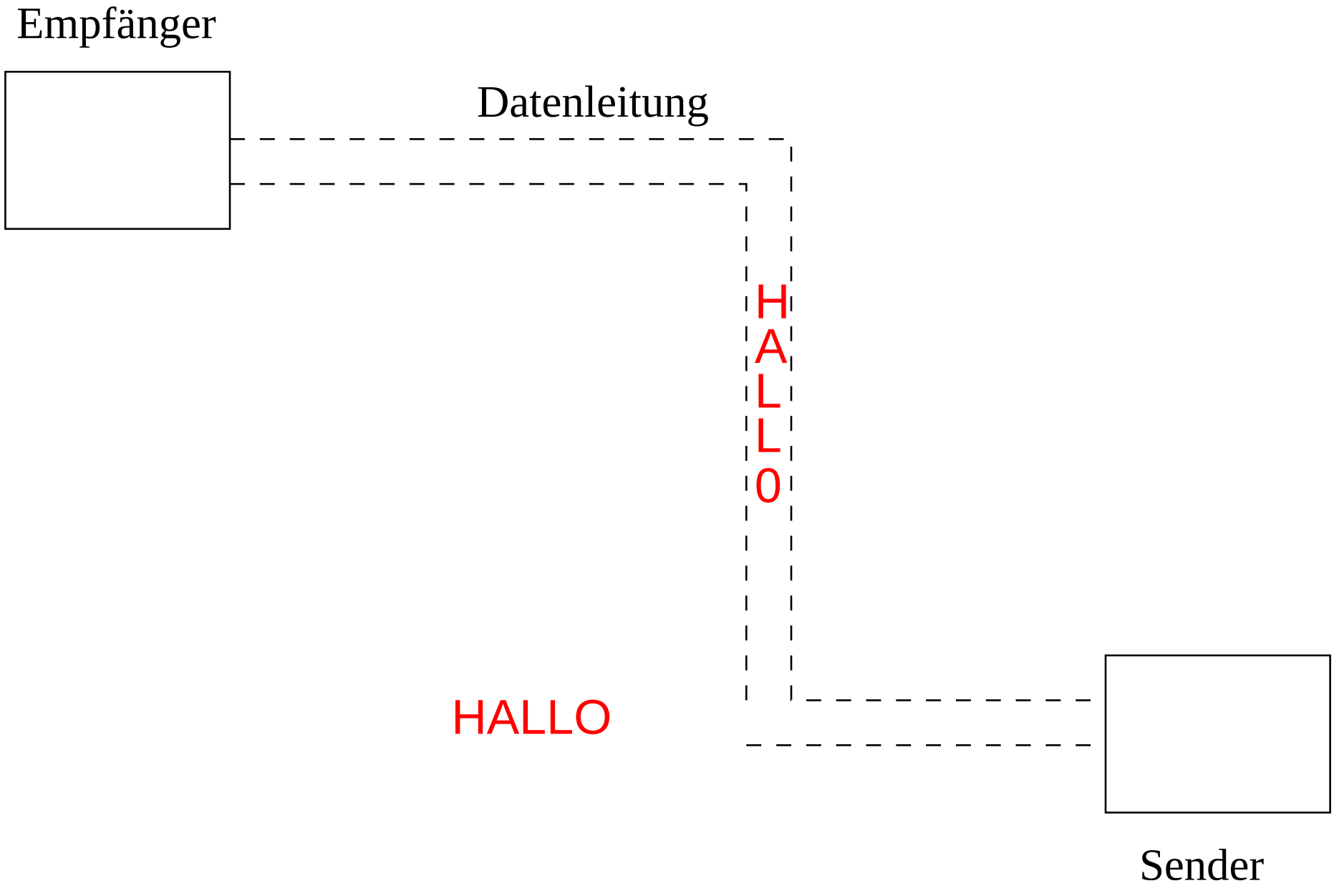
Übertragung von Daten



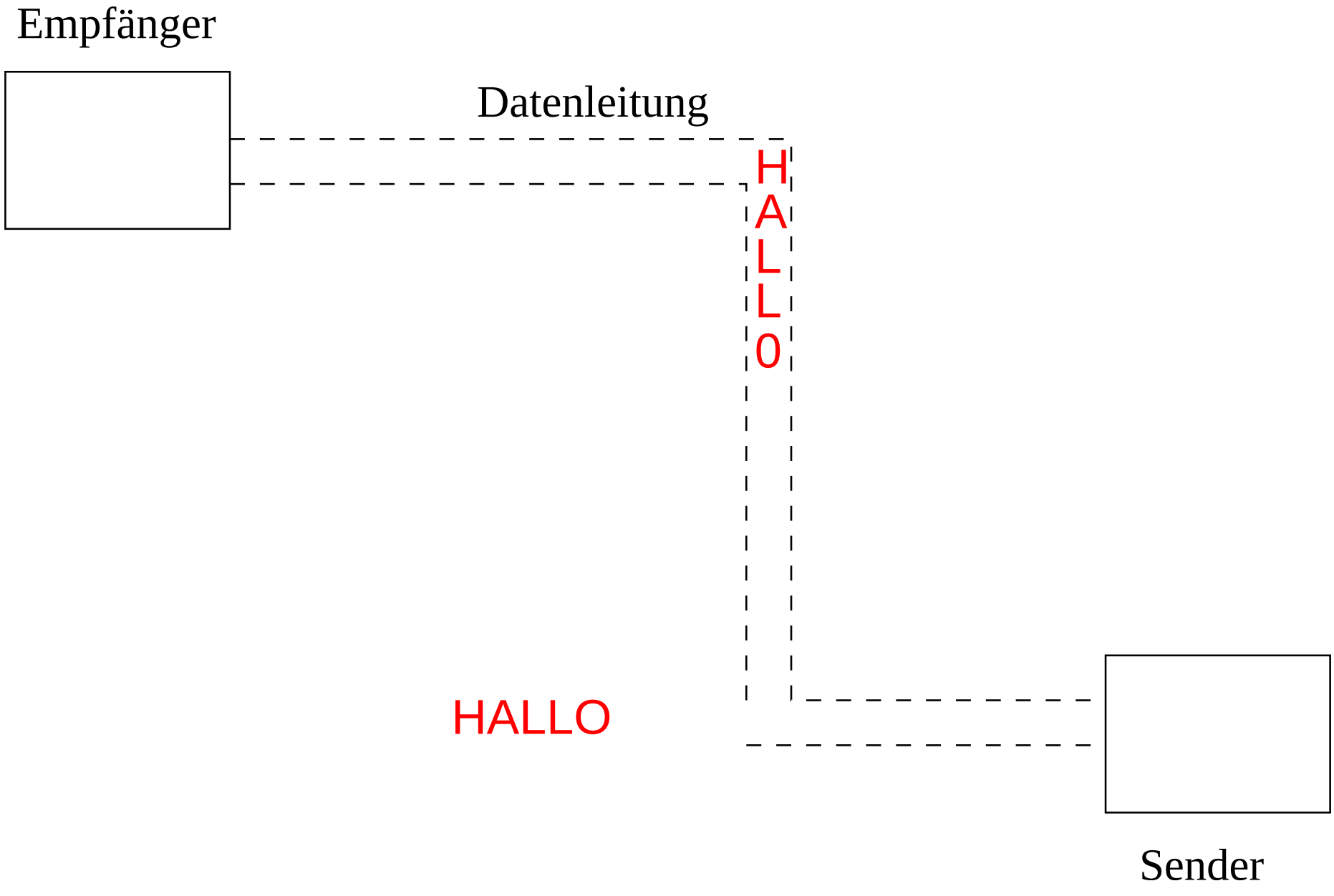
Übertragung von Daten



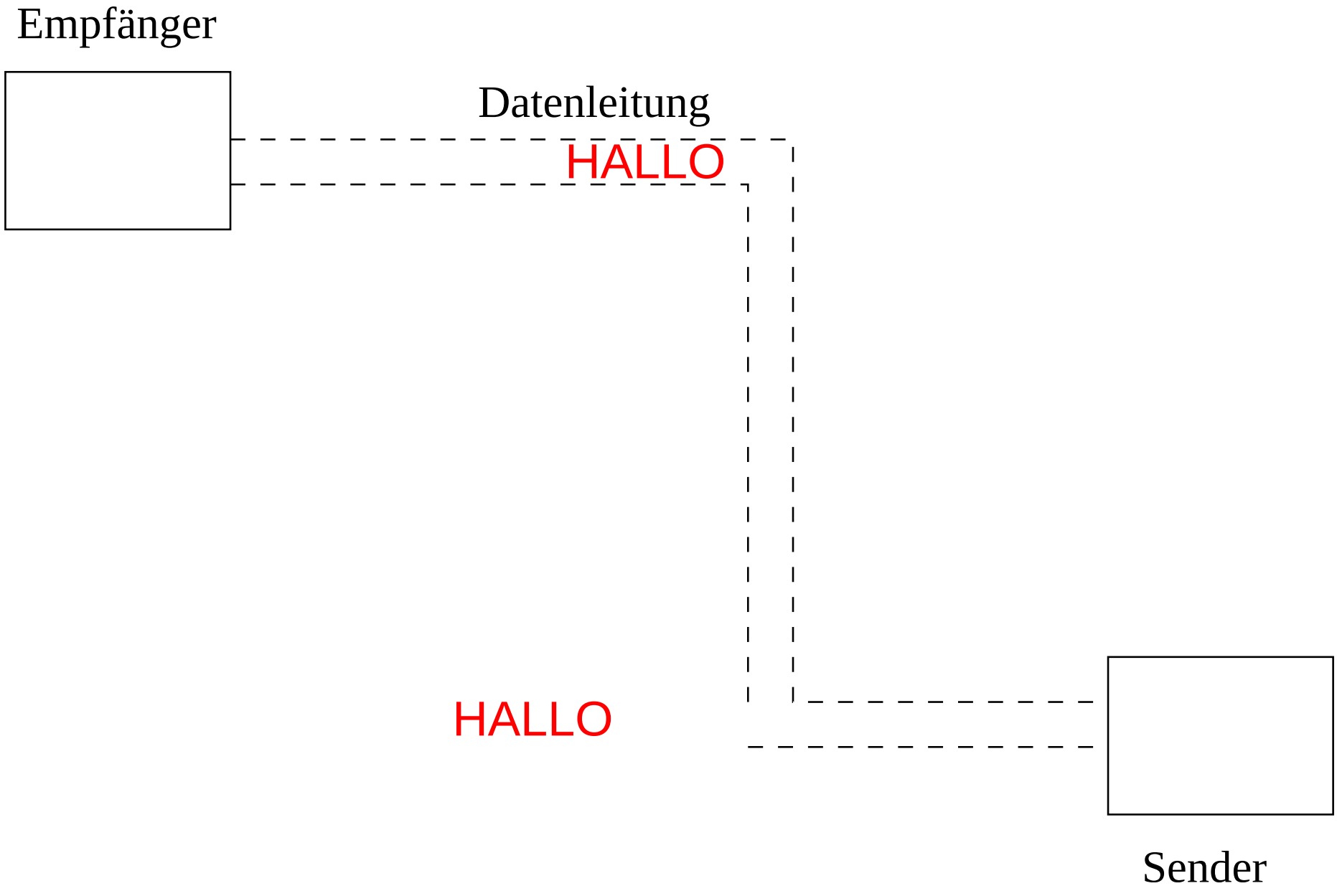
Übertragung von Daten



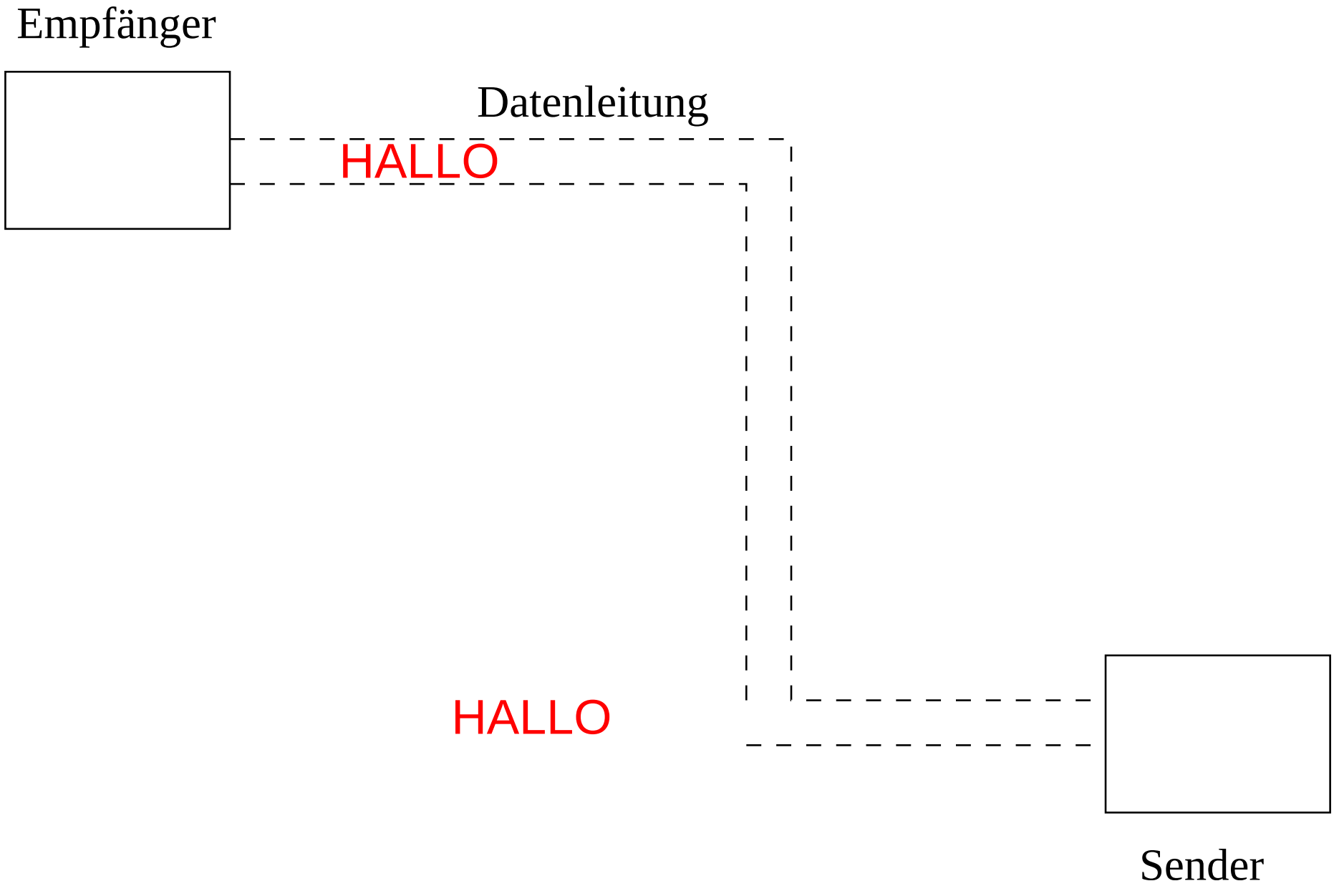
Übertragung von Daten



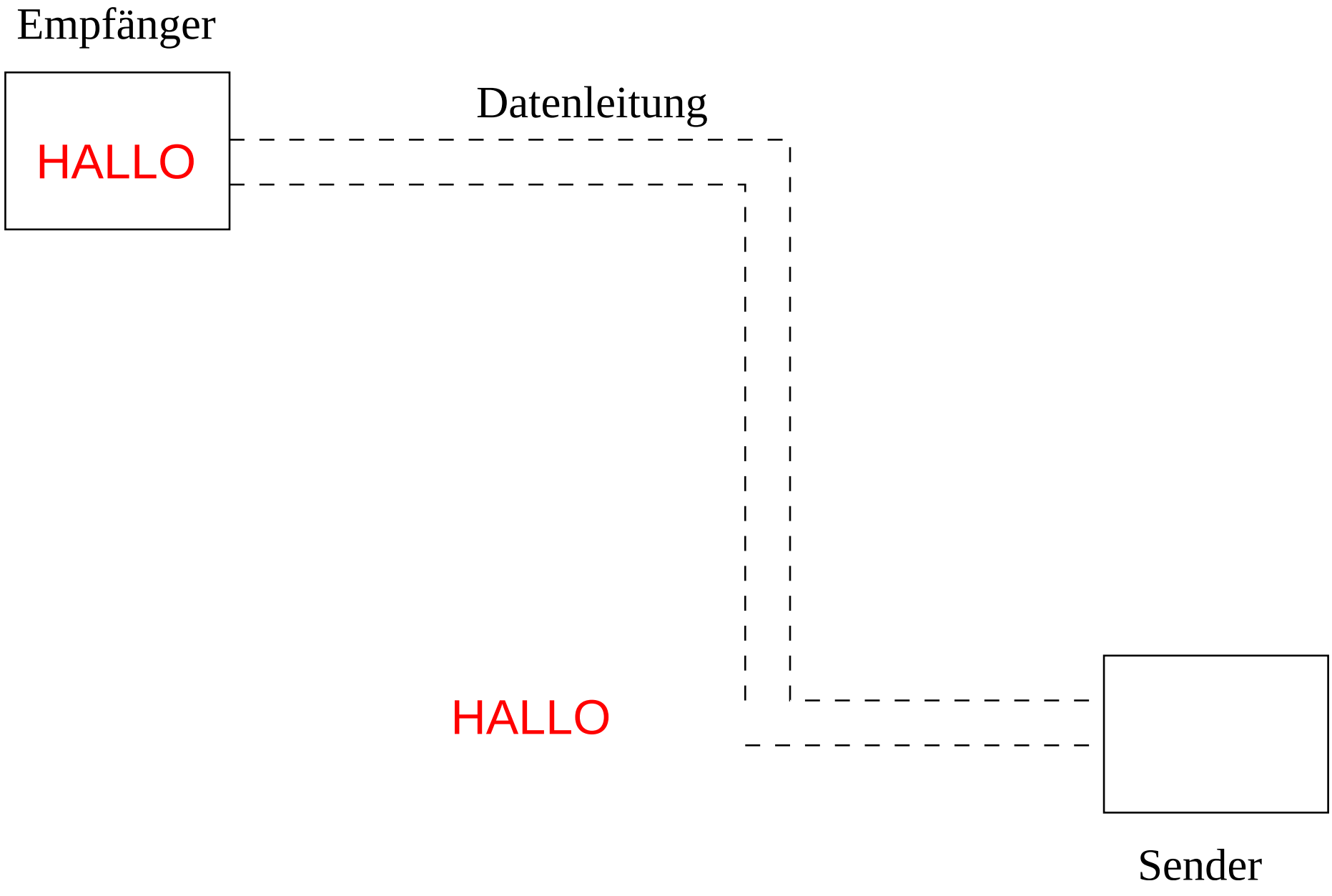
Übertragung von Daten



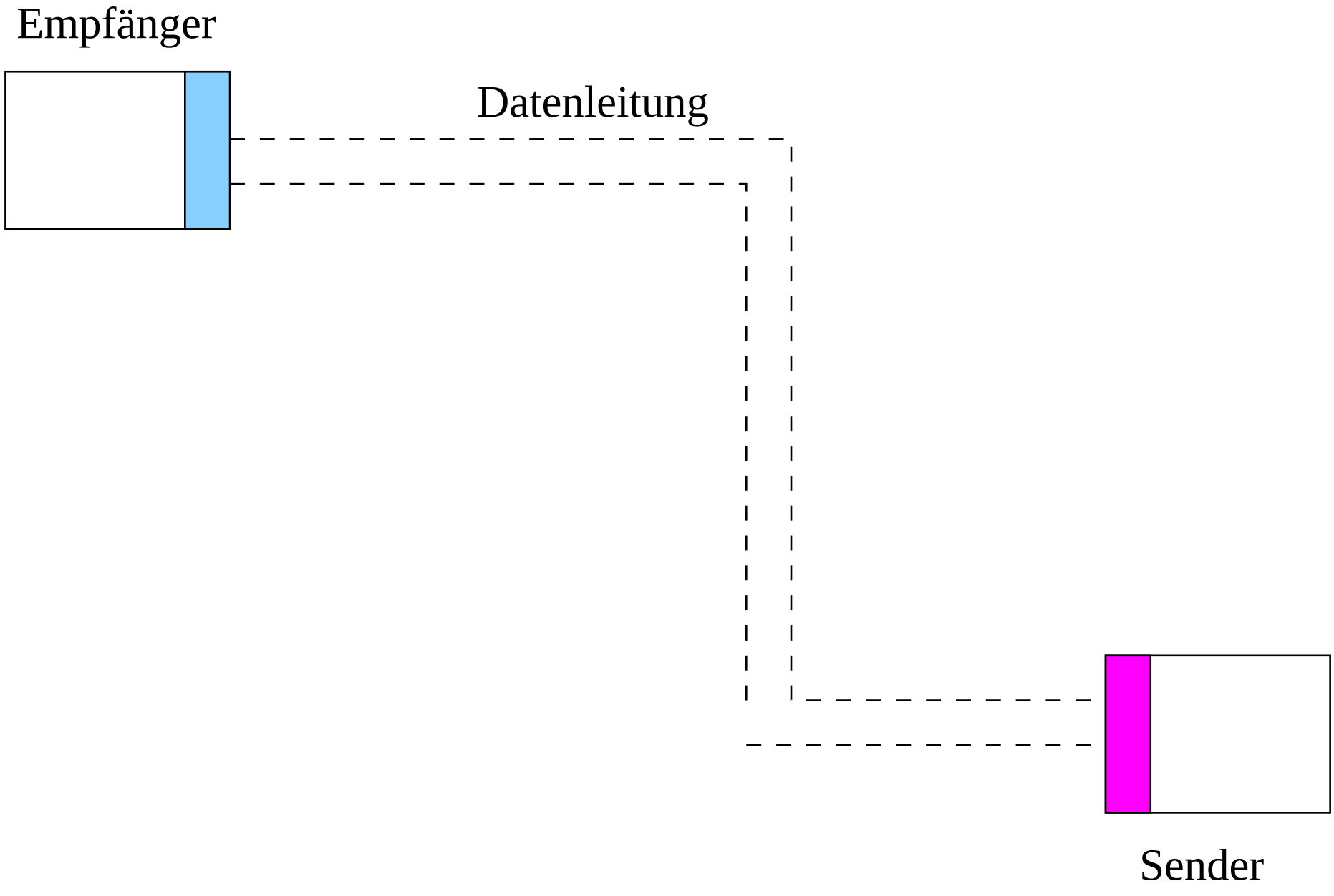
Übertragung von Daten



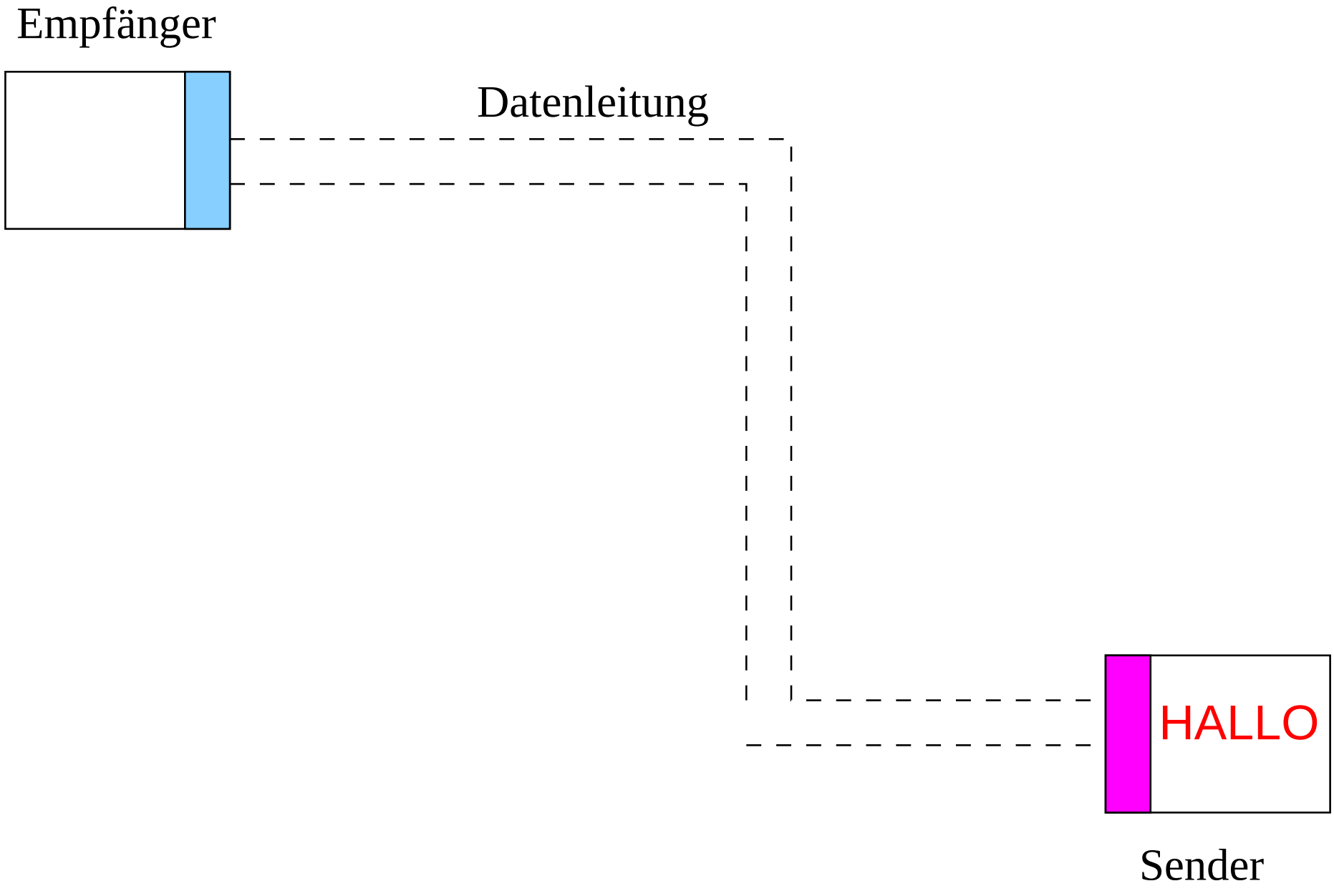
Übertragung von Daten



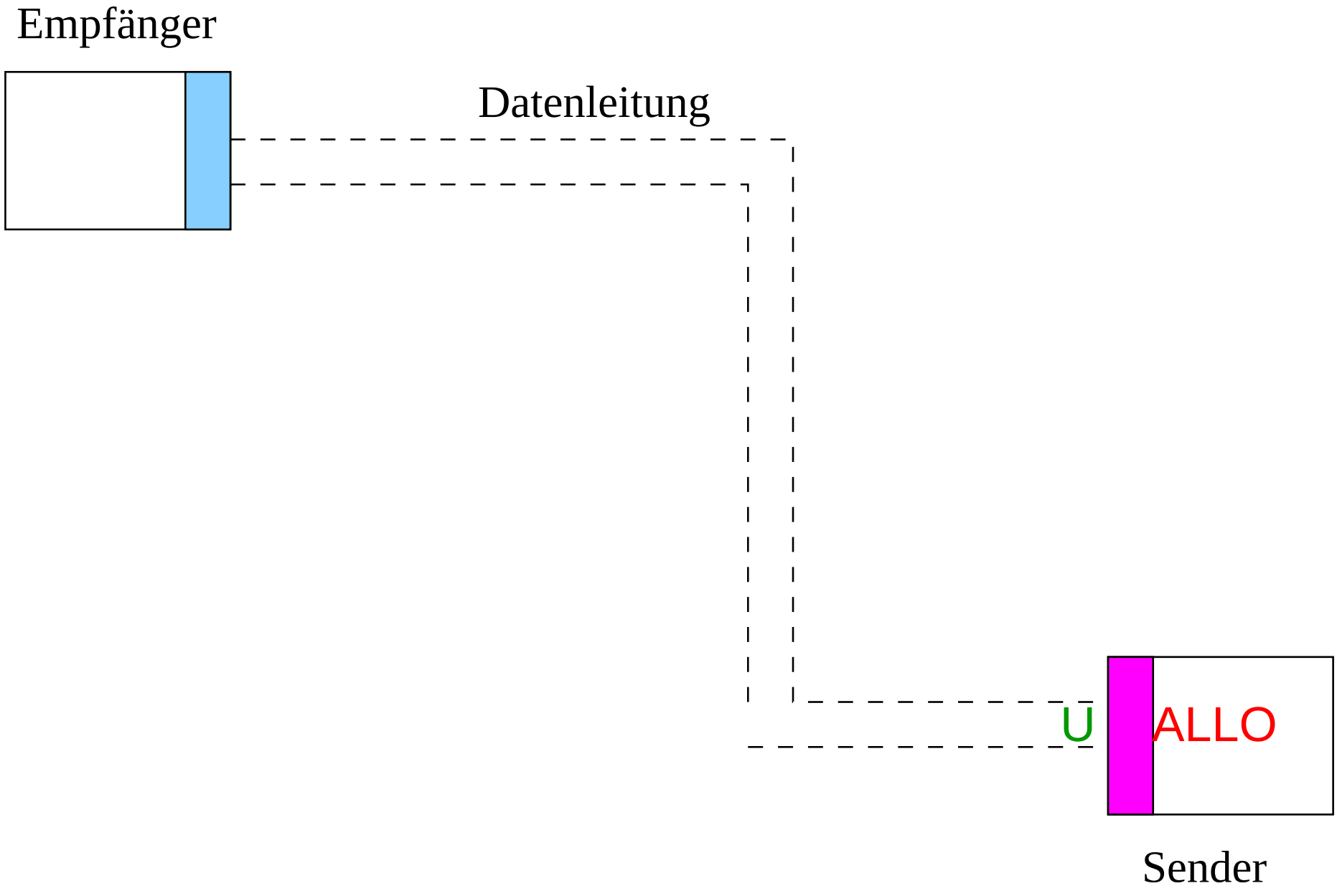
Verschlüsselte Übertragung



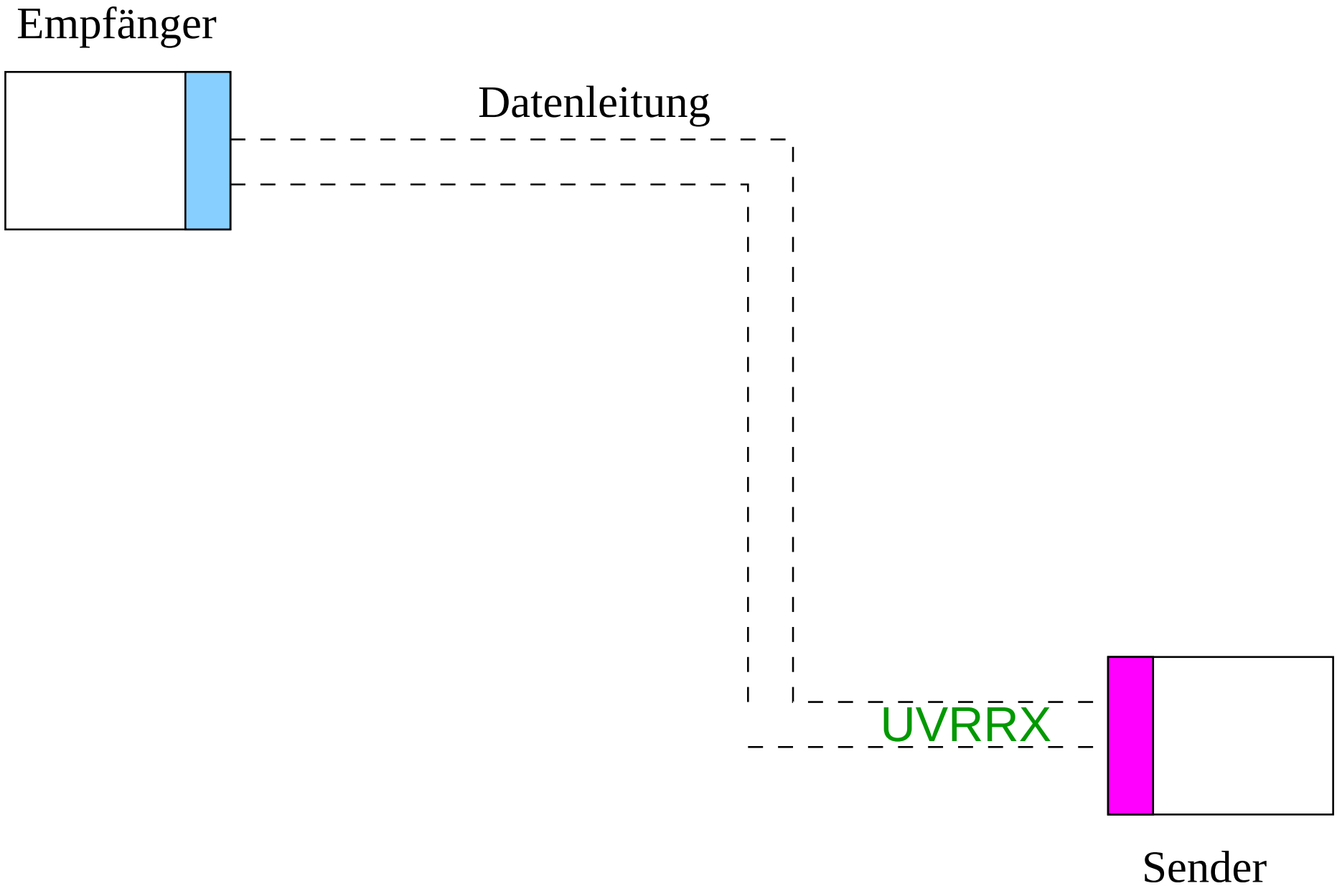
Verschlüsselte Übertragung



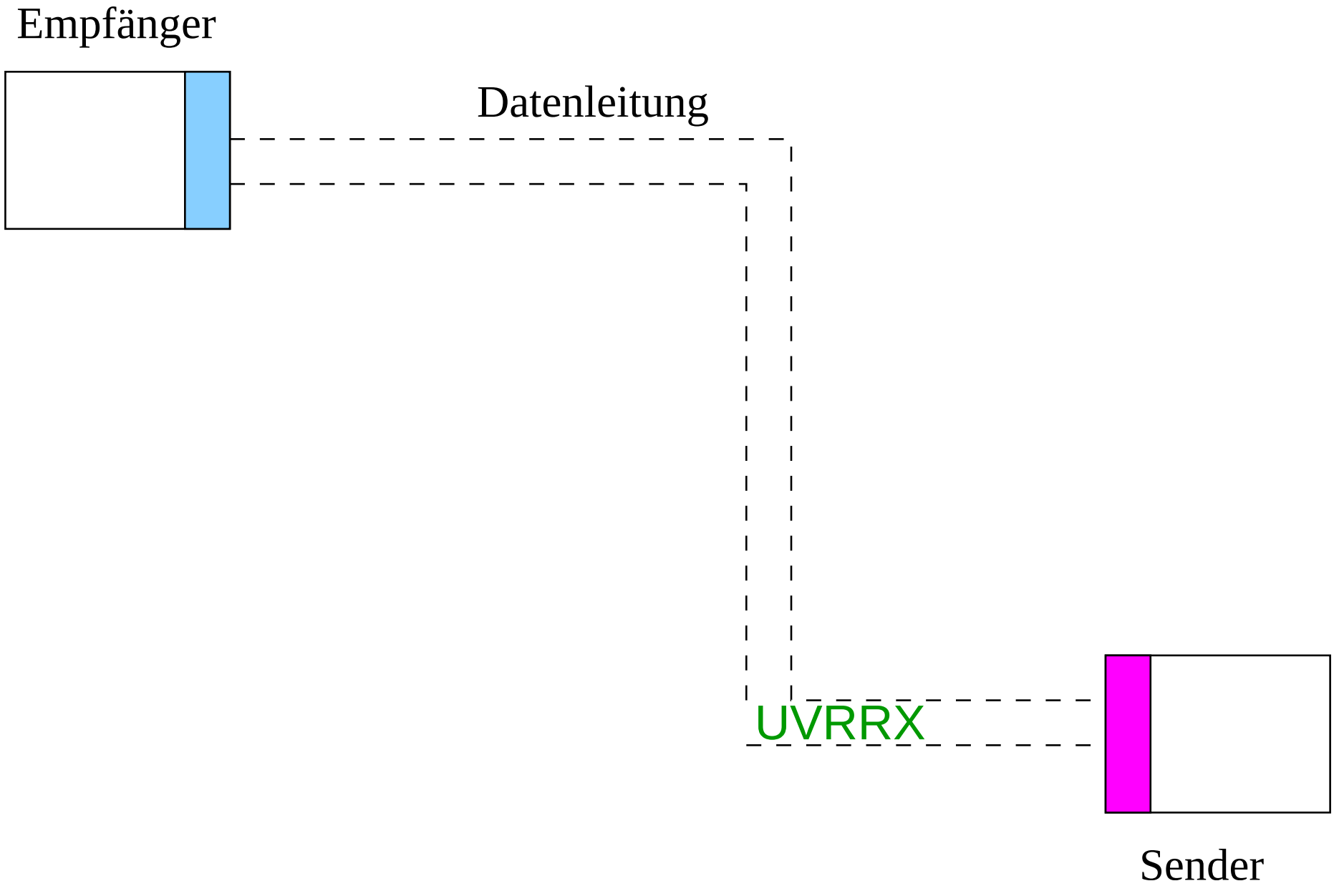
Verschlüsselte Übertragung



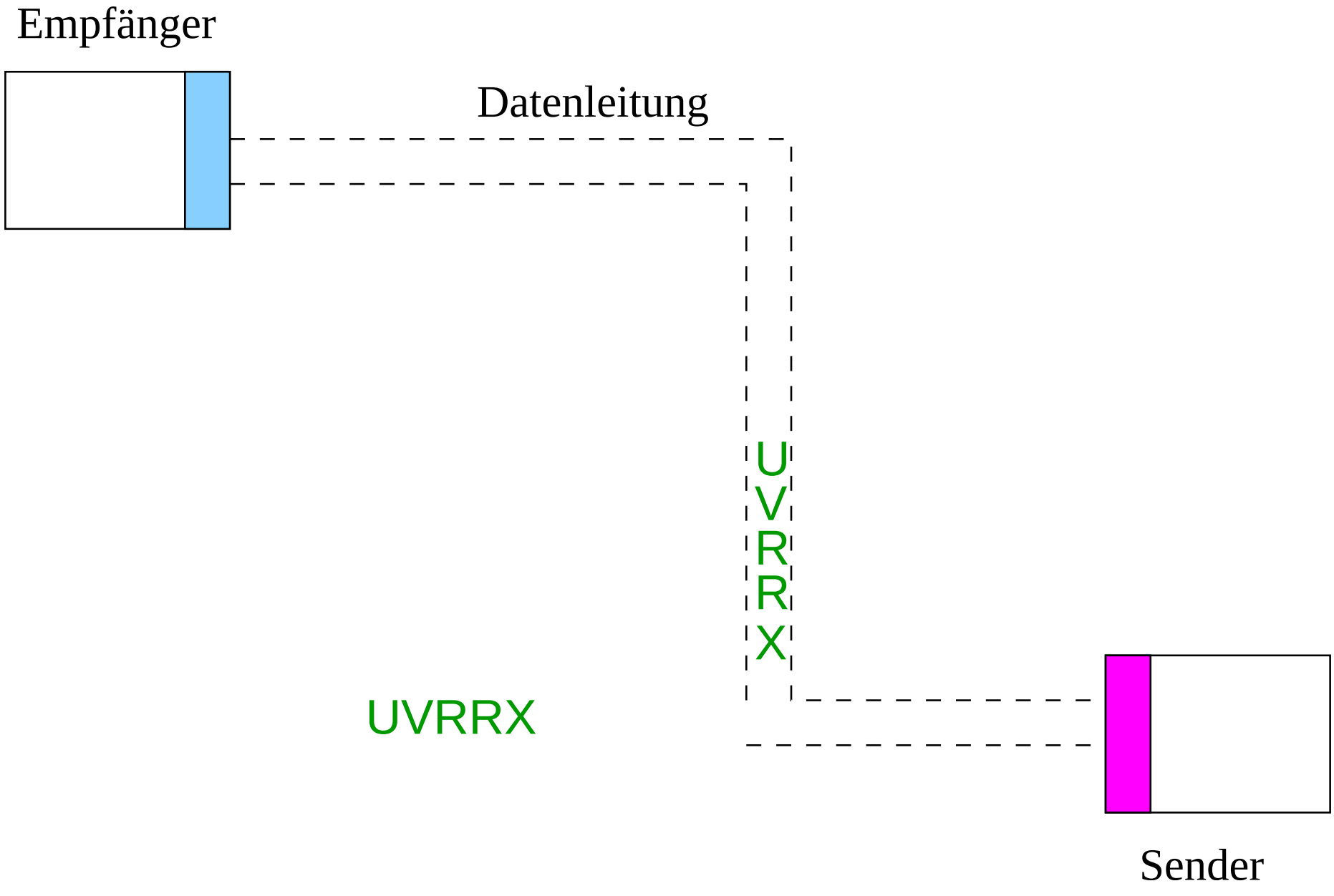
Verschlüsselte Übertragung



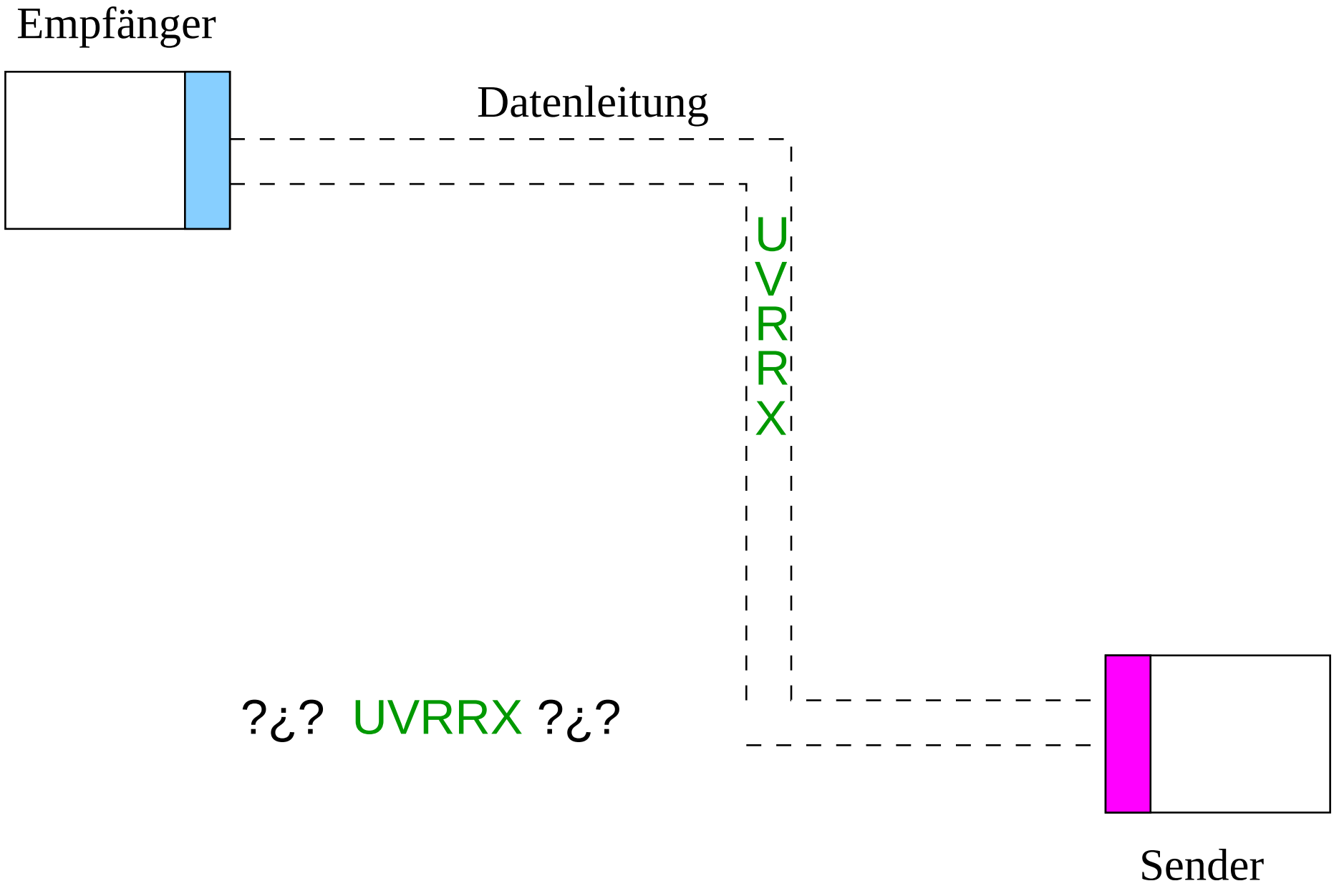
Verschlüsselte Übertragung



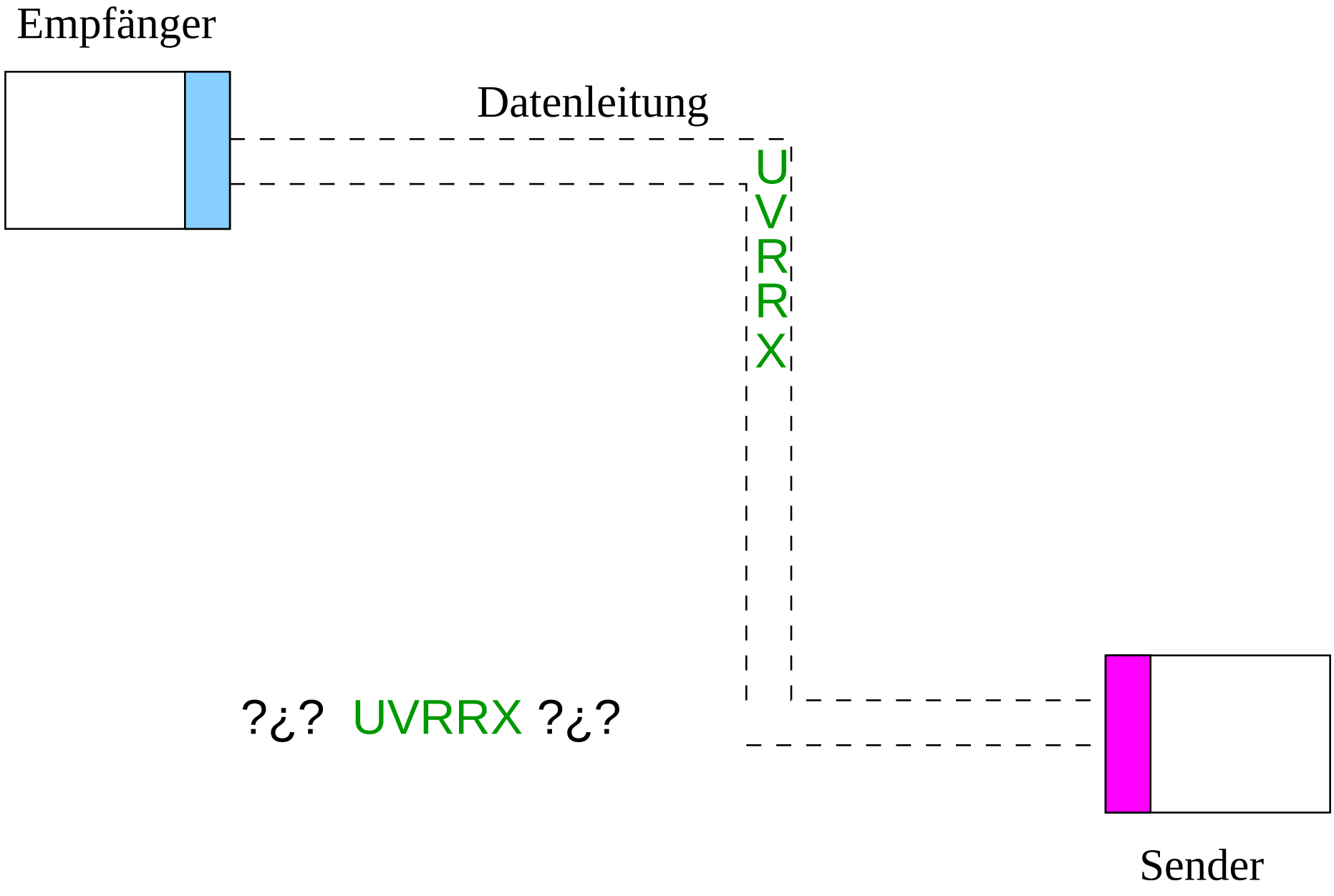
Verschlüsselte Übertragung



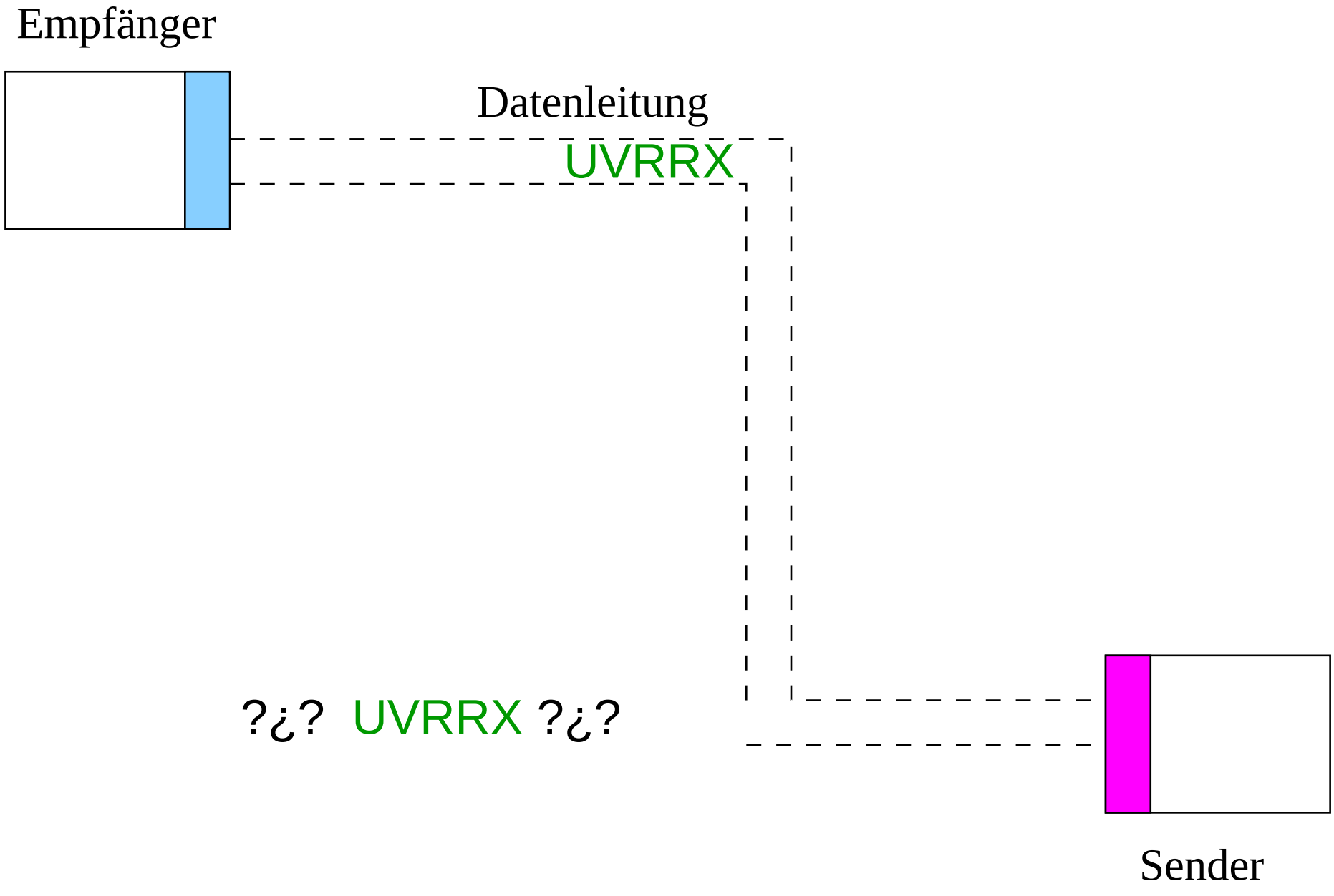
Verschlüsselte Übertragung



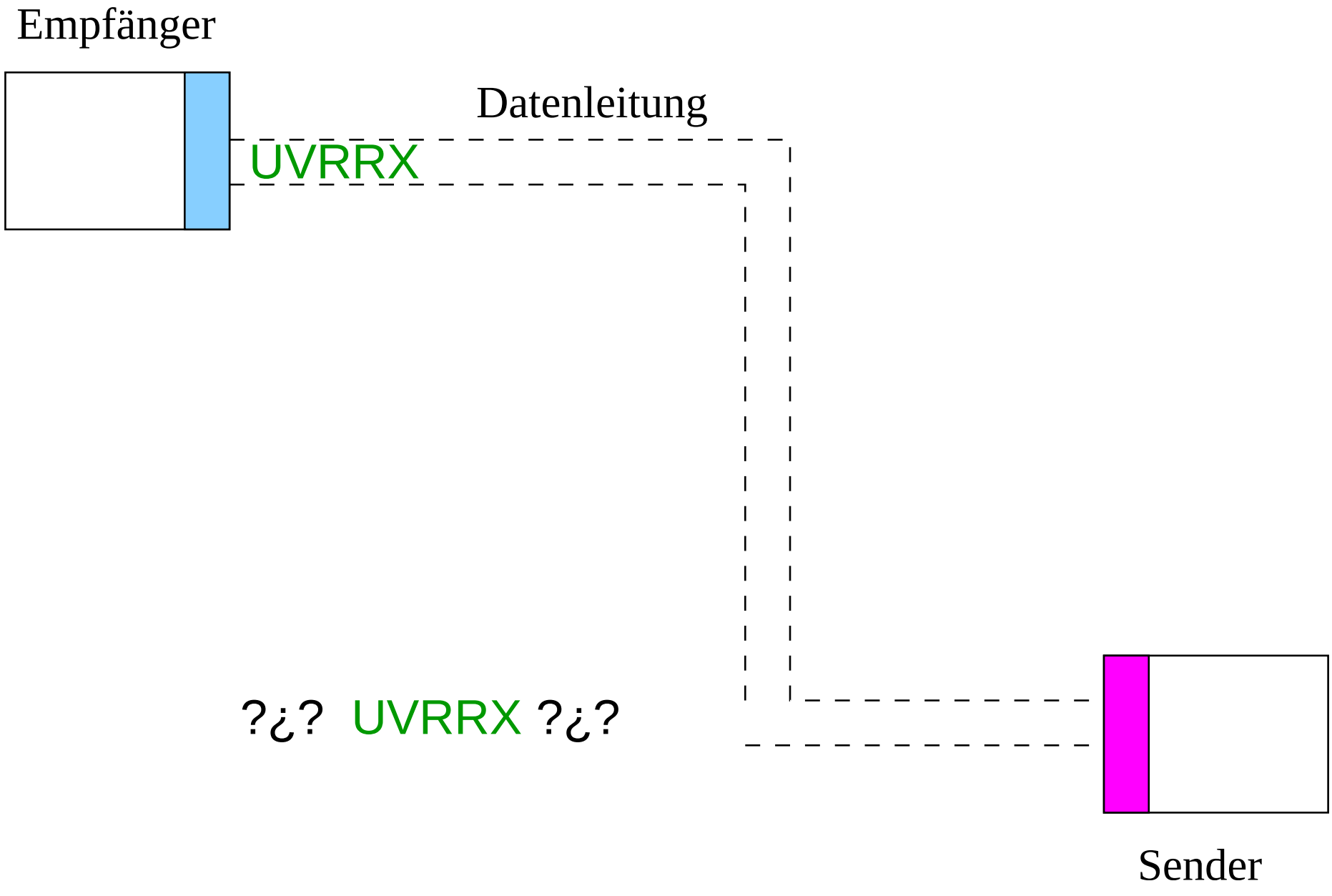
Verschlüsselte Übertragung



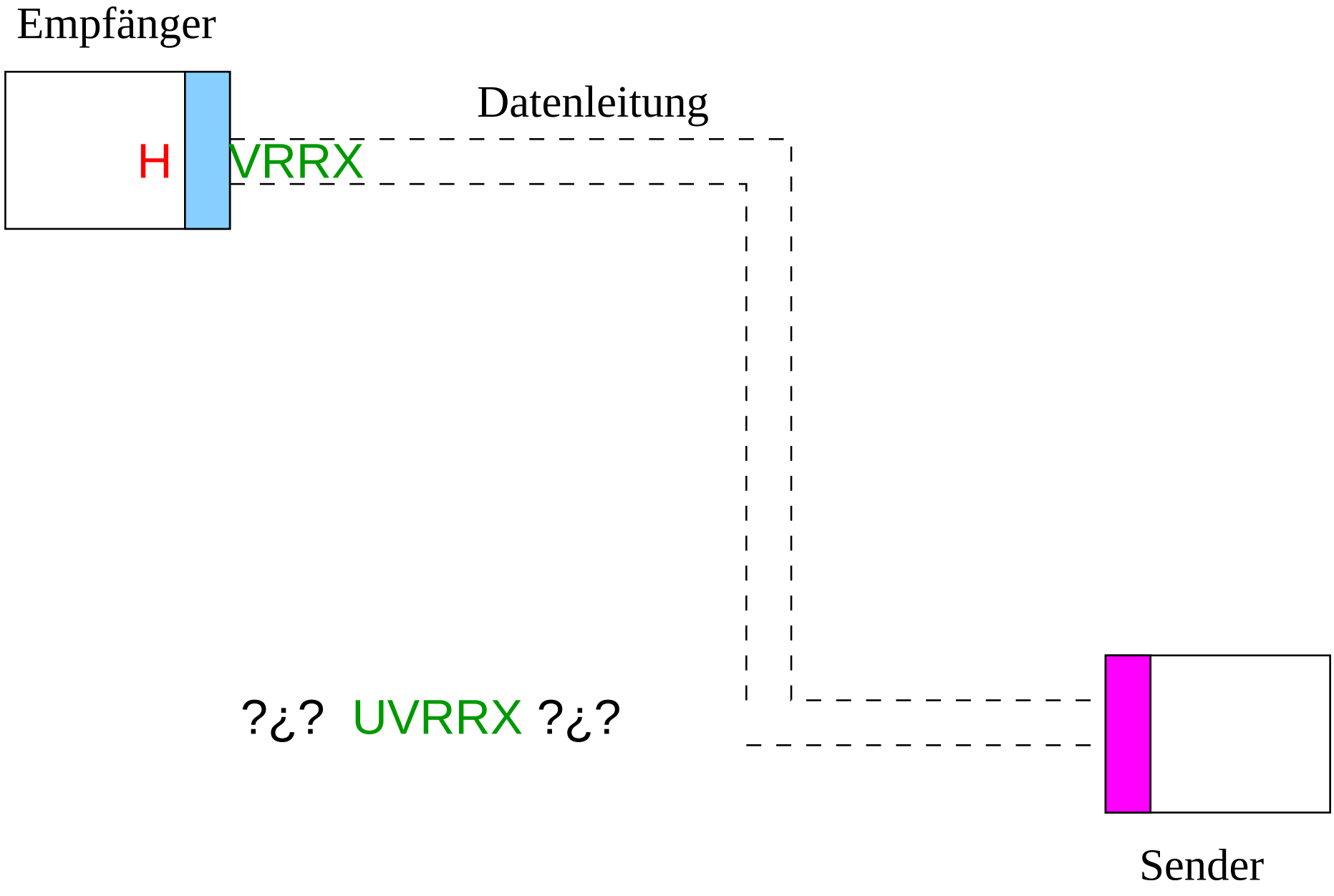
Verschlüsselte Übertragung



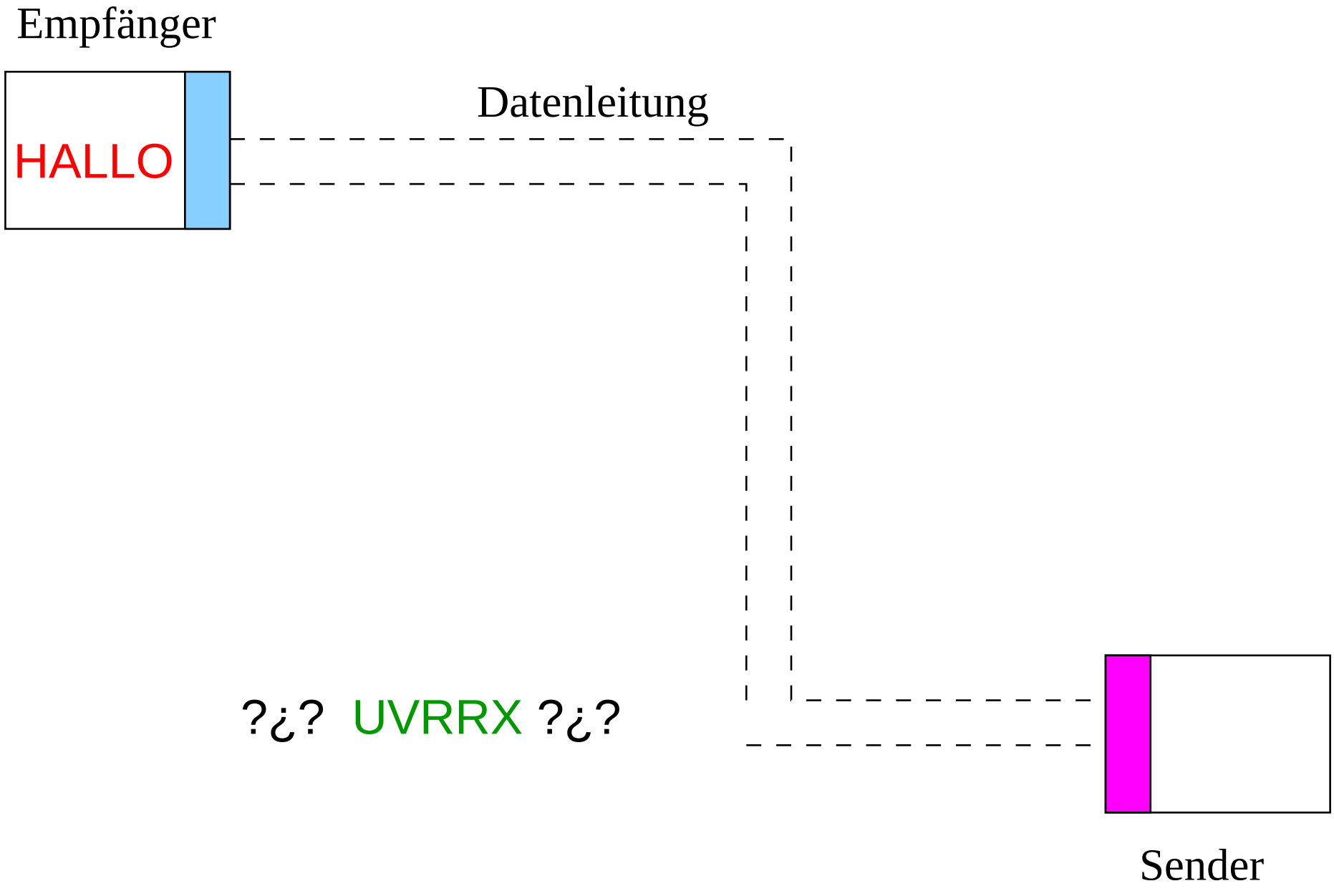
Verschlüsselte Übertragung



Verschlüsselte Übertragung



Verschlüsselte Übertragung

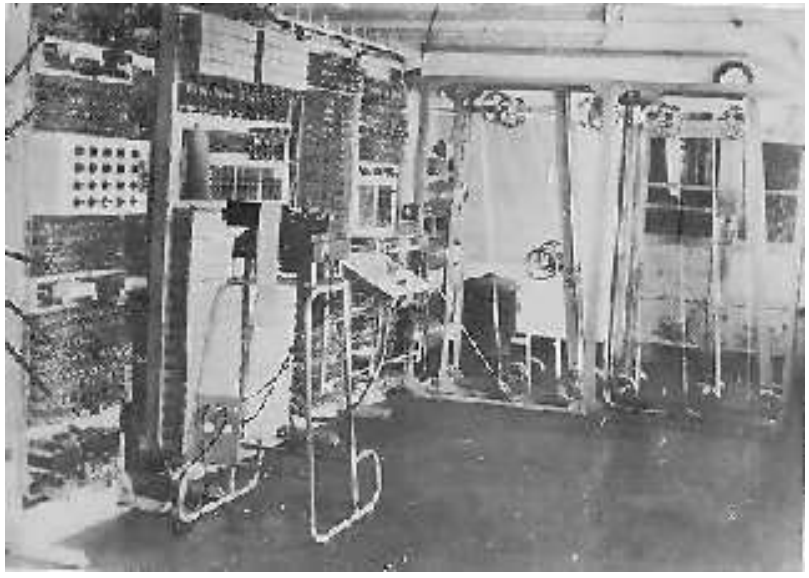


Enigma und Colossus

Empfänger

HALLO

Datenleitung



?¿? UVRRX ?¿?

Sender

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

