

# IT-Risk-Management und Kryptographie

## Theoretische Einführung und praktische Übungen

*Prof. Grimm, Prof. Paulus, Dipl.-inform. Droege, Dipl.-ing. Hundacker*

Seminar für die Debeka

8.12.2006

Universität Koblenz-Landau
http://www.uni-koblenz.de/FB4/



# Seminarplan

**9:00-10:30** **Theorie I:** Einführung, Begriffe, Modelle, Beispiele (Paulus)

**11:00-12:30** **Praxis I:** Schlüsselerzeugung, Schlüsselaustausch, Schlüsseleinsatz (Droege)

**13:30-15:00** **Theorie II:** Berechnungen einzelner Verschlüsselungsalgorithmen und Sicherheitsprotokolle (Grimm)

**15:30-17:00** **Praxis II:** Verschlüsselung in Beispielanwendungen WLAN, Email, Homebanking, Türschlösser usw. (Hundacker)

# Wo wird Verschlüsselung benötigt?

- Daten (z. B. auf Festplatte) vor fremdem Zugriff schützen
  Nur der Besitzer kennt den Schlüssel
- Kommunikation (z. B. per E-Mail) vor fremdem Zugriff schützen
  Die Schlüssel müssen ausgetauscht werden

Frage: Wie kann der Schlüsseltausch sicher durchgeführt werden?

# Wo wird Verschlüsselung benötigt?

- Daten (z. B. auf Festplatte) vor fremdem Zugriff schützen
  Nur der Besitzer kennt den Schlüssel
- Kommunikation (z. B. per E-Mail) vor fremdem Zugriff schützen
  Die Schlüssel müssen ausgetauscht werden

Frage: Wie kann der Schlüsseltausch sicher durchgeführt werden?

# Wo wird Verschlüsselung benötigt?

- Daten (z. B. auf Festplatte) vor fremdem Zugriff schützen
  Nur der Besitzer kennt den Schlüssel
- Kommunikation (z. B. per E-Mail) vor fremdem Zugriff schützen
  Die Schlüssel müssen ausgetauscht werden

Frage: Wie kann der Schlüsseltausch sicher durchgeführt werden?

# Wo wird Verschlüsselung benötigt?

- Daten (z. B. auf Festplatte) vor fremdem Zugriff schützen
  Nur der Besitzer kennt den Schlüssel
- Kommunikation (z. B. per E-Mail) vor fremdem Zugriff schützen
  Die Schlüssel müssen ausgetauscht werden

Frage: Wie kann der Schlüsseltausch sicher durchgeführt werden?

# Motivation

## Ausgangslage

Person A (Alice) will an Person B (Bob) wollen eine Nachricht ***m*** übermitteln

## Problem

- Person F (Fred) soll die Nachricht nicht lesen können



# Motivation

## Ausgangslage

Person A (Alice) will an Person B (Bob) wollen eine Nachricht ***m*** übermitteln

## Problem

- Person F (Fred) soll die Nachricht nicht lesen können



## Lösung 1

Alice versteckt die Botschaft ***m*** in einem unverfänglichen Brief. Bob kann die versteckte Botschaft entschlüsseln. Fred ahnt gar nicht, dass die unverfängliche Botschaft einen geheimen Teil hat.

## Lösung 2

- Alice und Bob vereinbaren einen Schlüssel und ein Verschlüsselungs- und Entschlüsselungsmethode und
- Alice und Bob vereinbaren einen Schlüssel
- Möglicherweise sind Schlüssel zur Verschlüsselung $k_e$ und Entschlüsselung $k_d$ ebenso wie Verschlüsselungsmethode $E$ und Entschlüsselungsmethode $D$ unterschiedlich

## Lösung 1

Alice versteckt die Botschaft ***m*** in einem unverfänglichen Brief. Bob kann die versteckte Botschaft entschlüsseln. Fred ahnt gar nicht, dass die unverfängliche Botschaft einen geheimen Teil hat.

## Lösung 2

- Alice und Bob vereinbaren einen Schlüssel und ein Verschlüsselungs- und Entschlüsselungsmethode und
- Alice und Bob vereinbaren einen Schlüssel
- Möglicherweise sind Schlüssel zur Verschlüsselung $k_e$ und Entschlüsselung $k_d$ ebenso wie Verschlüsselungsmethode $E$ und Entschlüsselungsmethode $D$ unterschiedlich



# Unverfänglicher Brief

## Der Text

Lieber Freund, hiermit schicke ich Dir eine streng geheime Botschaft.

## Die Datei: Text.pgm

```
P5
10 7
255
Lieber Freund,
hiermit schicke ich Dir eine streng
geheime Botschaft.
```

# Unverfänglicher Brief

## Der Text

Lieber Freund, hiermit schicke ich Dir eine streng geheime Botschaft.

## Die Datei: Text.pgm

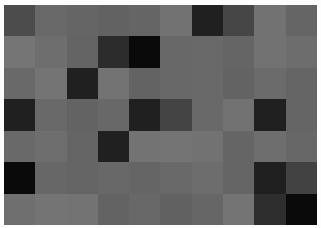
```
P5
10 7
255
Lieber Freund,
hiermit schicke ich Dir eine streng
geheime Botschaft.
```

## Der Trick: Text.png

`convert Text.pgm Text.png` [1]

### Das Ergebnis

Lieber Freund, sieh mal, was meine neue Kamera heute für ein merkwürdiges Bild geliefert hat.



### Die Technik

*Steganographie*

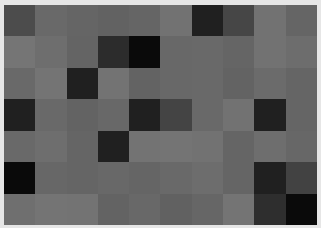
[1] PNG liefert eine verlustfreie Datenkompression, in der dann der Text nicht mehr direkt sichtbar ist.

## Der Trick: Text.png

`convert Text.pgm Text.png` [1]

## Das Ergebnis

Lieber Freund, sieh mal, was meine neue Kamera heute für ein merkwürdiges Bild geliefert hat.



## Die Technik

*Steganographie*

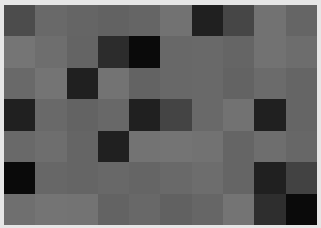
[1] PNG liefert eine verlustfreie Datenkompression, in der dann der Text nicht mehr direkt sichtbar ist.

## Der Trick: Text.png

`convert Text.pgm Text.png` [1]

## Das Ergebnis

Lieber Freund, sieh mal, was meine neue Kamera heute für ein merkwürdiges Bild geliefert hat.



## Die Technik

*Steganographie*

[1] PNG liefert eine verlustfreie Datenkompression, in der dann der Text nicht mehr direkt sichtbar ist.

# Historisches Beispiel

## Caesar und Kleopatra

- haben angeblich geheime Briefe ausgetauscht
- Zwei Alphabetscheiben gegeneinander verdreht
- Schlüssel: Richtung und Anzahl der Buchstaben



## Notation

| | |
|---|---|
| Nachricht | $\boldsymbol{m} = m_1, m_2 \ldots, m_n$ |
| Verschlüsselte Nachricht | $\boldsymbol{c}$ |
| Verschlüsselungsfunktion | $E$ |
| Entschlüsselungsfunktion | $D$ |
| Verschlüsselungsschlüssel | $k_e$ |
| Entschlüsselungsschlüssel | $k_d$ |



## Notation

| | |
|---|---|
| Nachricht | $\boldsymbol{m} = m_1, m_2 \ldots, m_n$ |
| Verschlüsselte Nachricht | $\boldsymbol{c}$ |
| Verschlüsselungsfunktion | $E$ |
| Entschlüsselungsfunktion | $D$ |
| Verschlüsselungsschlüssel | $k_e$ |
| Entschlüsselungsschlüssel | $k_d$ |
| Öffentlicher Schlüssel | $k_{\text{pu}}$ |
| Privater Schlüssel | $k_{\text{pr}}$ |



Damit:

$$\boldsymbol{m} \rightarrow E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$$
$$\boldsymbol{c} \rightarrow D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$$

Also:

$$D_{k_d}(E_{k_e}(\boldsymbol{m})) = \boldsymbol{m}$$



Damit:

$$\boldsymbol{m} \rightarrow E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$$

$$\boldsymbol{c} \rightarrow D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$$

Also:

$$D_{k_d}(E_{k_e}(\boldsymbol{m})) = \boldsymbol{m}$$



Damit:

$$\boldsymbol{m} \rightarrow E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$$
$$\boldsymbol{c} \rightarrow D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$$

Also:

$$D_{k_d}(E_{k_e}(\boldsymbol{m})) = \boldsymbol{m}$$



Damit:

$$\boldsymbol{m} \rightarrow E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$$
$$\boldsymbol{c} \rightarrow D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$$

Also:

$$D_{k_d}(E_{k_e}(\boldsymbol{m})) = \boldsymbol{m}$$



Damit:

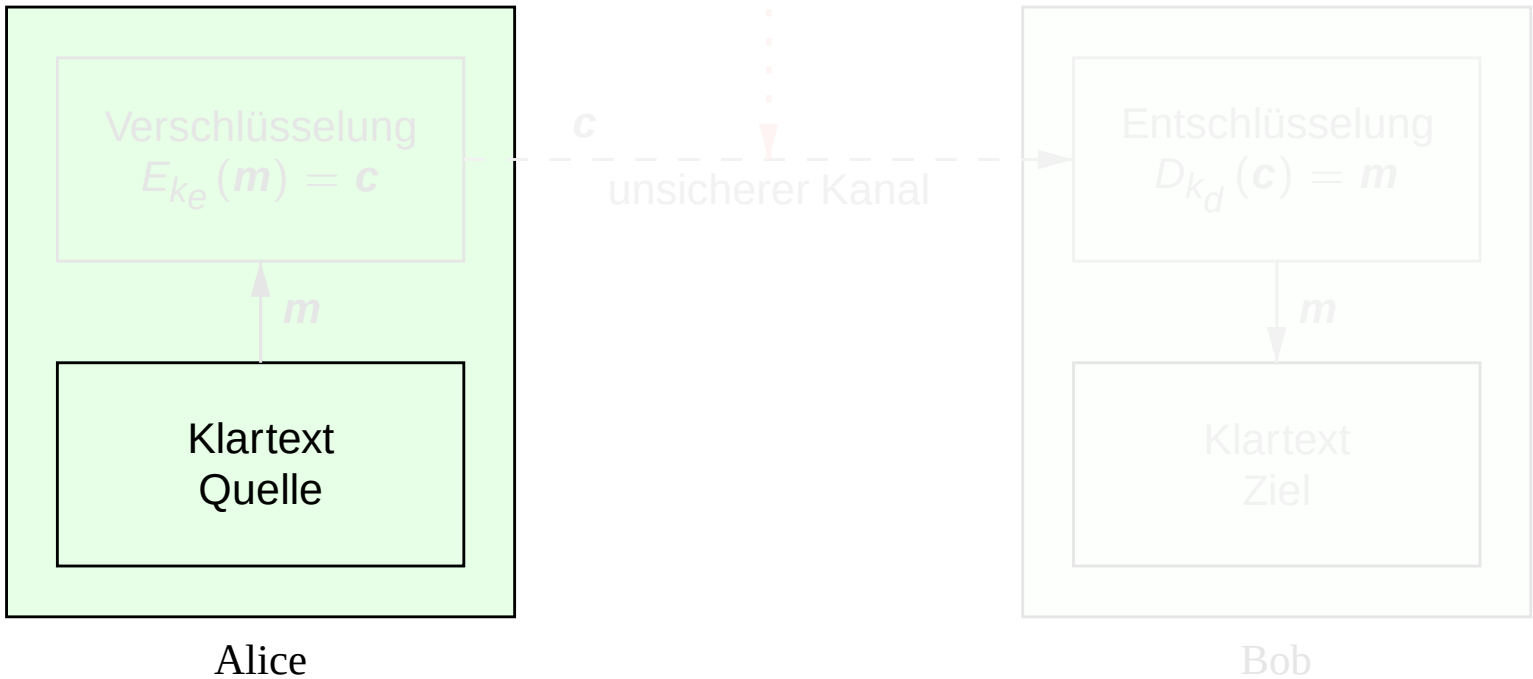
$$\boldsymbol{m} \rightarrow E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$$
$$\boldsymbol{c} \rightarrow D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$$

