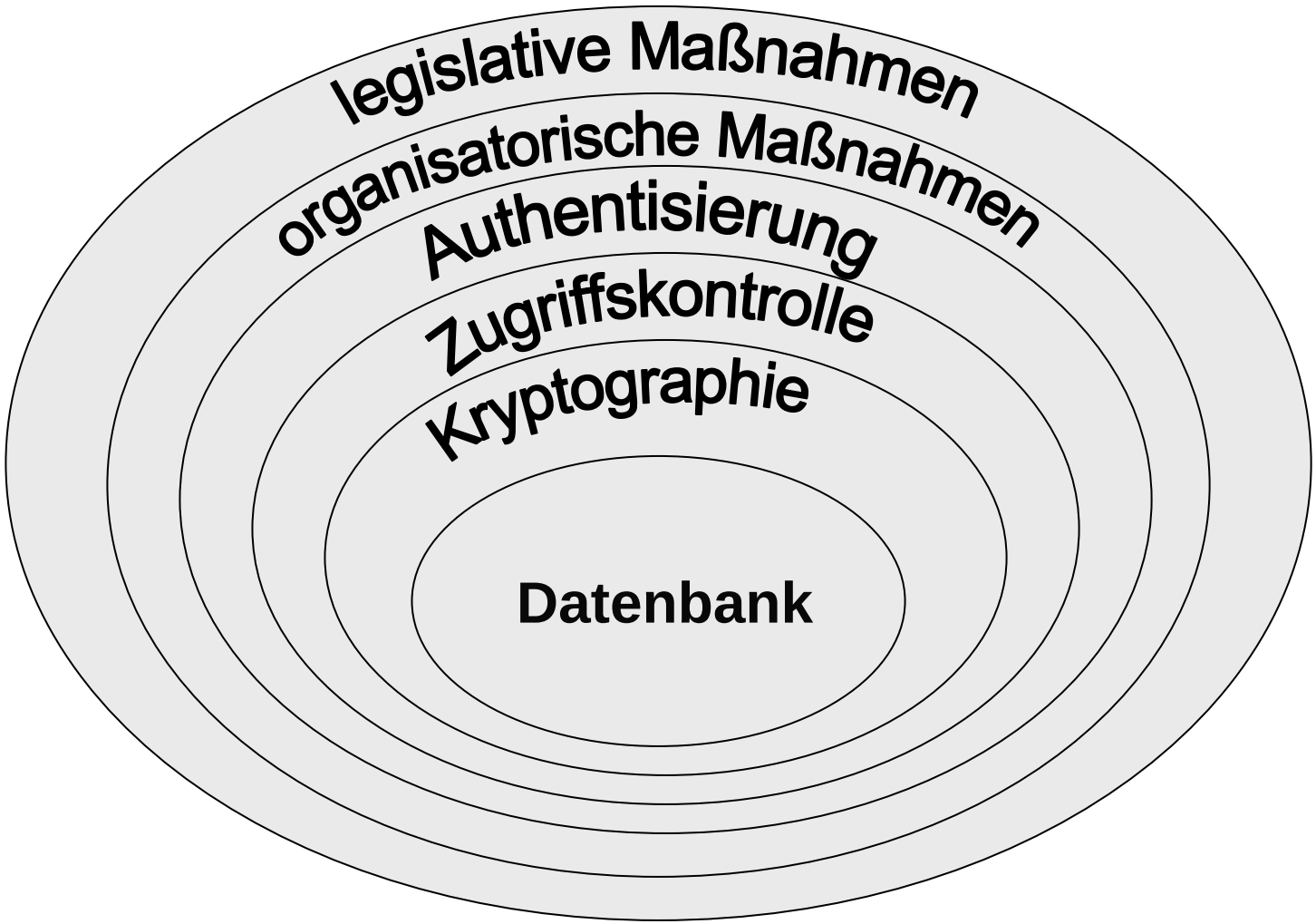
Kapitel 15:
Sicherheit

Datenschutz



Legislative Maßnahmen

Gesetz zum Schutz vor Missbrauch personenbezogener Daten

Organisatorische Maßnahmen

- bauliche Maßnahmen
- Pförtner
- Ausweiskontrolle
- Diebstahlsicherung
- Alarmanlage