A	B	C	...	H	...	L	...	O	...	Z
V	S	E	...	U	...	R	...	X	...	Q

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

A	B	C	...	H	...	L	...	O	...	Z
V	S	E	...	U	...	R	...	X	...	Q

HALLO wird verschlüsselt zu UVRRX

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

A	B	C	...	H	...	L	...	O	...	Z
V	S	E	...	U	...	R	...	X	...	Q

HALLO wird verschlüsselt zu UVRRX
ZOLL wird verschlüsselt zu QXRR

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

A	B	C	...	H	...	L	...	O	...	Z
V	S	E	...	U	...	R	...	X	...	Q

Vertauschen von Bitfolgen gemäß einer Tabelle

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

A	B	C	...	H	...	L	...	O	...	Z
V	S	E	...	U	...	R	...	X	...	Q

Vertauschen von Bitfolgen gemäß einer Tabelle

000	001	010	011	100	101	110	111
101	100	111	110	001	000	011	010

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

A	B	C	...	H	...	L	...	O	...	Z
V	S	E	...	U	...	R	...	X	...	Q

Vertauschen von Bitfolgen gemäß einer Tabelle

000	001	010	011	100	101	110	111
101	100	111	110	001	000	011	010

010011100 wird verschlüsselt zu 111110001

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

A	B	C	...	H	...	L	...	O	...	Z
V	S	E	...	U	...	R	...	X	...	Q

Vertauschen von Bitfolgen gemäß einer Tabelle

000	001	010	011	100	101	110	111
101	100	111	110	001	000	011	010

Vertauschen von Zahlen gemäß einer Tabelle

Verschlüsselung

Vertauschen von Buchstaben gemäß einer Tabelle

A	B	C	...	H	...	L	...	O	...	Z