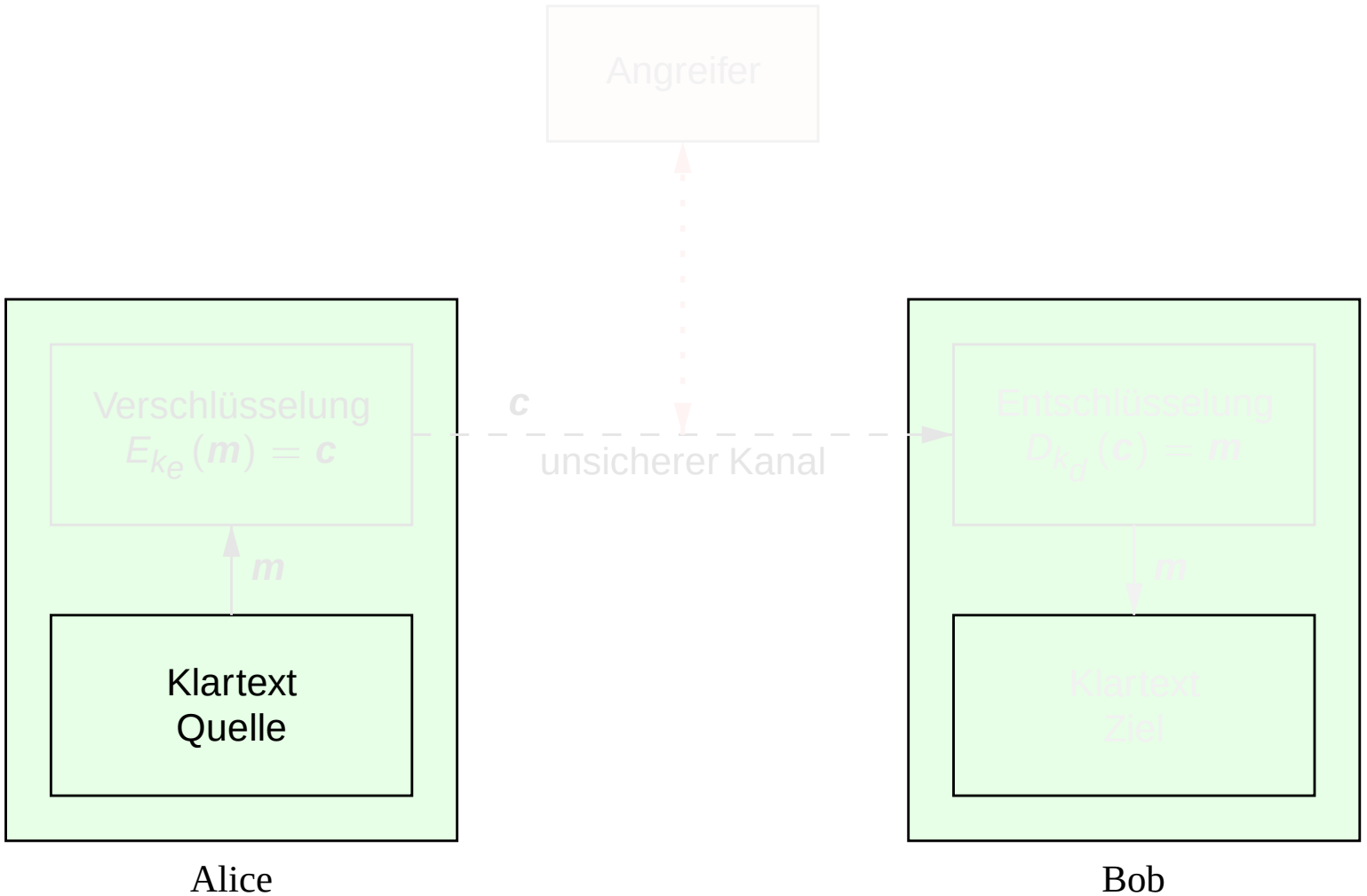
Also:

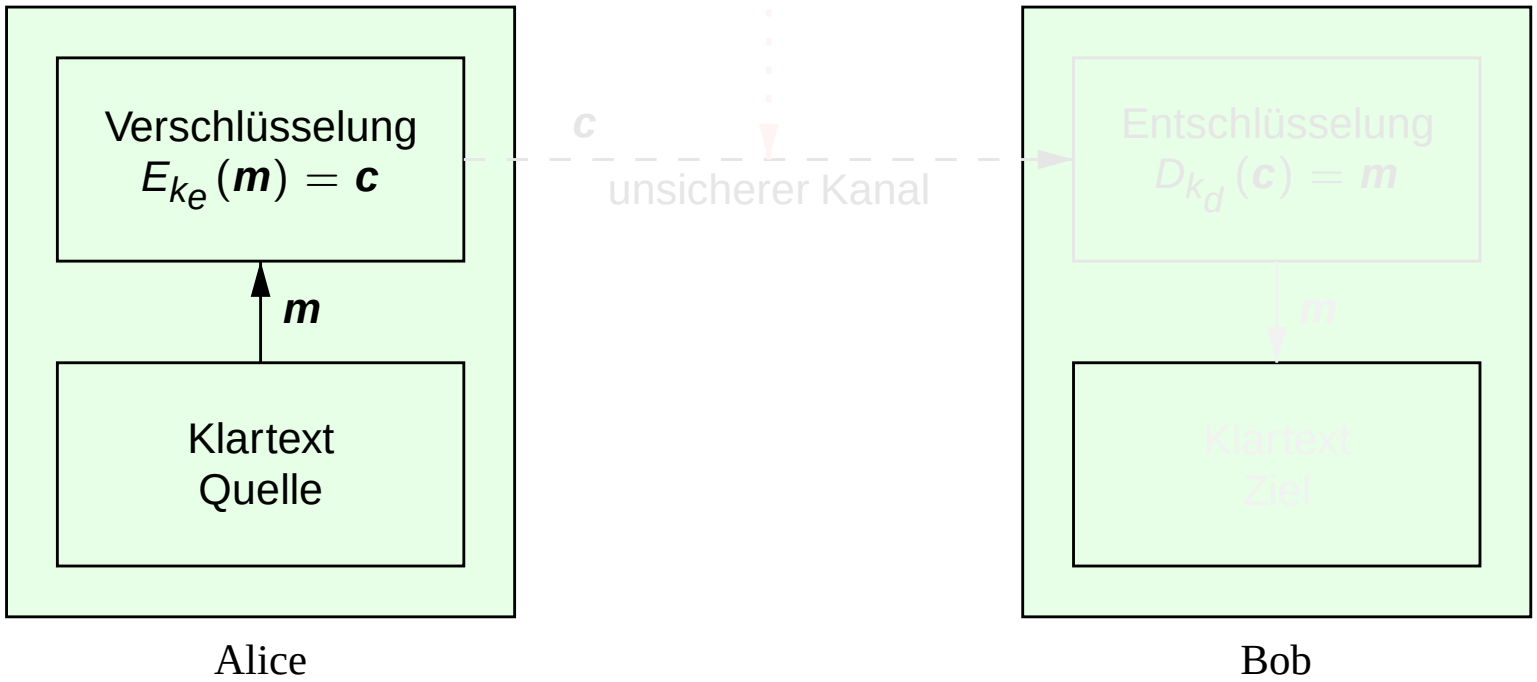
$$D_{k_d}(E_{k_e}(\boldsymbol{m})) = \boldsymbol{m}$$

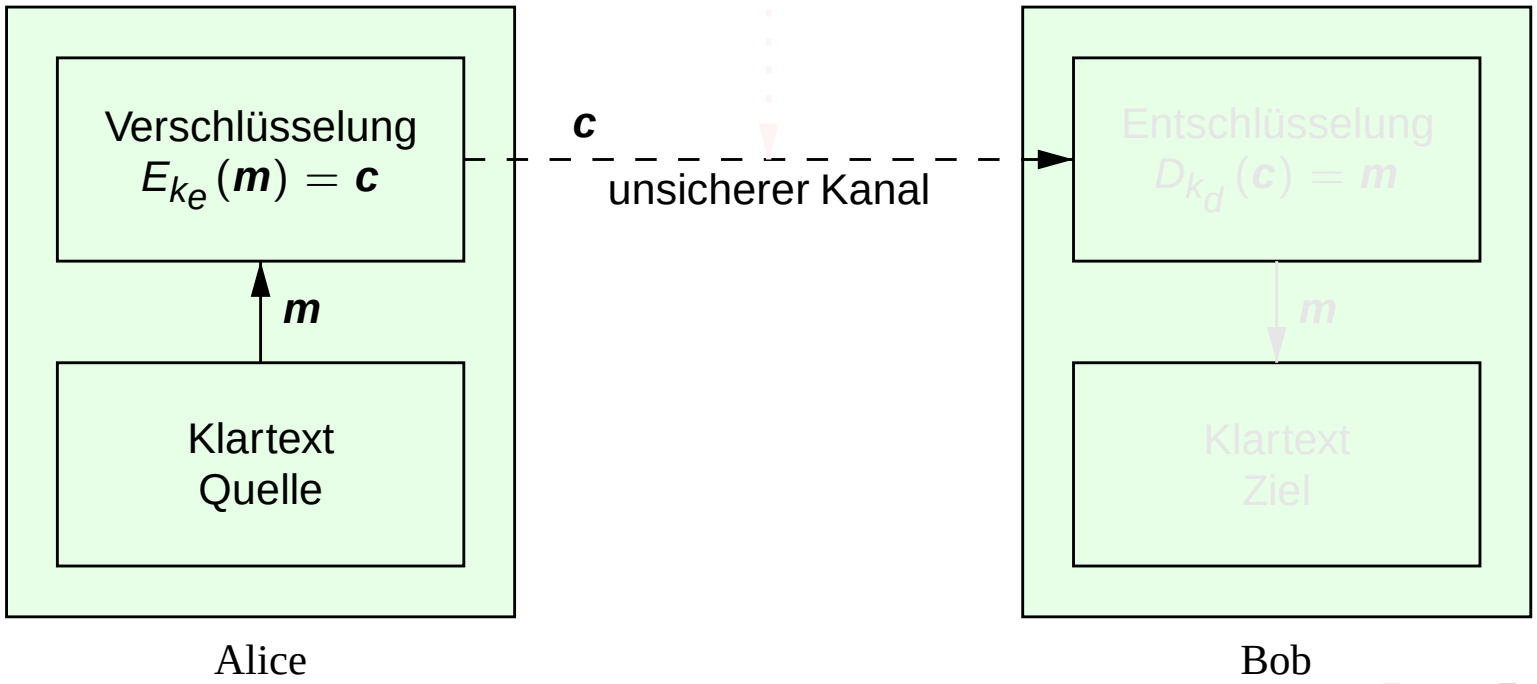
Das folgende Bild und viele weitere Informationen und Bilder aus diesem Kurs stammen aus [MVO96].