Authentisierung

- Magnetkarte
- Stimmanalyse
- Fingerabdruck
- Passwort
- dynamisches Passwort

Zugriffskontrolle durch Berechtigungsmatrix

Benutzer	Ang-Nr	Gehalt	Leistung
A (Manager)	R	R	RW
B (Personalchef)	RW	RW	R
C (Lohnbüro)	R	R	-

Zugriff (*A*, Gehalt): *R*: Gehalt < 10.000
W: Gehalt < 5.000

Zugriffskontrolle durch Sichten

```
define view v(angnr, gehalt) as
select angnr, gehalt from angest
where gehalt < 3000
```

Zugriffskontrolle durch Abfrageeinschränkung

```
deny (name, gehalt) where gehalt > 3000
                          +
select gehalt from angest where name = ‘Schmidt‘
                          =
select gehalt from angest
where name = ‘Schmidt‘ and not gehalt > 3000
```

Statistische Datenbanken

Nur aggregierte Werte erlaubt !

Obacht !

```
select sum (gehalt) from angest
```

```
select sum (gehalt) from angest
where gehalt <
              (select max(gehalt) from angest)
```

Zugriffsrechte in SQL-92

```
grant { select |
        insert |
        delete |
        update |
        references |
        all }
on <relation> to <user> [with grant option]
```

***A*: grant read, insert on angest to *B* with grant option**
***B*: grant read on angest to *C* with grant option**
B*: grant insert on angest to *C

Obacht bei Recht auf Fremdschlüssel:

Probeweises Einfügen kann Schlüssel finden !

```
create table Agententest(Kennung character(3) references Agenten)
```

Revoke-Anweisung

```
revoke { select |
         insert |
         delete |
         update |
         references |
         all }
on <relation> from <user>
```

B*: revoke all on angest from *C

Entzug eines Grant

... als wenn es nie gewährt worden wäre:

```
A: grant read, insert, update on angest to D
B: grant read, update         on angest to D with grant option
D: grant read, update         on angest to E
A: revoke insert, update      on angest from D
```

D verliert nur sein insert-Recht, weil update von B erh.

E verliert keine Rechte (weil indirekt von B erhalten)

```
B: revoke update              on angest from D
```

D verliert update-Recht

E verliert update-Recht

Auditing

```
audit delete any table

noaudit delete any table

audit update on professoren
whenever not successful
```