V	S	E	...	U	...	R	...	X	...	Q

Vertauschen von Bitfolgen gemäß einer Tabelle

000	001	010	011	100	101	110	111
101	100	111	110	001	000	011	010

Vertauschen von Zahlen gemäß einer Tabelle

0	1	2	3	4	5	6	7
5	4	7	6	1	0	3	2

Verschlüsselung von Bitfolgen

Wenn Vertauschen von Bitfolgen der Länge 200 unsicher wird,
dann wechselt man zu Bitfolgen der Länge 210.

Verschlüsselung von Bitfolgen

Wenn Vertauschen von Bitfolgen der Länge 200 unsicher wird, dann wechselt man zu Bitfolgen der Länge 210.

Problem: Um eine Vertauschungstabelle für Bitfolgen der Länge 210 vollständig anzugeben, benötigt man mindestens 2^{210} Bits.

- 128 Gigabyte = 2^{40} Bits
- 2^{170} Festplatten mit 128 Gigabyte
- es gibt 2^{170} Atome auf der Erde

Verschlüsselung von Bitfolgen

Wenn Vertauschen von Bitfolgen der Länge 200 unsicher wird, dann wechselt man zu Bitfolgen der Länge 210.

Problem: Um eine Vertauschungstabelle für Bitfolgen der Länge 210 vollständig anzugeben, benötigt man mindestens 2^{210} Bits.

- 128 Gigabyte = 2^{40} Bits
- 2^{170} Festplatten mit 128 Gigabyte
- es gibt 2^{170} Atome auf der Erde

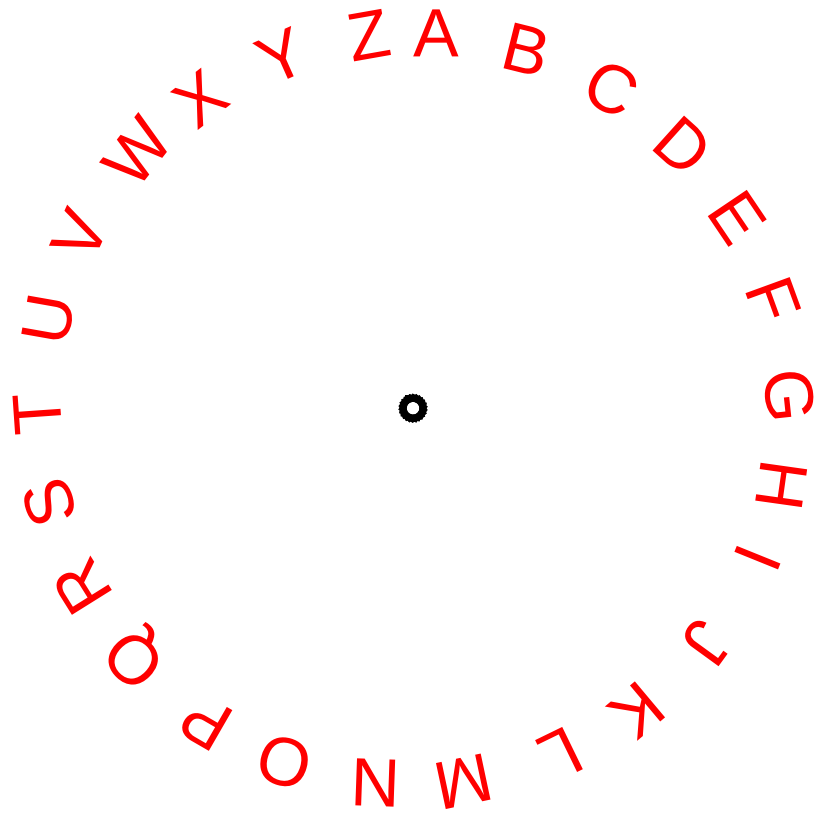
Statt Angabe der Vertauschungstabelle:

Angabe einer mathematischen Funktion.

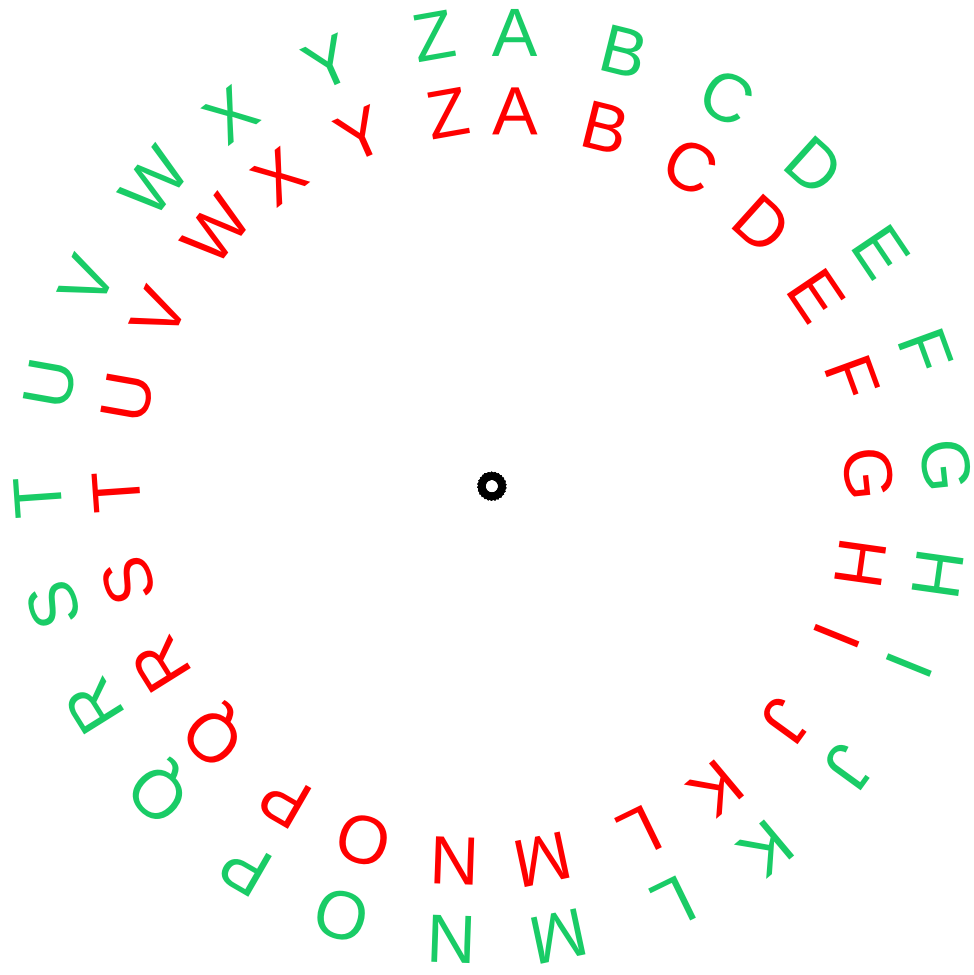
Übersicht

- Verschlüsselung als arithmetische Funktion
  - Addition
  - Multiplikation
  - Potenzierung
- Das RSA-Verfahren
- Sicherheit des RSA-Verfahrens

