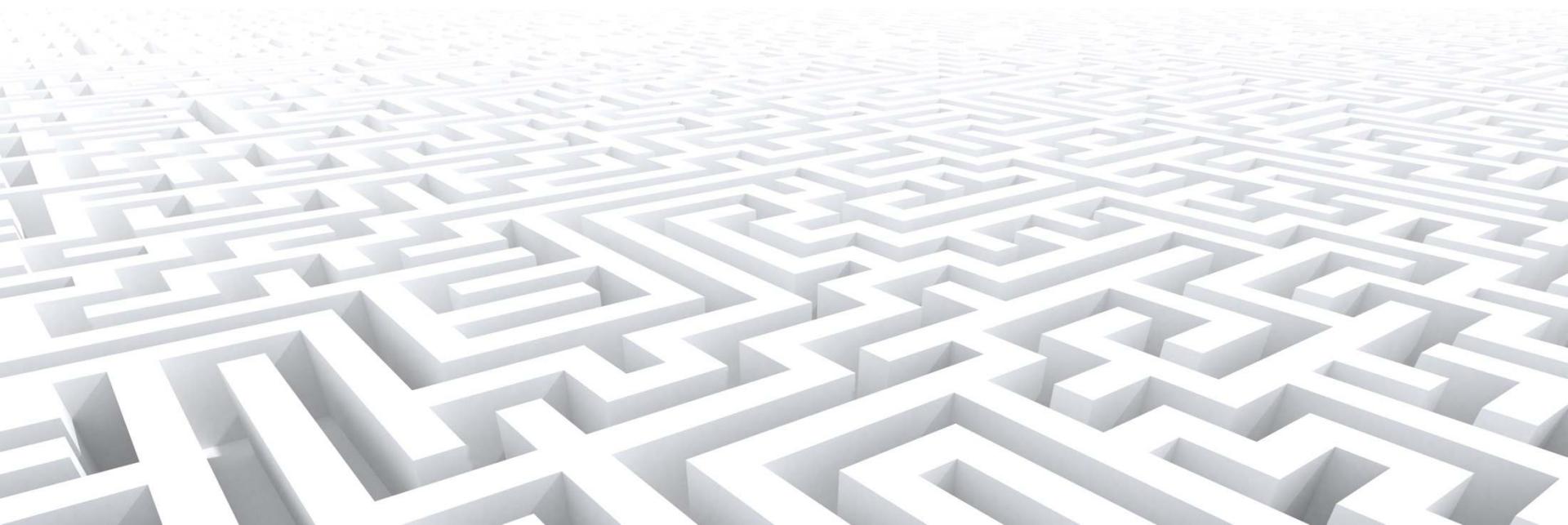


Vorlesung Security by Design (SbD)

Prof. Dr. Hannes Federrath
Sicherheit in verteilten Systemen (SVS)
<http://svs.informatik.uni-hamburg.de>

Wegweiser durch die Vorlesung

- Kryptographie: Vertiefung
- Public Key Infrastructures (PKI)
- Sniffing, Spoofing, Denial of Service,
Internet of Things and Security
- Mobile Security



Organisatorisches

Ziele der vorlesungsbegleitenden Übungen

- **Verständnisprobleme lösen**
 - Offene Fragen der Vorlesung können im kleineren Kreis geklärt werden.
- **Vertiefung und Anwendung des Vorlesungsstoffs**
 - In der Vorlesung wird teilweise nur eine Auswahl des Stoffs präsentiert.
 - Bearbeitung von Aufgaben mit Bezug zur Vorlesung
 - Bearbeitung von Aufgaben, die über die Vorlesung hinausgehen
- **Vorstellung und Diskussion der Lösungen innerhalb der Übungsgruppe**
 - Wissenslücken können nur beseitigt werden, wenn sie bekannt sind.
 - Aktive Beteiligung ist notwendig und sinnvoll.
- **Übungsstoff ist auch Klausurrelevant!**

Übungtleistung

- Übungsleistung wird Online erbracht
 - <https://lernen.min.uni-hamburg.de/course/>
 - SbDyy Security by Design
 - Einloggen mit Kennung und Passwort der Uniseiten
- Bewertung
 - Tests müssen bestanden werden 75%
 - beliebig viele Versuche bis zum Ablauf der Frist
 - bestes Ergebnis wird gewertet
 - Ergebnisse werden direkt angezeigt
- Zeitplan für die Abgaben
 - Freischaltung der Übung am Montag Mittag (vorher nicht sichtbar)
 - Bearbeitungszeit je Blatt verschieden (typ. 1 oder 2 Wochen)
 - Abgabefrist auf Blatt angegeben (Montag Mittag)

yy steht für das Jahr
(zweistellig).

Übungen

- Durchführung
 - Hintergründe zu den Übungsaufgaben werden erklärt
 - Aufgaben (Pflicht und optional) werden erläutert und besprochen
 - Gelegenheit zum Stellen von Fragen

Übungen zu machen hilft dabei, den Vorlesungsstoff besser zu verstehen.

- Es gibt typischerweise 5 Übungstermine bzw. -gruppen.
 - Aktuelle Termine siehe Stine:
 - Montag 10-12 Uhr in F-635 und G-102
 - Dienstag 14-16 Uhr in F-334 und F-009
 - Dienstag 16-18 Uhr in F-009

Klausur

- **Formaila**
 - 60 Minuten Dauer
 - Mit Unterlagen (»open book«)
 - Nicht-programmierbarer Taschenrechner erlaubt
- **Inhalte**
 - Inhalte der Vorlesung
 - Inhalte der Übung
- **Klausurtermine zu finden unter**
 - <https://www.inf.uni-hamburg.de/studies/orga/dates.html>
- Bitte prüfen Sie bei Bedarf die Barrierefreiheit des Zugangs zu den Räumen und nehmen frühzeitig Kontakt zu uns auf, sollten Sie hierbei Probleme feststellen.

Kommunikation zur Vorlesung und Übung

- Infos wo?
 - Ankündigungen, Verschiebungen in Stine und/oder im Moodle
 - Vorlesungsfolien in Stine
 - Übungsaufgaben im Moodle
- Fragen?
 - zum Übungsbetrieb: Übungsleiter des Vertrauens fragen
 - zur Vorlesung: Prof. Dr. Hannes Federrath fragen
 - gerne auch per E-Mail
- Alle zur Vorlesung und Übung angemeldet?
 - Wenn nicht: umgehend nachholen!