Verschlüsselung

Caesar-Chiffre

Verschiebe Buchstaben im Alphabet

z.B. 4

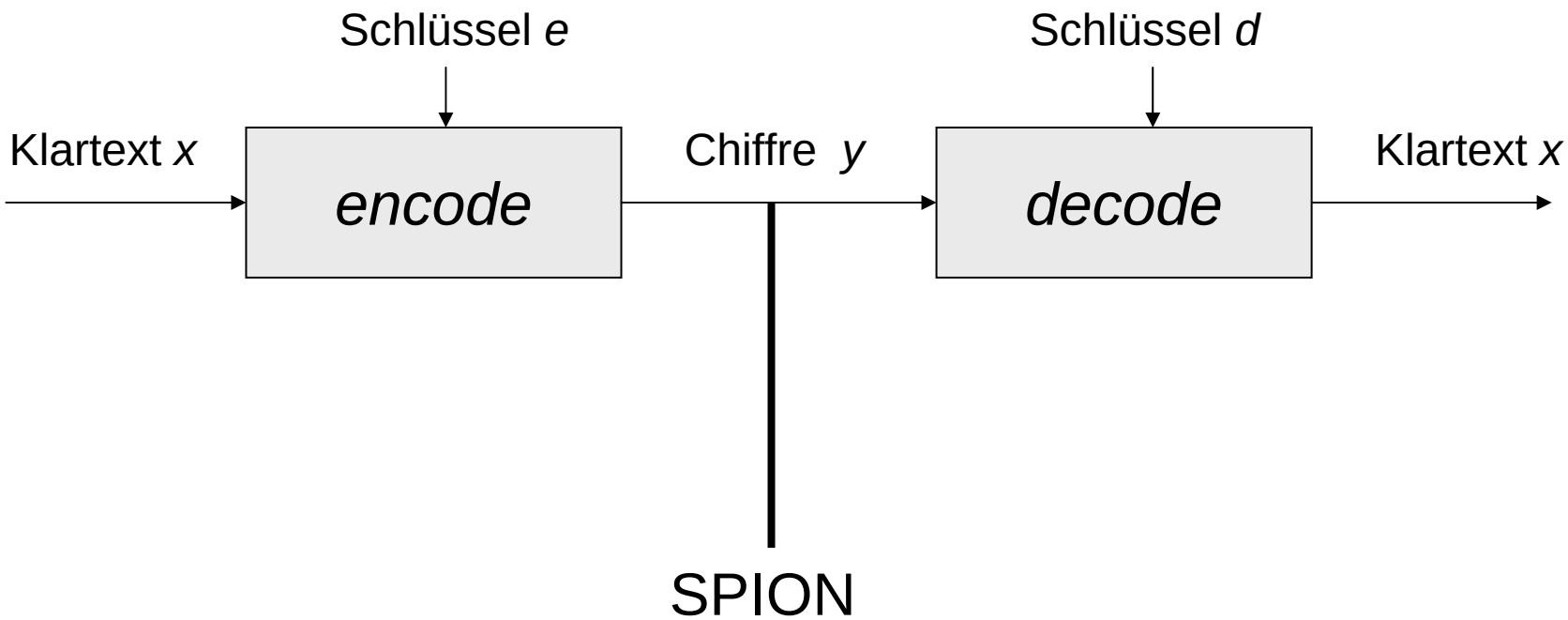
C A E S A R

G E I W E V

XOR-Verknüpfung

Klartext:	**0100101011110110**
Schlüssel:	**1101101101101110**
Chiffre:	**1001000110000000**
Schlüssel:	**1101101101101110**
Klartext:	**0100101011110110**