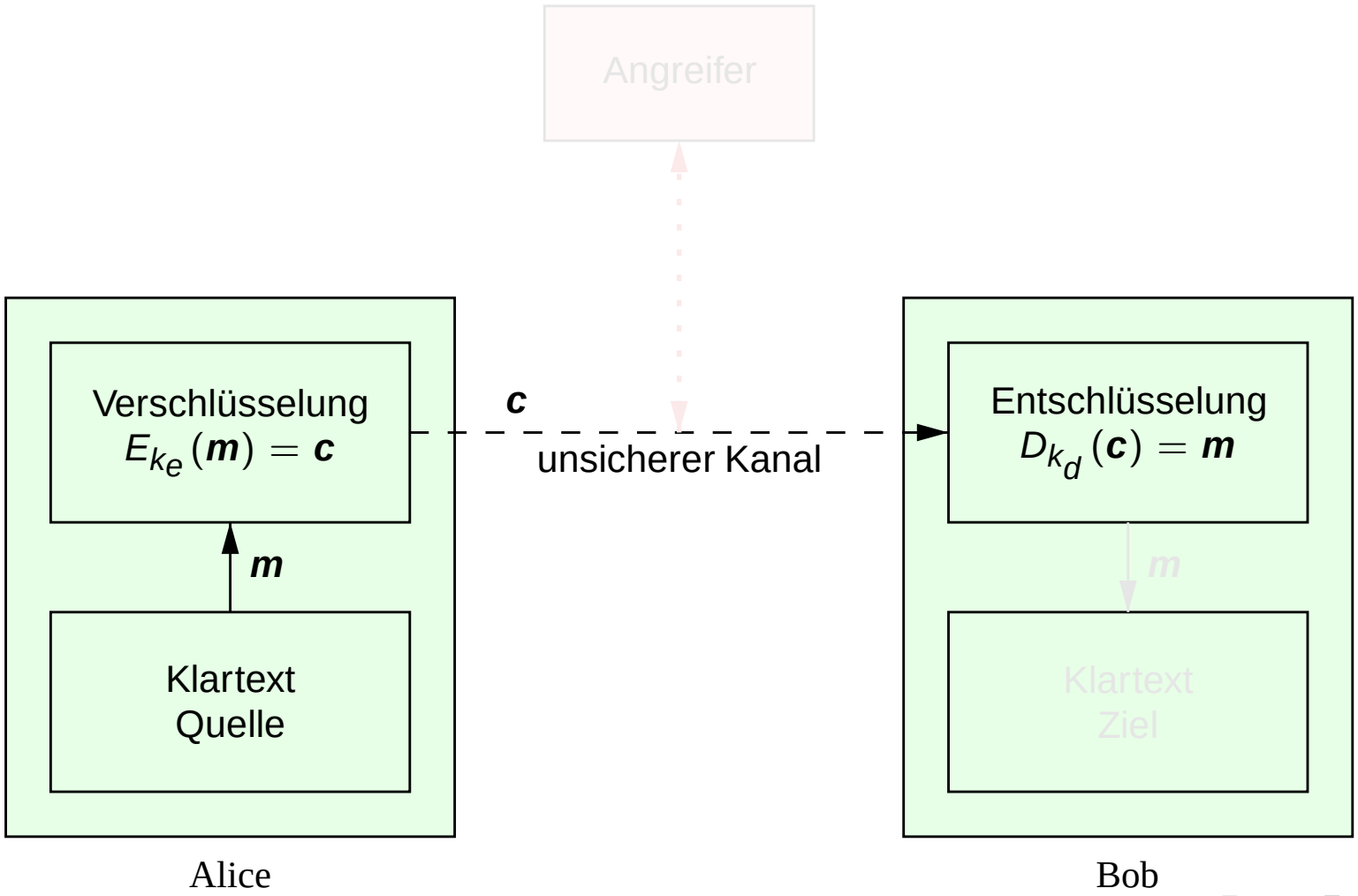


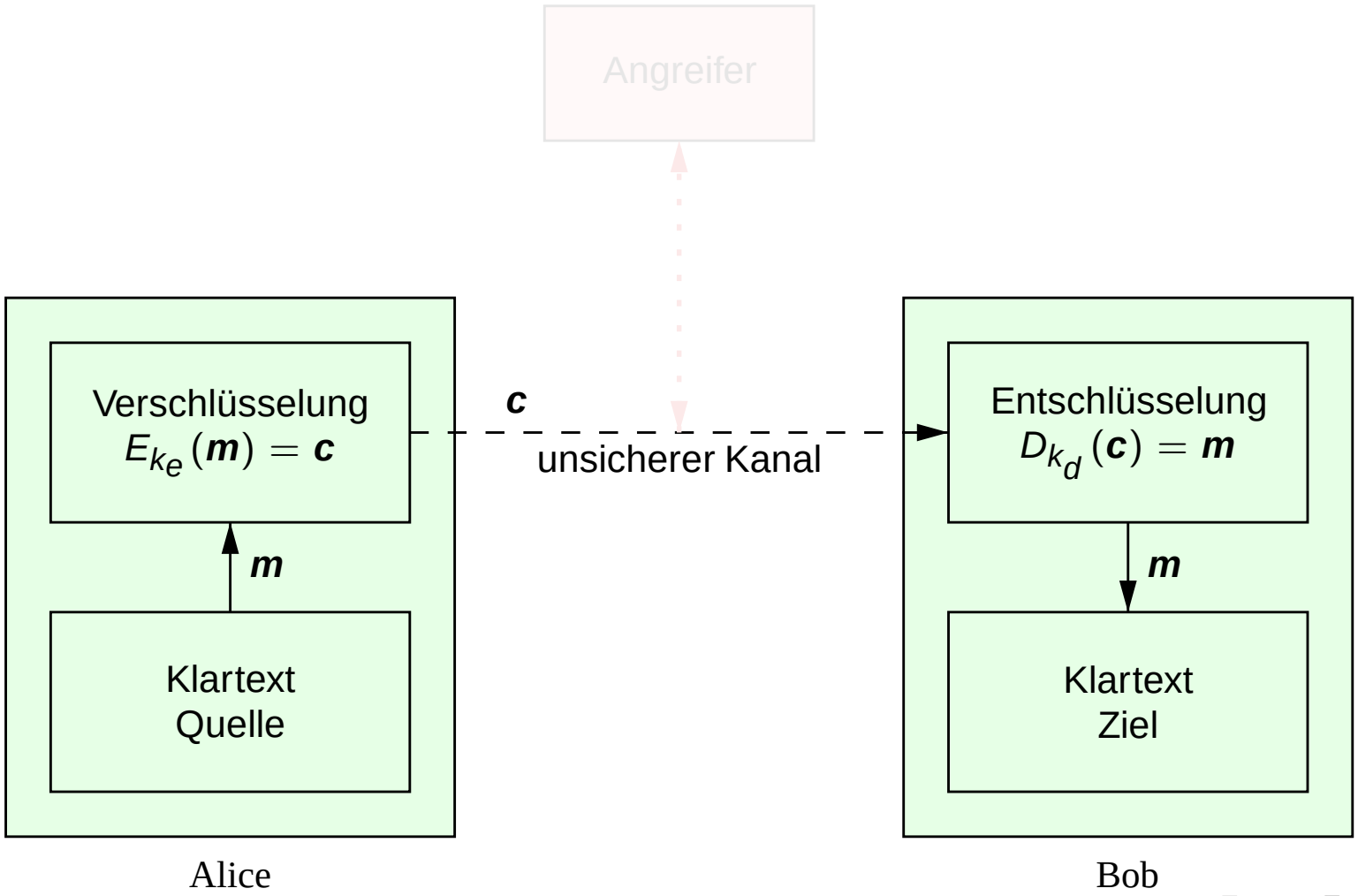

Angreifer
Verschlüsselung
$E_{k_e}(m) = c$
$c$
unsicherer Kanal
Entschlüsselung
$D_{k_d}(c) = m$
$m$
$m$
Klartext
Quelle
Klartext
Ziel
Alice
Bob

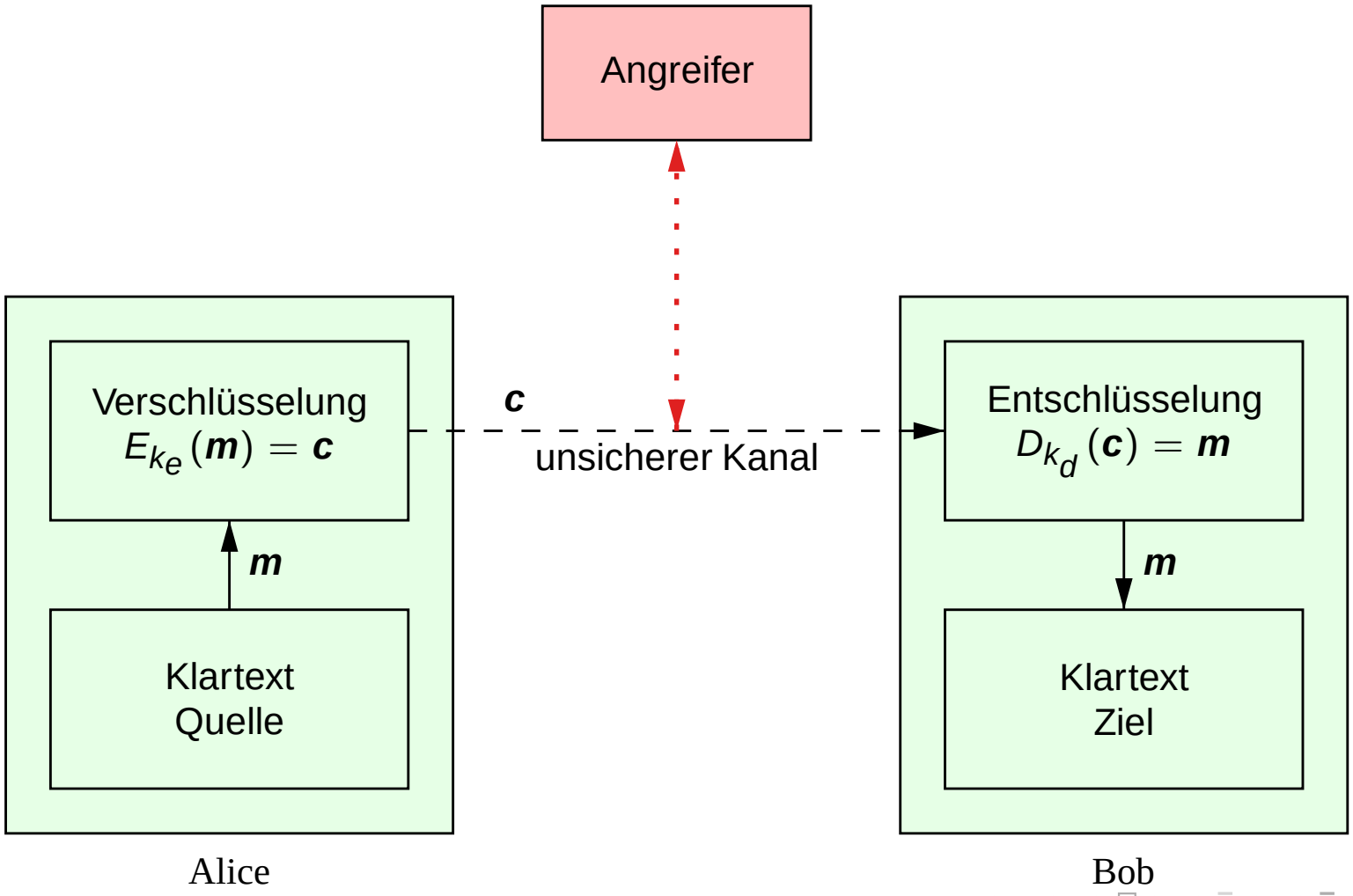

Angreifer
Verschlüsselung
$E_{k_e}(m) = c$
c
unsicherer Kanal
Entschlüsselung
$D_{k_d}(c) = m$
m
m
Klartext
Quelle
Klartext
Ziel
Alice
Bob


Angreifer
Verschlüsselung
$E_{k_e}(m) = c$
m
Klartext
Quelle
c
unsicherer Kanal
Entschlüsselung
$D_{k_d}(c) = m$
m
Klartext
Ziel
Alice
Bob


Angreifer
Verschlüsselung
$E_{k_e}(m) = c$
m
Klartext
Quelle
c
unsicherer Kanal
Entschlüsselung
$D_{k_d}(c) = m$
m
Klartext
Ziel
Alice
Bob


Angreifer
Verschlüsselung
$E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{c}$
unsicherer Kanal
Entschlüsselung
$D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
$\boldsymbol{m}$
Klartext
Quelle
Klartext
Ziel
Alice
Bob

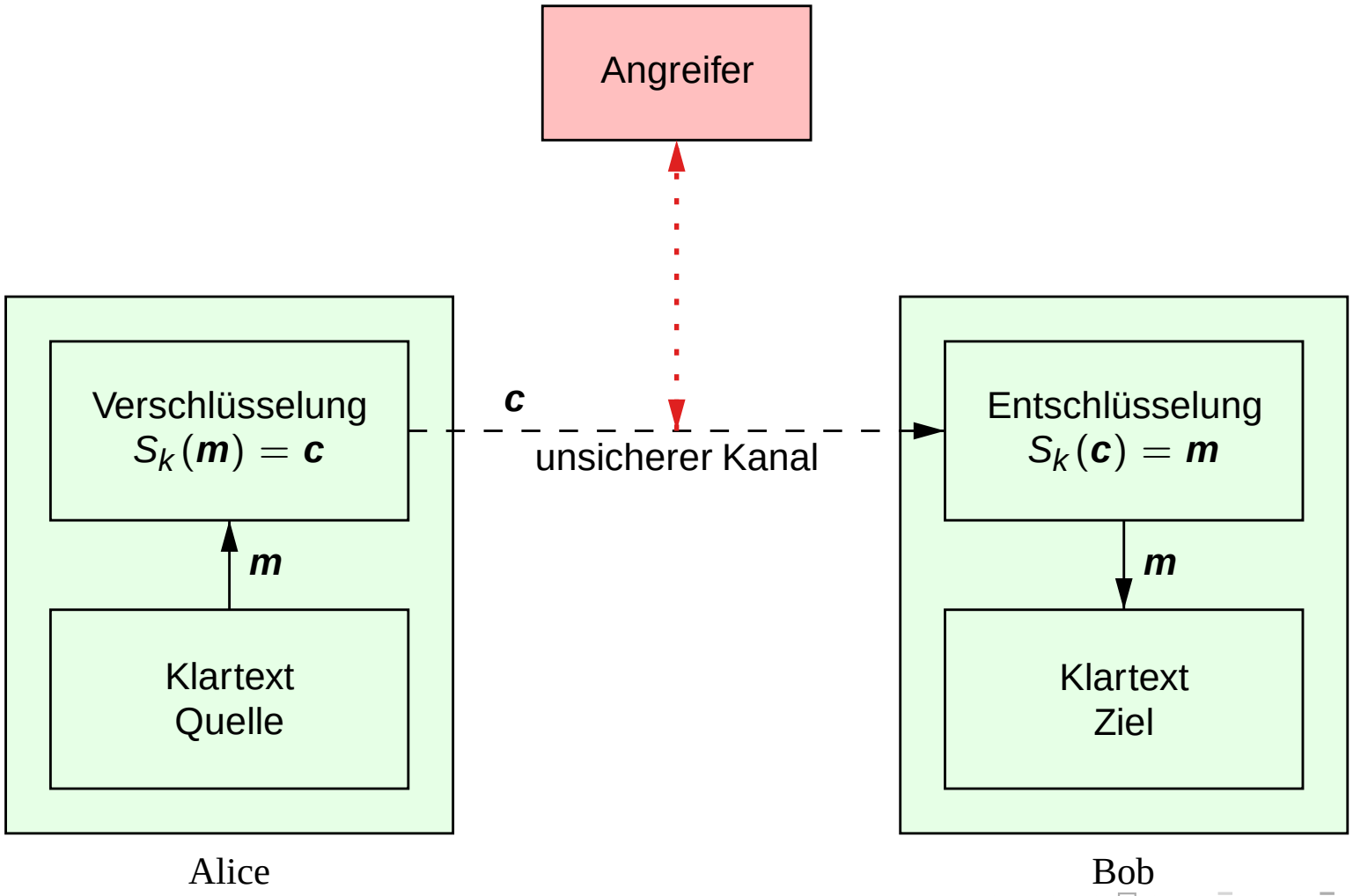

Angreifer
Verschlüsselung
$E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{c}$
unsicherer Kanal
Entschlüsselung
$D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
$\boldsymbol{m}$
Klartext
Quelle
Klartext
Ziel
Alice
Bob


Angreifer
Verschlüsselung
$E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{c}$
unsicherer Kanal
Entschlüsselung
$D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
$\boldsymbol{m}$
Klartext
Quelle
Klartext
Ziel
Alice
Bob

# Einfachster Fall

Entschlüsselungsverfahren = Verschlüsselungsverfahren
Verschlüsselungsfunktion $E$ = Entschlüsselungsfunktion $D$ = $S$

Entschlüsselungsschlüssel = Verschlüsselungsschlüssel
Verschlüsselung $k_e$ = Entschlüsselung $k_d$ = $k$


Angreifer
Verschlüsselung
$S_k(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{c}$
unsicherer Kanal
Entschlüsselung
$S_k(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
$\boldsymbol{m}$
Klartext
Quelle
Klartext
Ziel
Alice
Bob



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$

Damit: $a \otimes b \otimes b = a$

Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$

$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| c | 0x63 | 0110 0011 |
|---|---|---|
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$

# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = m$



# XOR

$0 \otimes 0 = 0$
$1 \otimes 0 = 1$
$0 \otimes 1 = 1$
$1 \otimes 1 = 0$
Damit: $a \otimes b \otimes b = a$
Bitweise auf Buchstaben angewendet: $a = \boldsymbol{m}, b = k_e = k_d$.

| | | |
|---|---|---|
| c | 0x63 | 0110 0011 |
| 1 | 0x31 | 0011 0001 |
| | 0x52 | 0101 0010 |

$\boldsymbol{c} = E_{k_e}(\boldsymbol{m}) := \boldsymbol{m} \otimes k_e$
$D_{k_d}(\boldsymbol{c}) := \boldsymbol{c} \otimes k_e = \boldsymbol{m} \otimes k_d \otimes k_e = \boldsymbol{m}$



| * | N | a | c | h | r | i | c | h | t | ! |
|---|---|---|---|---|---|---|---|---|---|---|
| 2a | 4e | 61 | 63 | 68 | 72 | 69 | 63 | 68 | 74 | 21 |
| G | e | h | e | i | m | 1 | G | e | h | e |
| 47 | 65 | 68 | 65 | 69 | 6d | 31 | 47 | 65 | 68 | 65 |
| 6d | 2b | 9 | 6 | 1 | 1f | 58 | 24 | d | 1c | 44 |
| 2a | 4e | 61 | 63 | 68 | 72 | 69 | 63 | 68 | 74 | 21 |



- einfach
- unsicher

Bessere Verfahren existieren.