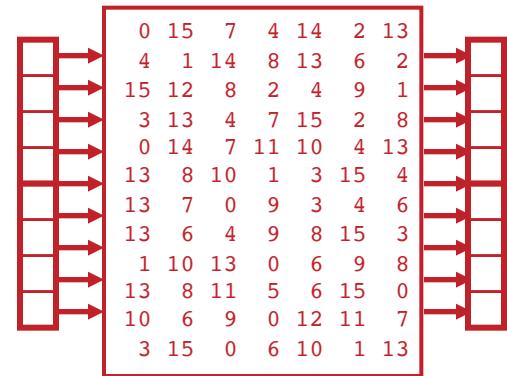
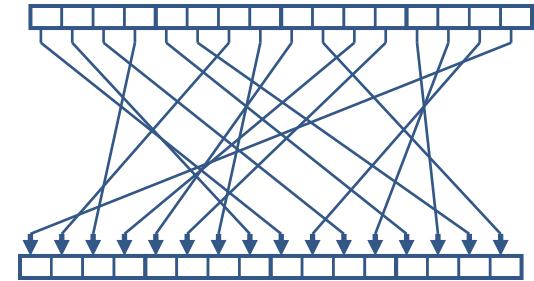
Kryptographie: Vertiefung

Definition Konzelationssystem

- Seien
 - $n, l \in \mathbb{N}$
 - A und B Alphabete und K eine endliche Menge
 - Klartext: $m \in A^n = M$ Schlüsseltext: $c \in B^l = C$
- Dann ist eine Kryptofunktion eine Abbildung
 - $e: A^n \times K \rightarrow B^l$
derart, dass die Abbildung
 - $e_k: A^n \rightarrow B^l$
definiert durch
 - $e_k(m) := e(m, k)$
 - für alle $k \in K$ injektiv ist.
 - \bullet injektiv: f^{-1} ist rechtseindeutig
 - \bullet rechtseindeutig (auch: partiell):
 $\forall x. \forall y. y': ((f(x)=y \wedge f(x)=y') \rightarrow (y=y'))$
- Für jedes e_k existiert eine Umkehrfunktion $d_{k'}$.
- Es gelten
 - $c = e_k(m)$ und $m = d_{k'}(c)$, d.h. $m = d_{k'}(e_k(m))$ oder
 - $c = e(m, k)$ und $m = d(c, k')$, d.h. $m = d(e(m, k), k')$

Klassische Chiffren: Systematik

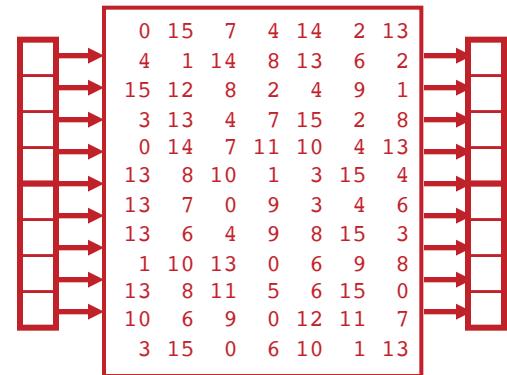
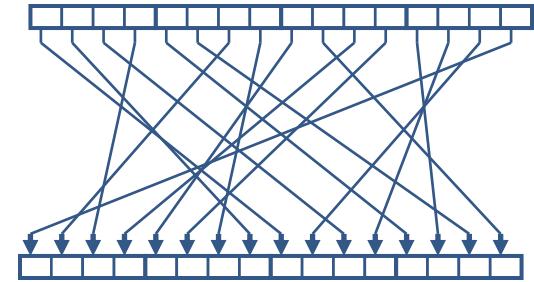
- **Transpositionschiffren**
 - Veränderung (Permutation) der Anordnung von Schriftzeichen
- **Substitutionschiffren**
 - Systematische Ersetzung von Schriftzeichen
- **Produktchiffren**
 - Kombination von Transpositionen und Substitutionen
 - Vorläufer der modernen symm. Kryptographie, bei denen Permutationen und Substitutionen (meist) iterativ angewendet werden



Klassische Chiffren: Konkrete Systeme

- Transpositionschiffren
 - Spalten-Transpositionen
 - freie Permutationen

- Substitutionschiffren
 - Schema von Polybios
 - Caesar-Chiffre
 - Vigenere-Chiffre
 - Vernam-Chiffre



■ Skytala

- ca. 2400 Jahre alte griechische Chiffre
- Zylinder mit gewickeltem Papierstreifen, schreiben, abwickeln
- Empfänger hat Zylinder mit gleichem Durchmesser



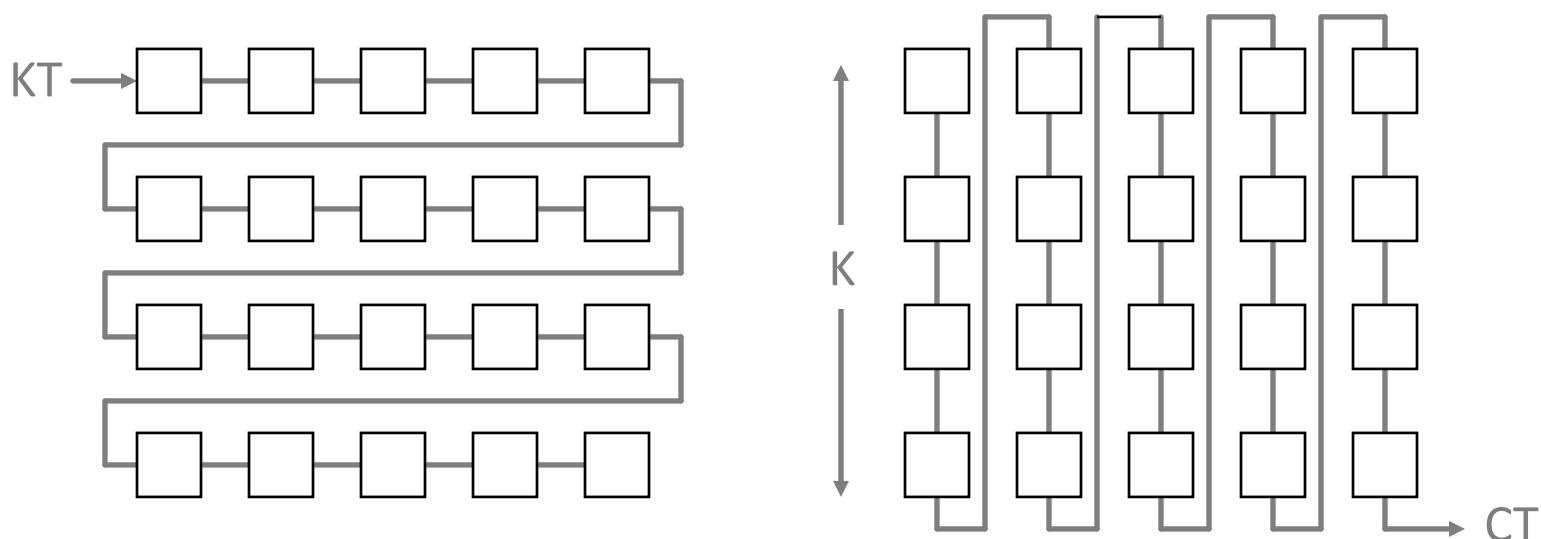
Bild: Wikipedia (CC BY-SA 3.0)