Ablauf

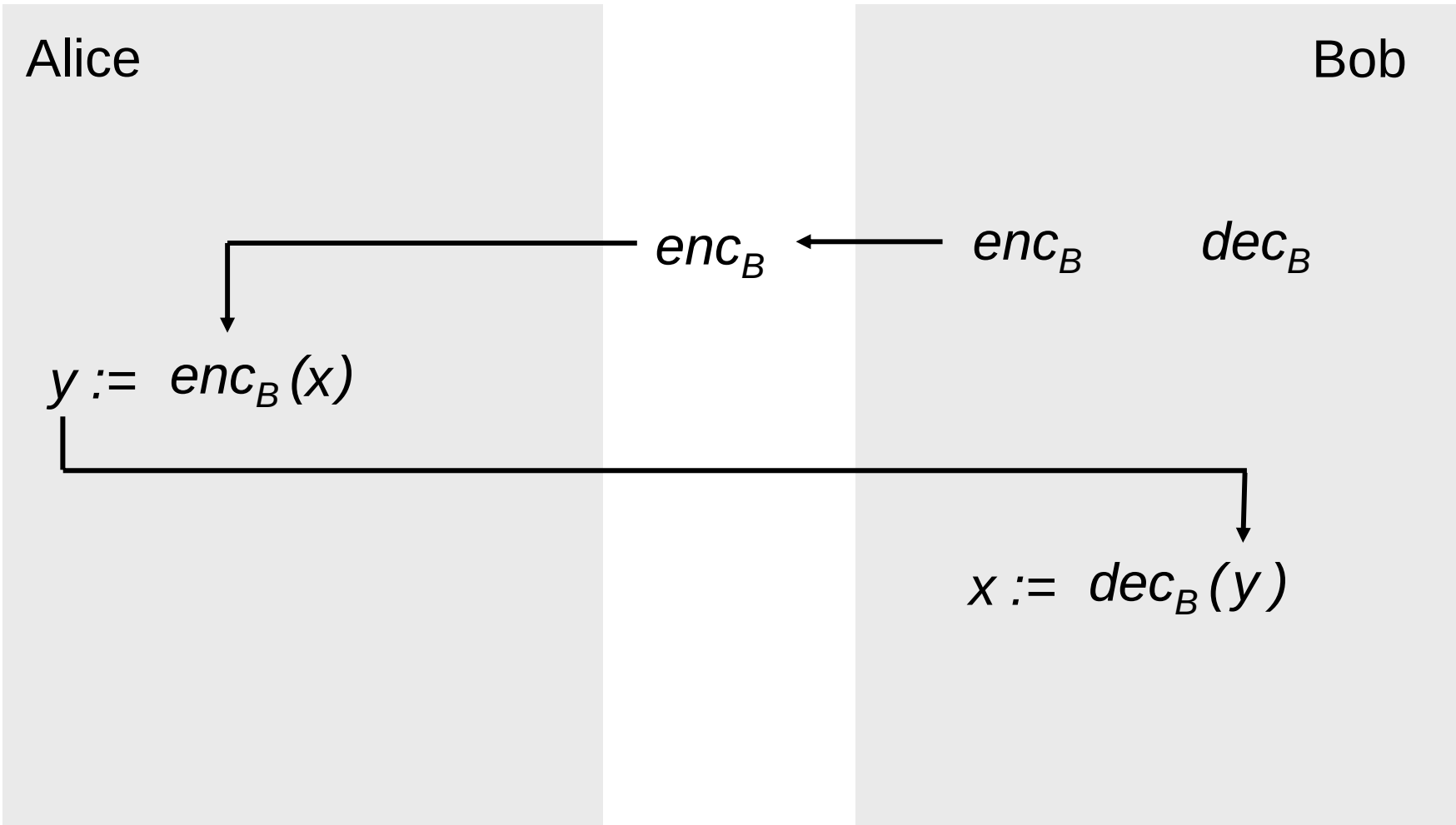

Schlüssel *e*
Schlüssel *d*
Klartext *x*
encode
Chiffre *y*
decode
Klartext *x*
SPION

Diffie & Hellman, 1976

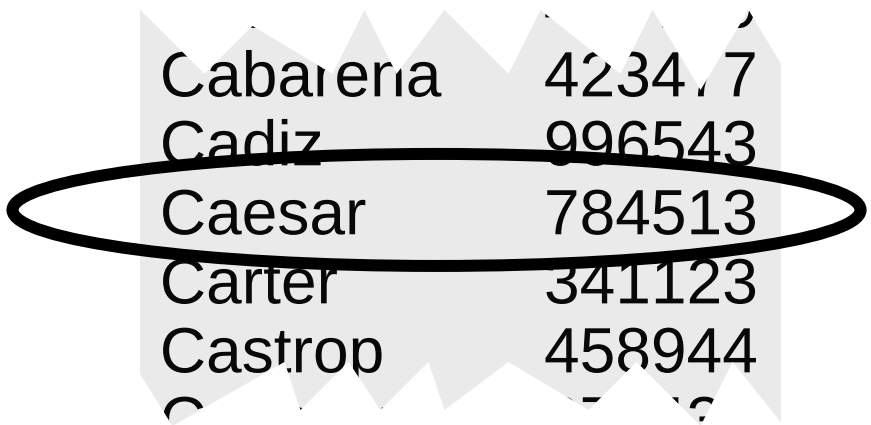
$$decode(encode(x)) = x$$

aus der Kenntnis von *encode*
läßt sich nicht *decode* ermitteln

Nachricht verschlüsseln



Einweg-Funktion



Rivest, Shamir, Adleman, 1978:

Wähle 2 Primzahlen p,q

Bestimme $n := p \cdot q$

Wähle Zahl d relativ prim zu φ (n) := (p-1) • (q-1)

Bestimme e mit $e \cdot d \equiv 1 \bmod \varphi$ (n)

$$encode(x) := x^{e} \bmod n$$

$$decode(y) := y^{d} \bmod n$$

Beispiel mit 2-stelligen Primzahlen:

$p := 11$

$q := 13$

$n := 143$

$d := 23$

$e := 47$

$encode(x) := x^{47} \bmod 143$

$decode(y) := y^{23} \bmod 143$

Beispiel mit 200-stelligen Primzahlen:

p=89909531058836627717440422858641402456684976683721255300590190073945028024188735646917834001445460697949352415335921952260900528540697407053331179637761895800615635806322855777396006499513330486337609