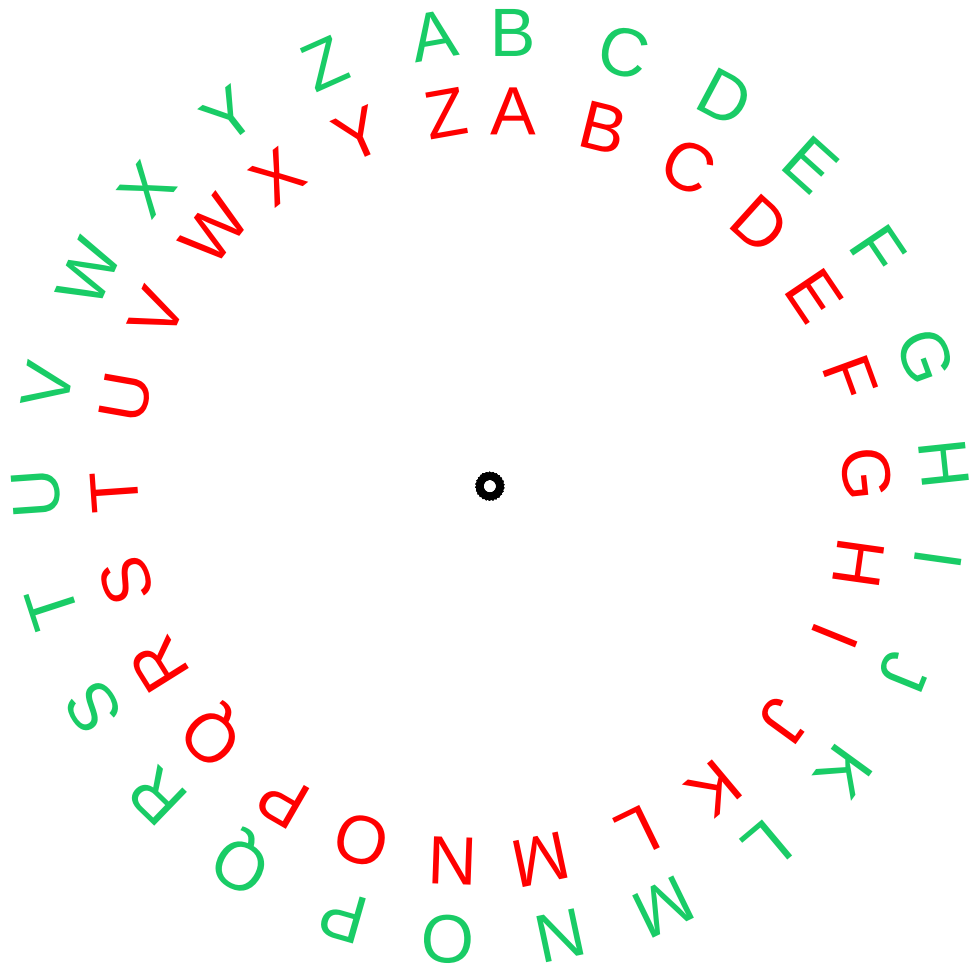
Wie Cäsar verschlüsselt hat



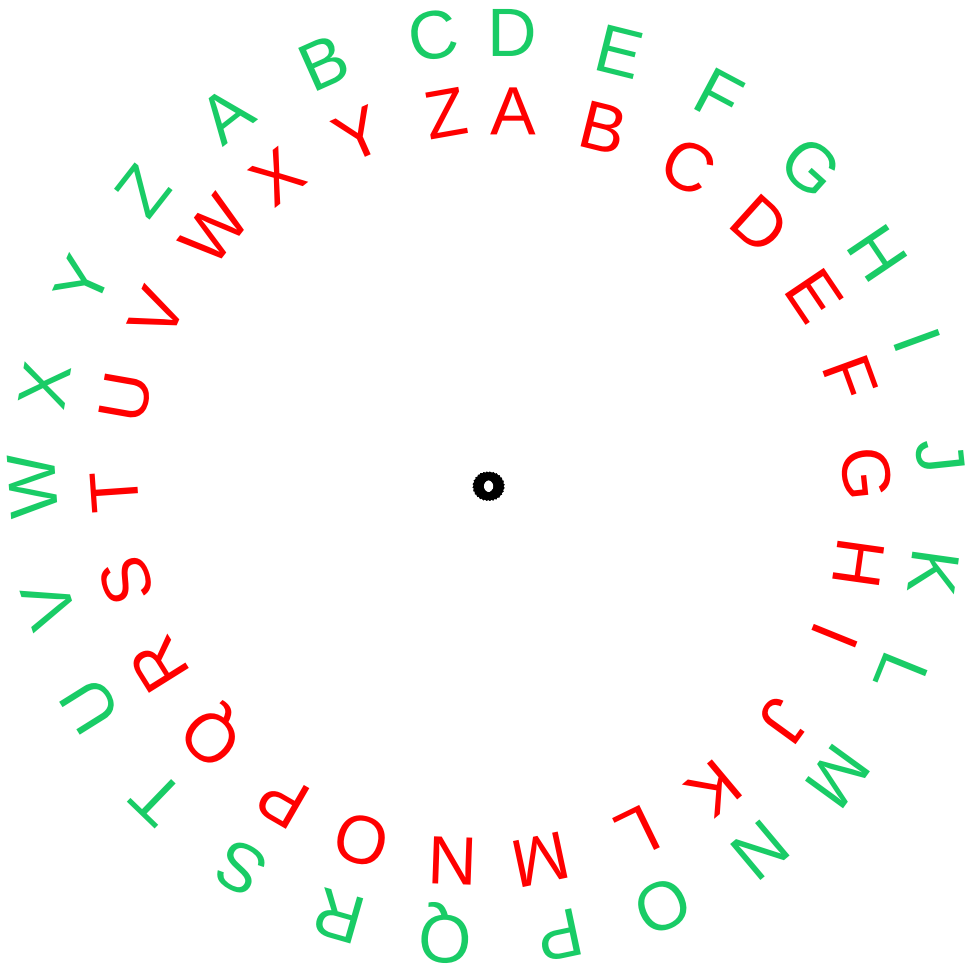
Wie Cäsar verschlüsselt hat



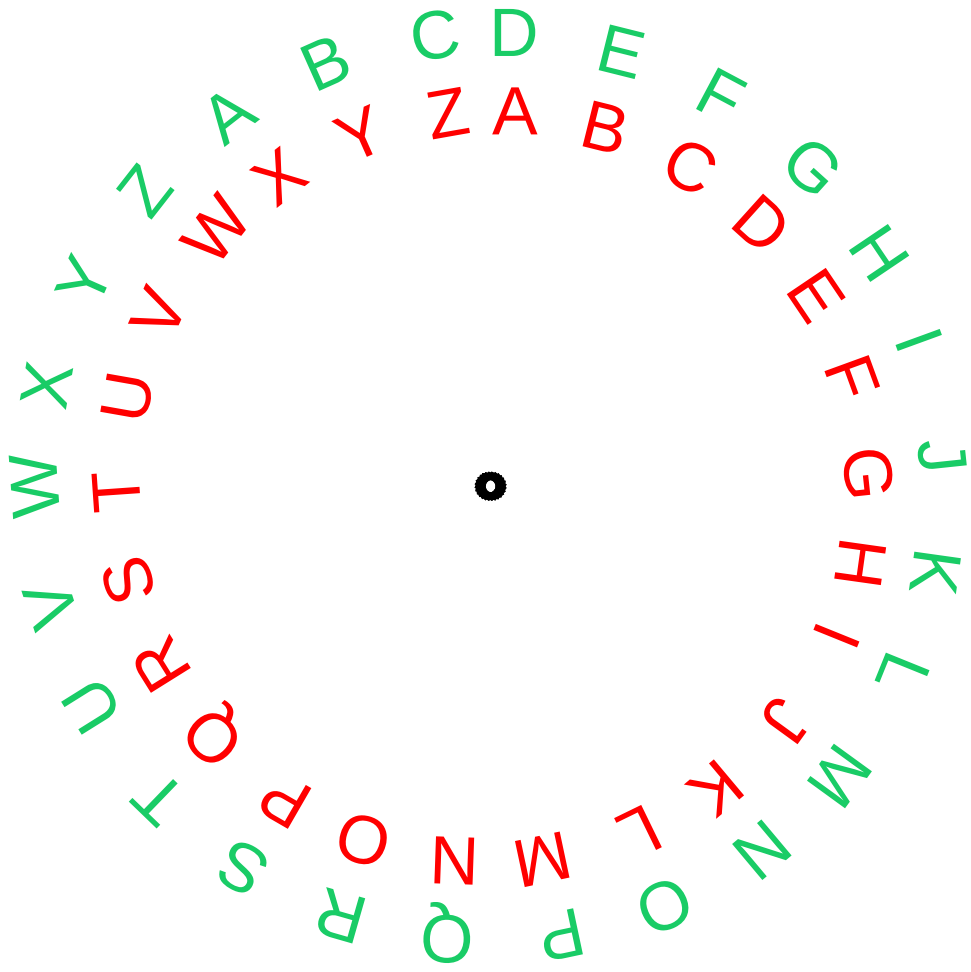
Wie Cäsar verschlüsselt hat



Wie Cäsar verschlüsselt hat

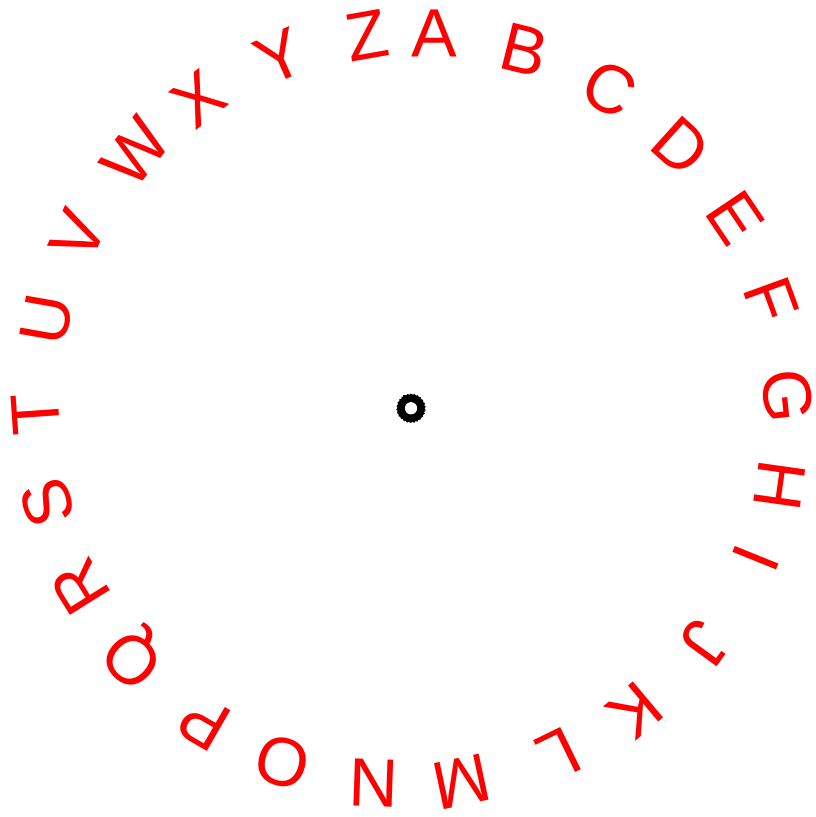


Wie Cäsar verschlüsselt hat

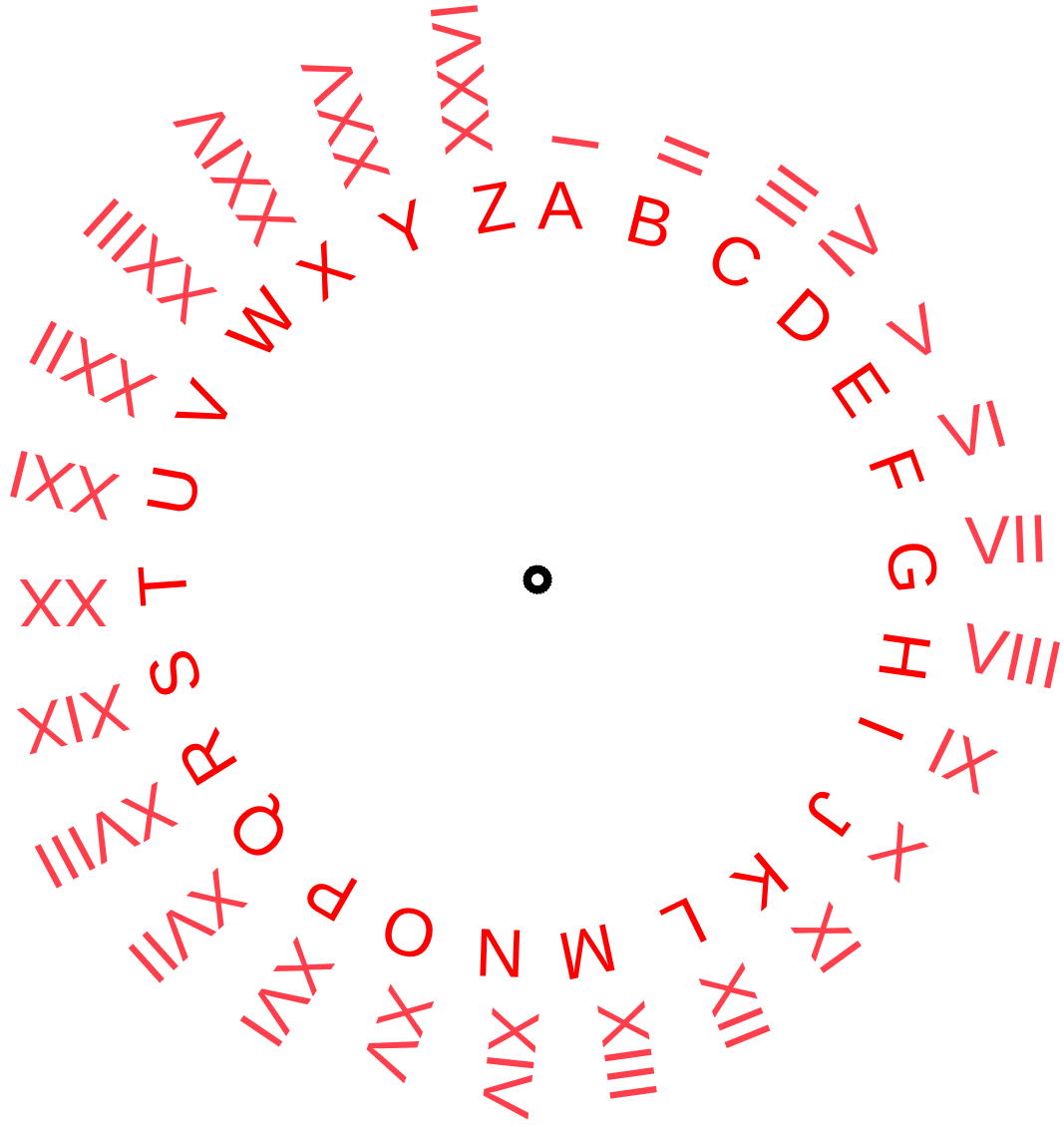


Klartext	VENI VIDI VICI
verschlüsselt	YHQL YLGL YLFL

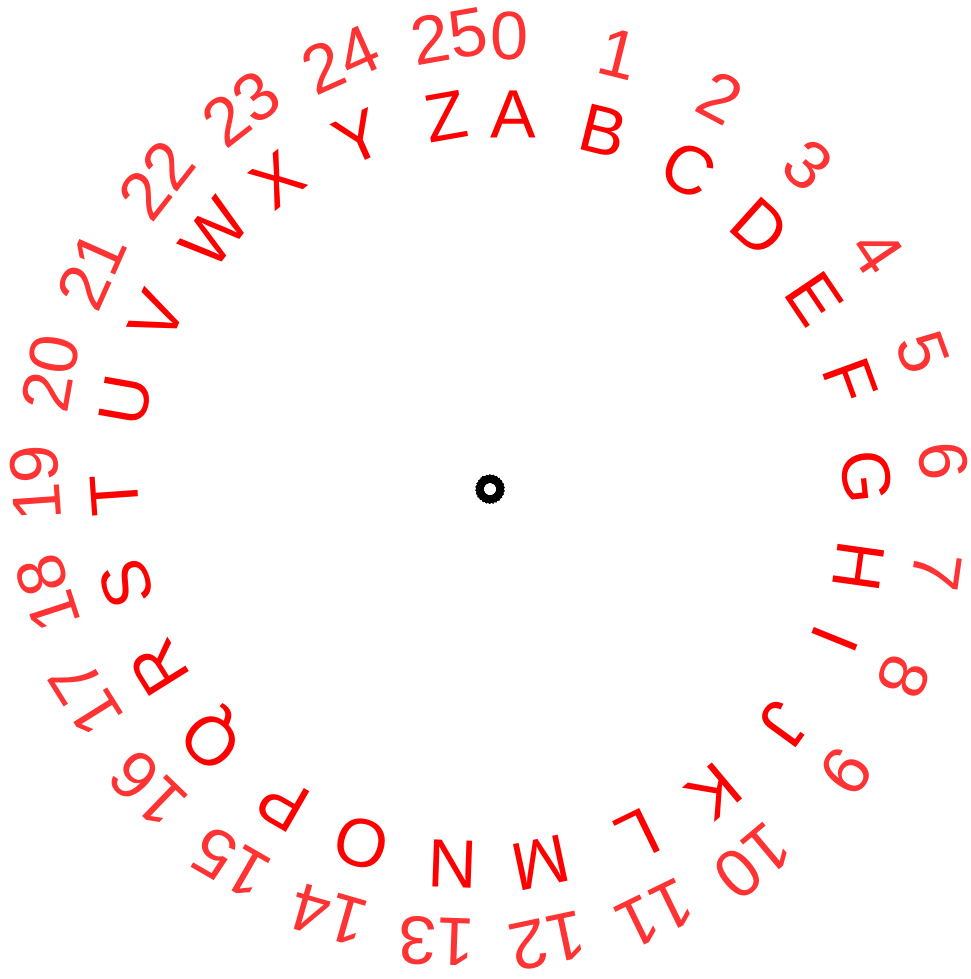
Wie C. mit Zahlen verschlüsselt hätte . . .



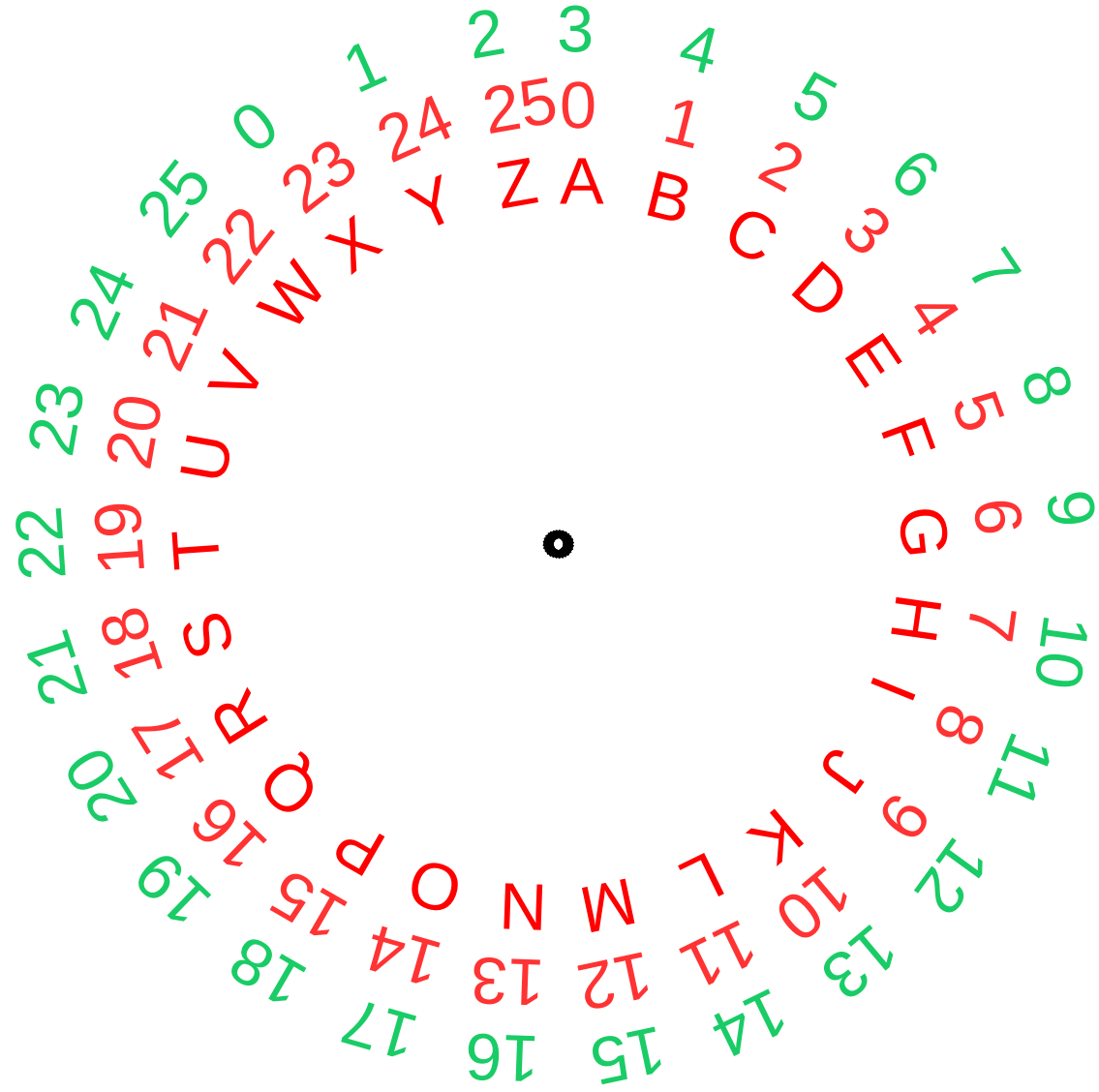
Wie C. mit Zahlen verschlüsselt hätte . . .



Wie C. mit Zahlen verschlüsselt hätte . . .

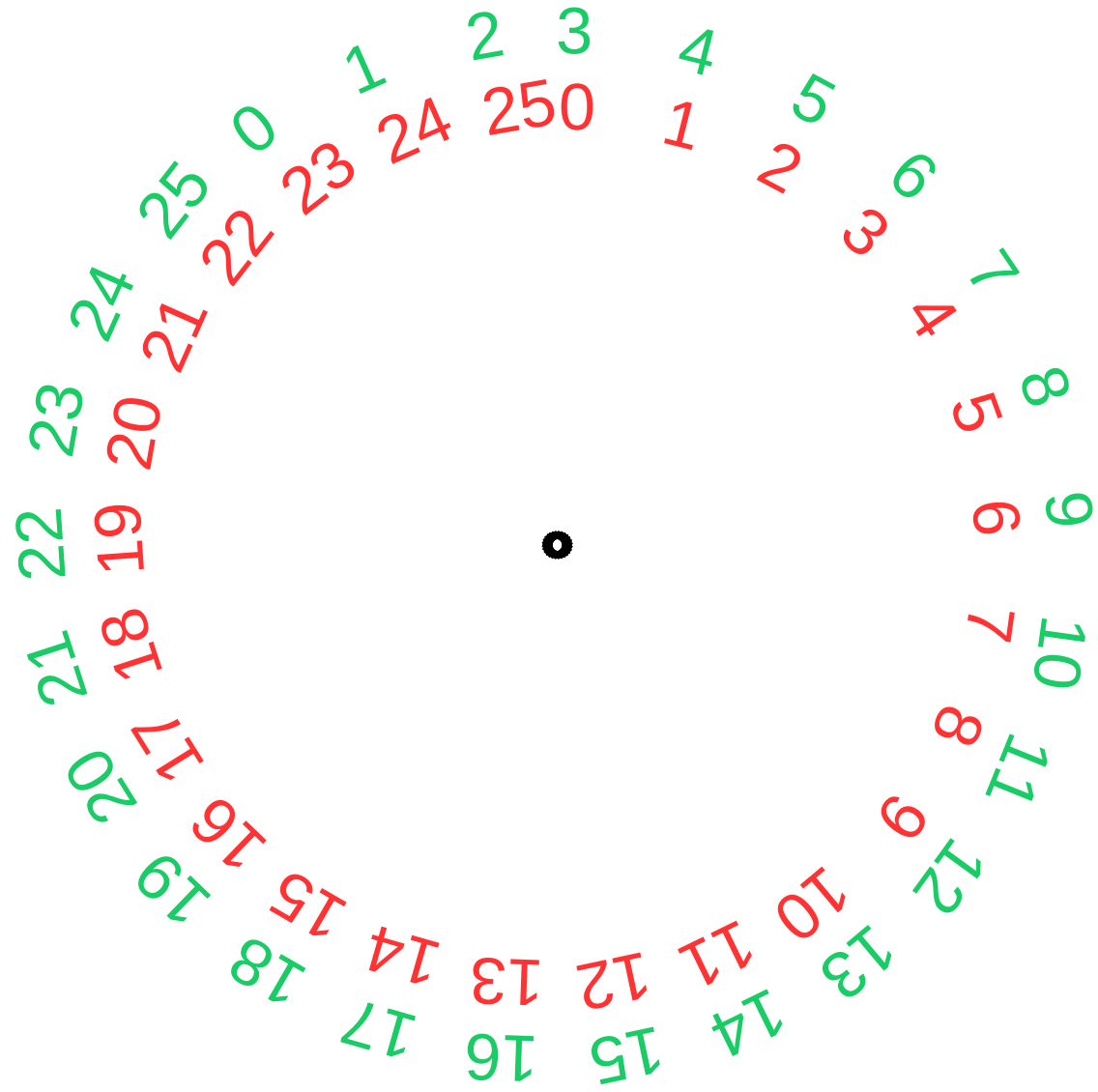


Wie C. mit Zahlen verschlüsselt hätte . . .



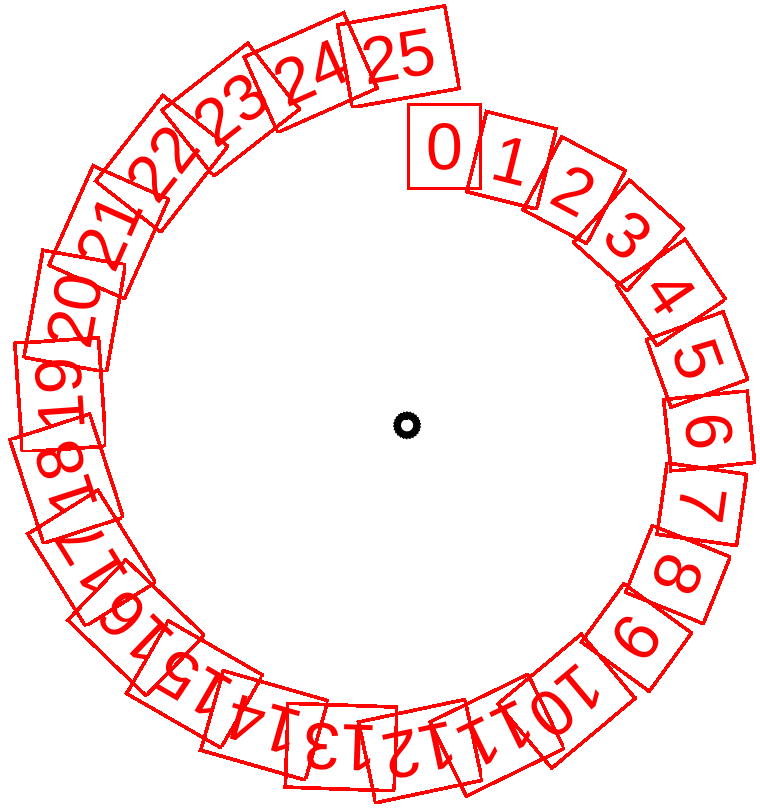
Klartext	VENI	→	21	4	13	8
verschlüsselt	YHQL	←	24	7	16	11

Wie C. mit Zahlen verschlüsselt hätte . . .