■ Kryptanalyse

- Heute: Durchprobieren (sehr kleiner Schlüsselraum)
- Statistische Analyse (Bigramme)

■ Skytala

- ca. 2400 Jahre alte griechische Chiffre
- Zylinder mit gewickeltem Papierstreifen, schreiben, abwickeln
- Empfänger hat Zylinder mit gleichem Durchmesser

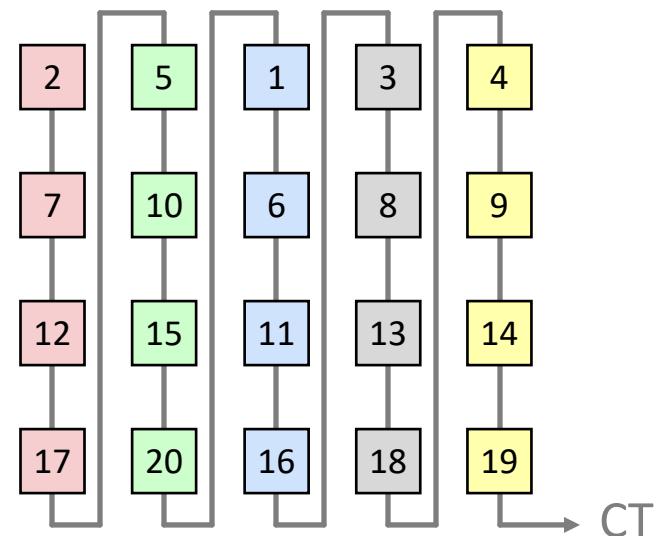
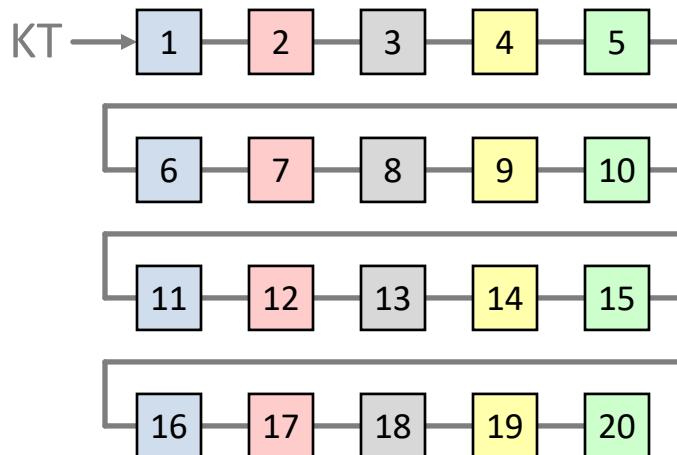


■ Kryptanalyse

- Heute: Durchprobieren (sehr kleiner Schlüsselraum)
- Statistische Analyse (Bigramme)

- Variante

- zusätzlich die Spalten transponieren, d.h. deren Reihenfolge vertauschen bzw. permutieren



- Übungsaufgabe

- Um welchen Faktor vergrößert sich der Schlüsselraum bei s Spalten?

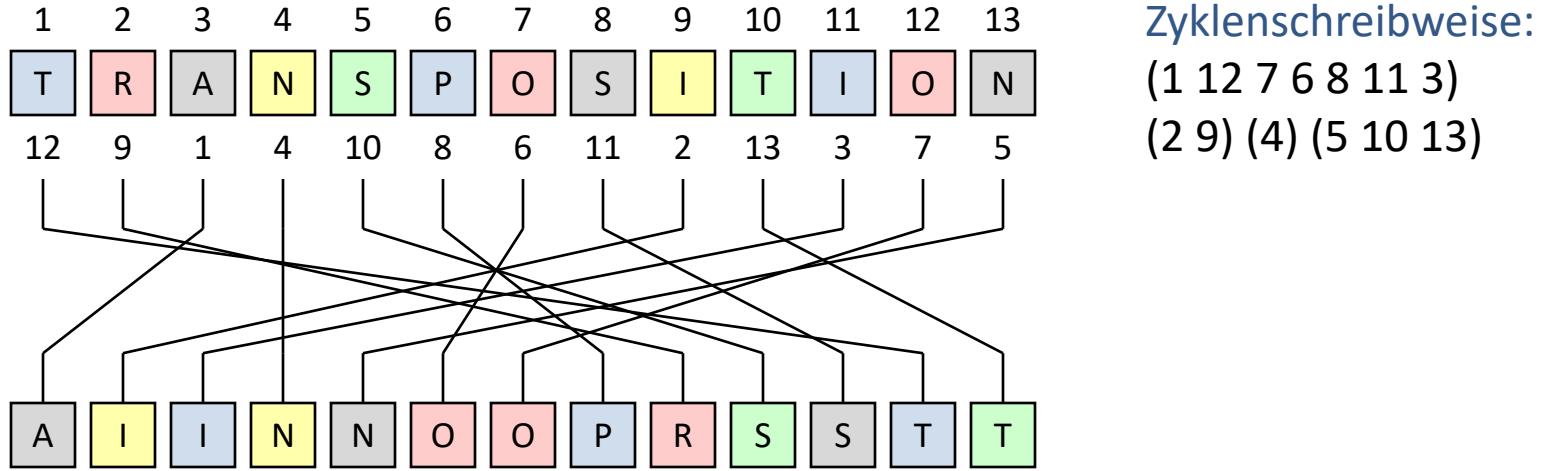
Skytala: Statistische Kryptanalyse

- **Vorgehen**
 - Suche typ. Bigramme (z.B. EN, ER, CH, ...) und ermittle die Häufigkeit der Buchstabenabstände. Beispiel:
- **Beispiel**
 - Klartext: VERSCHLUESSELNMACHTGROSSENSPASS
 - k=5 **VERSCHL**
 UESSELN
 MACHTGR
 OSSENSP
 ASSXYZX
 - Chiffretext: VUMOAEEASSRSCSSSSHEXCETNYHLGSZLNRPX