Dies ist eine *symmetrische Verschlüsselung*[2]

[2] Mehr dazu in Teil 3.

- einfach
- unsicher

Bessere Verfahren existieren.

Dies ist eine *symmetrische Verschlüsselung*[2]

[2] Mehr dazu in Teil 3.



# Modulare Arithmetik

„Rechnen mit der Uhr“
z. B.:
6 + 3 = 9; 9 + 3 = 12;
9 + 4 = 1 mod 12

Basis einer möglichen Verschlüsselung mit $k_d \neq k_e$
z. B.:
Verschlüsseln von $x \rightarrow y = x + 5;$ $k_e = 5$
Entschlüsseln von $y \rightarrow y + 7 = x;$ $k_d = 7$

Dies ist eine (triviale) assymetrische Verschlüsselung[3]

[3] Mehr dazu in Teil 3



# Modulare Arithmetik

„Rechnen mit der Uhr“
z. B.:
6 + 3 = 9; 9 + 3 = 12;
9 + 4 = 1 mod 12

Basis einer möglichen Verschlüsselung mit $k_d \neq k_e$
z. B.:
Verschlüsseln von $x \rightarrow y = x + 5;$ $k_e = 5$
Entschlüsseln von $y \rightarrow y + 7 = x;$ $k_d = 7$

Dies ist eine (triviale) assymetrische Verschlüsselung[3]

[3] Mehr dazu in Teil 3

# Modulare Arithmetik

„Rechnen mit der Uhr“
z. B.:
6 + 3 = 9; 9 + 3 = 12;
9 + 4 = 1 mod 12

Basis einer möglichen Verschlüsselung mit $k_d \neq k_e$
z. B.:
Verschlüsseln von $x \rightarrow y = x + 5;$ $k_e = 5$
Entschlüsseln von $y \rightarrow y + 7 = x;$ $k_d = 7$

Dies ist eine (triviale) assymetrische Verschlüsselung[3]

[3] Mehr dazu in Teil 3



# Modulare Arithmetik

„Rechnen mit der Uhr“
z. B.:
6 + 3 = 9; 9 + 3 = 12;
9 + 4 = 1 mod 12

Basis einer möglichen Verschlüsselung mit $k_d \neq k_e$
z. B.:
Verschlüsseln von $x \rightarrow y = x + 5;$ $k_e = 5$
Entschlüsseln von $y \rightarrow y + 7 = x;$ $k_d = 7$

Dies ist eine (triviale) assymetrische Verschlüsselung[3]

[3] Mehr dazu in Teil 3

# Modulare Arithmetik

„Rechnen mit der Uhr“
z. B.:
6 + 3 = 9; 9 + 3 = 12;
9 + 4 = 1 mod 12

Basis einer möglichen Verschlüsselung mit $k_d \neq k_e$
z. B.:
Verschlüsseln von $x \rightarrow y = x + 5$; $k_e = 5$
Entschlüsseln von $y \rightarrow y + 7 = x$; $k_d = 7$

Dies ist eine (triviale) assymetrische Verschlüsselung[3]

[3] Mehr dazu in Teil 3

# Modulare Arithmetik

„Rechnen mit der Uhr“
z. B.:
6 + 3 = 9; 9 + 3 = 12;
9 + 4 = 1 mod 12

Basis einer möglichen Verschlüsselung mit $k_d \neq k_e$
z. B.:
Verschlüsseln von $x \rightarrow y = x + 5$; $k_e = 5$
Entschlüsseln von $y \rightarrow y + 7 = x$; $k_d = 7$

Dies ist eine (triviale) assymetrische Verschlüsselung[3]

[3] Mehr dazu in Teil 3

# Modulare Arithmetik

„Rechnen mit der Uhr“
z. B.:
6 + 3 = 9; 9 + 3 = 12;
9 + 4 = 1 mod 12

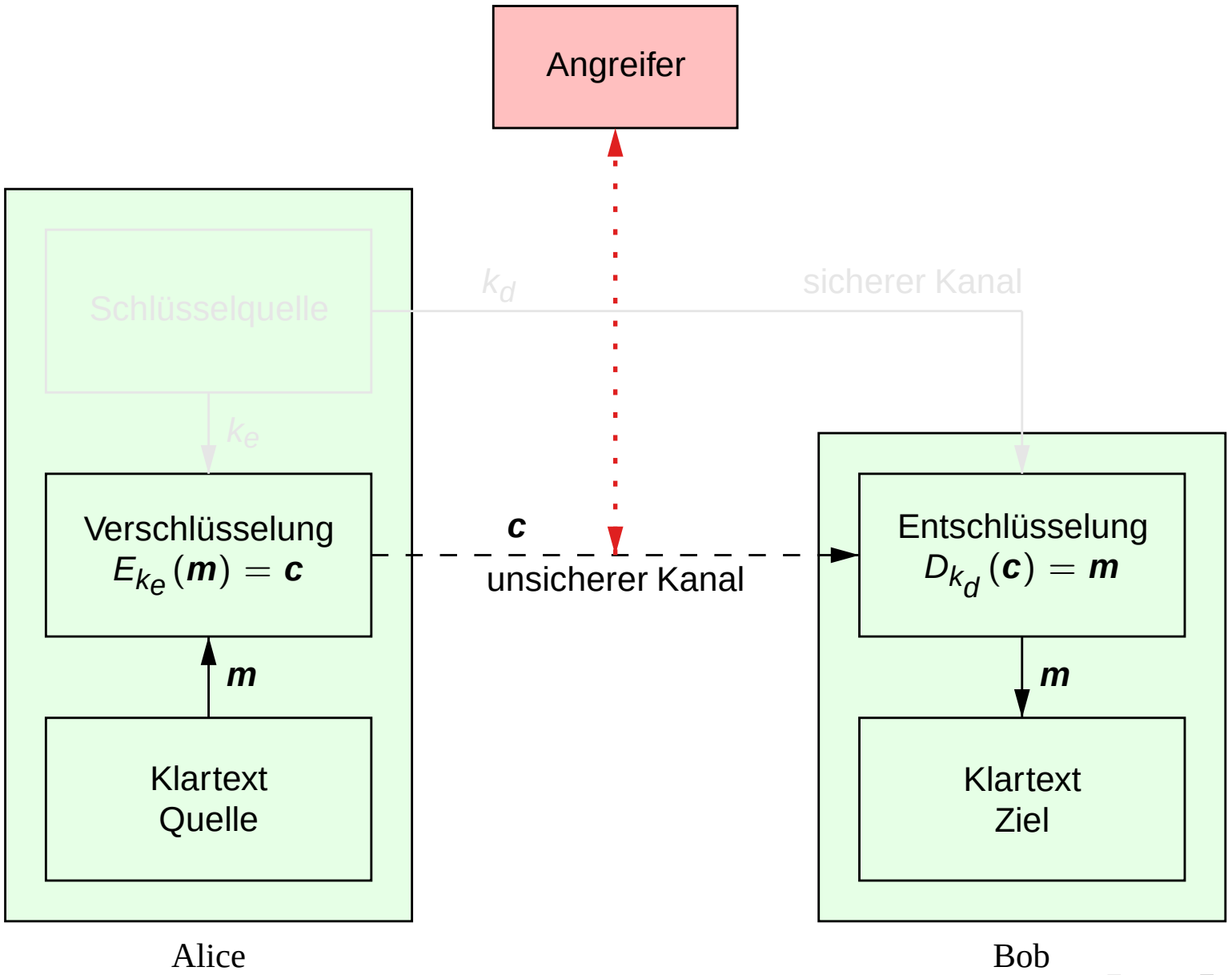
Basis einer möglichen Verschlüsselung mit $k_d \neq k_e$
z. B.:
Verschlüsseln von $x \rightarrow y = x + 5$; $k_e = 5$
Entschlüsseln von $y \rightarrow y + 7 = x$; $k_d = 7$

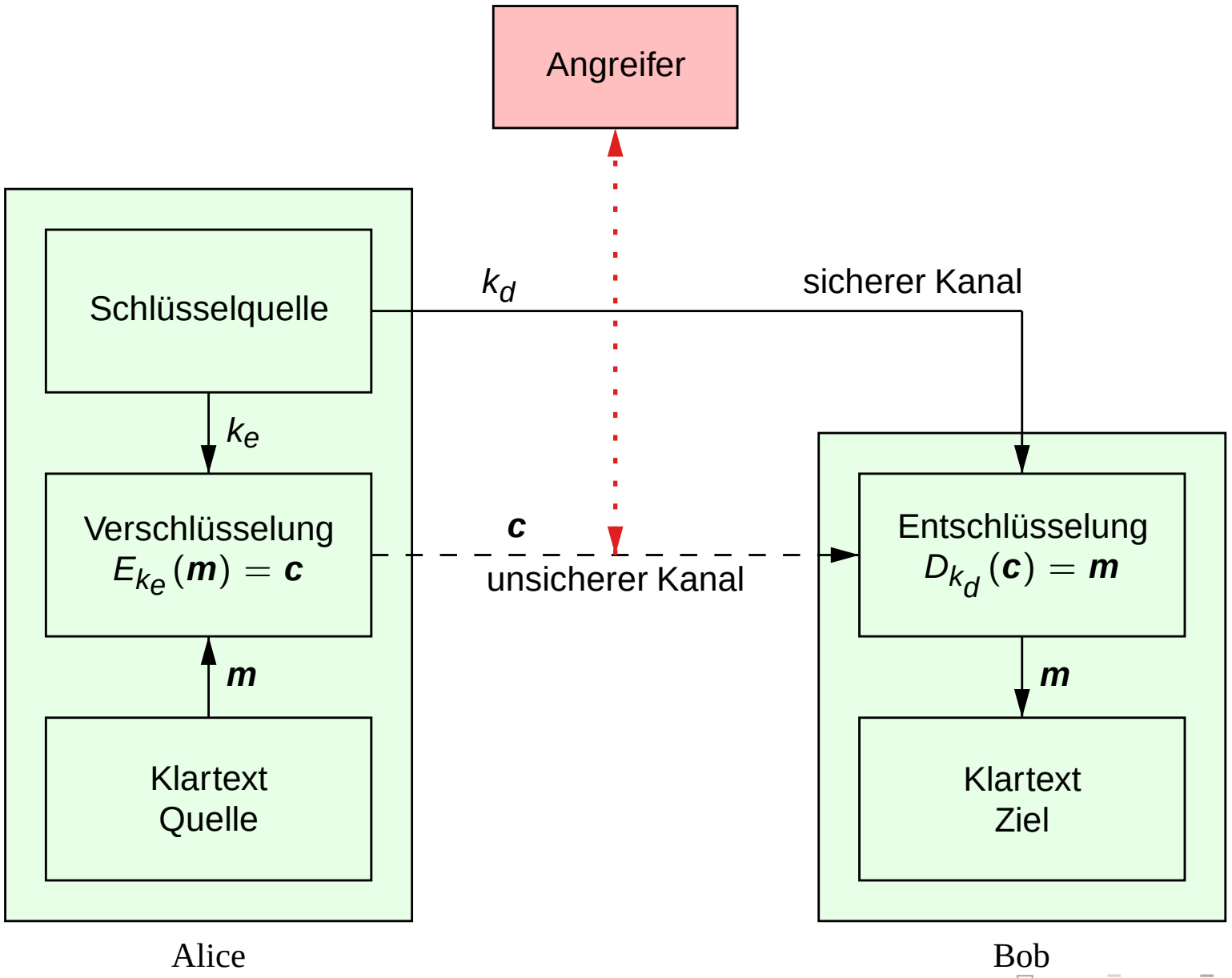
Dies ist eine (triviale) assymetrische Verschlüsselung[3]

[3] Mehr dazu in Teil 3

# Schlüsseltausch

In symmetrischer und unsymmetrischer Verschlüsselung:
Schlüsselaustausch muss über zuverlässige Kanäle erfolgen!