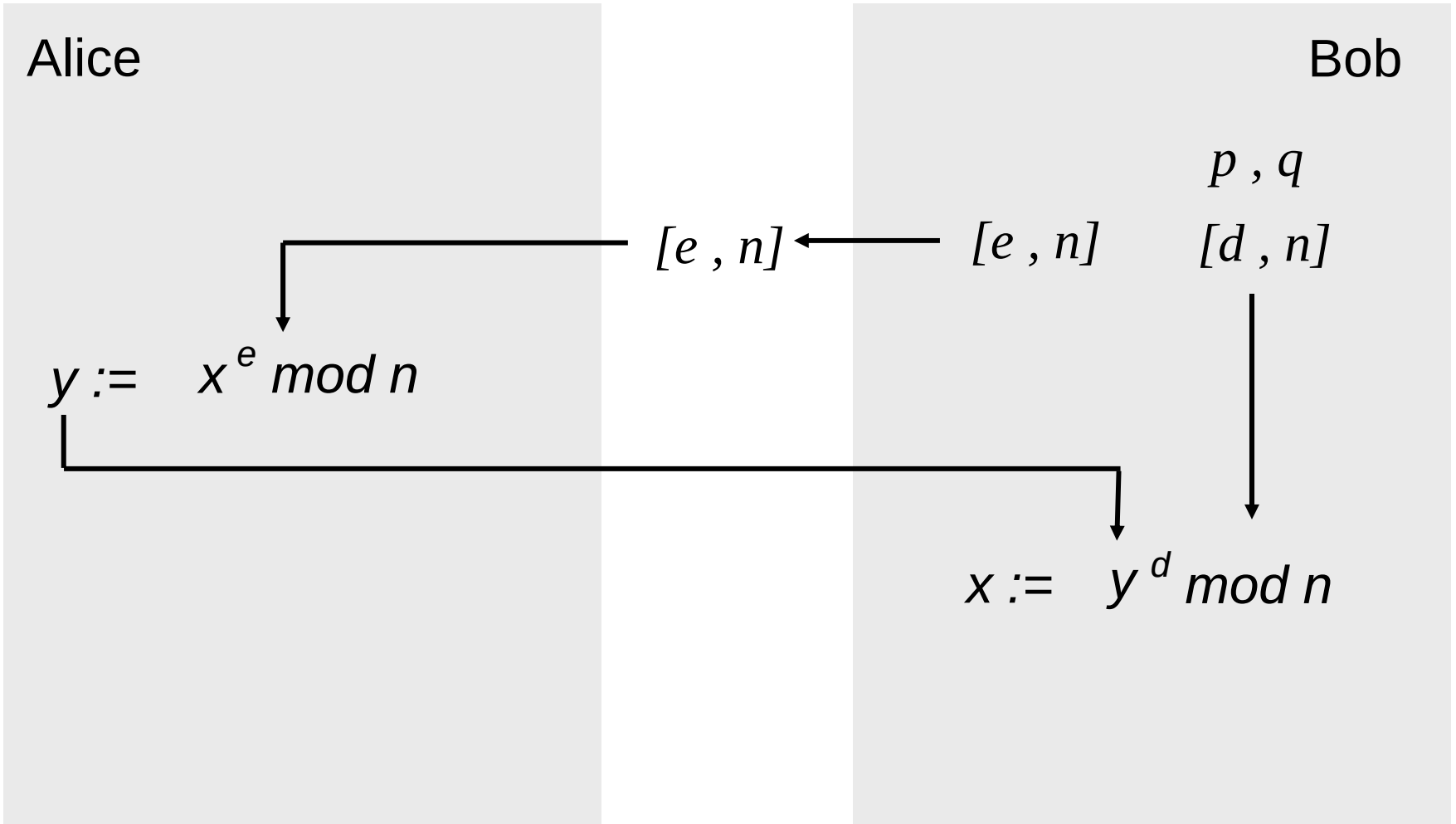
q=79902175078147244863793395126772763662312207522293560193498936718764209317443774306873658126877542468301012822975747137724559339338677640225632517923934437699039483170501603216985610255328776476846 83