Bigramm	Chiffretext	Abstand/Häufigk.
EN:	VUMOA EEASSRSCSSSSHEXCETNYHLGSZLNRPX:	2/1, 5/1, ...
ER:	VUMOA EEASSRSCSSSSHEXCETNYHLGSZLNRPX:	5/1, 4/1, ...
EI:	VUMOA EEASSRSCSSSSHEXCETNYHLGSZLNRPX:	0
CH:	VUMOA EEASSRSCSSSSHEXCETNYHLGSZLNRPX:	5/2, ...
 - Abstand 5 kommt am Häufigsten vor,
 - teste, ob Text in 5 Zeilen sinnvoll → gebrochen

Freie Permutationen

- Idee
 - Zeichen werden nach einer Vorschrift vertauscht
- Beispiel



- Übungsaufgabe
 - Schreiben Sie eine Skytala mit 4 Zeilen und 3 Spalten in Zyklenschreibweise.

Klassische Substitutionschiffren

- **Monoalphabetische Substitutionen**
 - Jedem Zeichen bzw. jeder Zeichenfolge über A ist eindeutig ein Zeichen bzw. eine Zeichenfolge über B zugeordnet.
- **Polyalphabetische Substitutionen**
 - Jedem Zeichen bzw. jeder Zeichenfolge über A ist eindeutig ein Zeichen bzw. eine Zeichenfolge über B_1, B_2, \dots, B_n zugeordnet.
- **Monographische Substitution**
 - Es werden einzelne Zeichen ersetzt.
- **Polygraphische Substitution**
 - Es werden Zeichenfolgen ersetzt.

Schema von Polybios

- Def.

- $A = \{A:Z\}$
- $B = \{(i,j) \mid i, j \in \{1:5\}\}$
- e/d:

	j →	1	2	3	4	5
i ↓		A	B	C	D	E
	1	F	G	H	I	J
	2	K	L	M	N	O
	3	P	Q	R	S	T
	4	U	V	W	X/	Z

Beispiel:

Chiffretext: 223515452315

Klartext:

- Eigenschaften

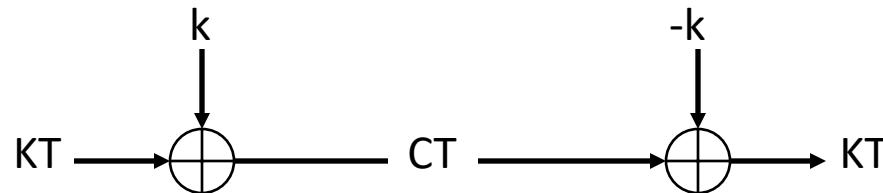
- monoalphabetische/monographische Substitution
- arbeitet »schlüssellos«

- Ableitungen

- Tabelle mit Zeichen (Freimaurer-Chiffre, Friedhofschieffre)

Verschiebechiffre

- auch: Caesar-Chiffre
 - Caesar, röm. Kaiser und Feldherr (100-44 v.Chr)
- Def.
 - $A=B=\{A:Z\}$
 - $K=\{A:Z\}$ oder allg. $K=\{0:n-1\}$ mit $n \leq \text{card}(\{A:Z\})$
 - e: $c=(m+k) \bmod n$
 - d: $m=(c-k) \bmod n$



$(x+y) \bmod 26$

Schlüssel

Klartext

A	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
B	B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
C	C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
D	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
E	E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
F	F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
G	G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
H	H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
I	I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
J	J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
K	K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
L	L M N O P Q R S T U V W X Y Z A B C D E F G H I J K
M	M N O P Q R S T U V W X Y Z A B C D E F G H I J K L
N	N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
O	O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
P	P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
Q	Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
R	R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
S	S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
T	T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U	U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V	V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
W	W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
X	X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Y	Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
Z	Z A B C D E F G H I J K L M N O P Q R S T U V W X Y

$$(x+y) \bmod 26$$

Beutelspacher, Kryptologie, 8. Aufl., S. 7



Verschiebechiffre

- auch: Caesar-Chiffre
 - Caesar, röm. Kaiser und Feldherr (100-44 v.Chr)
- Def.
 - $A=B=\{A:Z\}$
 - $K=\{A:Z\}$ oder allg. $K=\{0:n-1\}$ mit $n \leq \text{card}(\{A:Z\})$
 - e: $c=(m+k) \bmod n$
 - d: $m=(c-k) \bmod n$
- Eigenschaften
 - monoalphabetische/monographische Substitution
 - additive Chiffre
 - Buchstabenhäufigkeiten bleiben erhalten und bilden Ansatz zur Kryptanalyse

Beispiel für Verschiebechiffre – Spiegel Plus

Quelle:

<http://andreas-zeller.blogspot.de/2016/06/spiegel-online-nutzt-unsichere-casar.html>

The screenshot shows a web browser window with the URL spiegel.de. The page content is in German. A callout box highlights the encryption key **TQJFHF**.

SPIEGEL: Herr Cryan, die Briten haben für den Ausstieg aus der EU gestimmt.

Cryan **SPIEGEL** Klartext

Dszbo **TQJFHF** Schlüsseltext

ABCDEF**GHIJKL**MNOPQRSTUVWXYZ Verschiebung
BCDEF**GHIJKL**MNOPQRSTUVWXYZA um 1 Position

SPIEGEL: Wie wird die Deutsche Bank auf die Brexit-Entscheidung reagieren?

Dszbo; Ebt xjse ebwpo bciåohfo- xjf ejf Wfsiboemvohfo wfsmbvgfo/ Xjs tjoe eb tfis gmfyjcfm- eb xjs tpxpim jo Mpoepo bmt bvdi jo Gsbolgvsu tubsl wfsusfufo tjoe/ Ebevsdi xfsefo xjs gýs fvspqåjtdif Voufsofinfo vntp xjdiujhfs- hfsbef jo ejftfs Qibtf efs Votjdifsifju bo efo Lbqjubmnåslufo/

TQJFHF; Xfsefo Tjf Ufjmf eft Hftdiågut bo efo Nbjo wfsmbhfso@

Dszbo; Tpmmuf ft ubutådimjdi {v fjofn Bvtusu nfjoft Ifjnbumboeft bvt efs FV lpnnfo- eboo xjse ebt Mpoepo tdixådifo voe Gsbolgvsu tuålsfo/ Xbt ebt bcfs hfobv gýs ejf Djuz voe gýs vot ifjåu- måttu tji opdi ojdiu wpsifstbhfo/

Beispiel für Verschiebechiffre

The screenshot shows a web application for Caesar cipher encryption. The title bar reads "Cäsar Verschiebechiffre bzw. Cäsar Verschlüsselung".