Angreifer
Schlüsselquelle
$k_d$
sicherer Kanal
$k_e$
Verschlüsselung
$E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{c}$
unsicherer Kanal
Entschlüsselung
$D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
$\boldsymbol{m}$
Klartext
Quelle
Klartext
Ziel
Alice
Bob


Angreifer
Schlüsselquelle
$k_d$
sicherer Kanal
$k_e$
Verschlüsselung
$E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{c}$
unsicherer Kanal
Entschlüsselung
$D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
$\boldsymbol{m}$
Klartext
Quelle
Klartext
Ziel
Alice
Bob

# Mehrere Sender

## Ausgangslage

Person A (Alice) will an Person B (Bob) eine Nachricht $\boldsymbol{m}_{\text{Alice}}$,
Person C (Chris) will an Person B (Bob) eine Nachricht $\boldsymbol{m}_{\text{Chris}}$
übermitteln



## Ziel

- Person F (Fred) soll keine Nachricht lesen können, Person A (Alice) soll $\boldsymbol{m}_{\text{Chris}}$ nicht lesen können, Person C (Chris) soll $\boldsymbol{m}_{\text{Alice}}$ nicht lesen können

## Aber

In unserem Beispiel:
Kenntnis von $k_d$ ermöglicht Berechnung von $k_e = 12 - k_d$
d.h.: Bob muss mit Alice und Chris jeweils einen anderes Schlüsselpaar vereinbaren

## Ziel

- Person F (Fred) soll keine Nachricht lesen können, Person A (Alice) soll $\boldsymbol{m}_{\text{Chris}}$ nicht lesen können, Person C (Chris) soll $\boldsymbol{m}_{\text{Alice}}$ nicht lesen können

## Aber

In unserem Beispiel:
Kenntnis von $k_d$ ermöglicht Berechnung von $k_e = 12 - k_d$
d.h.: Bob muss mit Alice und Chris jeweils einen anderes Schlüsselpaar vereinbaren

# Viele Sender

**Problemanalyse**

Kenntnis von $k_d$ ermöglicht Berechnung von $k_e$

**Lösung 1**

finde Verfahren, bei dem Kenntnis von $k_d$ die Berechnung von $k_e$ **nicht** ermöglicht

# Viele Sender

## Problemanalyse

Kenntnis von $k_d$ ermöglicht Berechnung von $k_e$

## Lösung 1

finde Verfahren, bei dem Kenntnis von $k_d$ die Berechnung von $k_e$ **nicht** ermöglicht



## Lösung 2 - stärker

Kenntnis von $k_d, \boldsymbol{c}, \boldsymbol{m}$ erlaubt die Berechnung $k_e$ **nicht**! [4]

### Folge

Der Verschlüsselungsschlüssel $k_e$ muss **nicht** sicher übertragen werden – er kann öffentlich sein!

[4] Mehr dazu in Teil 3

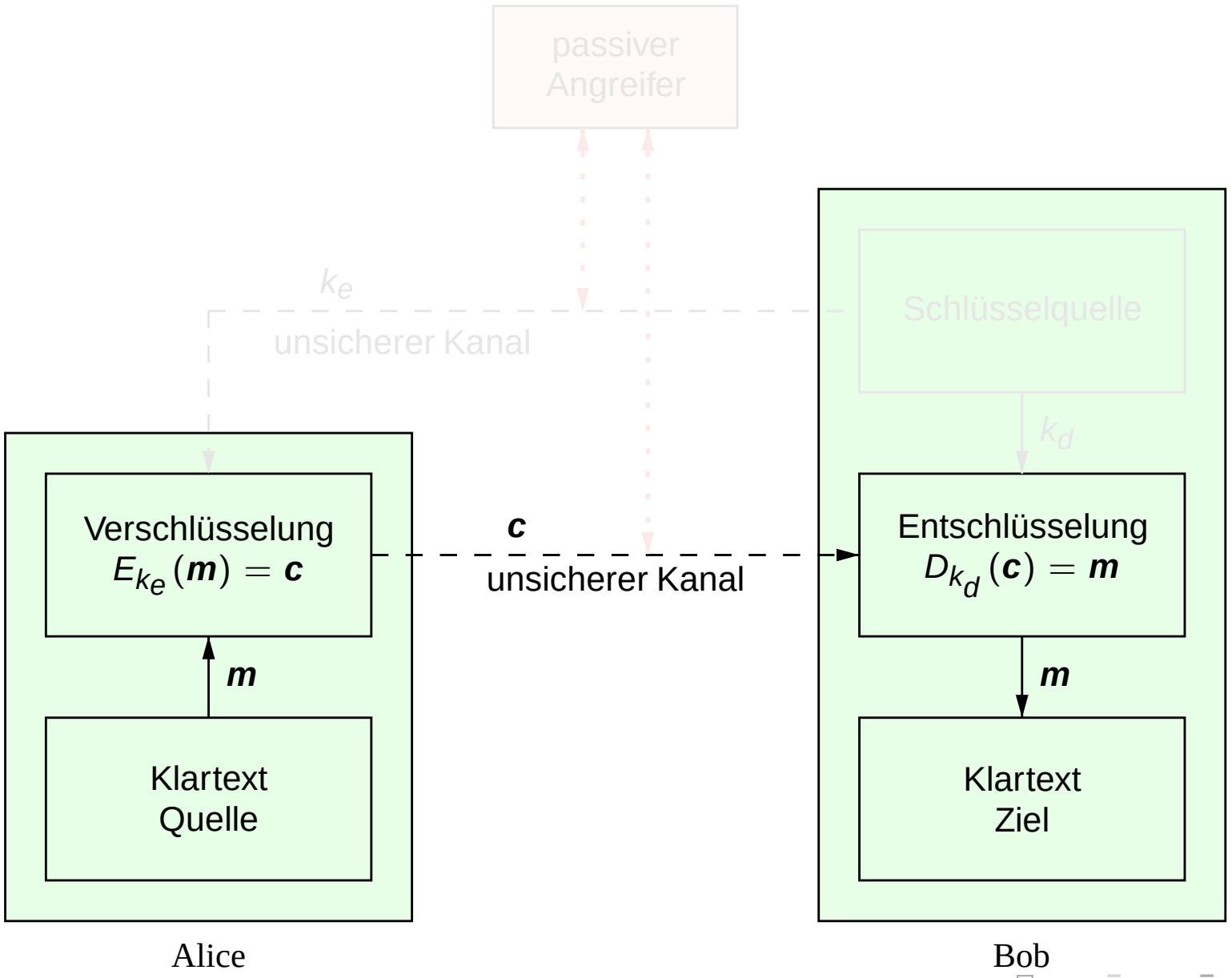
## Lösung 2 - stärker

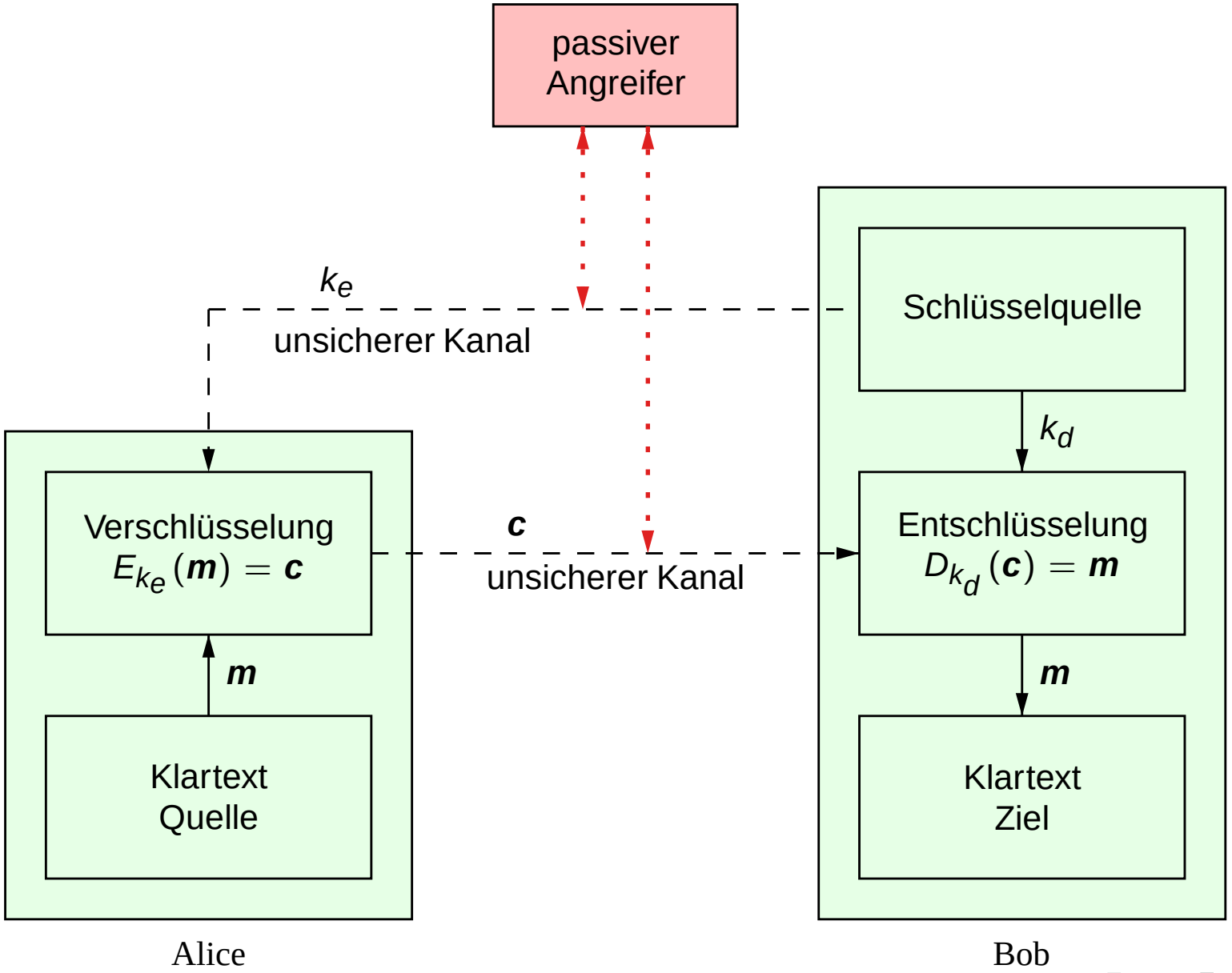
Kenntnis von $k_d, \boldsymbol{c}, \boldsymbol{m}$ erlaubt die Berechnung $k_e$ **nicht**! [4]

## Folge

Der Verschlüsselungsschlüssel $k_e$ muss **nicht** sicher übertragen werden – er kann öffentlich sein!

[4] Mehr dazu in Teil 3


passiver Angreifer
$k_e$
unsicherer Kanal
Schlüsselquelle
$k_d$
Verschlüsselung
$E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{c}$
unsicherer Kanal
Entschlüsselung
$D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
$\boldsymbol{m}$
Klartext Quelle
Klartext Ziel
Alice
Bob


passiver Angreifer
$k_e$
unsicherer Kanal
Schlüsselquelle
$k_d$
Verschlüsselung
$E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{c}$
unsicherer Kanal
Entschlüsselung
$D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
$\boldsymbol{m}$
Klartext Quelle
Klartext Ziel
Alice
Bob



# Viele Empfänger

## Beispiel

Person A (Alice) will an Person B (Bob) und Person C (Chris) eine Nachricht $\boldsymbol{m}_{\text{Alice}}$ verschlüsselt übermitteln

## Voraussetzung

Person A (Alice) kennt die Schlüssel

- $k_{e\,\text{Bob}}$ für Person B (Bob)
- $k_{e\,\text{Chris}}$ für Person C (Chris)

# Viele Empfänger

## Beispiel

Person A (Alice) will an Person B (Bob) und Person C (Chris) eine Nachricht $\boldsymbol{m}_{\text{Alice}}$ verschlüsselt übermitteln

## Voraussetzung

Person A (Alice) kennt die Schlüssel

- $k_{e\,\text{Bob}}$ für Person B (Bob)
- $k_{e\,\text{Chris}}$ für Person C (Chris)

## Methode 1

Person A (Alice) sendet die Nachricht $\boldsymbol{m}$

- als $E_{k_{e\,\mathrm{Bob}}}(\boldsymbol{m})$ an Person B (Bob)
- als $E_{k_{e\,\mathrm{Chris}}}(\boldsymbol{m})$ an Person C (Chris)

## Methode 2 (besser)

Person A erzeugt einen Zufallsschlüssel $k_r$ und berechnet

- $\boldsymbol{c}_{\mathrm{Bob}} = E_{k_{e\,\mathrm{Bob}}}(k_r)$
- $\boldsymbol{c}_{\mathrm{Chris}} = E_{k_{e\,\mathrm{Chris}}}(k_r)$

Person A sendet die Nachricht $\boldsymbol{m}$ an alle als $\boldsymbol{c}_{\mathrm{Bob}} + \boldsymbol{c}_{\mathrm{Bob}} + Sk_r(\boldsymbol{m})$

## Methode 1

Person A (Alice) sendet die Nachricht $\boldsymbol{m}$

- als $E_{k_{e\,\mathrm{Bob}}}(\boldsymbol{m})$ an Person B (Bob)
- als $E_{k_{e\,\mathrm{Chris}}}(\boldsymbol{m})$ an Person C (Chris)

## Methode 2 (besser)

Person A erzeugt einen Zufallsschlüssel $k_r$ und berechnet

- $\boldsymbol{c}_{\mathrm{Bob}} = E_{k_{e\,\mathrm{Bob}}}(k_r)$
- $\boldsymbol{c}_{\mathrm{Chris}} = E_{k_{e\,\mathrm{Chris}}}(k_r)$

Person A sendet die Nachricht $\boldsymbol{m}$ an alle als $\boldsymbol{c}_{\mathrm{Bob}} + \boldsymbol{c}_{\mathrm{Bob}} + Sk_r(\boldsymbol{m})$

Neues Problem: Zufallsschlüssel erzeugen $\rightarrow$ später



# Schlüsselaustausch

Idee: Wenn aus $k_d, \boldsymbol{c}, \boldsymbol{m}$ der Schlüssel $k_e$ nicht erraten werden kann, dann kann auch $k_d$ unsicher übertragen werden!