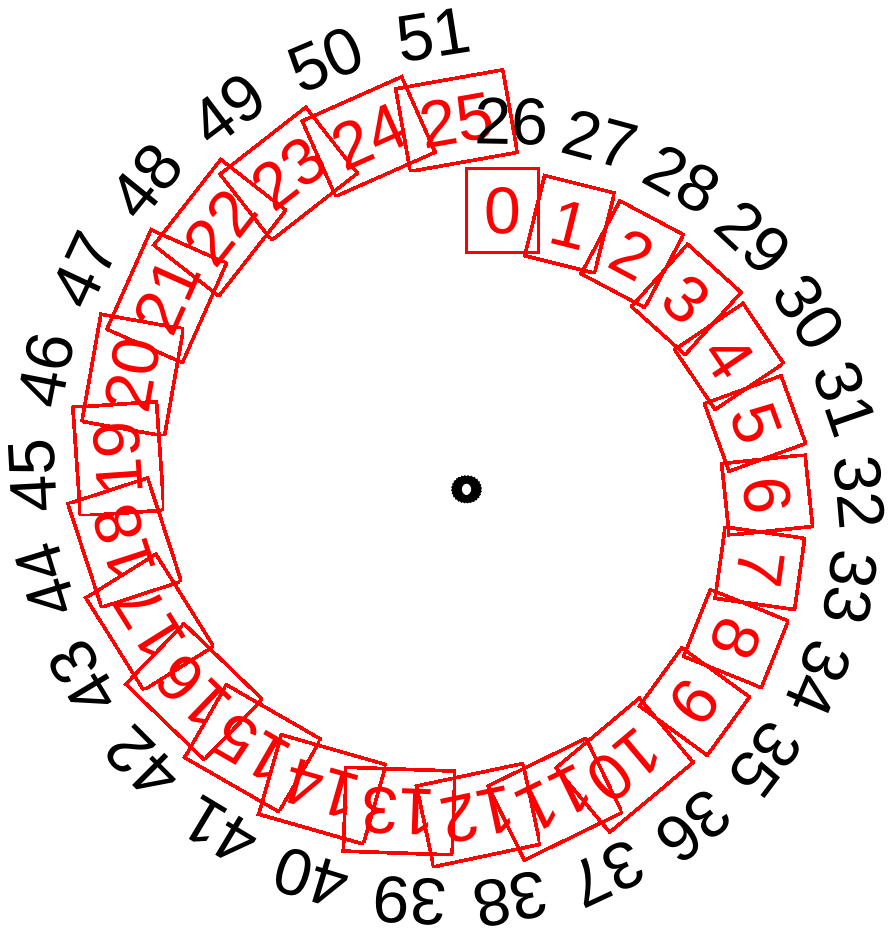


Klartext	VENI	→	21	4	13	8
verschlüsselt	YHQL	←	24	7	16	11

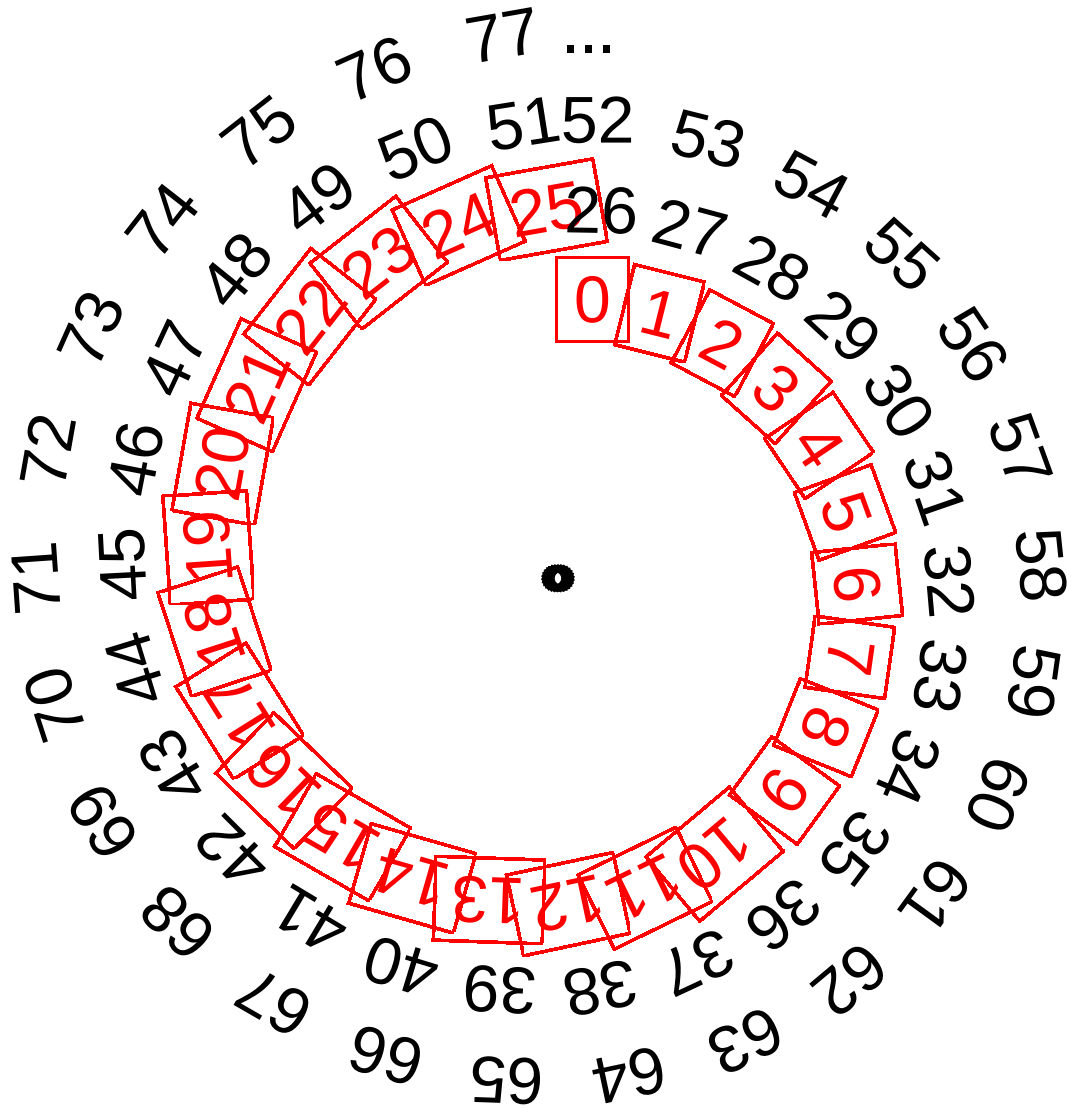
Modulare Arithmetik



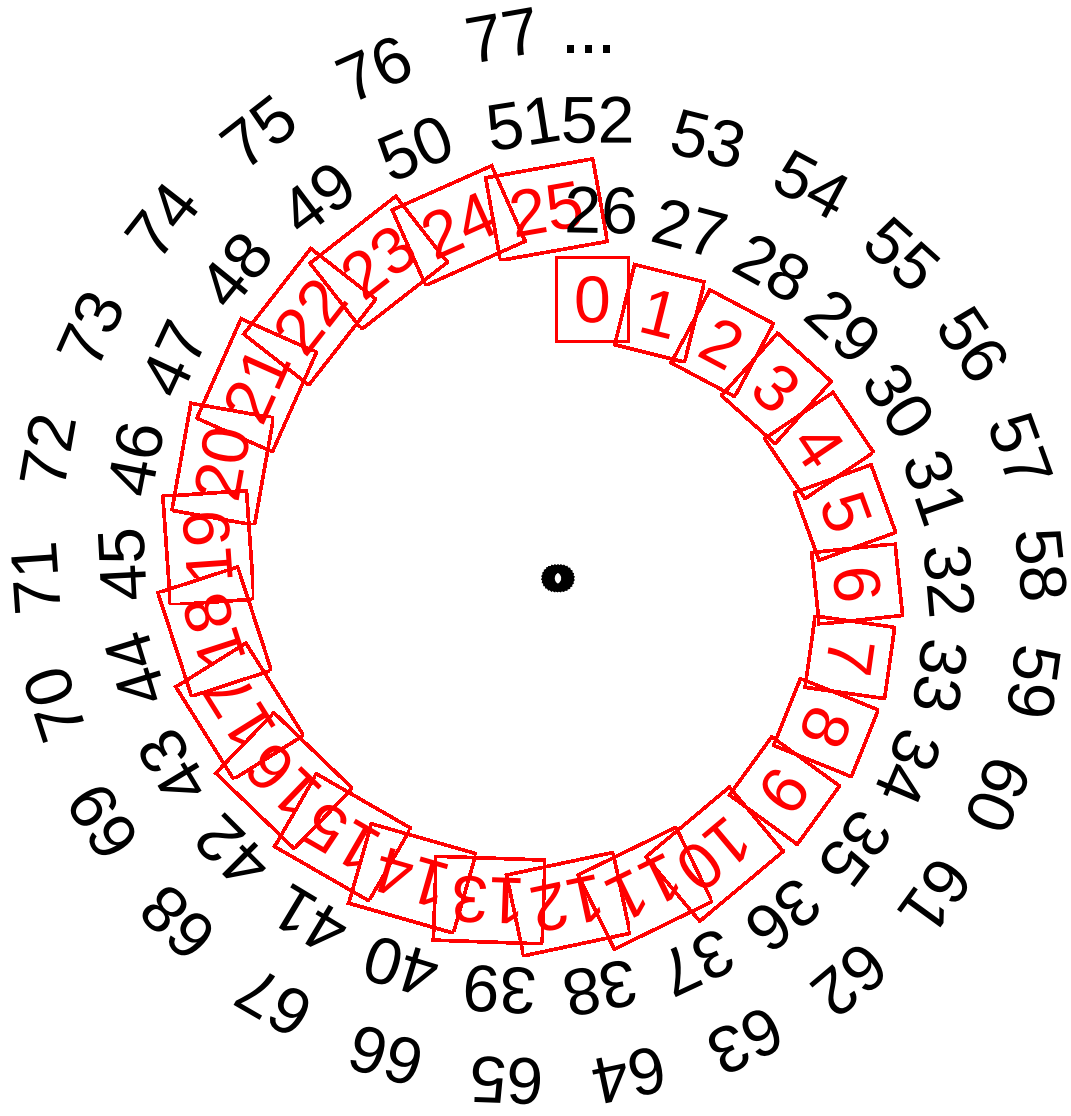
Modulare Arithmetik



Modulare Arithmetik



Modulare Arithmetik



x modulo 26 = der Rest beim Teilen von x durch 26

Cäsars Verschlüsselung à la Arithmetik

Verschlüsselungsfunktion

$$v(x) \;=\; (x+3) \mod 26$$

Cäsars Verschlüsselung à la Arithmetik

Verschlüsselungsfunktion

$$v(x) = (x + 3) \mod 26$$

A wird verschlüsselt zu D

$$v(0) = 3$$

Z wird verschlüsselt zu C

$$v(25) = 2$$

Cäsars Verschlüsselung à la Arithmetik

Verschlüsselungsfunktion

$$v(x) = (x + 3) \mod 26$$

A wird verschlüsselt zu D

$$v(0) = 3$$

Z wird verschlüsselt zu C

$$v(25) = 2$$

Entschlüsselungsfunktion

$$e(y) = (y - 3) \mod 26$$

Cäsars Verschlüsselung à la Arithmetik

Verschlüsselungsfunktion

$$v(x) = (x + 3) \mod 26$$

A wird verschlüsselt zu D

$$v(0) = 3$$

Z wird verschlüsselt zu C

$$v(25) = 2$$

Entschlüsselungsfunktion

$$e(y) = (y - 3) \mod 26$$

$$= (y + 23) \mod 26$$

Cäsars Verschlüsselung à la Arithmetik

Verschlüsselungsfunktion

$$v(x) = (x+3) \mod 26$$

A wird verschlüsselt zu D

$$v(0) = 3$$

Z wird verschlüsselt zu C

$$v(25) = 2$$

Entschlüsselungsfunktion

$$e(y) = (y-3) \mod 26$$

$$= (y+23) \mod 26$$

D wird entschlüsselt zu A

$$e(3) = 0$$

Cäsars Verschlüsselung à la Arithmetik

Allgemein:
Verschlüsselungsfunktion

$$v(x) = (x + a) \mod m$$

Entschlüsselungsfunktion

$$e(y) = (y + b) \mod m$$

für $b = m - a$ (d.h. $b + a = m$).

Cäsars Verschlüsselung à la Arithmetik

Allgemein:
Verschlüsselungsfunktion

$$v(x) = (x + a) \mod m$$

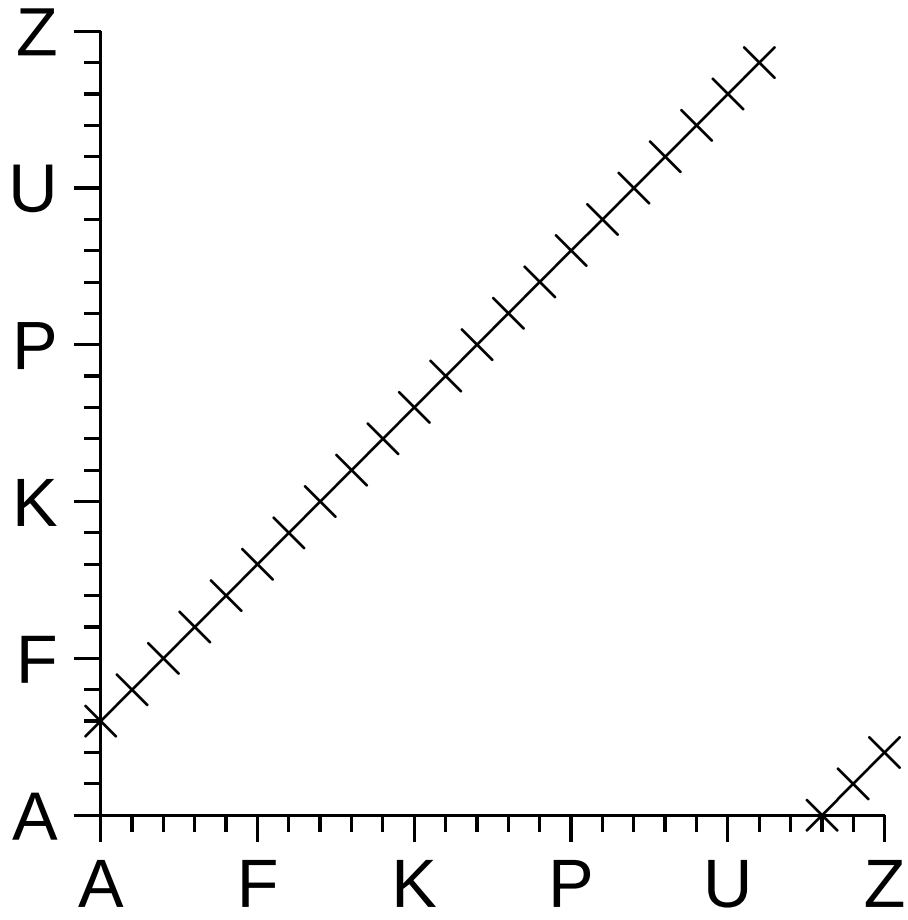
Entschlüsselungsfunktion

$$e(y) = (y + b) \mod m$$

für $b = m - a$ (d.h. $b + a = m$).

Diese Art der Verschlüsselung ist nicht sicher!

Verschlüsselung durch Addition



$$v(x) = (x + 3) \mod 26$$

Verschlüsselung durch Multiplikation

Multiplikation statt Addition

$$v(x) = a \cdot x \mod 26$$

Verschlüsselung durch Multiplikation

Multiplikation statt Addition

$$v(x) = a \cdot x \mod 26$$

Nicht jeder Faktor a liefert eine Verschlüsselungsfunktion:

mit $a = 4$ ergibt sich die Funktion

$$v(x) = 4 \cdot x \mod 26$$

Verschlüsselung durch Multiplikation

Multiplikation statt Addition

$$v(x) = a \cdot x \mod 26$$

Nicht jeder Faktor a liefert eine Verschlüsselungsfunktion:

mit $a = 4$ ergibt sich die Funktion

$$v(x) = 4 \cdot x \mod 26$$

$$4 \cdot 0 \mod 26 = 0 = 4 \cdot 13 \mod 26$$

A A N

Verschlüsselung durch Multiplikation

Multiplikation statt Addition

$$v(x) = a \cdot x \mod 26$$

Nicht jeder Faktor a liefert eine Verschlüsselungsfunktion:

mit $a = 4$ ergibt sich die Funktion

$$v(x) = 4 \cdot x \mod 26$$

$$4 \cdot 0 \mod 26 = 0 = 4 \cdot 13 \mod 26$$

A A N

ANNA wird verschlüsselt zu AAAA.

AAAA kann nicht eindeutig entschlüsselt werden

AAAA, NANA, NNAA, . . .

Verschlüsselung durch Multiplikation

Multiplikation statt Addition

$$v(x) = a \cdot x \mod 26$$

Nicht jeder Faktor a liefert eine Verschlüsselungsfunktion:

mit $a = 4$ ergibt sich die Funktion

$$v(x) = 4 \cdot x \mod 26$$

$$4 \cdot 0 \mod 26 = 0 = 4 \cdot 13 \mod 26$$

A A N

Besitzen a und m einen gemeinsamen Teiler > 1,
dann ist $v(x) = a \cdot x \mod m$ keine Verschlüsselungsfunktion.

Verschlüsselung mit Multiplikation

Aber: $v(x) = 7 \cdot x \mod 26$

ist eine Verschlüsselungsfunktion!

Verschlüsselung mit Multiplikation

Aber: $v(x) = 7 \cdot x \mod 26$

ist eine Verschlüsselungsfunktion!

Wie sieht die passende Entschlüsselungsfunktion aus?

Verschlüsselung mit Multiplikation

Aber: $v(x) = 7 \cdot x \mod 26$

ist eine Verschlüsselungsfunktion!

Wie sieht die passende Entschlüsselungsfunktion aus?

Kann man modular dividieren?

Verschlüsselung mit Multiplikation

Aber: $v(x) = 7 \cdot x \mod 26$

ist eine Verschlüsselungsfunktion!

Wie sieht die passende Entschlüsselungsfunktion aus?

Kann man modular dividieren?

Es gilt

$$7 \cdot 15 \mod 26 = 105 \mod 26 = 1 \mod 26$$

Verschlüsselung mit Multiplikation

Aber: $v(x) = 7 \cdot x \mod 26$

ist eine Verschlüsselungsfunktion!

Wie sieht die passende Entschlüsselungsfunktion aus?

Kann man modular dividieren?

Es gilt

$$7 \cdot 15 \mod 26 = 105 \mod 26 = 1 \mod 26$$

Also ist $e(y) = 15 \cdot y \mod 26$

die zu v passende Entschlüsselungsfunktion.

Multiplikation statt Division!

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

Formulierung 300 v.Chr.:

Nimmt man immer wieder das Kleinere vom Größeren weg, dann muss der Rest schließlich die vorhergehende Größe messen.

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

Formulierung 300 v.Chr.:

Nimmt man immer wieder das Kleinere vom Größeren weg, dann muss der Rest schließlich die vorhergehende Größe messen.

Formulierung heutzutage:

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

Berechnung des größten gemeinsamen Teilers von 26 und 7:

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

Berechnung des größten gemeinsamen Teilers von 26 und 7:

ggT(26,7)

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

Berechnung des größten gemeinsamen Teilers von 26 und 7:

ggT(26,7)

ggT(7,

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

Berechnung des größten gemeinsamen Teilers von 26 und 7:

ggT(26,7)

ggT(7,5)

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

Berechnung des größten gemeinsamen Teilers von 26 und 7:

ggT(26,7)

ggT(7,5)

ggT(5,2)

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

Berechnung des größten gemeinsamen Teilers von 26 und 7:

ggT(26,7)
ggT(7,5)
ggT(5,2)
ggT(2,1)

Der Algorithmus von Euklid

zur Berechnung des größten gemeinsamen Teilers

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

Berechnung des größten gemeinsamen Teilers von 26 und 7:

ggT(26,7)
ggT(7,5)
ggT(5,2)
ggT(2,1)
1

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$

$\mathrm{ggT}(7,5)$

$\mathrm{ggT}(5,2)$

$\mathrm{ggT}(2,1)$

1

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$
$\mathrm{ggT}(7,5)$
$\mathrm{ggT}(5,2)$ $\quad 1 = 5 - 2 \cdot 2$
$\mathrm{ggT}(2,1)$
$1 \quad =$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$
$\mathrm{ggT}(7,5)$
$\mathrm{ggT}(5,2)$ $\quad 1 = 5 - 2 \cdot 2$
$\mathrm{ggT}(2,1)$

$$1 = 5 - 2 \cdot 2$$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$

$\mathrm{ggT}(7,5)$ $\quad 2 = 7 - 1 \cdot 5$

$\mathrm{ggT}(5,2)$

$\mathrm{ggT}(2,1)$

$1 \quad = \quad 5 - 2 \cdot 2$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$

$\mathrm{ggT}(7,5)$ $\quad 2 = 7 - 1 \cdot 5$

$\mathrm{ggT}(5,2)$

$\mathrm{ggT}(2,1)$

$1 \quad = \quad 5 - 2 \cdot 2 = 5 - 2 \cdot (7 - 5)$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$
$\mathrm{ggT}(7,5)$
$\mathrm{ggT}(5,2)$
$\mathrm{ggT}(2,1)$

$$1 = 3 \cdot 5 - 2 \cdot 7$$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$ $5 = 26 - 3 \cdot 7$

$\mathrm{ggT}(7,5)$

$\mathrm{ggT}(5,2)$

$\mathrm{ggT}(2,1)$

$1 = 3 \cdot 5 - 2 \cdot 7$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$ $5 = 26 - 3 \cdot 7$

$\mathrm{ggT}(7,5)$

$\mathrm{ggT}(5,2)$

$\mathrm{ggT}(2,1)$

$1 = 3 \cdot 5 - 2 \cdot 7 = 3 \cdot (26 - 3 \cdot 7) - 2 \cdot 7$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$
$\mathrm{ggT}(7,5)$
$\mathrm{ggT}(5,2)$
$\mathrm{ggT}(2,1)$

$$1 = 3 \cdot 26 - 11 \cdot 7$$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$\mathrm{ggT}(26,7)$
$\mathrm{ggT}(7,5)$
$\mathrm{ggT}(5,2)$
$\mathrm{ggT}(2,1)$

$$1 = 3 \cdot 26 - 11 \cdot 7$$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$$\mathrm{ggT}(26,7) \quad = \quad 1 \quad = \quad 3 \cdot 26 - 11 \cdot 7$$

Berechnung des multiplikativ Inversen

mit dem Algorithmus von Euklid

$$\mathrm{ggT}(a,b) = \begin{cases} b, & \text{falls } a \bmod b = 0 \\ \mathrm{ggT}(b, a \bmod b), & \text{sonst} \end{cases}$$

$$\mathrm{ggT}(26,7) \quad = \quad 1 \quad = \quad 3 \cdot 26 - 11 \cdot 7$$

Für alle positiven ganzen Zahlen a und b
gibt es ganze Zahlen q und r, so dass

$$\mathrm{ggT}(a,b) = q \cdot a + r \cdot b \quad .$$

[Für teilerfremde Zahlen: Lemma von Bachet (1581-1638)]

Anwendung des Lemmas von Bachet

Seien a und 26 teilerfremd (also ggT$(a, 26) = 1$).
Dann gibt es q und r mit

$$1 = q \cdot a + r \cdot 26 \quad .$$

Anwendung des Lemmas von Bachet

Seien a und 26 teilerfremd (also ggT$(a, 26) = 1$).
Dann gibt es q und r mit

$$1 = q \cdot a + r \cdot 26 \quad .$$

Betrachte alles modulo 26:

$$1 \quad \equiv \quad (q \cdot a + r \cdot 26) \bmod 26 \quad \equiv \quad (q \cdot a) \bmod 26$$

Anwendung des Lemmas von Bachet

Seien a und 26 teilerfremd (also ggT$(a, 26) = 1$).
Dann gibt es q und r mit

$$1 = q \cdot a + r \cdot 26 \quad .$$

Betrachte alles modulo 26:

$$1 \quad \equiv \quad (q \cdot a + r \cdot 26) \bmod 26 \quad \equiv \quad (q \cdot a) \bmod 26$$

Also gilt für jedes z

$$\begin{aligned} z &\equiv z \cdot 1 \bmod 26 \\ &\equiv z \cdot (q \cdot a) \bmod 26 \\ &\equiv (z \cdot a) \cdot q \bmod 26 \end{aligned}$$

Anwendung des Lemmas von Bachet

Seien a und 26 teilerfremd (also ggT$(a, 26) = 1$).
Dann gibt es q und r mit

$$1 = q \cdot a + r \cdot 26 \quad .$$

Betrachte alles modulo 26:

$$1 \quad \equiv \quad (q \cdot a + r \cdot 26) \bmod 26 \quad \equiv \quad (q \cdot a) \bmod 26$$

Also gilt für jedes z

$$z \quad \equiv \quad (z \cdot a) \cdot q \quad \bmod 26$$

Eine Multiplikation mit a wird durch
eine weitere Multiplikation mit q wieder rückgängig gemacht!

q heißt *multiplikativ Inverses zu a (*mod 26*)*.

Multiplikative Verschlüsselung

Verschlüsselungsfunktion: $v(x) = a \cdot x \mod m$

für ein zu m teilerfremdes a.

Entschlüsselungsfunktion: $e(y) = q \cdot y \mod m$

für das multiplikativ Inverse q zu a ($\mod m$).

Multiplikative Verschlüsselung

Verschlüsselungsfunktion: $v(x) = a \cdot x \mod m$

für ein zu m teilerfremdes a.

Entschlüsselungsfunktion: $e(y) = q \cdot y \mod m$

für das multiplikativ Inverse q zu a ($\mod m$).

a teilerfremd zu 26	1	3	5	7	9	11	15	17	19	21	23
mult. Inverses von a	1	9	21	15	3	19	7	23	11	5	17

Multiplikative Verschlüsselung

Verschlüsselungsfunktion: $v(x) = a \cdot x \mod m$

für ein zu m teilerfremdes a.

Entschlüsselungsfunktion: $e(y) = q \cdot y \mod m$

für das multiplikativ Inverse q zu a ($\mod m$).

a teilerfremd zu 26	1	3	5	7	9	11	15	17	19	21	23
mult. Inverses von a	1	9	21	15	3	19	7	23	11	5	17

$v(x) = 3 \cdot x \mod 26$ $\quad$ $e(y) = 9 \cdot y \mod 26$

Multiplikative Verschlüsselung

Verschlüsselungsfunktion: $v(x) = a \cdot x \mod m$

für ein zu m teilerfremdes a.

Entschlüsselungsfunktion: $e(y) = q \cdot y \mod m$

für das multiplikativ Inverse q zu $a \pmod m$.

$v(x) = 3 \cdot x \mod 26$ $\quad$ $e(y) = 9 \cdot y \mod 26$

Klartext `VENI` $\rightarrow$ `21 4 23 8`

Multiplikative Verschlüsselung

Verschlüsselungsfunktion: $v(x) = a \cdot x \mod m$

für ein zu m teilerfremdes a.

Entschlüsselungsfunktion: $e(y) = q \cdot y \mod m$

für das multiplikativ Inverse q zu a ($\mod m$).

$v(x) = 3 \cdot x \mod 26$ $\quad$ $e(y) = 9 \cdot y \mod 26$

Klartext	VENI	$\rightarrow$	21	4	23	8
verschlüsselt	JKPX	$\leftarrow$	11	12	17	24

Multiplikative Verschlüsselung

Verschlüsselungsfunktion: $v(x) = a \cdot x \mod m$

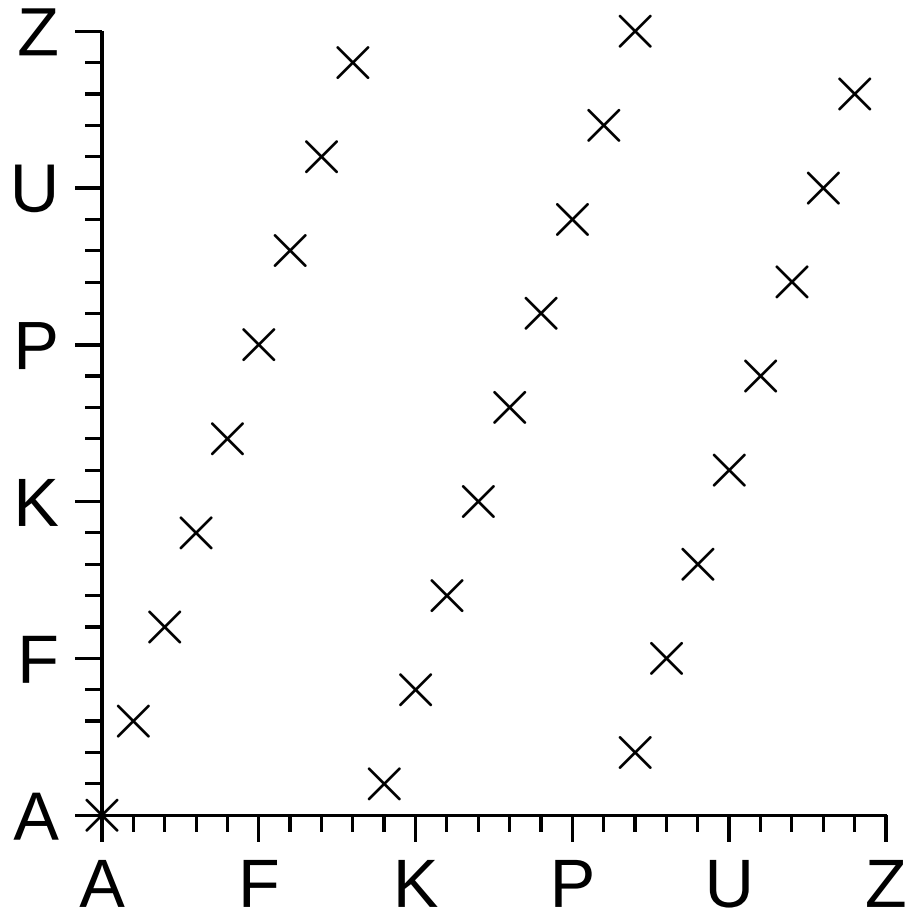
für ein zu m teilerfremdes a.

Entschlüsselungsfunktion: $e(y) = q \cdot y \mod m$

für das multiplikativ Inverse q zu a ($\mod m$).

Multiplikative Verschlüsselung ist nicht sicher.

Verschlüsselung durch Multiplikation



$$v(x) = 3 \cdot x \mod 26$$

Verschlüsselung durch Potenzierung

Beispiel:

Verschlüsselungsfunktion: $v(x) = x^3 \mod 29$

Entschlüsselungsfunktion: $e(y) = y^{19} \mod 29$

Verschlüsselung durch Potenzierung

Beispiel:

Verschlüsselungsfunktion: $v(x) = x^3 \mod 29$

Entschlüsselungsfunktion: $e(y) = y^{19} \mod 29$

Wie müssen die Exponenten gewählt werden?

Verschlüsselung durch Potenzierung

Beispiel:

Verschlüsselungsfunktion: $v(x) = x^3 \mod 29$

Entschlüsselungsfunktion: $e(y) = y^{19} \mod 29$

Wie müssen die Exponenten gewählt werden?

$$x = (x^3)^{19} \mod 29 = x^{3 \cdot 19} \mod 29$$

Verschlüsselung durch Potenzierung

Beispiel:

Verschlüsselungsfunktion: $v(x) = x^3 \mod 29$

Entschlüsselungsfunktion: $e(y) = y^{19} \mod 29$

Wie müssen die Exponenten gewählt werden?

$$x = (x^3)^{19} \mod 29 = x^{3 \cdot 19} \mod 29$$

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p$ gilt:

$$x^p \equiv x \mod p \ .$$

[Kleiner Satz von Fermat (1601-1665)]

Beweis des kleinen Satzes von Fermat

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Beweis des kleinen Satzes von Fermat

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei $x \in \mathbb{Z}_p^+$, und $a < b$ seien aus $\mathbb{Z}_p^+$. Dann gilt

$x \cdot (b-a) \not\equiv 0 \mod p$ und folglich $b \cdot x \not\equiv a \cdot x \mod p$.

Beweis des kleinen Satzes von Fermat

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei $x \in \mathbb{Z}_p^+$, und $a < b$ seien aus $\mathbb{Z}_p^+$. Dann gilt

$x \cdot (b-a) \not\equiv 0 \mod p$ und folglich $b \cdot x \not\equiv a \cdot x \mod p$.