Original Text:

```
Cryan; Das wird davon abhängen- wie die Verhandlungen verlaufen/ Wir sind da sehr flexibel- da wir sowohl in London als auch in Frankfurt stark vertreten sind/ Dadurch werden wir fyr europäische Unternehmen umso wichtiger- gerade in dieser Phase der Unsicherheit an den Kapitalmärkten/  
SPIEGEL; Werden Sie Teile des Geschäfts an den Main verlagern@  
Cryan; Sollte es tatsächlich {u einem Austritt meines Heimatlandes aus der EU kommen- dann wird das London
```

Methode: Cäsar Verschiebechiffre

Hilfe: A-Z,a-z werden um die gewünschte Anzahl von Positionen im Alphabet zyklisch nach rechts oder links verschoben, alle anderen Zeichen bleiben unverändert. ROT13 ist eine Sonderform der Cäsar Verschiebechiffre mit einer Verschiebung um 13 Positionen. Die Umwandlung funktioniert in beide Richtungen. Bei einem Verschiebewert von '0' werden alle Verschiebemöglichkeiten von 1-25 ausgegeben.

Verschiebung: 1

Kodiert:

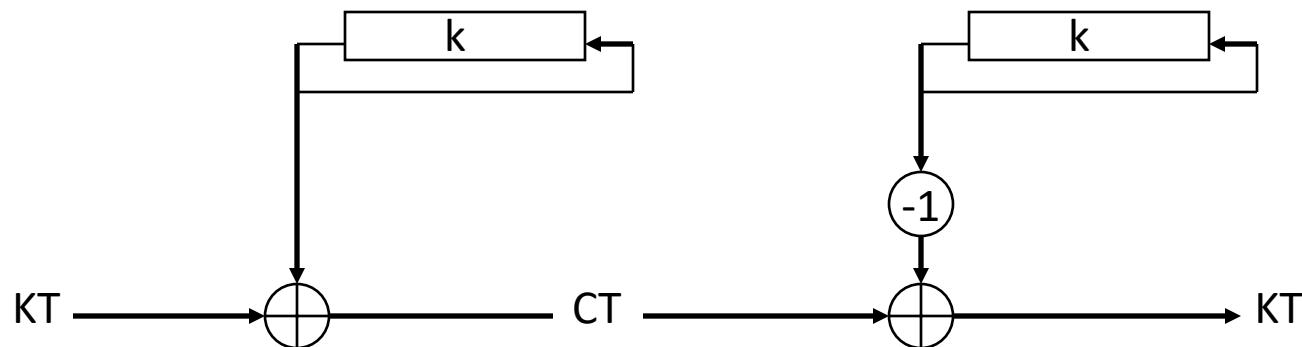
```
Dszbo; Ebt xjse ebwpo bciähohfo- xjf ejf wfsiboemvohfo wfsmbvgfo/ Xjs tjoе eb tfis gmfyjcfm- eb xjs tpxpim jo Mpoepo bmt bvdi jo Gsbolgvsu tubsl wfsusfufo tjoе/ Ebevsdi xfsefo xjs gýs fvspqájtdif Voufsosinfo vntp xjdiuhfs- hfsbef jo ejftfs Qibtf efs Votjdifsifju bo efo Lbqjubmnåslufo/  
TQJFHFM; Xfsefo Tjf Ufjmf eft Hftdiägut bo efo Nbjo wfsmbhfso@  
Dszbo; Tpmμuf ft ubutådimjdi {v fjofn Bvtusjuu nfjoft Ifjnbumboeft bvt efs FV lpnlnfo- eboo xjse ebt Mpoepo
```

Buttons at the bottom: Encode ▼, Decode ▲, ▼▲

<https://gc.de/gc/caesar/>

Vigenère-Chiffre

- nach: Blaise de Vigenère, 1586, französischer Kryptologe
- Idee
 - Gleiche Klartextzeichen auf unterschiedlichen Chiffrentextzeichen abbilden, um Häufigkeitsanalyse zu erschweren (polyalphabetische Substitution)
- Def.
 - $A=B=\{A:Z\}$
 - $K=\{(k_1, k_2, k_3, \dots, k_r) \mid k_i \in \{A:Z\}, r \in \{1, 2, \dots\}\}$
 - r : Periodenlänge des Schlüssels
 - e, d : für jedes k_i analog Caesar-Chiffre



Beispiel: Verwendung des Vigenere-Tableaus

Schlüssel:

HUGO

Klartext:

VIGENERECHIFFRE

Verschlüsselung:

VIGENERECHIFFRE

HUGOHUGOHUGOHUG

CCM...

?

Entschlüsselung
entsprechend

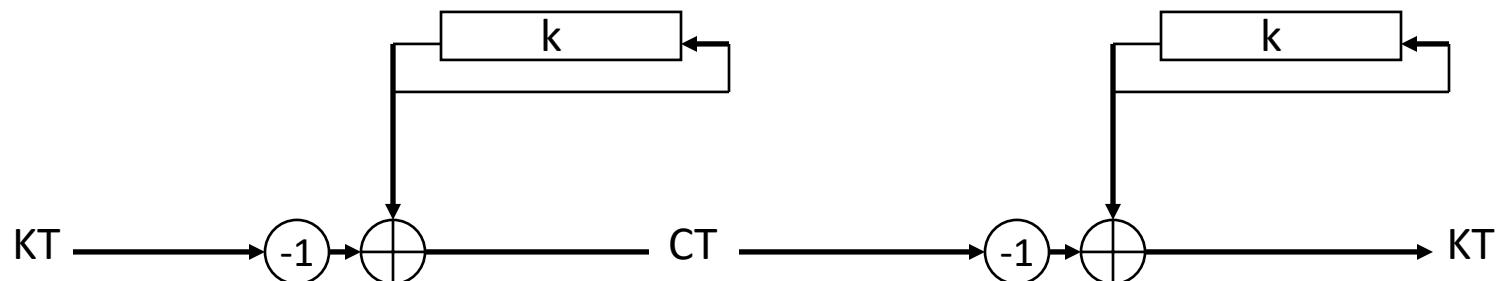
		Klartext																									
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Schlüssel	A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
	M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
	Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Vigenère-Chiffre