Noch besser: $k_d$ darf allgemein bekannt sein!

$\rightarrow$ Öffentliche Schlüssel-Verzeichnisse (z. B. `keyserver.net`)



# Schlüsselaustausch

Idee: Wenn aus $k_d, \boldsymbol{c}, \boldsymbol{m}$ der Schlüssel $k_e$ nicht erraten werden kann, dann kann auch $k_d$ unsicher übertragen werden!
Noch besser: $k_d$ darf allgemein bekannt sein!
$\rightarrow$ Öffentliche Schlüssel-Verzeichnisse (z. B. `keyserver.net`)



# Schlüsselaustausch

Idee: Wenn aus $k_d, \boldsymbol{c}, \boldsymbol{m}$ der Schlüssel $k_e$ nicht erraten werden kann, dann kann auch $k_d$ unsicher übertragen werden!
Noch besser: $k_d$ darf allgemein bekannt sein!
$\rightarrow$ Öffentliche Schlüssel-Verzeichnisse (z. B. `keyserver.net`)



# Digitale Signatur

## Beispiel

Person A (Alice) will an Person B (Bob) eine öffentliche Nachricht $\boldsymbol{m}_{\text{Alice}}$ schicken. Für Bob soll überprüfbar sein, ob die Nachricht von Alice stammt und nicht verändert wurde.

## Versand - Methode 1

Person A (Alice) signiert die Nachricht $\boldsymbol{m}$ mit dem (geheimen!) Schlüssel $k_{d\,\text{Alice}}$:
$s = E_{k_{d\,\text{Alice}}}(\boldsymbol{m})$
und sendet die Nachricht $\boldsymbol{m}$ als $s + \boldsymbol{m}$

# Digitale Signatur

## Beispiel

Person A (Alice) will an Person B (Bob) eine öffentliche Nachricht $\boldsymbol{m}_{\text{Alice}}$ schicken. Für Bob soll überprüfbar sein, ob die Nachricht von Alice stammt und nicht verändert wurde.

## Versand - Methode 1

Person A (Alice) signiert die Nachricht $\boldsymbol{m}$ mit dem (geheimen!) Schlüssel $k_{d\,\text{Alice}}$:
$s = E_{k_{d\,\text{Alice}}}(\boldsymbol{m})$
und sendet die Nachricht $\boldsymbol{m}$ als $s + \boldsymbol{m}$

## Überprüfung - Methode 1

Person B (Bob) erhält $s + \boldsymbol{m} = E_{k_d\,\text{Alice}}(\boldsymbol{m}) + \boldsymbol{m}$
und berechnet $s_t = D_{k_e\,\text{Alice}}(\boldsymbol{m})$
(mit dem öffentlichen **Ver**schlüsselungsschlüssel von Alice)
Wenn $s = s_t$ dann ist alles o.k.

## Voraussetzung

d. h.: $D(E(\boldsymbol{m})) = \boldsymbol{m}$
und $E(D(\boldsymbol{m})) = \boldsymbol{m}$

## Nachteil - Methode 1

Nachrichtenlänge verdoppelt sich

## Überprüfung - Methode 1

Person B (Bob) erhält $s + \boldsymbol{m} = E_{k_d\,\text{Alice}}(\boldsymbol{m}) + \boldsymbol{m}$
und berechnet $s_t = D_{k_e\,\text{Alice}}(\boldsymbol{m})$
(mit dem öffentlichen **Ver**schlüsselungsschlüssel von Alice)
Wenn $s = s_t$ dann ist alles o.k.

## Voraussetzung

***m*** und ***c*** sind aus dem gleichen Definitionsbereich (z. B. Zeichenketten)
d. h.: $D(E(\boldsymbol{m})) = \boldsymbol{m}$
und $E(D(\boldsymbol{m})) = \boldsymbol{m}$

## Nachteil - Methode 1

Nachrichtenlänge verdoppelt sich

## Überprüfung - Methode 1

Person B (Bob) erhält $s + \boldsymbol{m} = E_{k_d\,\text{Alice}}(\boldsymbol{m}) + \boldsymbol{m}$
und berechnet $s_t = D_{k_e\,\text{Alice}}(\boldsymbol{m})$
(mit dem öffentlichen **Ver**schlüsselungsschlüssel von Alice)
Wenn $s = s_t$ dann ist alles o.k.

## Voraussetzung

$\boldsymbol{m}$ und $\boldsymbol{c}$ sind aus dem gleichen Definitionsbereich (z. B. Zeichenketten)
d. h.: $D(E(\boldsymbol{m})) = \boldsymbol{m}$
und $E(D(\boldsymbol{m})) = \boldsymbol{m}$

## Nachteil - Methode 1

Nachrichtenlänge verdoppelt sich

## Überprüfung - Methode 1

Person B (Bob) erhält $s + \boldsymbol{m} = E_{k_d\,\text{Alice}}(\boldsymbol{m}) + \boldsymbol{m}$
und berechnet $s_t = D_{k_e\,\text{Alice}}(\boldsymbol{m})$
(mit dem öffentlichen **Ver**schlüsselungsschlüssel von Alice)
Wenn $s = s_t$ dann ist alles o.k.

## Voraussetzung

$\boldsymbol{m}$ und $\boldsymbol{c}$ sind aus dem gleichen Definitionsbereich (z. B. Zeichenketten)
d. h.: $D(E(\boldsymbol{m})) = \boldsymbol{m}$
und $E(D(\boldsymbol{m})) = \boldsymbol{m}$

## Nachteil - Methode 1

Nachrichtenlänge verdoppelt sich

## Versand - Methode 2

Person A (Alice) signiert den Extrakt der Nachricht $\boldsymbol{m}$ mit dem (geheimen!) Schlüssel $k_{d\,\text{Alice}}$:
$s = h(E_{k_{d\,\text{Alice}}}(\boldsymbol{m}))$
und sendet die Nachricht $\boldsymbol{m}$ als $s + \boldsymbol{m}$

## Überprüfung - Methode 2

Person B (Bob) erhält $s + \boldsymbol{m} = h(E_{k_{d\,\text{Alice}}}(\boldsymbol{m})) + \boldsymbol{m}$
und berechnet $s_t = h(D_{k_{e\,\text{Alice}}}(\boldsymbol{m}))$
(mit dem öffentlichen **Ver**schlüsselungsschlüssel von Alice)
Wenn $s = s_t$ dann ist alles o.k.

## Vorteil - Methode 2

Nachrichtenlänge erhöht sich nur geringfügig



## Versand - Methode 2

Person A (Alice) signiert den Extrakt der Nachricht $\boldsymbol{m}$ mit dem (geheimen!) Schlüssel $k_{d\,\mathrm{Alice}}$:
$s = h(E_{k_{d\,\mathrm{Alice}}}(\boldsymbol{m}))$
und sendet die Nachricht $\boldsymbol{m}$ als $s + \boldsymbol{m}$

## Überprüfung - Methode 2

Person B (Bob) erhält $s + \boldsymbol{m} = h(E_{k_{d\,\mathrm{Alice}}}(\boldsymbol{m})) + \boldsymbol{m}$
und berechnet $s_t = h(D_{k_{e\,\mathrm{Alice}}}(\boldsymbol{m}))$
(mit dem öffentlichen **Ver**schlüsselungsschlüssel von Alice)
Wenn $s = s_t$ dann ist alles o.k.

## Vorteil - Methode 2

Nachrichtenlänge erhöht sich nur geringfügig



## Versand - Methode 2

Person A (Alice) signiert den Extrakt der Nachricht $\boldsymbol{m}$ mit dem (geheimen!) Schlüssel $k_{d\,\mathrm{Alice}}$:
$s = h(E_{k_{d\,\mathrm{Alice}}}(\boldsymbol{m}))$
und sendet die Nachricht $\boldsymbol{m}$ als $s + \boldsymbol{m}$

## Überprüfung - Methode 2

Person B (Bob) erhält $s + \boldsymbol{m} = h(E_{k_{d\,\mathrm{Alice}}}(\boldsymbol{m})) + \boldsymbol{m}$
und berechnet $s_t = h(D_{k_{e\,\mathrm{Alice}}}(\boldsymbol{m}))$
(mit dem öffentlichen **Ver**schlüsselungsschlüssel von Alice)
Wenn $s = s_t$ dann ist alles o.k.

## Vorteil - Methode 2

Nachrichtenlänge erhöht sich nur geringfügig

# Hash-Funktion

Einfaches Beispiel:

$\boldsymbol{m} = \boldsymbol{m}_1\boldsymbol{m}_2 \ldots \boldsymbol{m}_n$

$h(\boldsymbol{m}) = \boldsymbol{m}_1 \otimes \boldsymbol{m}_2 \otimes \ldots \otimes \boldsymbol{m}_n$

Mehr dazu in später.



# Hash-Funktion

Einfaches Beispiel:

$\boldsymbol{m} = \boldsymbol{m}_1\boldsymbol{m}_2 \ldots \boldsymbol{m}_n$

$h(\boldsymbol{m}) = \boldsymbol{m}_1 \otimes \boldsymbol{m}_2 \otimes \ldots \otimes \boldsymbol{m}_n$

Mehr dazu in später.



# Hash-Funktion

Einfaches Beispiel:

$\boldsymbol{m} = \boldsymbol{m}_1\boldsymbol{m}_2 \ldots \boldsymbol{m}_n$

$h(\boldsymbol{m}) = \boldsymbol{m}_1 \otimes \boldsymbol{m}_2 \otimes \ldots \otimes \boldsymbol{m}_n$

Mehr dazu in später.

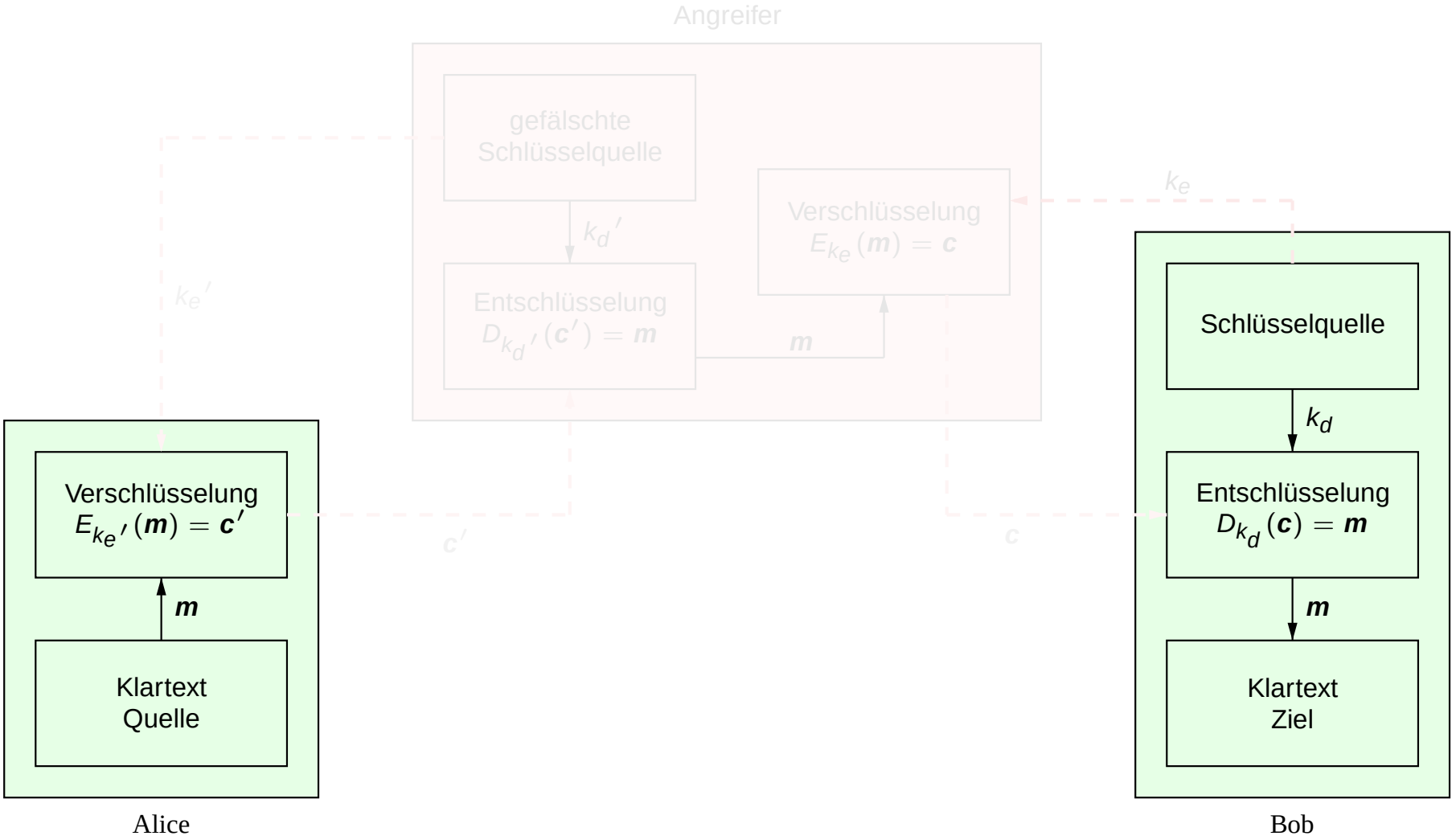
# Verschlüsselte und signierte Nachricht