n=71839670918572816475813040657559735262321716708748242340277642202639496129088670447644099480131986398124330716615728276265478293852457052897783239457461047674983301220394641493156652067377450793150290305058461964899175688693355447773536898140164511893805950504508350916992315431396618655071631147077493846714932072156438227755850165980354818200241894088517817612942193204044377518884613889031161429 47

d=69128807929831342017950975755926091290941171429517321559161131347080616192706093760902842100138662184459301019618950565163973523245851202287749007208501965412618330704286209316613491555159456129603960003081996462783443775626069899316740802382389457735204920763142529064393961291856199261761066183915948635159960663276590349868671369113560140768594069170088703418956801673314290795153429037197870521 13

e=32060956321701259905596578736460047365111500078235979150808651783912548293819432597831619829917499109195252931659284324831218397277144058830127546998961582106443297700554088899665948071647548434168832003527652121631893839546797171960967191537076338226340888927287616084554407880875931958510259177246251661227907691999195827804405610391749679780186113

Nachricht verschlüsseln



RSA-Applet

Korrektheit

Satz von Fermat/Euler:

x relativ prim zu $n \quad \Rightarrow \quad x^{\varphi(n)} \equiv 1 \bmod n$

Komplexität

$x^e \bmod n$

$$x^{13} = ((x^2 \bullet x)^2)^2 \bullet x$$

$e \cdot d \equiv 1 \bmod (p-1)\cdot(q-1)$

ggt (a,b) := ggt(b, a mod b)

(p-1)•(q-1)	d
120	19
19	6
6	1
1	0

Primzahl erkennen

$$\text{Prim} \overset{?}{\in} \text{P}$$

$$\text{Prim} \in \text{NP}$$

$$\overline{\text{Prim}} \in \text{NP}$$

$$\overline{\text{Prim}} \in \text{RP}$$

Primzahlen erzeugen

würfel ungerades x (500 Bit)

teste x, $x+2$, $x+4$, ...

mittlerer Abstand $ln(x) \Rightarrow ln(2^{500}) \approx 350$

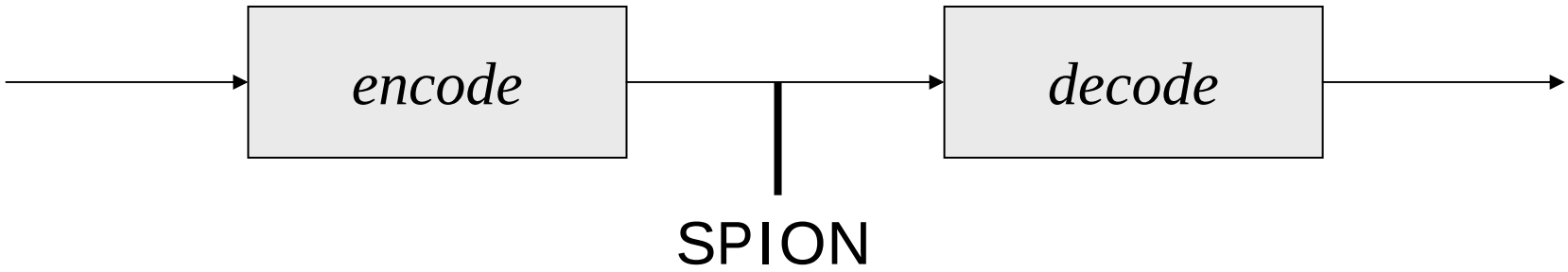
x ist Zeuge für die Zusammengesetztheit von n

n zusammengesetzt $\Rightarrow$ mehr als ¾ n Zeugen

```
z=0;
repeat
      z++;
      würfel x
until (x ist Zeuge für n) or (z=50)
```

Fehler = $¼^{50} \approx 10^{-30}$

Sicherheit



Der Spion kennt *e*, *n*, *y*

Faktorisieren von *n* würde *p* und *q* liefern.

Damit könnte er *d* ausrechnen.

Aber: Faktorisieren von *n* dauert $n^{\sqrt{\frac{\ln \ln (n)}{\ln (n)}}}$ Schritte !