Also haben die $p-1$ Zahlen

$$x,\ 2 \cdot x,\ 3 \cdot x, \ldots,\ (p-1) \cdot x$$

alle möglichen Reste ungleich 0 beim Teilen durch p

$$1,\ 2,\ 3, \ldots,\ (p-1) \quad .$$

Beweis des kleinen Satzes von Fermat

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Also haben die $p-1$ Zahlen

$$x,\ 2\cdot x,\ 3\cdot x, \ldots,\ (p-1)\cdot x$$

alle möglichen Reste ungleich 0 beim Teilen durch p

$$1,\ 2,\ 3, \ldots,\ (p-1) \quad .$$

Dann gilt

$$x\cdot(2\cdot x)\cdot(3\cdot x)\cdot\ldots\cdot((p-1)\cdot x) \quad \equiv \quad 1\cdot 2\cdot 3\cdot\ldots\cdot(p-1) \mod p$$

Beweis des kleinen Satzes von Fermat

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Also haben die $p-1$ Zahlen

$$x,\ 2\cdot x,\ 3\cdot x, \ldots,\ (p-1)\cdot x$$

alle möglichen Reste ungleich 0 beim Teilen durch p

$$1,\ 2,\ 3, \ldots,\ (p-1) \quad .$$

Dann gilt

$$\begin{aligned} x\cdot(2\cdot x)\cdot(3\cdot x)\cdot\ldots\cdot((p-1)\cdot x) &\equiv 1\cdot 2\cdot 3\cdot\ldots\cdot(p-1) \mod p \\ 1\cdot 2\cdot 3\cdot\ldots\cdot(p-1)\cdot x^{p-1} &\equiv 1\cdot 2\cdot 3\cdot\ldots\cdot(p-1) \mod p \end{aligned}$$

Beweis des kleinen Satzes von Fermat

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Also haben die $p-1$ Zahlen

$$x,\ 2\cdot x,\ 3\cdot x, \ldots,\ (p-1)\cdot x$$

alle möglichen Reste ungleich 0 beim Teilen durch p

$$1,\ 2,\ 3, \ldots,\ (p-1) \quad .$$

Dann gilt

$$\begin{aligned} x\cdot(2\cdot x)\cdot(3\cdot x)\cdot\ldots\cdot((p-1)\cdot x) &\equiv 1\cdot 2\cdot 3\cdot\ldots\cdot(p-1) \mod p \\ 1\cdot 2\cdot 3\cdot\ldots\cdot(p-1)\cdot x^{p-1} &\equiv 1\cdot 2\cdot 3\cdot\ldots\cdot(p-1) \mod p \\ x^{p-1} &\equiv 1 \mod p \end{aligned}$$

Verschlüsselung durch Potenzierung

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei a teilerfremd zu $p-1$, und b multiplikativ Inverses zu a.

Also gilt $a \cdot b \equiv 1 \mod (p-1)$.

Verschlüsselung durch Potenzierung

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei a teilerfremd zu $p-1$, und b multiplikativ Inverses zu a.

Also gilt $a \cdot b \equiv 1 \mod (p-1)$.

$$(x^a)^b$$

Verschlüsselung durch Potenzierung

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei a teilerfremd zu $p-1$, und b multiplikativ Inverses zu a.

Also gilt $a \cdot b \equiv 1 \mod (p-1)$.

$$(x^a)^b = x^{a \cdot b}$$

Verschlüsselung durch Potenzierung

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei a teilerfremd zu $p-1$, und b multiplikativ Inverses zu a.

Also gilt $a \cdot b \equiv 1 \mod (p-1)$.

$$(x^a)^b = x^{a \cdot b} = x^{(p-1) \cdot t + 1}$$

Verschlüsselung durch Potenzierung

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei a teilerfremd zu $p-1$, und b multiplikativ Inverses zu a.

Also gilt $a \cdot b \equiv 1 \mod (p-1)$.

$$(x^a)^b = x^{a \cdot b} = x^{(p-1) \cdot t + 1} = (x^{p-1})^t \cdot x^1$$

Verschlüsselung durch Potenzierung

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei a teilerfremd zu $p-1$, und b multiplikativ Inverses zu a.

Also gilt $a \cdot b \equiv 1 \mod (p-1)$.

$$(x^a)^b = x^{a \cdot b} = x^{(p-1) \cdot t+1} = (x^{p-1})^t \cdot x^1 \equiv x \mod p$$

Verschlüsselung durch Potenzierung

Für jede Primzahl p und jedes $x \in \mathbb{Z}_p^+$ gilt: $x^{p-1} \equiv 1 \mod p$.

Sei a teilerfremd zu $p-1$, und b multiplikativ Inverses zu a.

Also gilt $a \cdot b \equiv 1 \mod (p-1)$.

$$(x^a)^b = x^{a \cdot b} = x^{(p-1) \cdot t+1} = (x^{p-1})^t \cdot x^1 \equiv x \mod p$$

Dann ist

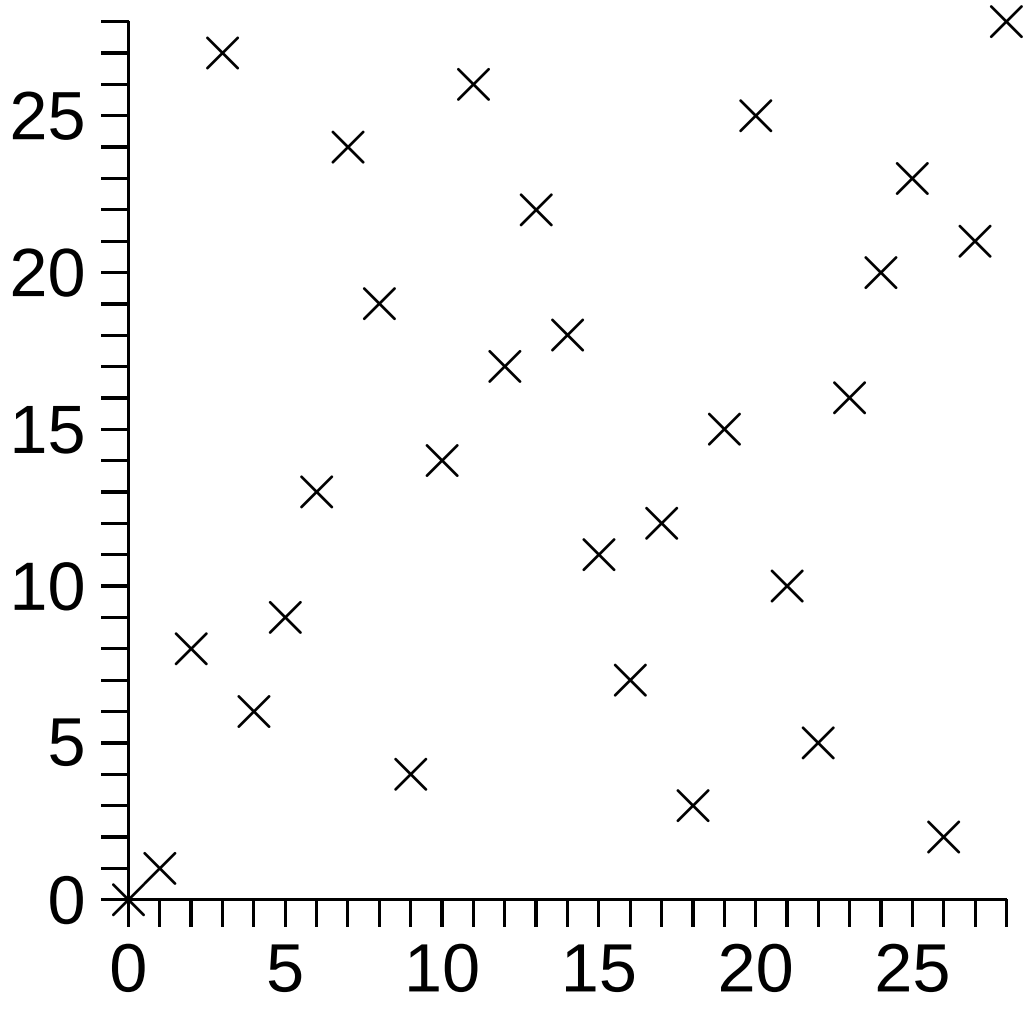
$$v(x) = x^a \mod p$$

eine Verschlüsselungsfunktion und

$$e(y) = y^b \mod p$$

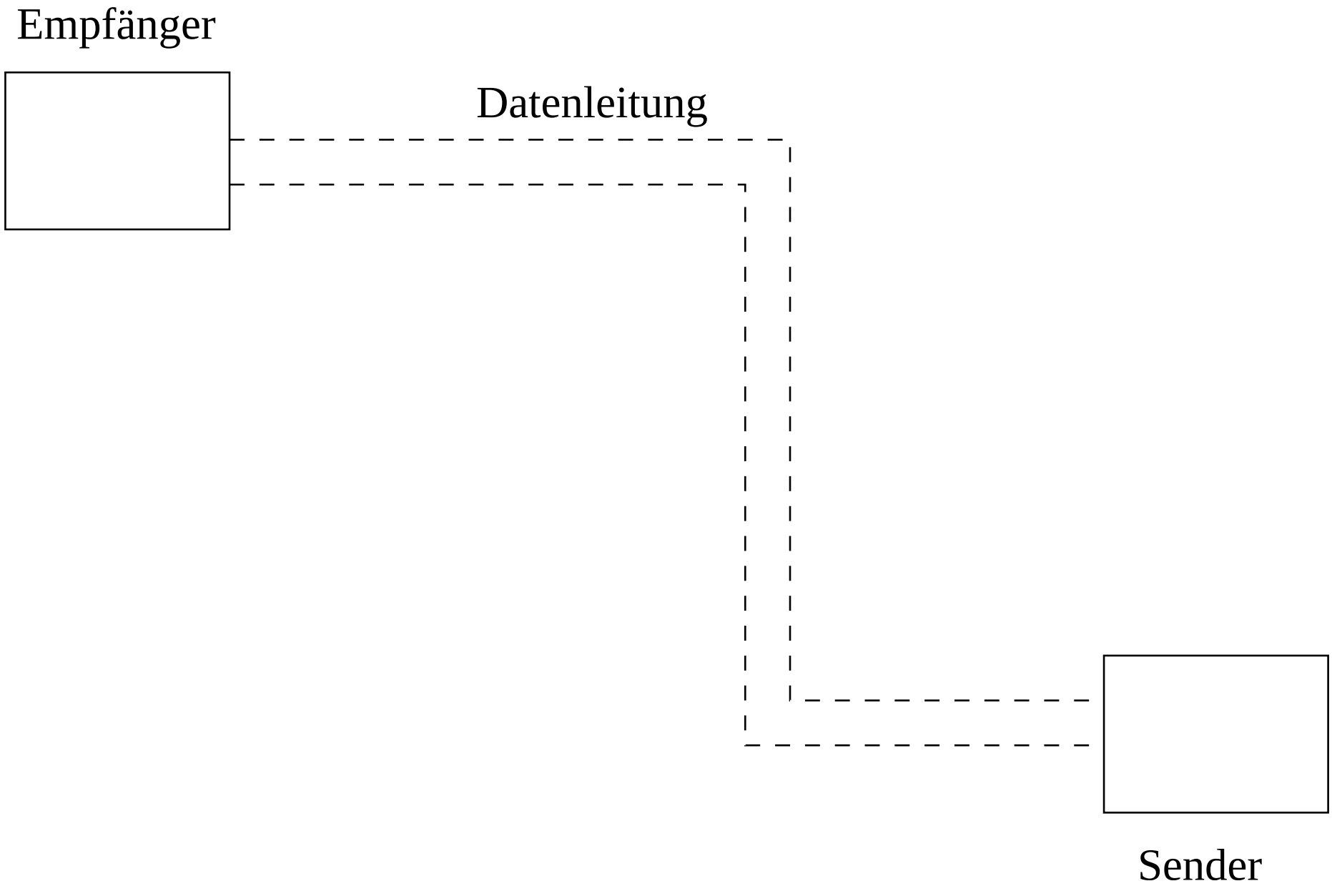
die dazugehörige Entschlüsselungsfunktion.

Verschlüsselung durch Potenzierung

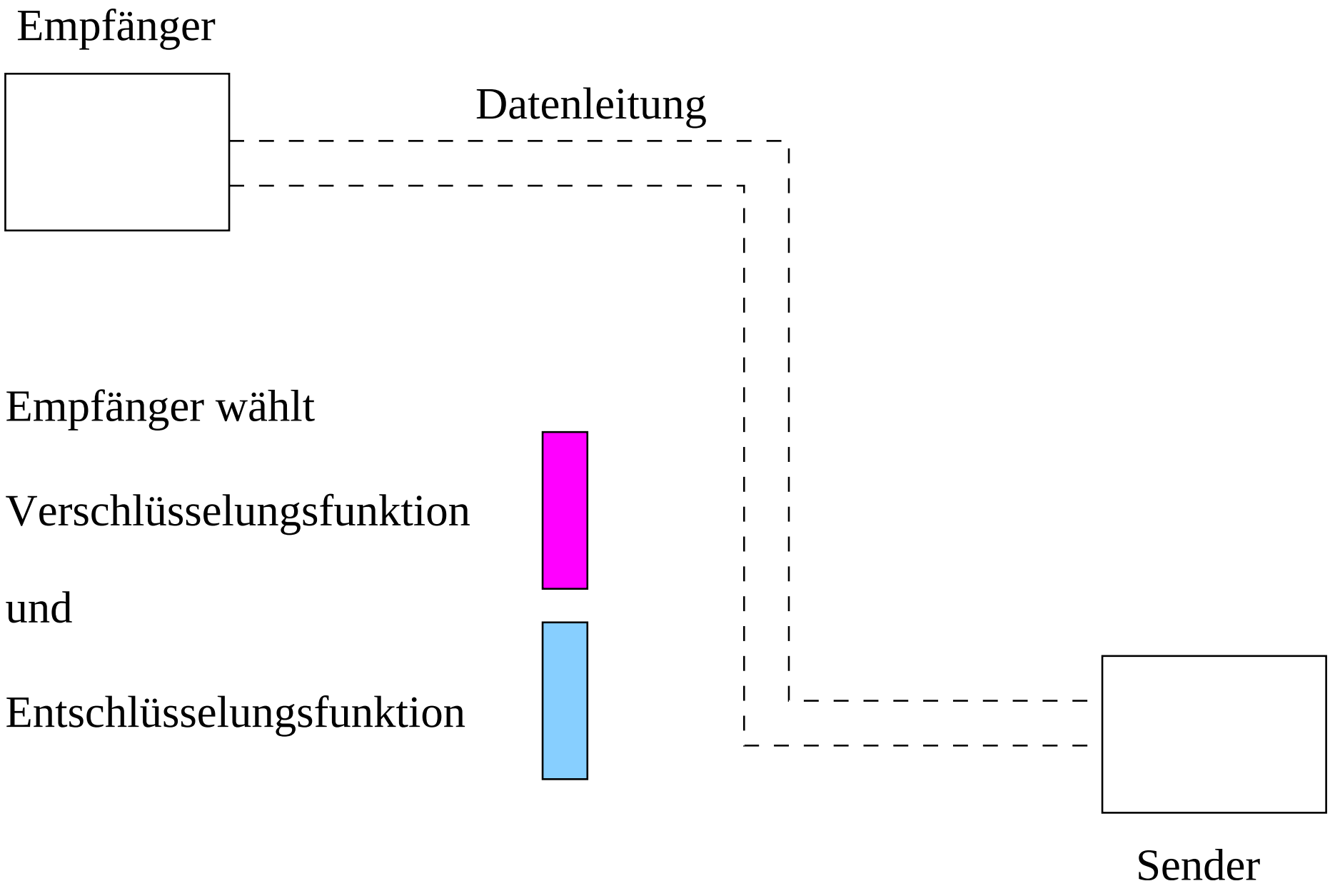


$$v(x) = x^3 \bmod 29$$

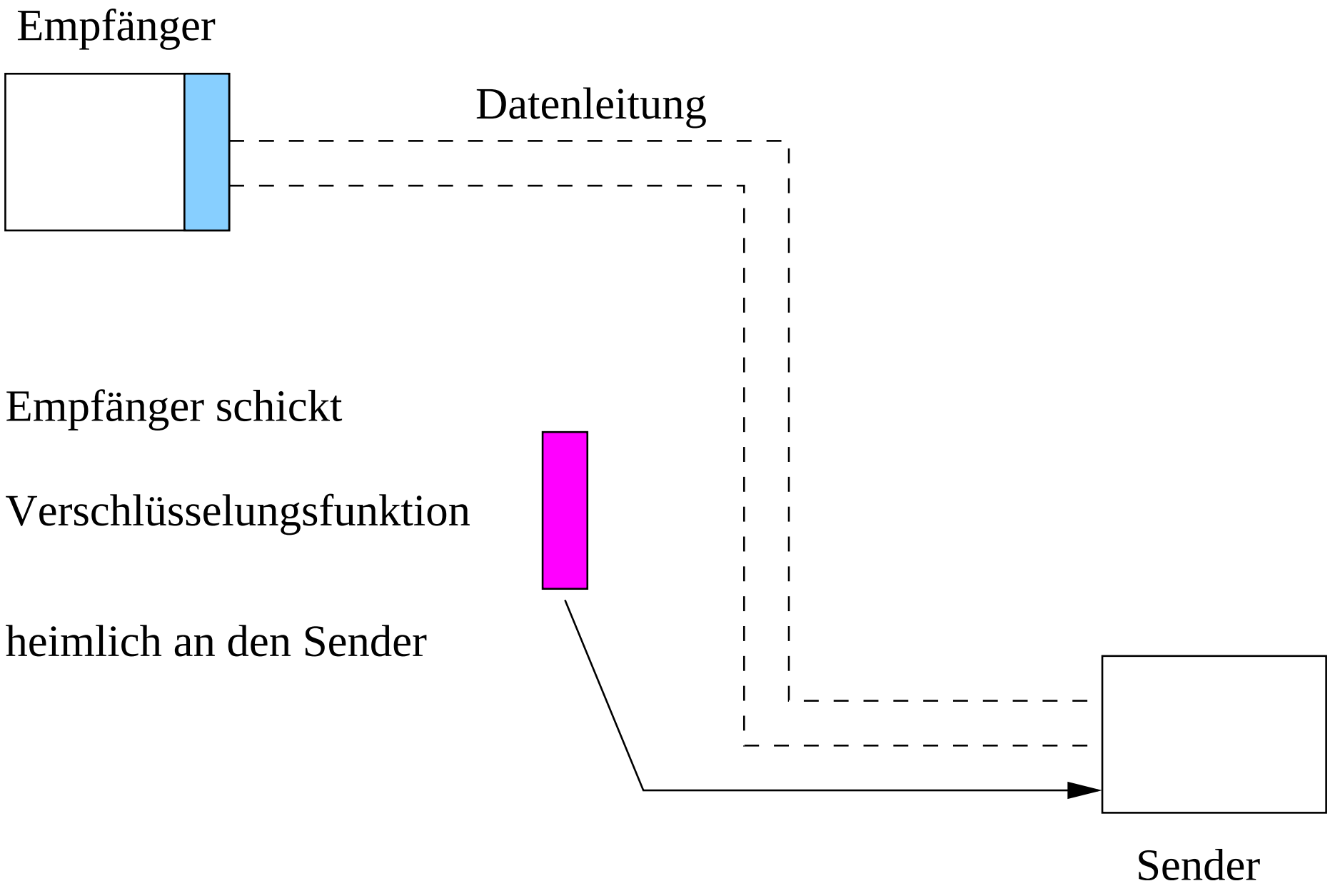
Vereinbarung des Schlüssels



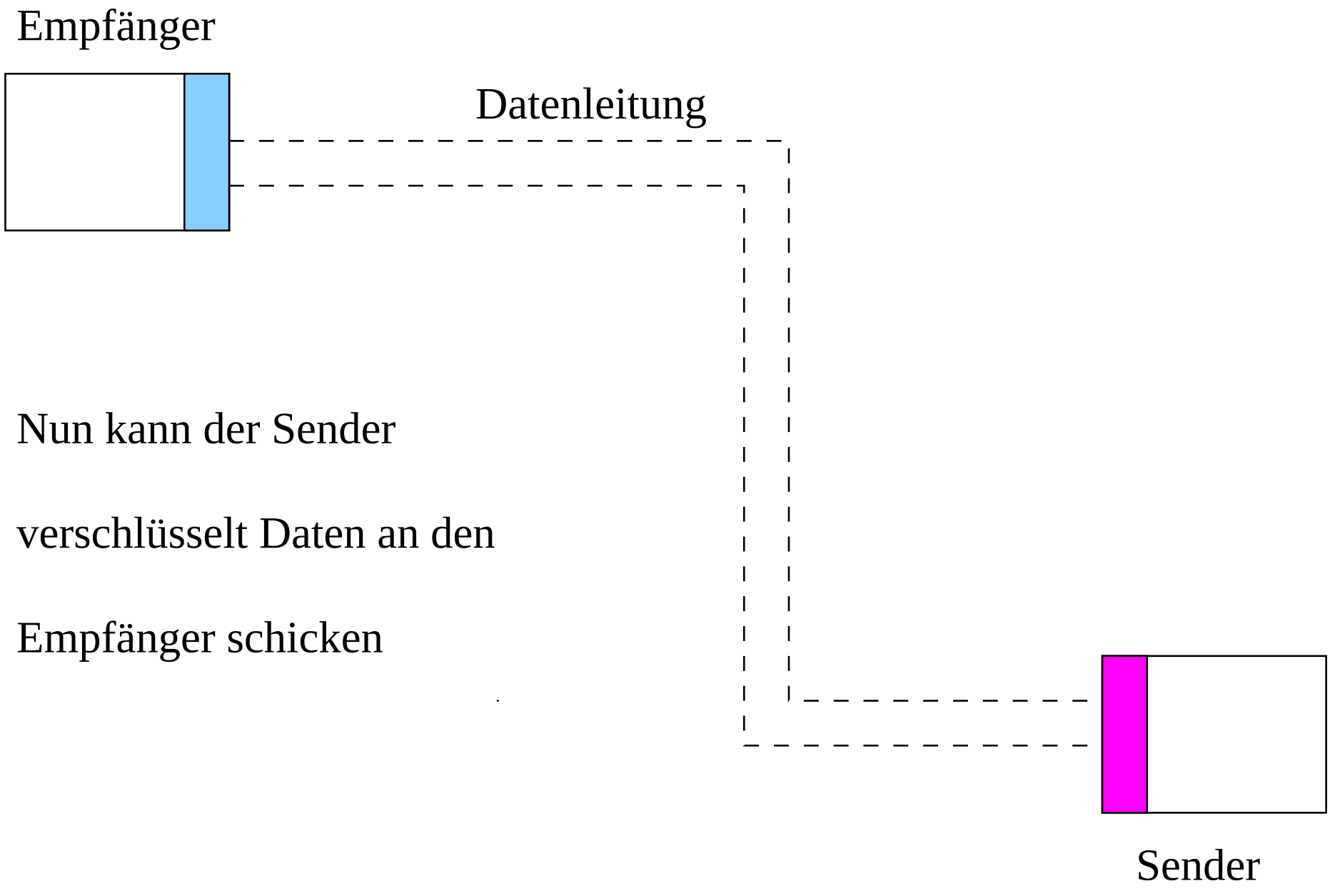
Vereinbarung des Schlüssels



Vereinbarung des Schlüssels



Vereinbarung des Schlüssels



Geht’s auch sicherer?

Jeder, der die Verschlüsselungsfunktion kennt,

kann auch die Entschlüsselungsfunktion ausrechnen.

Geht's auch sicherer?

Seien p und q unterschiedliche Primzahlen.

Dann gilt für jedes $x \in \mathbb{Z}_{p \cdot q}$:

$$x^{(p-1)\cdot(q-1)+1} \equiv x \mod p \cdot q \ .$$

[Satz von Fermat und Euler (1707–1783)]

Geht's auch sicherer?

Seien p und q unterschiedliche Primzahlen.

Dann gilt für jedes $x \in \mathbb{Z}_{p \cdot q}$:

$$x^{(p-1)\cdot(q-1)+1} \equiv x \mod p \cdot q \ .$$

[Satz von Fermat und Euler (1707–1783)]

Verschlüsselungsfunktion $v(x) = x^a \mod p \cdot q$

für zu $(p-1)\cdot(q-1)$ teilerfremdes a.

Entschlüsselungsfunktion $e(y) = y^b \mod p \cdot q$

für mult. Inverses b zu a ($\mod (p-1)\cdot(q-1)$).

Das RSA-Verfahren

Rivest, Shamir, Adleman (1978)

Das RSA-Verfahren

Rivest, Shamir, Adleman (1978)

1. Empfänger
   - wählt Primzahlen p und q

Das RSA-Verfahren

Rivest, Shamir, Adleman (1978)

1. Empfänger
   - wählt Primzahlen p und q
   - berechnet Produkt $m = p \cdot q$

Das RSA-Verfahren

Rivest, Shamir, Adleman (1978)

1. Empfänger
   - wählt Primzahlen p und q
   - berechnet Produkt $m = p \cdot q$
   - bestimmt zu $(p-1)\cdot(q-1)$ teilerfremdes a

Das RSA-Verfahren

Rivest, Shamir, Adleman (1978)

1. Empfänger
   - wählt Primzahlen p und q
   - berechnet Produkt $m = p \cdot q$
   - bestimmt zu $(p-1)\cdot(q-1)$ teilerfremdes a
   - berechnet mult. Inverses b zu a ($\bmod (p-1)\cdot(q-1)$)

Das RSA-Verfahren

Rivest, Shamir, Adleman (1978)

1. Empfänger
   - wählt Primzahlen p und q
   - berechnet Produkt $m = p \cdot q$
   - bestimmt zu $(p-1)\cdot(q-1)$ teilerfremdes a
   - berechnet mult. Inverses b zu a ($\bmod (p-1)\cdot(q-1)$)

   schickt a und m an Sender

Das RSA-Verfahren

Rivest, Shamir, Adleman (1978)

1. Empfänger
   - wählt Primzahlen p und q
   - berechnet Produkt $m = p \cdot q$
   - bestimmt zu $(p-1)\cdot(q-1)$ teilerfremdes a
   - berechnet mult. Inverses b zu a ($\bmod (p-1)\cdot(q-1)$)

   schickt a und m an Sender

2. Sender schickt Nachrichten verschlüsselt mit

   $v(x) = x^a \bmod m$

Das RSA-Verfahren

Rivest, Shamir, Adleman (1978)

1. Empfänger
   - wählt Primzahlen p und q
   - berechnet Produkt $m = p \cdot q$
   - bestimmt zu $(p-1)\cdot(q-1)$ teilerfremdes a
   - berechnet mult. Inverses b zu a ($\bmod (p-1)\cdot(q-1)$)

   schickt a und m an Sender
2. Sender schickt Nachrichten verschlüsselt mit

   $v(x) = x^a \bmod m$
3. Empfänger entschlüsselt Nachrichten mit

   $e(y) = y^b \bmod m$

Das RSA-Verfahren: ein Beispiel

Alice will Bob eine Nachricht als Folge von Zahlen aus $\{0, 1, 2 \ldots, 127\}$ schicken.

1. Bob
   - wählt Primzahlen p und q
   - berechnet Produkt $m = p \cdot q$
   - bestimmt zu $(p-1) \cdot (q-1)$ teilerfremdes a
   - berechnet mult. Inverses b zu a ($\bmod (p-1) \cdot (q-1)$)
2. Alice verschlüsselt mit $v(x) = x^a \bmod m$
3. Bob entschlüsselt Nachrichten mit $e(y) = y^b \bmod m$

Das RSA-Verfahren: ein Beispiel

Alice will Bob eine Nachricht als Folge von Zahlen aus $\{0, 1, 2 \ldots, 127\}$ schicken.

1. Bob
   - wählt Primzahlen p und q : $p = 37$ und $q = 5$
   - berechnet Produkt $m = p \cdot q$
   - bestimmt zu $(p-1) \cdot (q-1)$ teilerfremdes a
   - berechnet mult. Inverses b zu a ($\bmod (p-1) \cdot (q-1)$)
2. Alice verschlüsselt mit $v(x) = x^a \bmod m$
3. Bob entschlüsselt Nachrichten mit $e(y) = y^b \bmod m$

Das RSA-Verfahren: ein Beispiel

Alice will Bob eine Nachricht als Folge von Zahlen aus $\{0, 1, 2 \ldots, 127\}$ schicken.