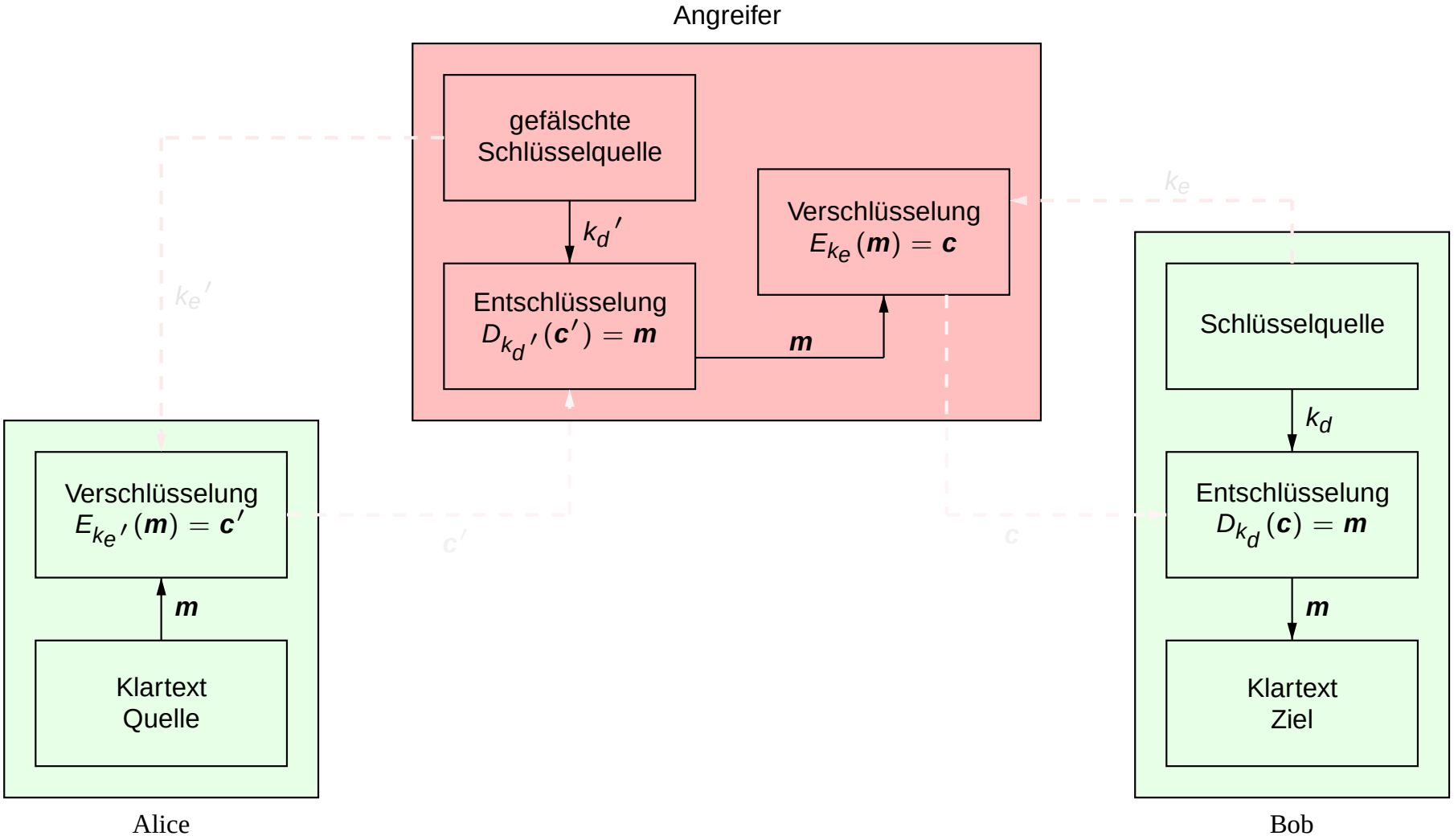
Die Verfahren lassen sich kombinieren (Alice$\rightarrow$ Bob):

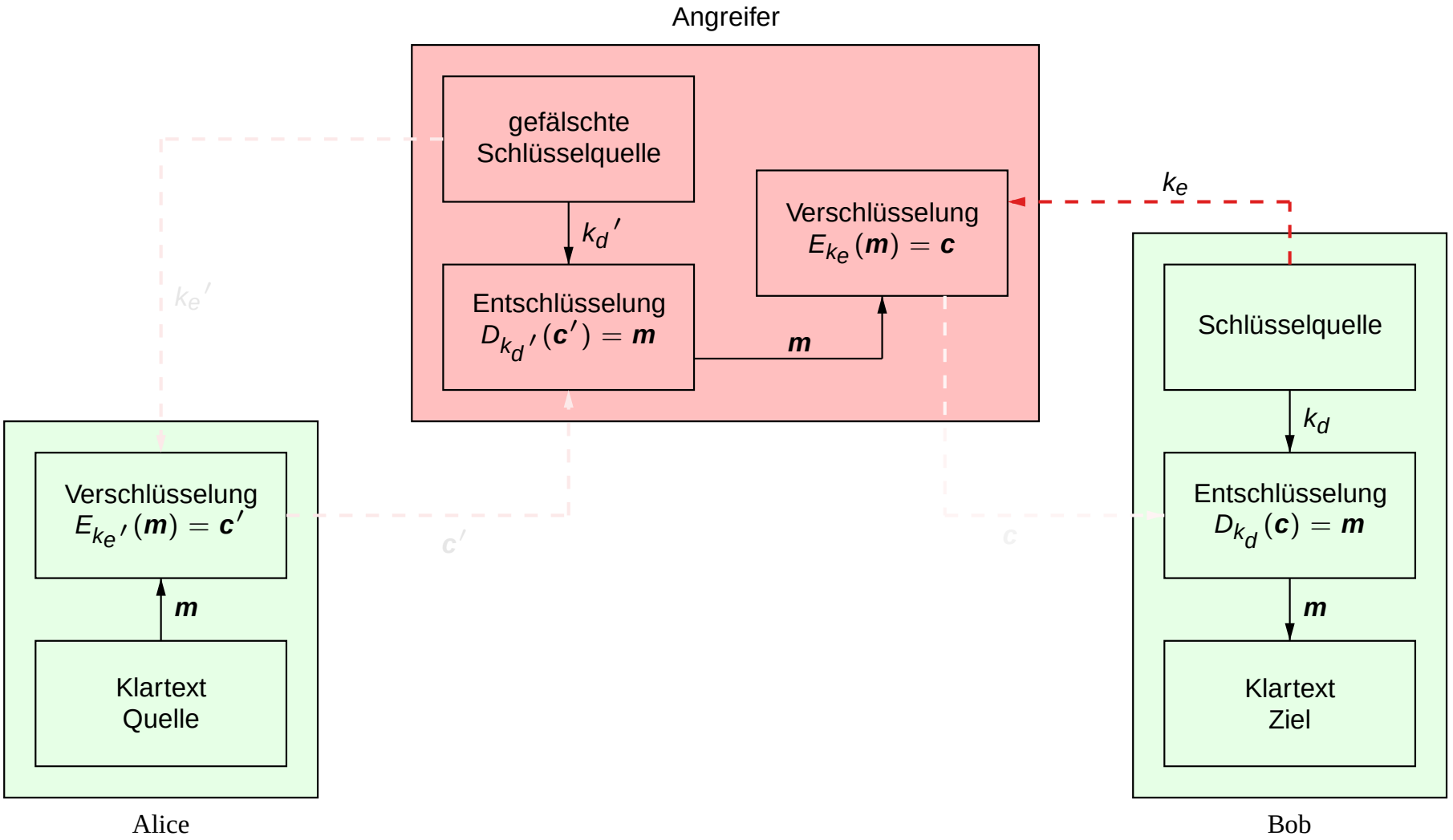
- Gegeben Nachricht $\boldsymbol{m}$
- Verschlüsselte Nachricht $\boldsymbol{c} = E_{ke\,\mathrm{Bob}}(\boldsymbol{m})$
- Signatur der verschlüsselten Nachricht $s = h(E_{kd\,\mathrm{Alice}}(\boldsymbol{c}))$
- Signierte, verschlüsselten Nachricht $\boldsymbol{c} + s$