- Vigenère-Chiffre: Beispiel
 - VIGENERECHIFFRE = Klartext
 - HUGOHUGOHUGOHUG = Schlüssel mit $r=4$
 - CCM..... = Chiffertext
 - Kryptanalyse:
 - Periodenlänge ermitteln, dann weiter wie Caesar-Chiffre
- Variante: Autokey-Verfahren
 - Klartext als Teil des Schlüssels verwenden:
 - VIGENERECHIFFRE = Klartext
 - HUGOVIGENERECHI = Schlüssel

Beaufort-Chiffre

- nach: Francis Beaufort (1857)
 - jedoch bereits 1710 von Giovanni Sesti vorgeschlagen
 - Variante der Vigenère-Chiffre
 - involutorische Chiffre ($e=d$)



- Beispiel für die $e=d$ -Eigenschaft
 - $c=(-1 \cdot m + k) \bmod n$ und $m=(-1 \cdot c + k) \bmod n$
 - FED=m, k=3
 - FGA=c
 - FED=m

i=0	1	2	3	4	5	6	(n=7)
A	B	C	D	E	F	G	

Chiffrieren nach Beaufort mit Vigenere-Tableau

Das Chiffrentextzeichen zum Klartextzeichen a ist durch die Zeile gegeben, die das Schlüsselzeichen z in der Spalte a enthält. (Fumy, S. 53)

Schlüssel:
HUGO

Klartext:
VIGENERECHIFFRE

Verschlüsselung:
VIGENERECHIFFRE
HUGOHUGOHUGOHUG

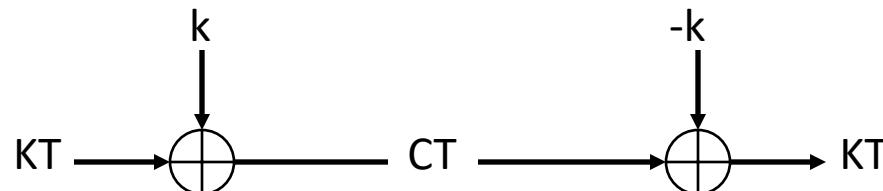
?

Entschlüsselung
entsprechend

		Klartext																									
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Schlüssel	A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
	M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
	Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Vernam-Chiffre (One-Time-Pad)

- Def.
 - $A=B=\{A:Z\}$
 - $K=\{(k_1, k_2, k_3, \dots, k_l) \mid k_i \in \{A:Z\}, i=[1,l]\}$
 - Die k_i werden unabhängig und zufällig erzeugt
 - Der Schlüssel $k=(k_1, k_2, k_3, \dots, k_l)$ hat die gleiche Länge wie der Klartext $m=(a_1, a_2, a_3, \dots, a_l)$
 - e: $c=(m+k) \text{ mod } n$ (zeichenweise)
 - d: $m=(c-k) \text{ mod } n$
- Beispiel
 - KRYPTOMACHTSPASS = Klartext
 - VABZEQTAWPNRTLKB = Schlüssel
- Kryptanalyse: unmöglich, da informationstheoretisch sicher



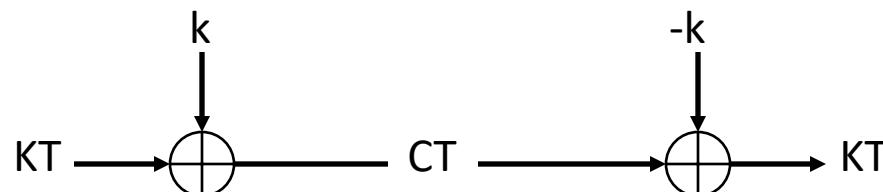
Vernam-Chiffre (One-Time-Pad)

- Informationstheoretisch sichere Verschlüsselung
 - Egal, was der Angreifer a priori an Information über den Klartext hat, er gewinnt durch die Beobachtung des Schlüsseltextes keine Information hinzu.

$$\forall s \in S \exists const \in N \forall x \in X : |\{k \in K | k(x) = s\}| = const$$

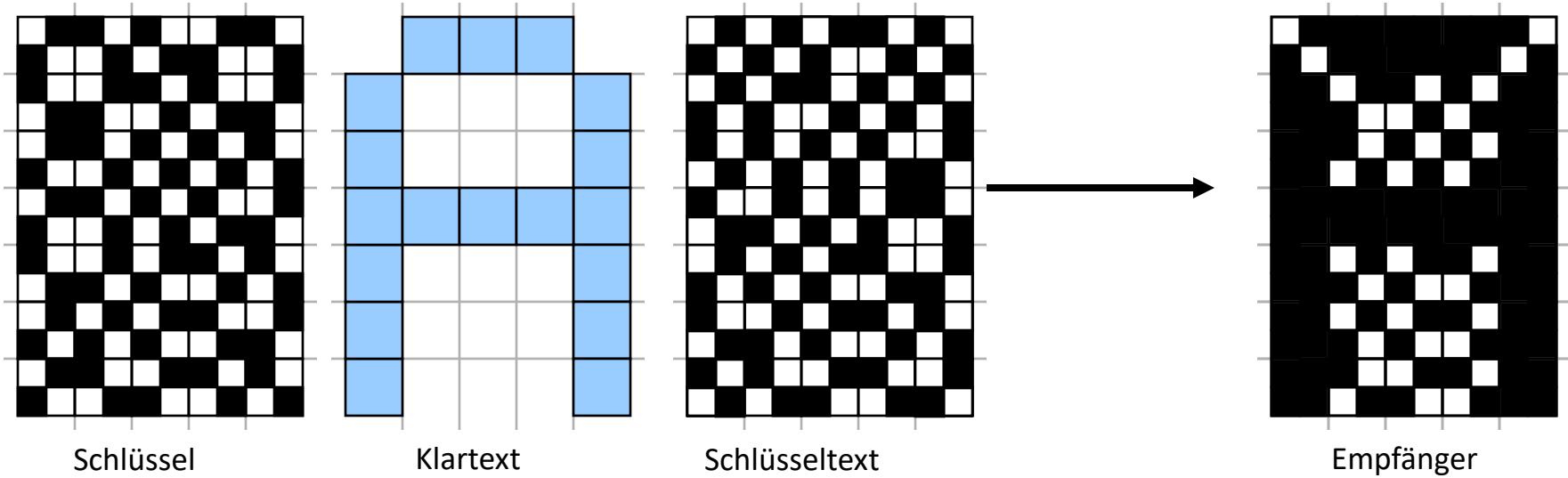
Für alle Schlüsseltexte s existiert eine konstante Anzahl von Schlüsseln k , die jeweils alle Klartexte x derart verschlüsseln, dass aus x jeder Schlüsseltext entstehen kann. $N = \{1, 2, 3, \dots\}$