1. Bob
   - wählt Primzahlen p und q : $p = 37$ und $q = 5$
   - berechnet Produkt $m = p \cdot q$ $= 37 \cdot 5 = 185$
   - bestimmt zu $(p-1) \cdot (q-1)$ teilerfremdes a
   - berechnet mult. Inverses b zu a ($\bmod (p-1) \cdot (q-1)$)
2. Alice verschlüsselt mit $v(x) = x^a \bmod m$
3. Bob entschlüsselt Nachrichten mit $e(y) = y^b \bmod m$

Das RSA-Verfahren: ein Beispiel

Alice will Bob eine Nachricht als Folge von Zahlen aus $\{0, 1, 2 \ldots, 127\}$ schicken.

1. Bob
   - wählt Primzahlen p und q : $p = 37$ und $q = 5$
   - berechnet Produkt $m = p \cdot q$ $= 37 \cdot 5 = 185$
   - bestimmt zu $(p-1) \cdot (q-1)$ teilerfremdes a

     65 ist teilerfremd zu 144
   - berechnet mult. Inverses b zu a ($\bmod (p-1) \cdot (q-1)$)

2. Alice verschlüsselt mit $v(x) = x^a \mod m$
3. Bob entschlüsselt Nachrichten mit $e(y) = y^b \mod m$

Das RSA-Verfahren: ein Beispiel

Alice will Bob eine Nachricht als Folge von Zahlen aus $\{0, 1, 2 \ldots, 127\}$ schicken.

1. Bob
   - wählt Primzahlen p und q : $p = 37$ und $q = 5$
   - berechnet Produkt $m = p \cdot q$ $= 37 \cdot 5 = 185$
   - bestimmt zu $(p-1) \cdot (q-1)$ teilerfremdes a

     65 ist teilerfremd zu 144
   - berechnet mult. Inverses b zu a ($\bmod (p-1) \cdot (q-1)$)

     $b = 113$ ist mult.Inverses zu 65 ($\bmod$ 144)
2. Alice verschlüsselt mit $v(x) = x^a \mod m$
3. Bob entschlüsselt Nachrichten mit $e(y) = y^b \mod m$

Das RSA-Verfahren: ein Beispiel

Alice will Bob eine Nachricht als Folge von Zahlen aus $\{0, 1, 2 \ldots, 127\}$ schicken.

1. Bob
   - wählt Primzahlen p und q : $p = 37$ und $q = 5$
   - berechnet Produkt $m = p \cdot q$ $= 37 \cdot 5 = 185$
   - bestimmt zu $(p-1) \cdot (q-1)$ teilerfremdes a

     65 ist teilerfremd zu 144
   - berechnet mult. Inverses b zu a ($\bmod (p-1) \cdot (q-1)$)

     $b = 113$ ist mult.Inverses zu 65 ($\bmod$ 144)

   schickt $a = 65$ und $m = 185$ an Sender

   merkt sich $b = 113$ und $m = 185$
2. Alice verschlüsselt mit $v(x) = x^a \bmod m$

Das RSA-Verfahren: ein Beispiel

Alice will Bob eine Nachricht als Folge von Zahlen aus $\{0, 1, 2 \ldots, 127\}$ schicken.

1. Bob

   schickt $a = 65$ und $m = 185$ an Sender

   merkt sich $b = 113$ und $m = 185$

2. Alice verschlüsselt mit $v(x) = x^a \mod m$

   $v(x) = x^{65} \mod 185$

3. Bob entschlüsselt Nachrichten mit $e(y) = y^b \mod m$

Das RSA-Verfahren: ein Beispiel

Alice will Bob eine Nachricht als Folge von Zahlen aus $\{0, 1, 2 \ldots, 127\}$ schicken.

1. Bob

   schickt $a = 65$ und $m = 185$ an Sender

   merkt sich $b = 113$ und $m = 185$

2. Alice verschlüsselt mit $v(x) = x^a \mod m$

   $v(x) = x^{65} \mod 185$

3. Bob entschlüsselt Nachrichten mit $e(y) = y^b \mod m$

   $v(x) = x^{113} \mod 185$

Schnelles modulares Exponenzieren

Ersetze viele Multiplikationen durch wenige Quadrierungen.

$$3^8 = 3^{2\cdot 4} = \left(3^2\right)^4 = \left(\left(3^2\right)^2\right)^2 .$$

Statt 8 Multiplikationen nur 3 Quadrierungen.
Modular kann sofort reduziert werden.

$$\begin{aligned} 3^8 \bmod 11 &= \left(\left(3^2\right)^2\right)^2 \bmod 11 \\ &= \left(9^2\right)^2 \bmod 11 \\ &= 4^2 \bmod 11 \\ &= 5 \end{aligned}$$

Schnelles modulares Exponenzieren

Die zu multiplizierenden Quadrate ergeben sich aus der Binärdarstellung des Exponenten.
Darstellung von 10 als Binärzahl: $b_3 b_2 b_1 b_0 = 1010$.
Damit gilt

$$n^{10} = \underbrace{\left(\left(n^2\right)^2\right)^2}_{b_3=1} \cdot \underbrace{n^2}_{b_1=1} .$$

Allgemein: sei $b_m b_{m-1} \cdots b_0$ die Binärdarstellung von a. Dann gilt

$$n^a = \prod_{i=0}^{m} b_i \cdot n^{(2^i)}$$

Berechnung von $v(27) = 27^{15} \bmod 47$

Die Binärdarstellung von 15 ist $b_3 b_2 b_1 b_0 = 1111$. Daraus ergibt sich

$$n^{15} = \underbrace{\left(\left(n^2\right)^2\right)^2}_{b_3=1} \cdot \underbrace{\left(n^2\right)^2}_{b_2=1} \cdot \underbrace{n^2}_{b_1=1} \cdot \underbrace{n}_{b_0=1} .$$

Berechnung von $v(27) = 27^{15} \bmod 47$

Die Binärdarstellung von 15 ist $b_3 b_2 b_1 b_0 = 1111$. Daraus ergibt sich

$$n^{15} = \underbrace{\left(\left(n^2\right)^2\right)^2}_{b_3=1} \cdot \underbrace{\left(n^2\right)^2}_{b_2=1} \cdot \underbrace{n^2}_{b_1=1} \cdot \underbrace{n}_{b_0=1} .$$

$$27^{15} \bmod 47 = \left(\left(27^2\right)^2\right)^2 \cdot \left(27^2\right)^2 \cdot 27^2 \cdot 27 \bmod 47$$

Berechnung von $v(27) = 27^{15} \bmod 47$

$$
\begin{aligned}
27^{15} \bmod 47 &= \left(\left(27^2\right)^2\right)^2 \cdot \left(27^2\right)^2 \cdot 27^2 \cdot 27 \mod 47 \\
&= \left(24^2\right)^2 \cdot 24^2 \cdot 24 \cdot 27 \mod 47 \\
&= 12^2 \cdot 12 \cdot 24 \cdot 27 \mod 47 \\
&= 3 \cdot 12 \cdot 24 \cdot 27 \mod 47 \\
&= 3 \cdot 12 \cdot 37 \mod 47 \\
&= 3 \cdot 21 \mod 47 \\
&= 16
\end{aligned}
$$

Also ist $v(27) = 16$.

Berechnung von $e(16) = 16^{43} \bmod 47$

Die Binärdarstellung von 43 ist

$$b_5 b_4 b_3 b_2 b_1 b_0 = 101011 \, .$$

Daraus ergibt sich

$$n^{43} = \underbrace{\left(\left(\left(\left(n^2\right)^2\right)^2\right)^2\right)}_{b_5=1} \cdot \underbrace{\left(\left(\left(n^2\right)^2\right)^2\right)}_{b_3=1} \cdot \underbrace{n^2}_{b_1=1} \cdot \underbrace{n}_{b_0=1} \, .$$

Berechnung von $e(16) = 16^{43} \bmod 47$

Die Binärdarstellung von 43 ist

$$b_5 b_4 b_3 b_2 b_1 b_0 = 101011 \, .$$

Daraus ergibt sich

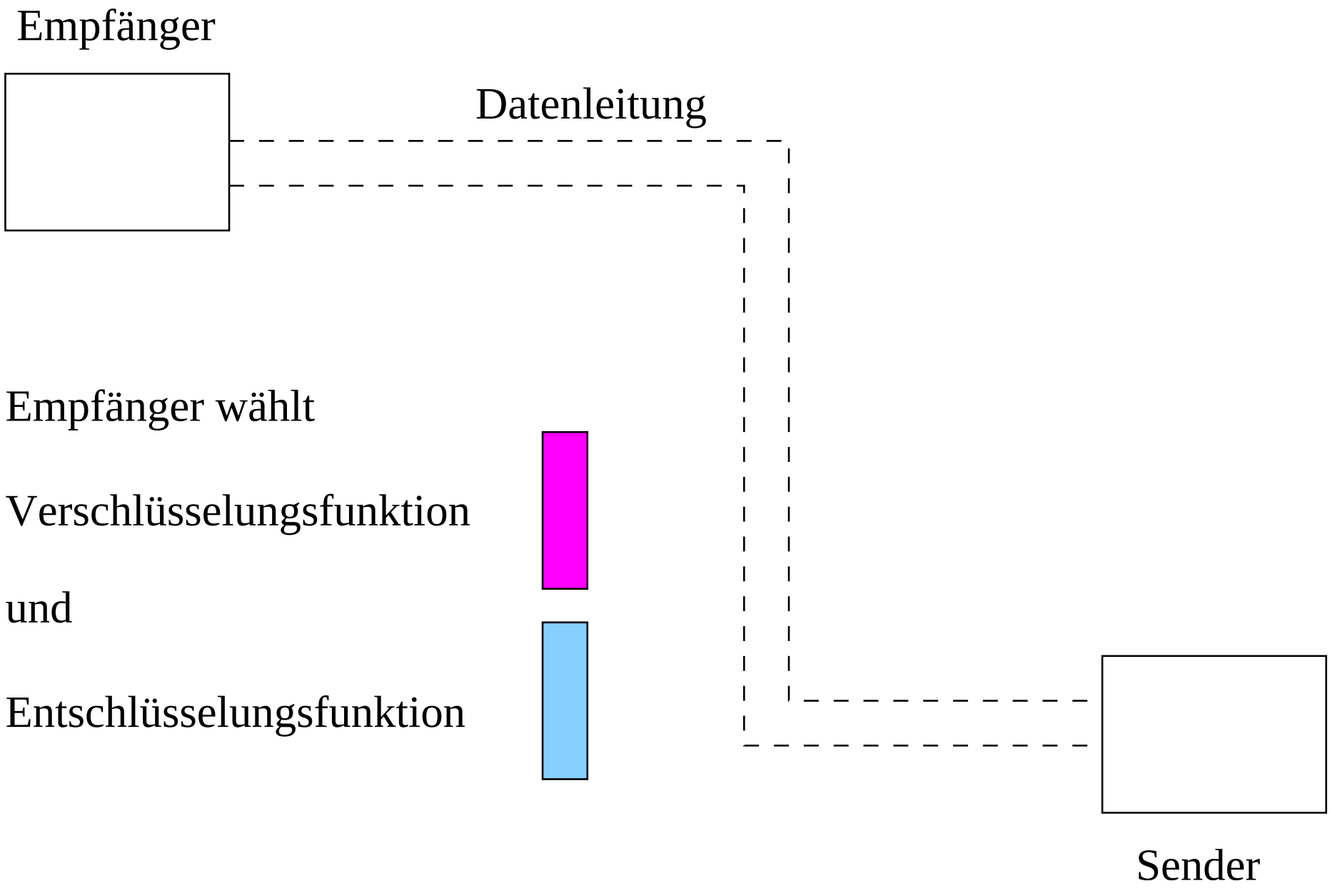
$$n^{43} = \underbrace{\left(\left(\left(\left(n^2\right)^2\right)^2\right)^2\right)}_{b_5=1} \cdot \underbrace{\left(\left(\left(n^2\right)^2\right)^2\right)}_{b_3=1} \cdot \underbrace{n^2}_{b_1=1} \cdot \underbrace{n}_{b_0=1} \, .$$

$$16^{43} \bmod 47 = \left(\left(\left(\left(16^2\right)^2\right)^2\right)^2\right) \cdot \left(\left(\left(16^2\right)^2\right)^2\right) \cdot 16^2 \cdot 16 \mod 4$$

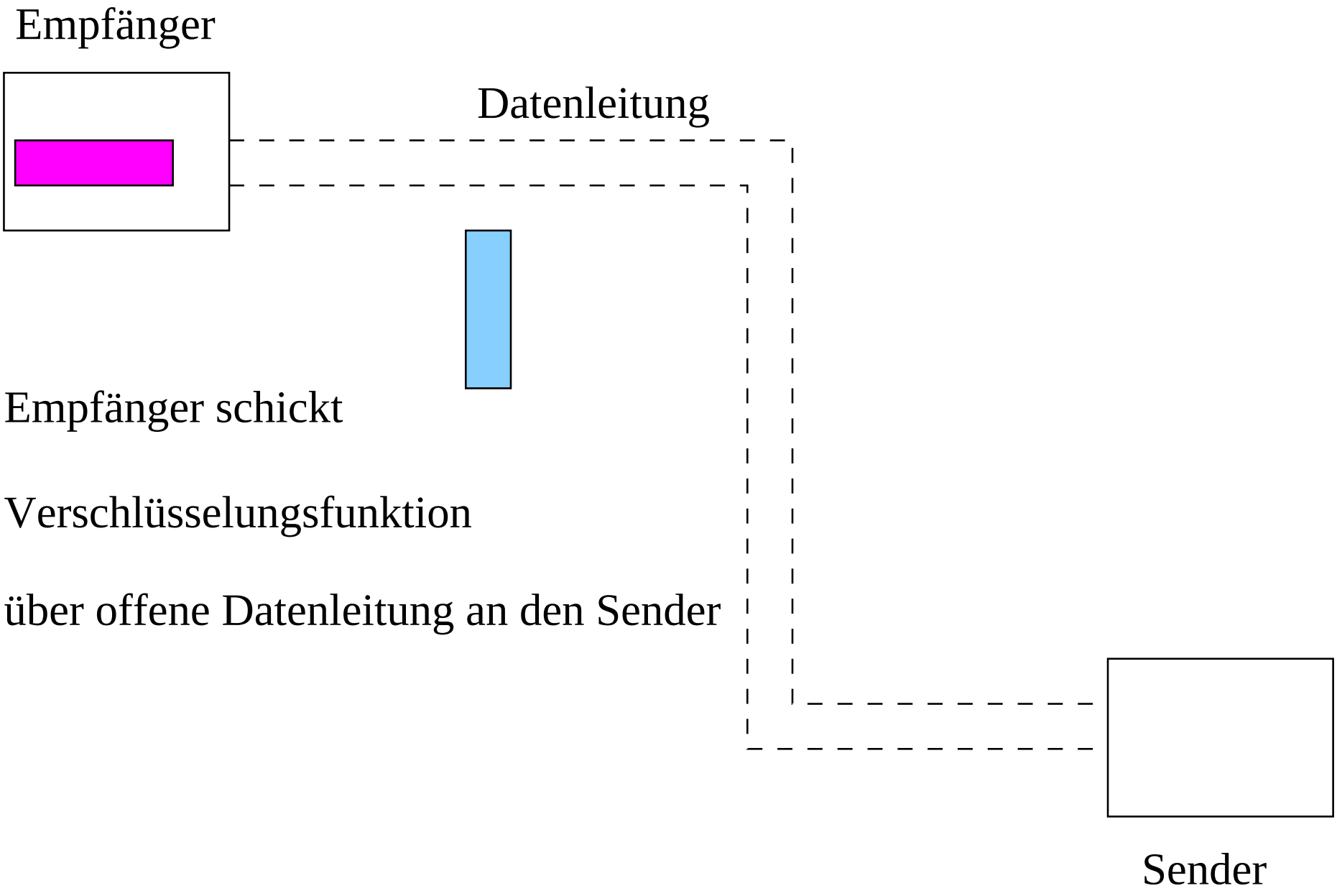
Berechnung von $e(16) = 16^{43} \bmod 47$

$$
\begin{aligned}
16^{43} \bmod 47 &= \left(\left(\left(\left(16^2\right)^2\right)^2\right)^2\right)^2 \cdot \left(\left(\left(16^2\right)^2\right)^2\right)^2 \cdot 16^2 \cdot 16 \mod 4 \\
&= \left(\left(\left(21^2\right)^2\right)^2\right)^2 \cdot \left(21^2\right)^2 \cdot 21 \cdot 16 \mod 47 \\
&= \left(\left(18^2\right)^2\right)^2 \cdot 18^2 \cdot 21 \cdot 16 \mod 47 \\
&= \left(42^2\right)^2 \cdot 42 \cdot 21 \cdot 16 \mod 47 \\
&= 25^2 \cdot 42 \cdot 21 \cdot 16 \mod 47 \\
&= \ldots \\
&= 27
\end{aligned}
$$