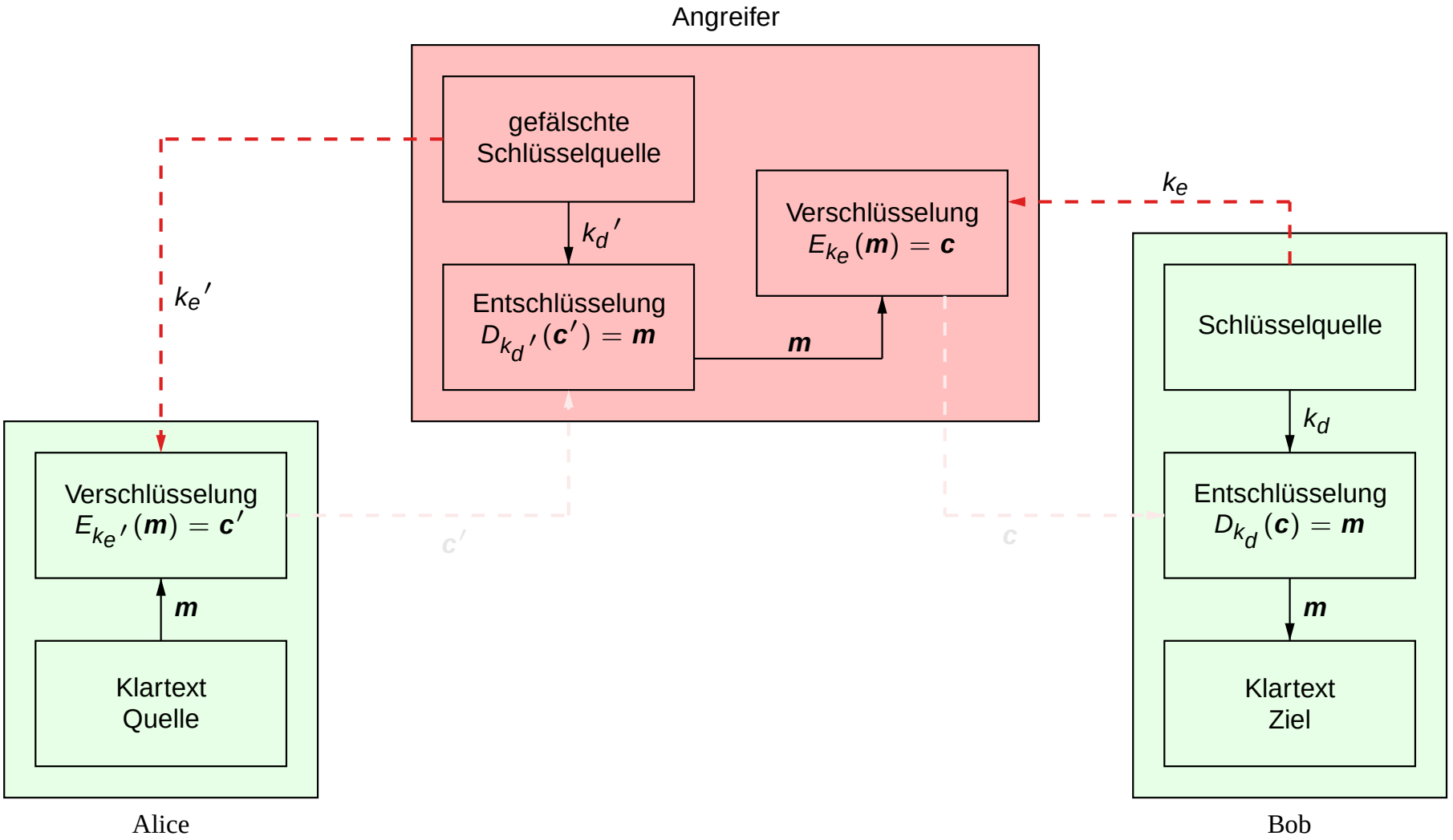
# Angriffe

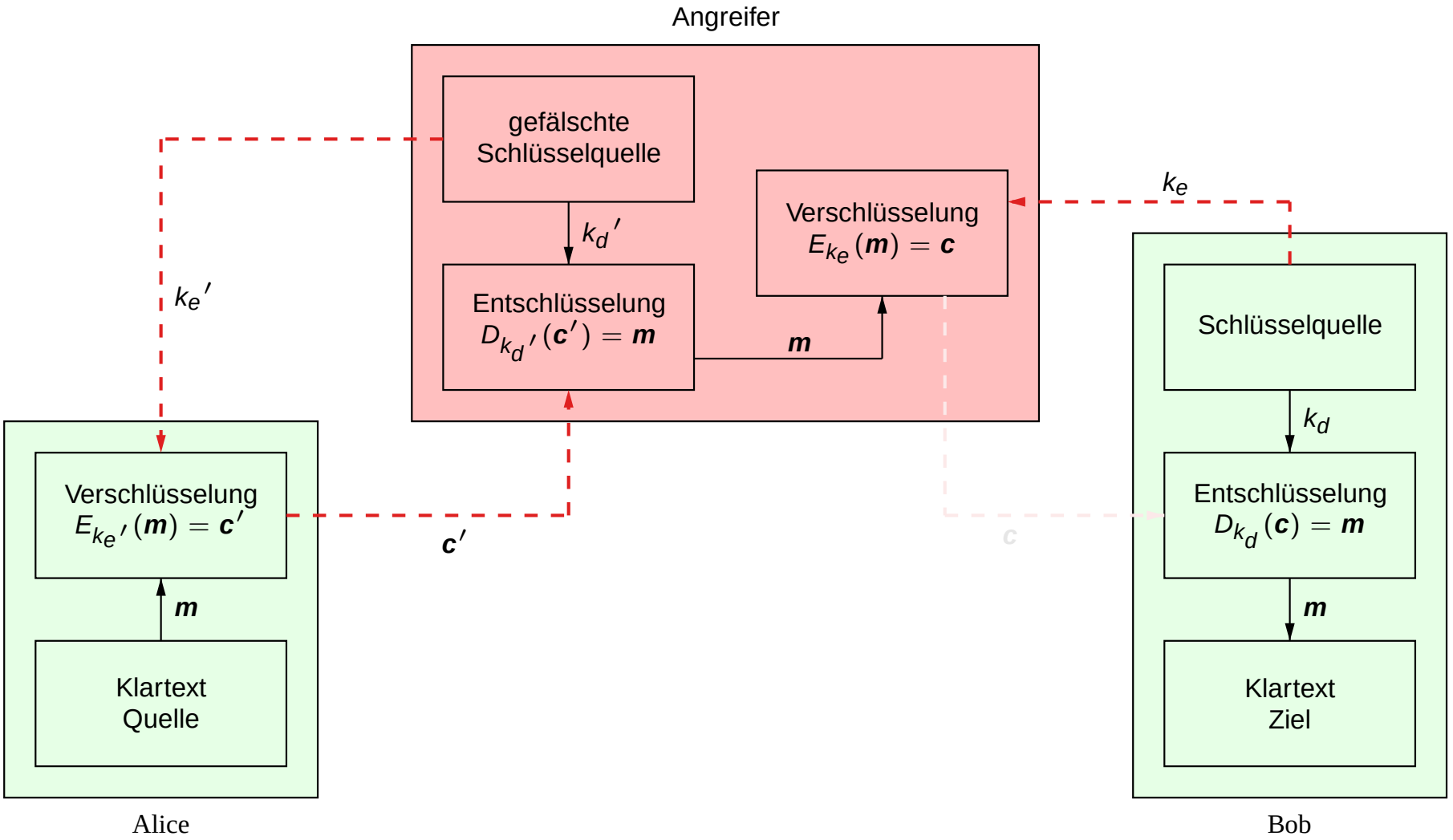
- Brute Force
- Lexikon
- Man in the middle
- ...

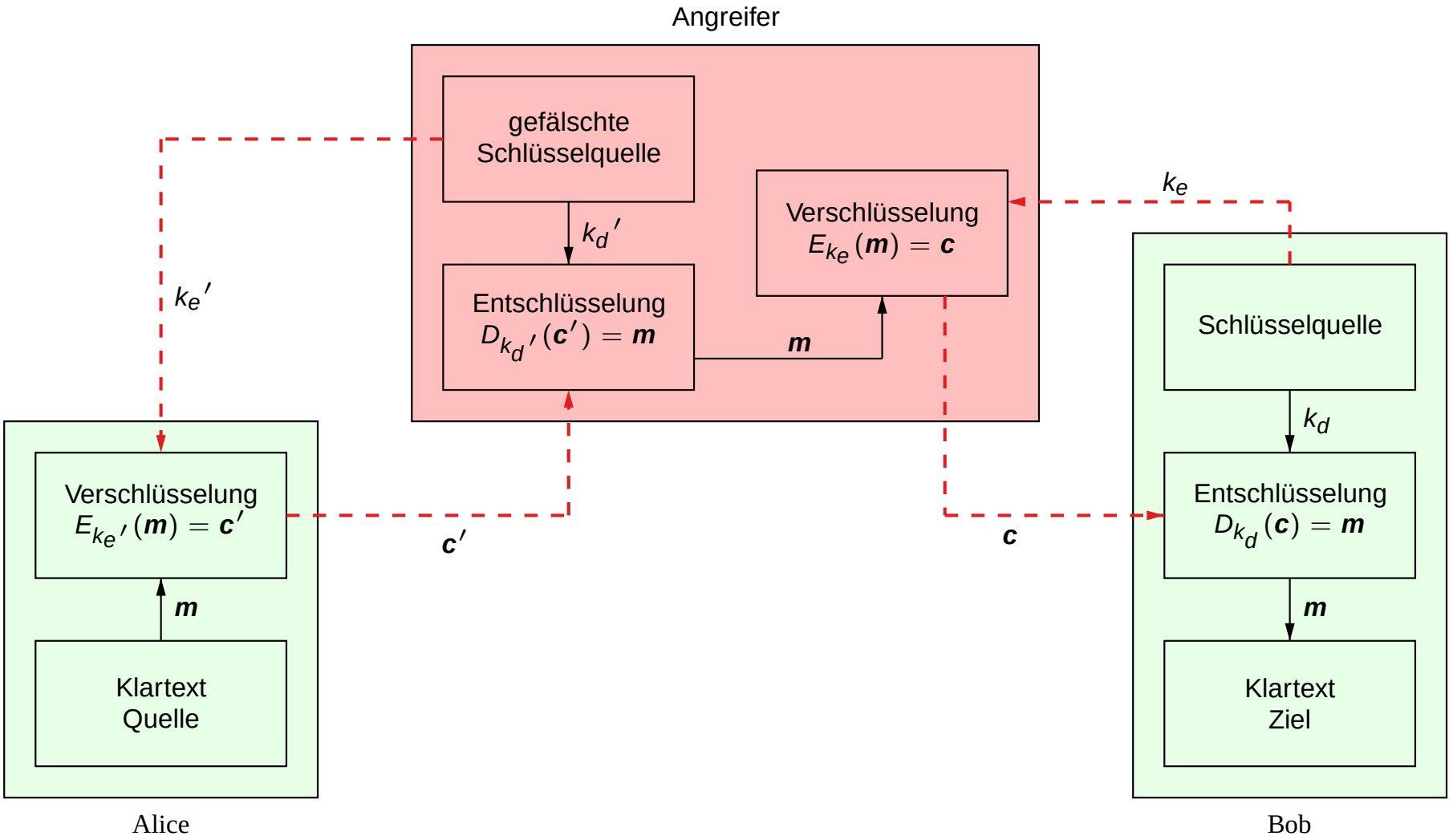

Angreifer
gefälschte Schlüsselquelle
$k_d{}'$
Entschlüsselung $D_{k_d{}'}(c') = m$
Verschlüsselung $E_{k_e}(m) = c$
$k_e$
$k_e{}'$
$m$
$c'$
$c$
Verschlüsselung $E_{k_e{}'}(m) = c'$
$m$
Klartext Quelle
Alice
Schlüsselquelle
$k_d$
Entschlüsselung $D_{k_d}(c) = m$
$m$
Klartext Ziel
Bob


Angreifer
gefälschte Schlüsselquelle
$k_d{}'$
Entschlüsselung $D_{k_d{}'}(\boldsymbol{c}') = \boldsymbol{m}$
Verschlüsselung $E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{m}$
$k_e$
$k_e{}'$
$\boldsymbol{c}'$
$\boldsymbol{c}$
Verschlüsselung $E_{k_e{}'}(\boldsymbol{m}) = \boldsymbol{c}'$
$\boldsymbol{m}$
Klartext Quelle
Alice
Schlüsselquelle
$k_d$
Entschlüsselung $D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
Klartext Ziel
Bob


Angreifer
gefälschte Schlüsselquelle
$k_d{}'$
Entschlüsselung $D_{k_d{}'}(\boldsymbol{c}') = \boldsymbol{m}$
Verschlüsselung $E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{m}$
$k_e$
$k_e{}'$
$\boldsymbol{c}'$
$\boldsymbol{c}$
Verschlüsselung $E_{k_e{}'}(\boldsymbol{m}) = \boldsymbol{c}'$
$\boldsymbol{m}$
Klartext Quelle
Alice
Schlüsselquelle
$k_d$
Entschlüsselung $D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
Klartext Ziel
Bob


Angreifer
gefälschte Schlüsselquelle
$k_d{}'$
Entschlüsselung $D_{k_d{}'}(\boldsymbol{c}') = \boldsymbol{m}$
Verschlüsselung $E_{k_e}(\boldsymbol{m}) = \boldsymbol{c}$
$\boldsymbol{m}$
$k_e$
$k_e{}'$
Verschlüsselung $E_{k_e{}'}(\boldsymbol{m}) = \boldsymbol{c}'$
$\boldsymbol{m}$
Klartext Quelle
Alice
$\boldsymbol{c}'$
$\boldsymbol{c}$
Schlüsselquelle
$k_d$
Entschlüsselung $D_{k_d}(\boldsymbol{c}) = \boldsymbol{m}$
$\boldsymbol{m}$
Klartext Ziel
Bob


Angreifer
gefälschte Schlüsselquelle
$k_d{}'$
Entschlüsselung $D_{k_d{}'}(c') = m$
Verschlüsselung $E_{k_e}(m) = c$
$m$
$k_e$
$k_e{}'$
Verschlüsselung $E_{k_e{}'}(m) = c'$
$c'$
$m$
Klartext Quelle
Alice
Schlüsselquelle
$k_d$
Entschlüsselung $D_{k_d}(c) = m$
$m$
Klartext Ziel
Bob


Angreifer
gefälschte Schlüsselquelle
$k_d{}'$
Entschlüsselung $D_{k_d{}'}(c') = m$
Verschlüsselung $E_{k_e}(m) = c$
$m$
$k_e$
$k_e{}'$
Verschlüsselung $E_{k_e{}'}(m) = c'$
$m$
Klartext Quelle
$c'$
$c$
Schlüsselquelle
$k_d$
Entschlüsselung $D_{k_d}(c) = m$
$m$
Klartext Ziel
Alice
Bob



Alice möchte Chris versichern, dass $k_{d\,\mathrm{Bob}}$ tatsächlich von Bob stammt.

Alice signiert $k_{d\,\mathrm{Bob}}$: $s = E_{k_{d\,\mathrm{Alice}}}(k_{d\,\mathrm{Bob}})$
und veröffentlicht $s$ im Netz.
Wer den Schlüssel von Alice hat und ihm vertraut, kann nun auch Bob vertrauen



Alice möchte Chris versichern, dass $k_{d\,\mathrm{Bob}}$ tatsächlich von Bob stammt.

Alice signiert $k_{d\,\mathrm{Bob}}$: $s = E_{k_{d\,\mathrm{Alice}}}(k_{d\,\mathrm{Bob}})$

und veröffentlicht $s$ im Netz.

Wer den Schlüssel von Alice hat und ihm vertraut, kann nun auch Bob vertrauen



Alice möchte Chris versichern, dass $k_{d\,\mathrm{Bob}}$ tatsächlich von Bob stammt.
Alice signiert $k_{d\,\mathrm{Bob}}$: $s = E_{k_{d\,\mathrm{Alice}}}(k_{d\,\mathrm{Bob}})$
und veröffentlicht $s$ im Netz.
Wer den Schlüssel von Alice hat und ihm vertraut, kann nun auch Bob vertrauen



Alice möchte Chris versichern, dass $k_{d\,\mathrm{Bob}}$ tatsächlich von Bob stammt.
Alice signiert $k_{d\,\mathrm{Bob}}$: $s = E_{k_{d\,\mathrm{Alice}}}(k_{d\,\mathrm{Bob}})$
und veröffentlicht $s$ im Netz.
Wer den Schlüssel von Alice hat und ihm vertraut, kann nun auch Bob vertrauen

ECKERT, CLAUDIA: *IT-Sicherheit: Konzepte - Verfahren - Protokolle*.

Oldenbourg Verlag, 4. überarbeitete Auflage, 4 2006.

MENEZES, ALFRED J., SCOTT A. VANSTONE und PAUL C. VAN OORSCHOT: *Handbook of Applied Cryptography*.

CRC Press, Inc., Boca Raton, FL, USA, 1996.

SCHNEIER, BRUCE: *Applied Cryptography: Protocols, Algorithms, and Source Code in C*.

John Wiley & Sons, Inc., New York, NY, USA, 1993.

SCHMEH, KLAUS: *Kryptografie und Public-Key-Infrastrukturen im Internet*.

dpunkt.verlag GmbH, 2. aktualisierte und erweiterte Auflage, 6 2001.

SCHNEIER, BRUCE: *Beyond Fear: Thinking Sensibly about Security in an Uncertain World*.

Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

SCHNEIER, BRUCE: *Secrets and Lies: Digital Security in a Networked World*.

John Wiley & Sons Inc., 2004.