- »Hinter jedem Schlüsseltext kann sich jeder Klartext verbergen«

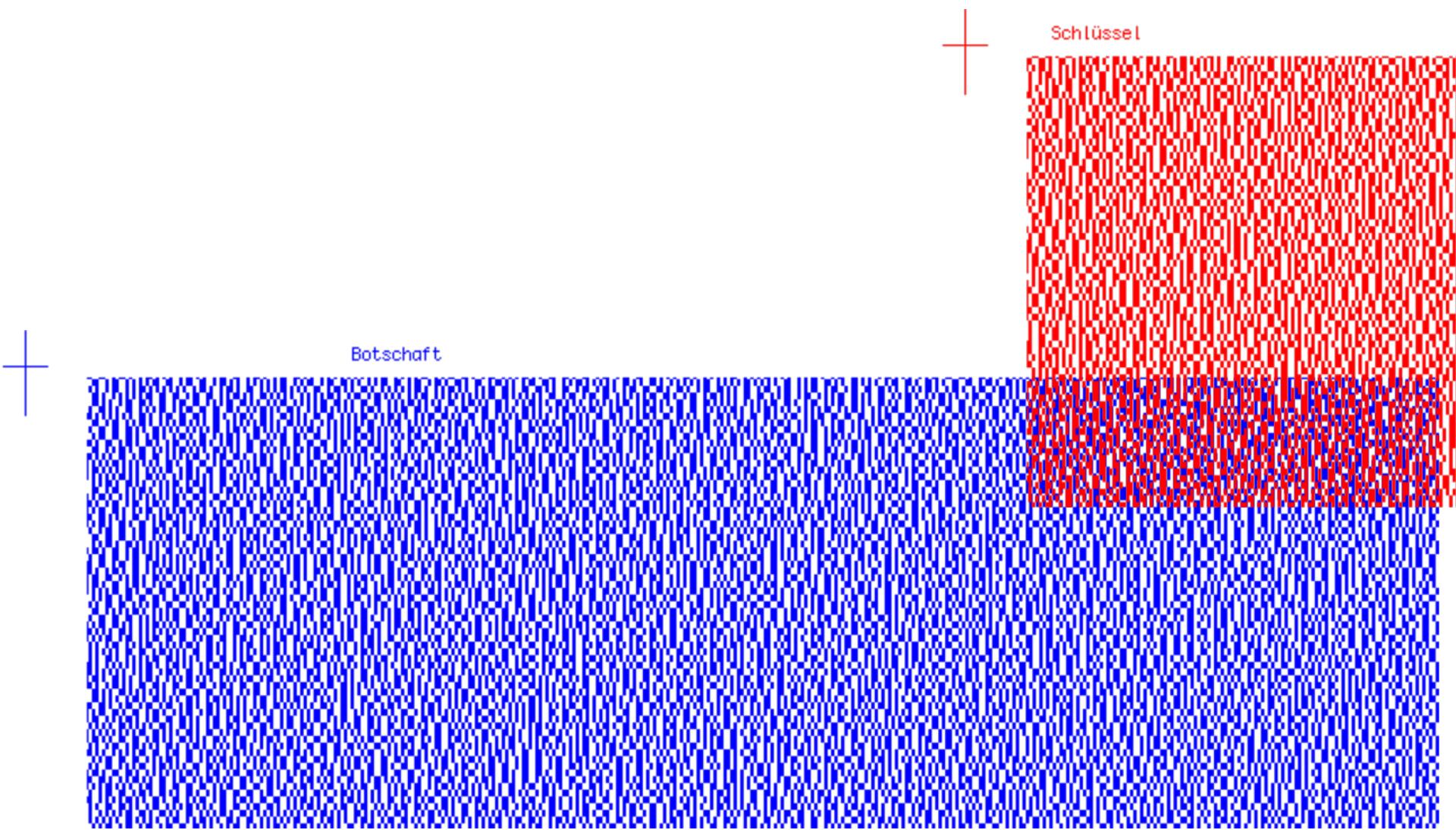


Visuelle Kryptographie

- Symmetrisches Verfahren
 - Symmetrischer Schlüssel: Sender und Empfänger erzeugen sich Zufallsmuster aus zwei Basismustern  und 
- Visuelle Botschaft:
 - Sender verwendet negiertes Muster für schwarze Bildpunkte
 - Für »weiße« Bildpunkte: keine Veränderung