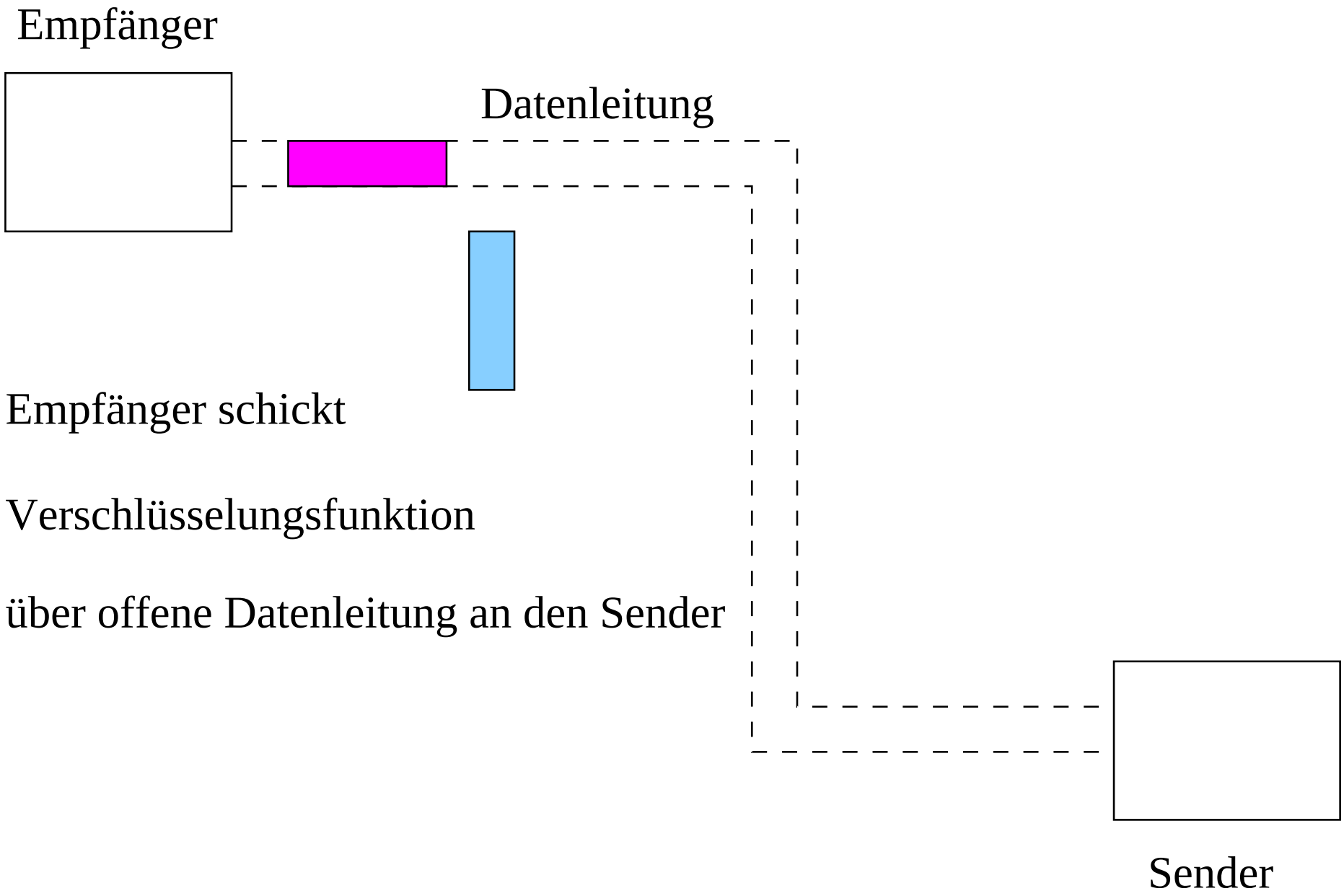
Öffentlicher Schlüsselaustausch



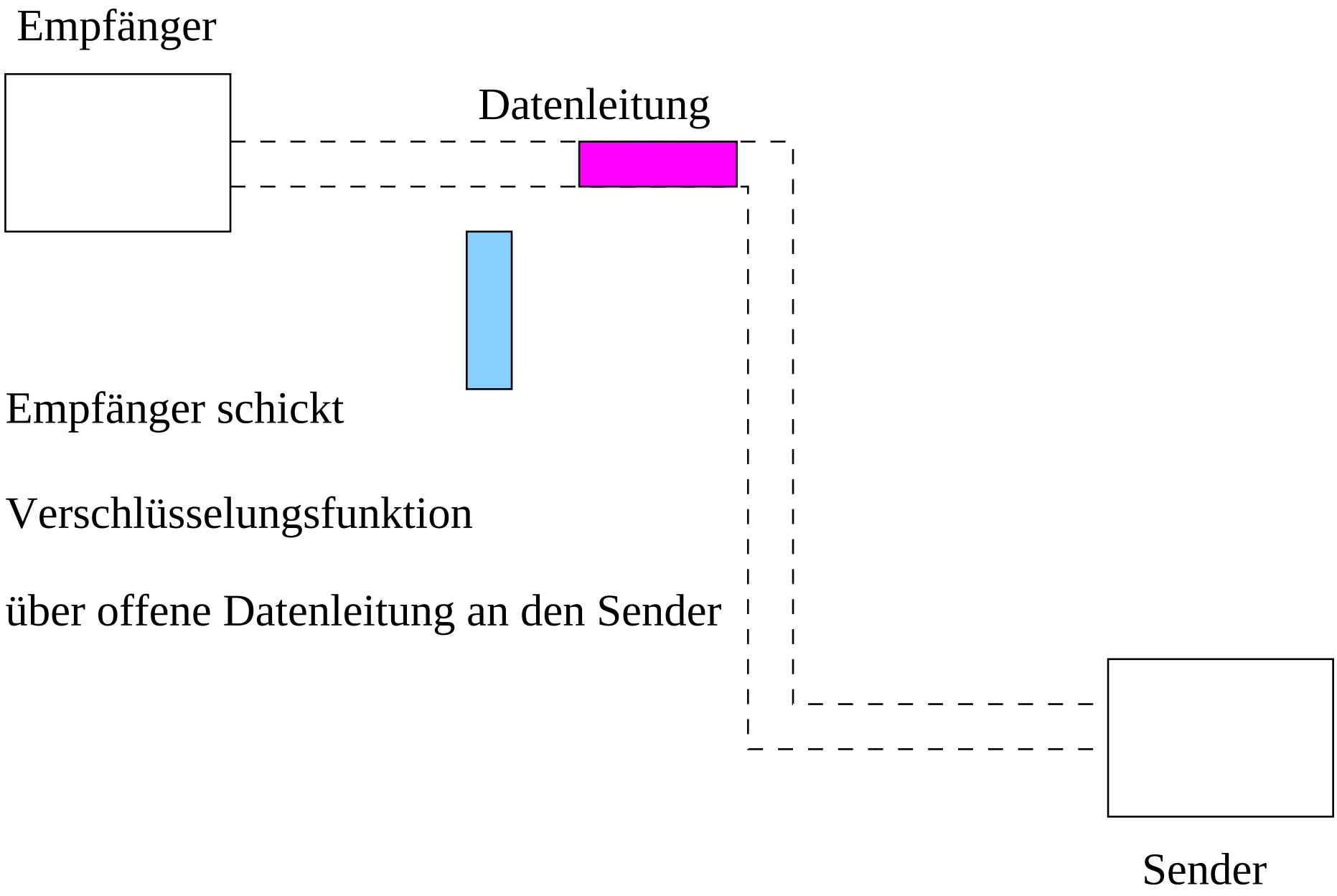
Öffentlicher Schlüsselaustausch



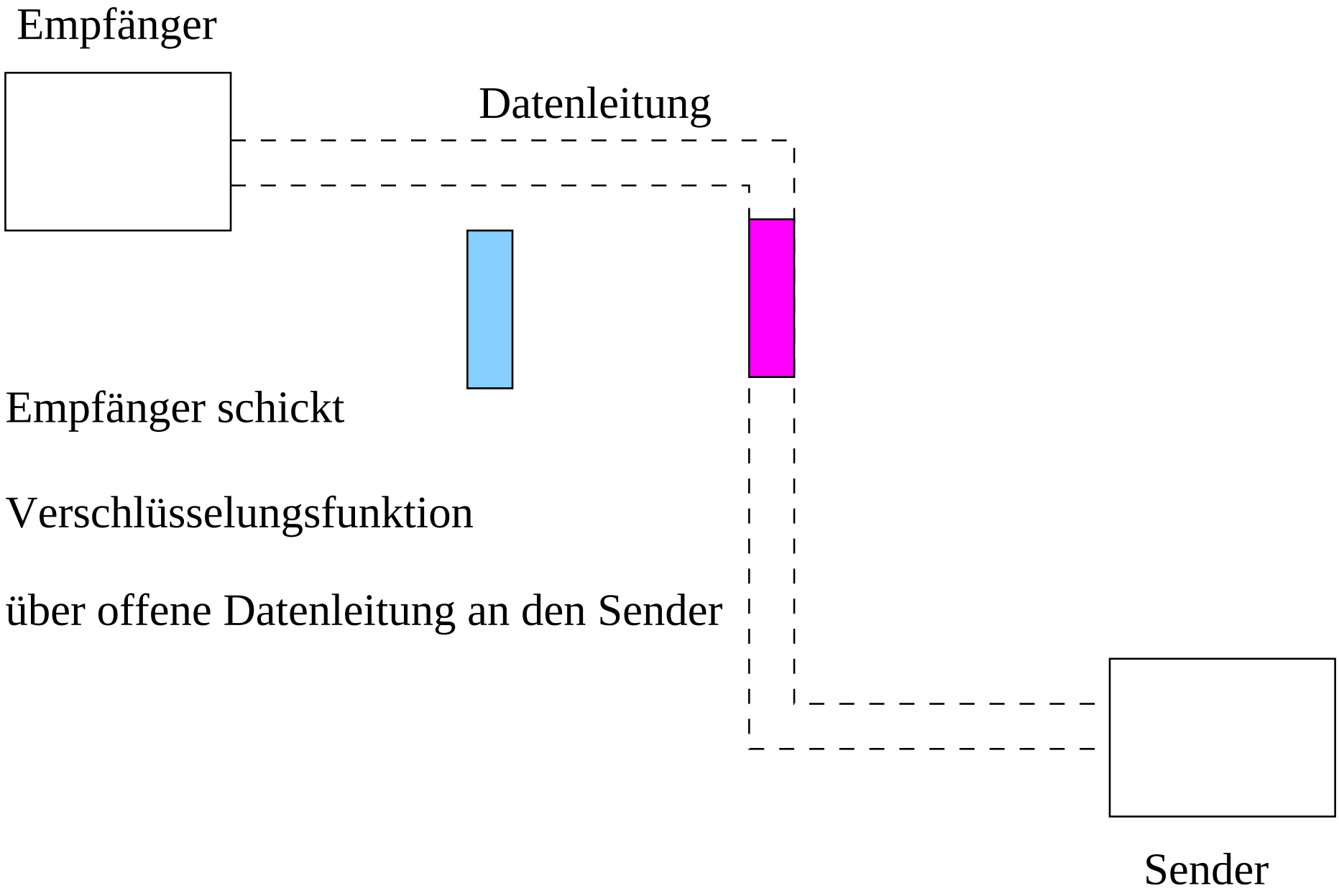
Öffentlicher Schlüsselaustausch



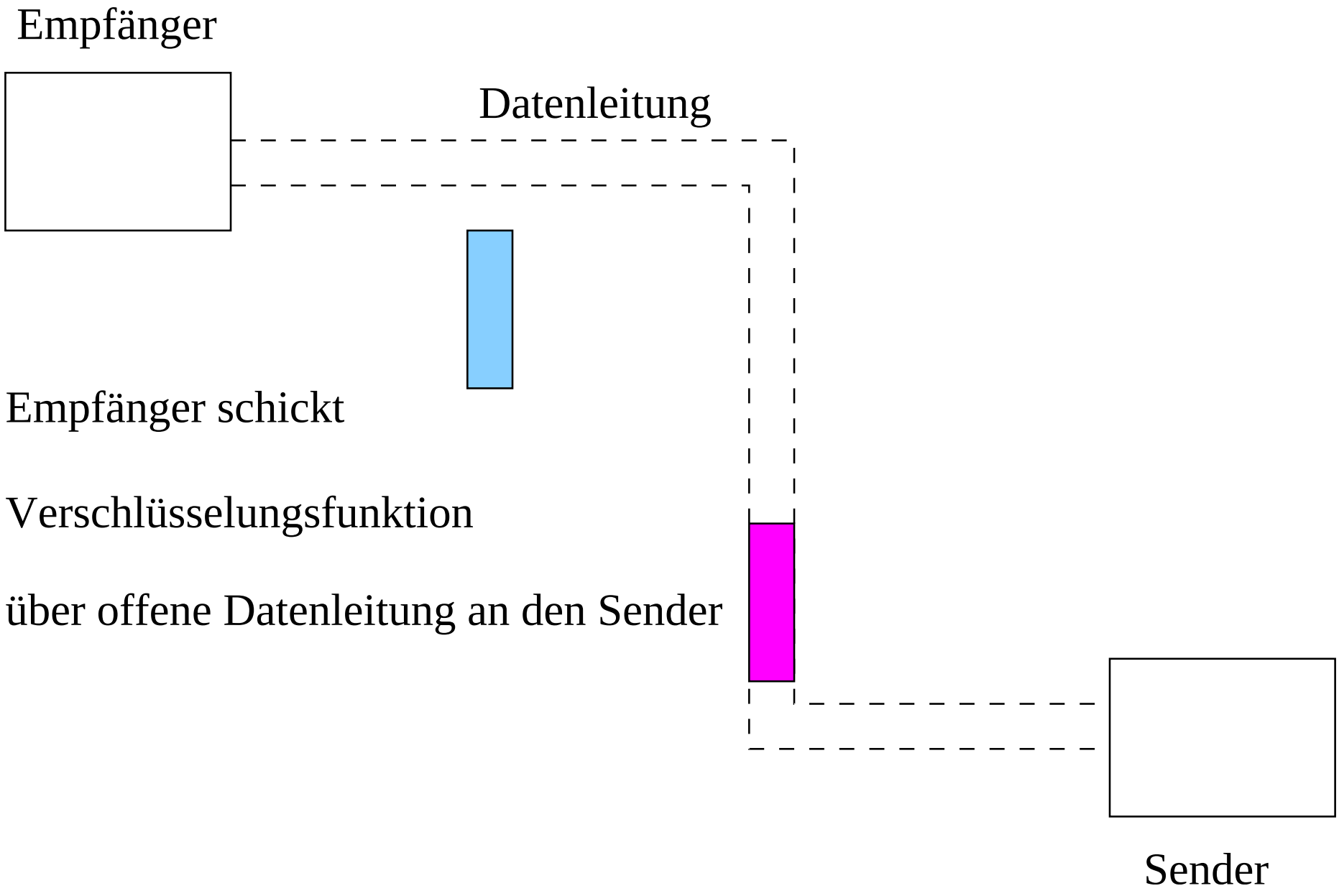
Öffentlicher Schlüsselaustausch



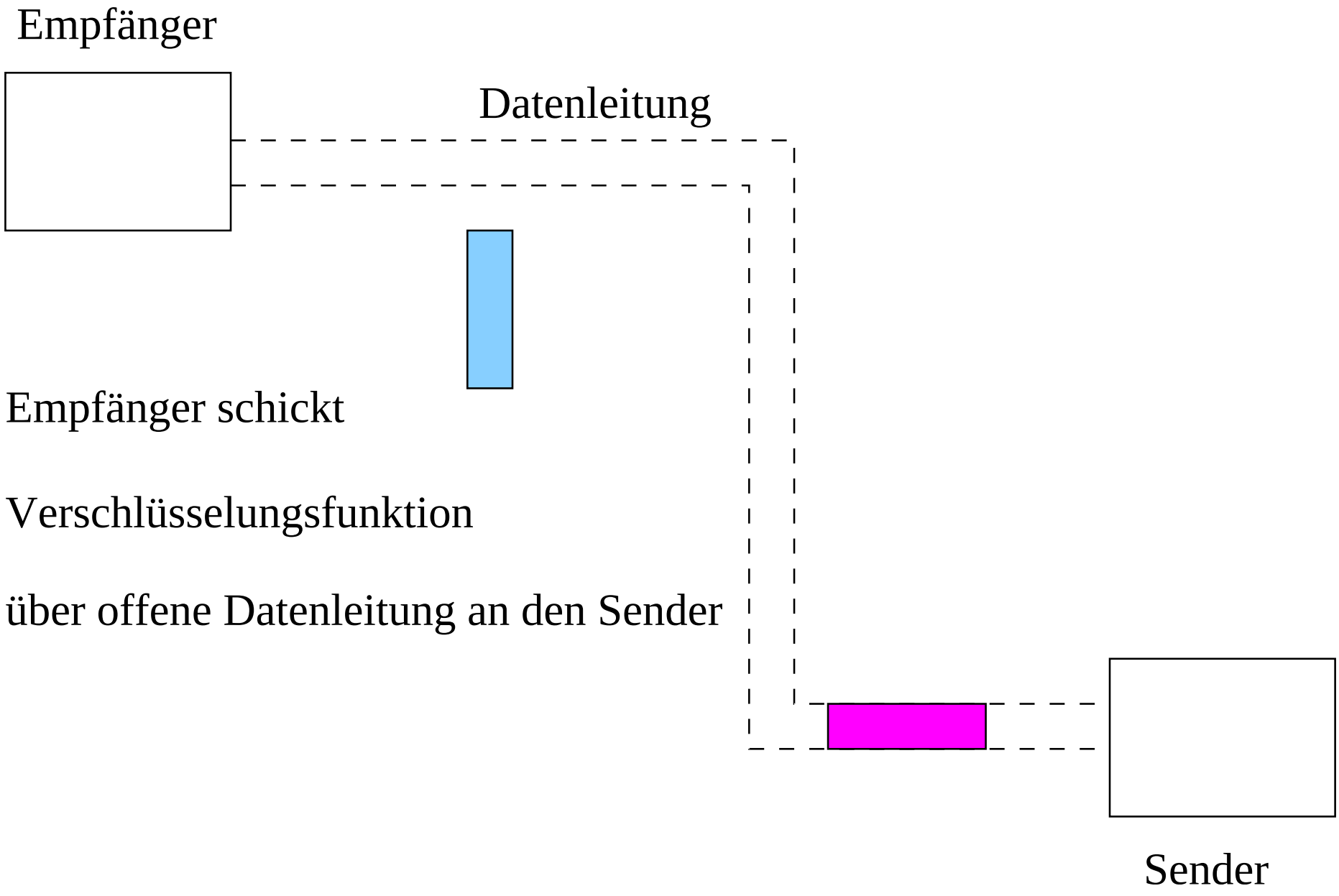
Öffentlicher Schlüsselaustausch



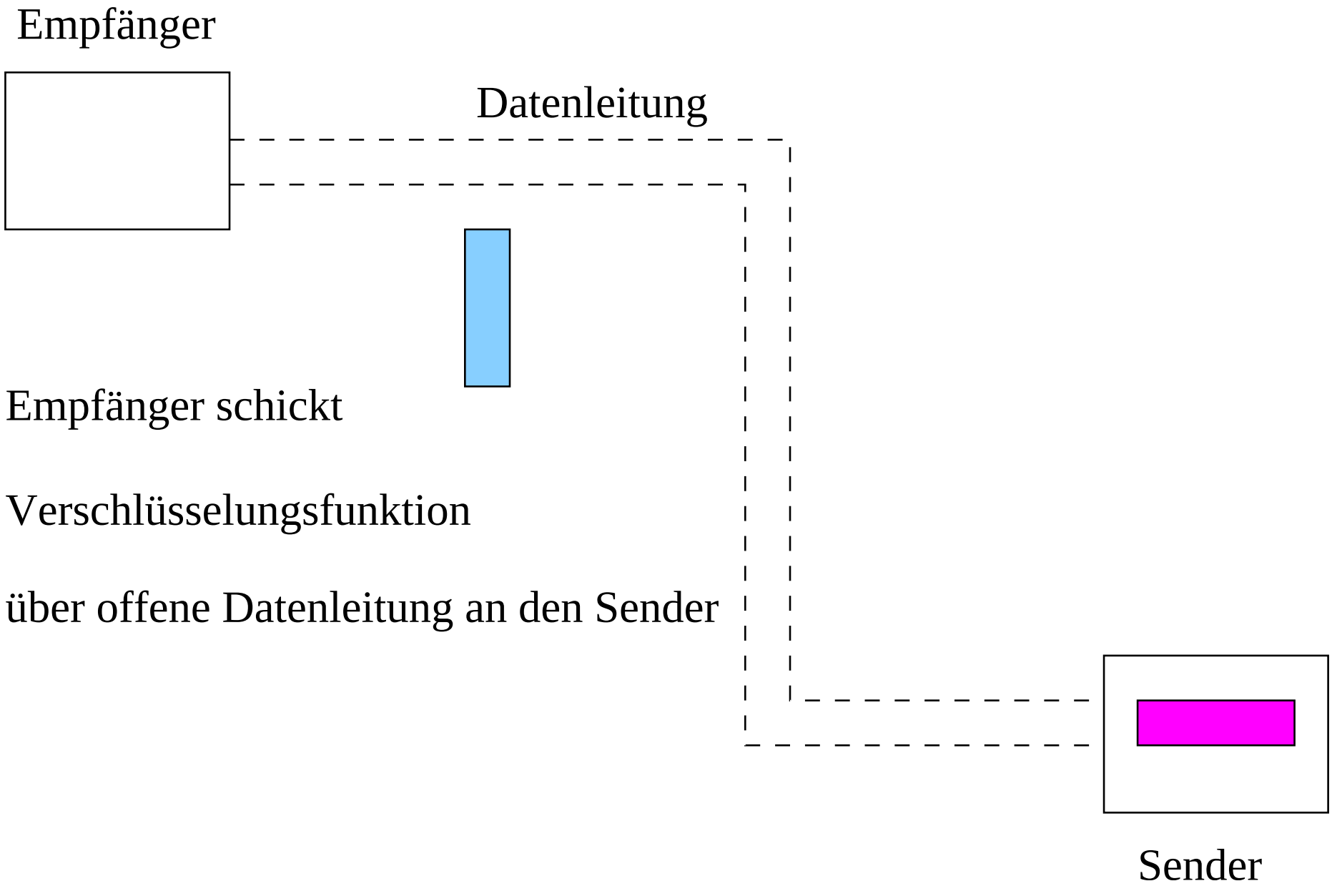
Öffentlicher Schlüsselaustausch



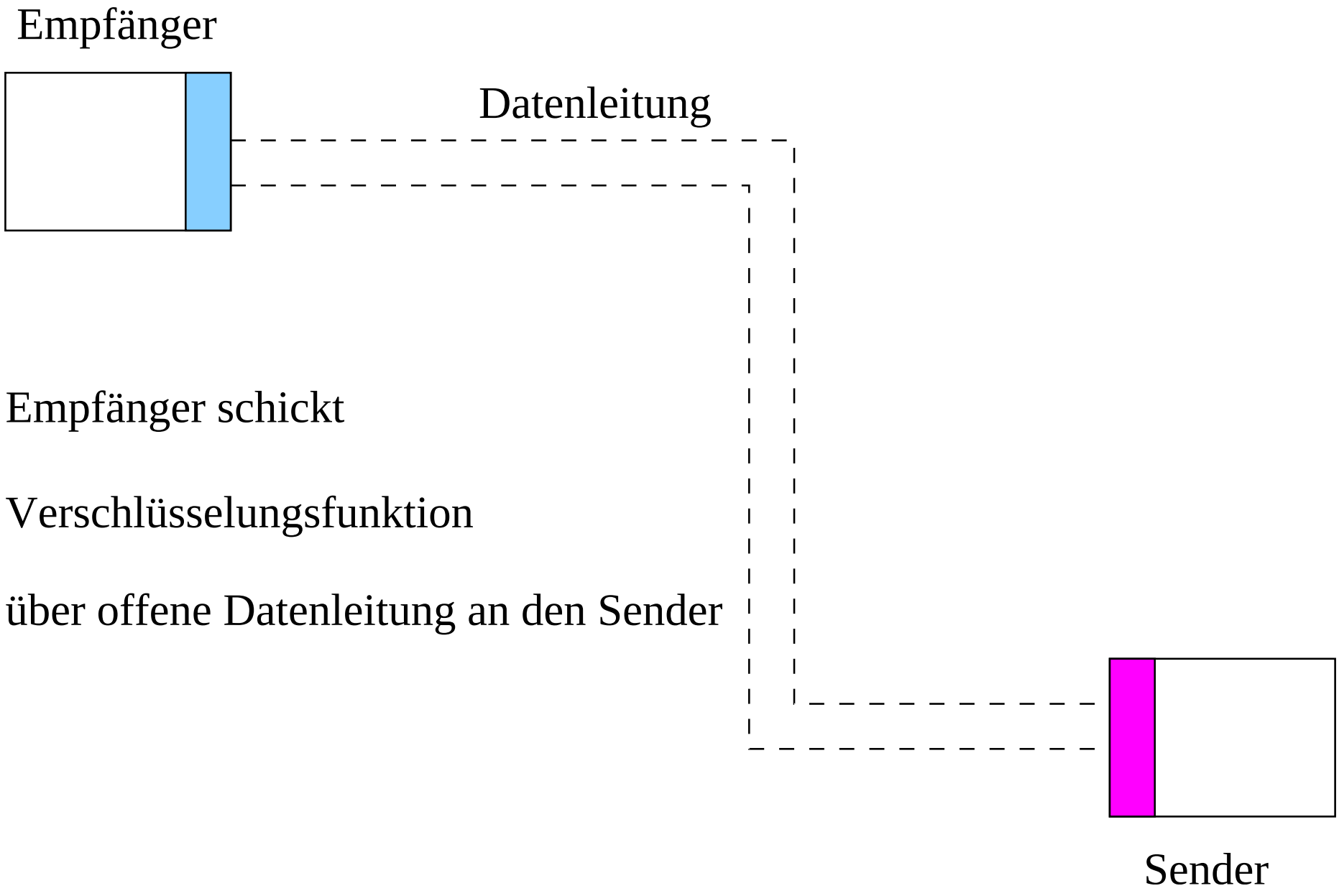
Öffentlicher Schlüsselaustausch



Öffentlicher Schlüsselaustausch



Öffentlicher Schlüsselaustausch



Warum scheint das RSA-Verfahren sicher?

Diese 99stellige Zahl ist das Produkt zweier Primzahlen:

354689941437726254971626578592511763023548841046378845501625087126557585352413960447471067588977753

Welche sind das?

Warum scheint das RSA-Verfahren sicher?

Diese 99stellige Zahl ist das Produkt zweier Primzahlen:

$$354689941437726254971626578592511763023548841046378845501625087126557585352413960447471067588977753$$

$$=$$

$$22953686867719691230002707821868552601124472329079$$

$$\times$$

$$15452417011775787851951047309563159388840946309807$$

Warum scheint das RSA-Verfahren sicher?

Diese 99stellige Zahl ist das Produkt zweier Primzahlen:

$$35468994143772625497162657859251176302354884104637$$
$$88455016250871265575853524139604474710675889777753$$

$$=$$

$$229536868677196912300027078218685526011244723290 79$$

$$\times$$

$$15452417011775787851951047309563159388840946309807$$

Es geht schnell, zwei Zahlen zu multiplizieren.

Warum scheint das RSA-Verfahren sicher?

Diese 99stellige Zahl ist das Produkt zweier Primzahlen:

$$35468994143772625497162657859251176302354884104637$$
$$88455016250871265575853524139604474710675889777753$$

$$=$$

$$229536868677196912300027078218685526011244723290 79$$

$$\times$$

$$15452417011775787851951047309563159388840946309807$$

Es geht schnell, zwei Zahlen zu multiplizieren.

Es geht nicht schnell, eine Zahl in ihre Faktoren zu zerlegen.

Was heißt schnell?

Die Laufzeit eines Algorithmus wird abhängig von der Länge n der Eingabe gemessen.

Laufzeit-Funktion	Länge n der Eingabe 10	50	100	1000
n	0.00001 Sek	0.00005 Sek	0.0001 Sek	0.001 Se
n^2	0.0001 Sek	0.0025 Sek	0.01 Sek	1 Sek
n^3	0.001 Sek	0.125 Sek	1 Sek	16.6 Mi
$2^{n/\log n}$	0.000009 Sek	0.0001 Sek	0.001 Sek	$4 \cdot 10^6$ Jah
2^n	0.001 Sek	35.7 Jahre	$4 \cdot 10^6$ J.	$3 \cdot 10^{287}$

auf einem Computer, der 1.000.000 Befehle pro Sekunde ausführt

Was heißt schnell?

Die Laufzeit eines Algorithmus wird abhängig von der Länge n der Eingabe gemessen.

Laufzeit-Funktion	Länge n der Eingabe 10	50	100	1000
n	0.00001 Sek	0.00005 Sek	0.0001 Sek	0.001 Se
n^2	0.0001 Sek	0.0025 Sek	0.01 Sek	1 Sek
n^3	0.001 Sek	0.125 Sek	1 Sek	16.6 Mi
$2^{n/\log n}$	0.000009 Sek	0.0001 Sek	0.001 Sek	$4 \cdot 10^6$ Jah
2^n	0.001 Sek	35.7 Jahre	$4 \cdot 10^6$ J.	$3 \cdot 10^{287}$

Polynomielle Laufzeit n, n^2, n^3 . . . ist schnell.

Was heißt schnell?

Die Laufzeit eines Algorithmus wird abhängig von der Länge n der Eingabe gemessen.

Laufzeit-Funktion	Länge n der Eingabe 10	50	100	1000
n	0.00001 Sek	0.00005 Sek	0.0001 Sek	0.001 Se
n^2	0.0001 Sek	0.0025 Sek	0.01 Sek	1 Sek
n^3	0.001 Sek	0.125 Sek	1 Sek	16.6 Mi
$2^{n/\log n}$	0.000009 Sek	0.0001 Sek	0.001 Sek	$4 \cdot 10^6$ Jah
2^n	0.001 Sek	35.7 Jahre	$4 \cdot 10^6$ J.	$3 \cdot 10^{287}$

Polynomielle Laufzeit n, n^2, n^3 . . . ist schnell.

Exponentielle Laufzeit 2^n . . . ist langsam.

Schnell oder langsam?

Finden einer Primzahl mit n Stellen ??

schnell

langsam

Schnell oder langsam?

schnell

langsam

Finden einer Primzahl

Schnell oder langsam?

Multiplizieren von Zahlen ??

schnell

langsam

Finden einer Primzahl

Schnell oder langsam?

schnell

langsam

Finden einer Primzahl

Multiplizieren von Zahlen

Schnell oder langsam?

Finden einer teilerfremden Zahl ??

schnell

Finden einer Primzahl

Multiplizieren von Zahlen

langsam

Schnell oder langsam?

schnell

Finden einer Primzahl

Multiplizieren von Zahlen

teilerfremde Zahl

langsam

Schnell oder langsam?

Berechnung des multiplikativ Inversen ??

schnell

Finden einer Primzahl

Multiplizieren von Zahlen

teilerfremde Zahl

langsam

Schnell oder langsam?

schnell

Finden einer Primzahl

Multiplizieren von Zahlen

teilerfremde Zahl

multiplikativ Inverses

langsam

Schnell oder langsam?

Potenzieren modulo n ??

schnell

langsam

Finden einer Primzahl

Multiplizieren von Zahlen

teilerfremde Zahl

multiplikativ Inverses

Schnell oder langsam?

schnell

Finden einer Primzahl

Multiplizieren von Zahlen

teilerfremde Zahl

multiplikativ Inverses

modulares Potenzieren

langsam

Schnell oder langsam?

Zerlegung einer Zahl in ihre Faktoren ??

schnell

Finden einer Primzahl

Multiplizieren von Zahlen

teilerfremde Zahl

multiplikativ Inverses

modulares Potenzieren

langsam

Schnell oder langsam?

schnell

Finden einer Primzahl
Multiplizieren von Zahlen
teilerfremde Zahl
multiplikativ Inverses
modulares Potenzieren

langsam

Faktorisieren

Primzahlen

Die beiden größten bekannten Primzahlen (and friends)

$2^{30402457}-1$	9152052	Dezimalstellen	(2005)
$2^{25964951}-1$	7816230		(2005)
$2^{13466917}-1$	4053946		(2001)
$2^{2976221}-1$	895932		(1997)
$3\cdot 2^{2145353}+1$	645817		(2003)
$62722^{131072}+1$	628808		(2003)

Primzahlen

Die beiden größten bekannten Primzahlen (and friends)

$2^{30402457}-1$	9152052	Dezimalstellen	(2005)
$2^{25964951}-1$	7816230		(2005)
$2^{13466917}-1$	4053946		(2001)
$2^{2976221}-1$	895932		(1997)
$3\cdot 2^{2145353}+1$	645817		(2003)
$62722^{131072}+1$	628808		(2003)

Der Pentium-Bug (470 Millionen US$) wurde 1995 bei der Suche nach Primzahl-Zwillingen gefunden.

Faktorisieren

Gibt es einen schnellen Algorithmus zum Faktorisieren ?

Faktorisieren

Gibt es einen schnellen Algorithmus zum Faktorisieren ?

Kann man das RSA-Verfahren knacken,
ohne schnell faktorisieren zu können ?

Alternativen zu RSA

Sicherheit und Schnelligkeit

Alternativen zu RSA

Sicherheit und Schnelligkeit

- Verfahren von Rabin
  (Sicherheit äquiv. zur Faktorisierung)

Alternativen zu RSA

Sicherheit und Schnelligkeit

- Verfahren von Rabin
  (Sicherheit äquiv. zur Faktorisierung)
- ElGamal
  (unsicher, falls Diskreter Logarithmus einfach)

Alternativen zu RSA

Sicherheit und Schnelligkeit

- Verfahren von Rabin
  (Sicherheit äquiv. zur Faktorisierung)
- ElGamal
  (unsicher, falls Diskreter Logarithmus einfach)
- Feistel-Netzwerke (DES)

Alternativen zu RSA

Sicherheit und Schnelligkeit

- Verfahren von Rabin
  (Sicherheit äquiv. zur Faktorisierung)
- ElGamal
  (unsicher, falls Diskreter Logarithmus einfach)
- Feistel-Netzwerke (DES)
- *Pretty good privacy* (PGP)

Alternativen zu RSA

Sicherheit und Schnelligkeit

- Verfahren von Rabin
  (Sicherheit äquiv. zur Faktorisierung)
- ElGamal
  (unsicher, falls Diskreter Logarithmus einfach)
- Feistel-Netzwerke (DES)
- *Pretty good privacy* (PGP)

  benutzt RSA zum Austausch einer Verschlüsselungsfunktion, die nur kurz verwendet wird

Alternativen

Alternativen

- Verfahren, deren Sicherheit mit schwierigen und gut untersuchten Problemen zusammenhängt

  (NP-vollständige Probleme nicht möglich)

Alternativen

- Verfahren, deren Sicherheit mit schwierigen und gut untersuchten Problemen zusammenhängt

  (NP-vollständige Probleme nicht möglich)
- Elliptische Kurven

  kürzere Schlüssel (163 Bit statt 1024), geeignet für Smart-Cards

Alternativen

- Verfahren, deren Sicherheit mit schwierigen und gut untersuchten Problemen zusammenhängt

  (NP-vollständige Probleme nicht möglich)
- Elliptische Kurven

  kürzere Schlüssel (163 Bit statt 1024), geeignet für Smart-Cards
- No-Key (Shamir)

Resumée

- Verschlüsselung ist wichtig

Resumée

- Verschlüsselung ist wichtig
- perfekt sichere Verschlüsselung ist praktisch unbekannt (Shannon, 1949)

Resumée

- Verschlüsselung ist wichtig
- perfekt sichere Verschlüsselung ist praktisch unbekannt (Shannon, 1949)
- RSA ist eines der derzeit besten Verfahren

Resumée

- Verschlüsselung ist wichtig
- perfekt sichere Verschlüsselung ist praktisch unbekannt (Shannon, 1949)
- RSA ist eines der derzeit besten Verfahren

  (hat aber auch seine Nachteile)

Resumée

- Verschlüsselung ist wichtig
- perfekt sichere Verschlüsselung ist praktisch unbekannt (Shannon, 1949)
- RSA ist eines der derzeit besten Verfahren

  (hat aber auch seine Nachteile)
- Die Suche nach beweisbar sicheren und schnellen Verfahren läuft

Resumée

- Verschlüsselung ist wichtig
- perfekt sichere Verschlüsselung ist praktisch unbekannt (Shannon, 1949)
- RSA ist eines der derzeit besten Verfahren

  (hat aber auch seine Nachteile)
- Die Suche nach beweisbar sicheren und schnellen Verfahren läuft
- (Elektronische Signatur, elektr. Geld, zero-knowledge Authentifizierung . . .)

Mixhgr_$isf oqr(ask*jwTzasdkjkqz?.op

Mixhgr_$isf oqr(ask*jwTzasdkjkqz?.op

Vielen Dank für Ihre Aufmerksamkeit!

Vielen Dank für Ihre Aufmerksamkeit!