$n = 2^{1000}$

$\Rightarrow n^{0.1}$ Schritte

$= 10^{30}$ Schritte = 10^{13} Jahre (bei 10^{9} Schritte/sec)

Reihenfolge vertauschen

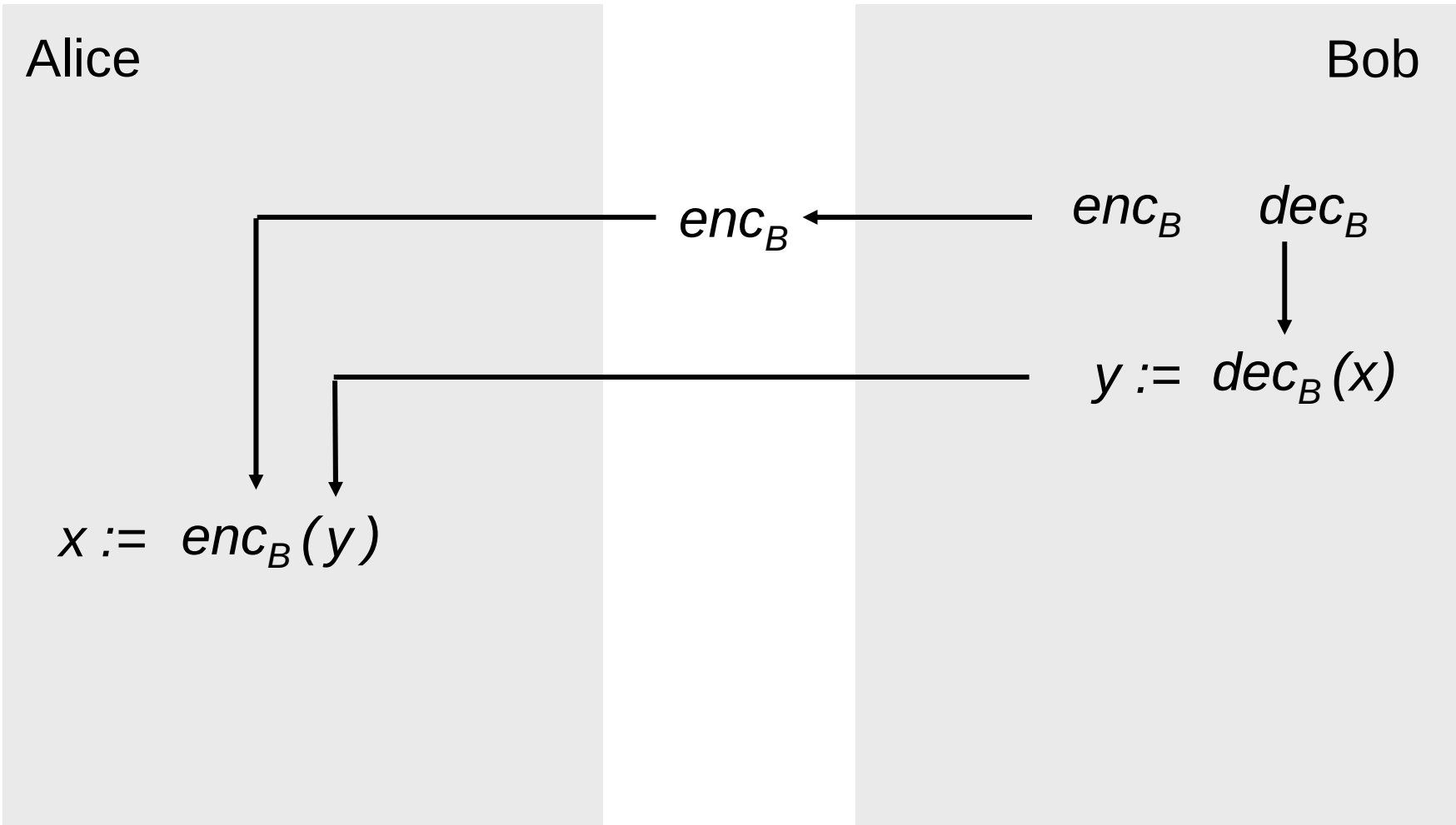
$$decode(encode(x)) = x$$

$$= x^{ed} \bmod n$$

$$= x^{de} \bmod n$$

$$= encode(decode(x))$$

Nachricht signieren



Und wenn sie nicht gestorben sind,
entschlüsseln sie noch heute !