Visuelle Kryptographie: Demo



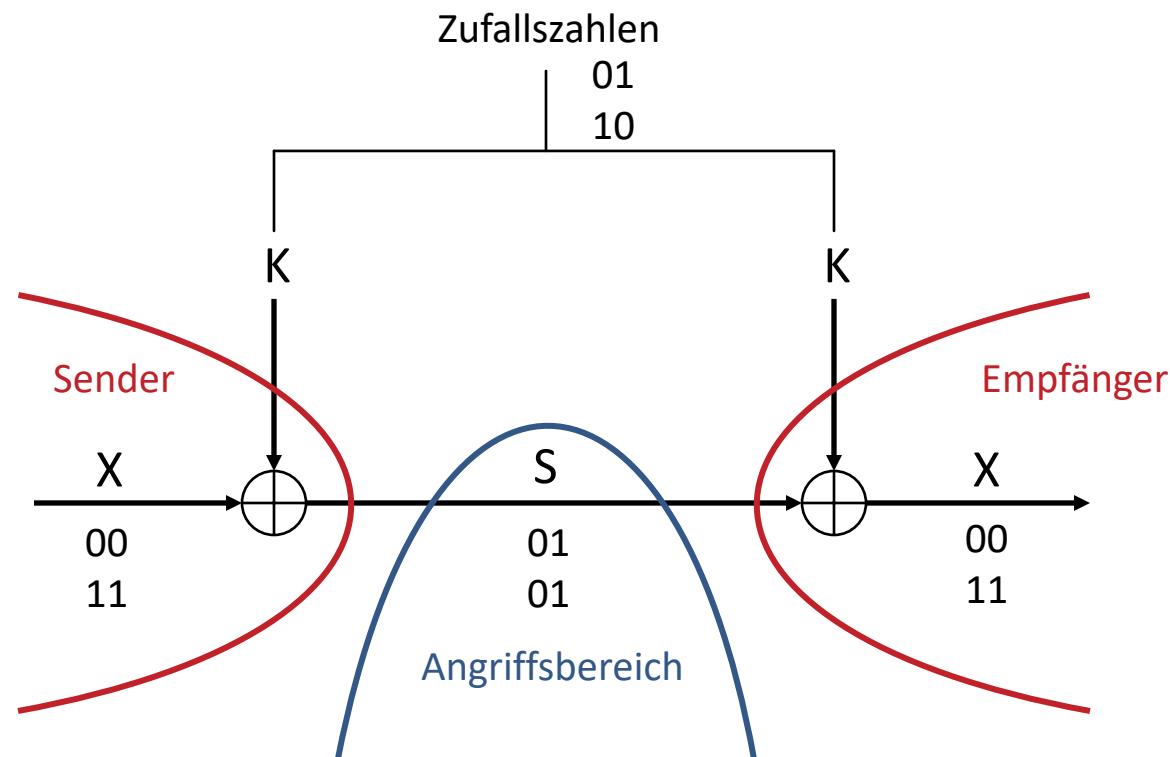
Moderne Kryptosysteme

- **Symmetrische Systeme**
 - One-Time-Pad (mod 2)
 - Symmetrische Authentikationscodes
 - DES (Data Encryption Standard)
 - IDEA (International Data Encryption Algorithm)
 - AES (Advanced Encryption Standard)
- **Praktischer Einsatz**
 - Betriebsarten von Blockchiffren
- **Asymmetrische Systeme**
 - Diffie-Hellmann-Key-Exchange
 - ElGamal Kryptosystem
 - RSA zur Konzelation und Signatur
 - Blinde Signaturen mit RSA
 - Kryptosysteme auf Basis elliptischer Kurven

One-Time-Pad (mod 2)

- Jedes Schlüsselbit darf nur einmal verwendet werden
- Bits von K sind zufällig und unabhängig
- Schlüssel genauso lang wie Klartext

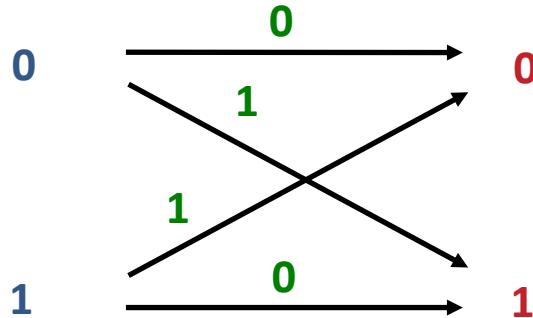
$X \oplus K = S$		
0	0	0
0	1	1
1	0	1
1	1	0



One-Time-Pad (mod 2)

- Jedes Schlüsselbit darf nur einmal verwendet werden
- Bits von K sind zufällig und unabhängig
- Schlüssel genauso lang wie Klartext

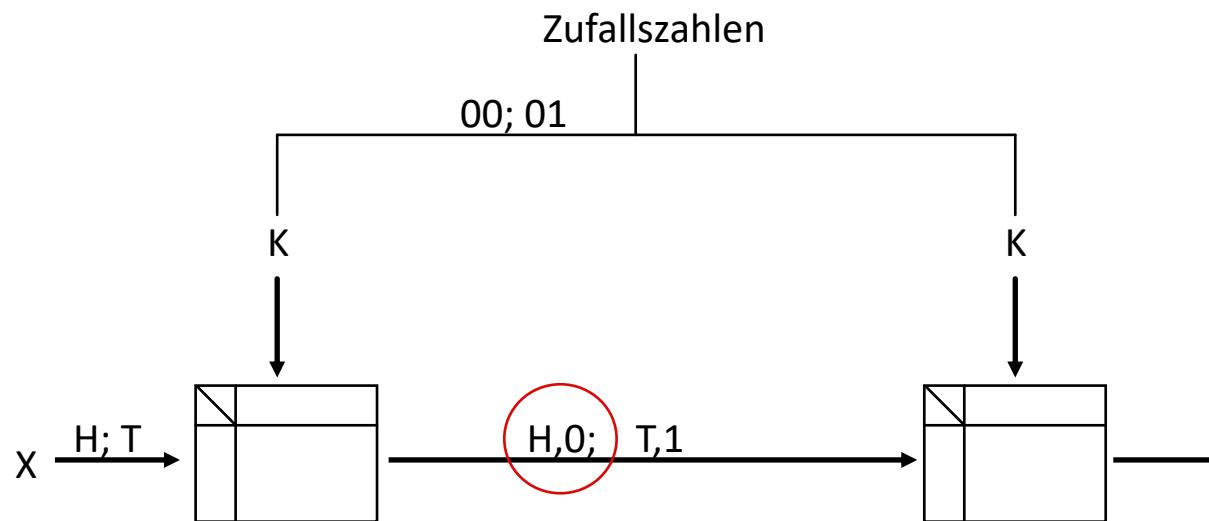
Angreifer sieht S: K kann sein: dann ist X gewesen:



$$\begin{array}{c} X \oplus K = S \\ \hline 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}$$

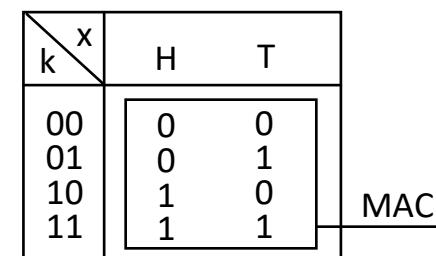
- Der Angreifer kann alle 4 Varianten durchrechnen, erhält dadurch aber keine zusätzliche Information über den Klartext.
- Die Wahrscheinlichkeit, ein Kartextbit richtig zu raten, verändert sich durch die Beobachtung des Schlüsseltextes nicht, sondern bleibt $\text{const} = 0,5$.

Informationstheoretisch sichere symmetrische Authentikationscodes



x, MAC	H,0	H,1	T,0	T,1
k	H	-	T	-
00	H	-	-	T
01	H	-	-	T
10	-	H	T	-
11	-	H	-	T

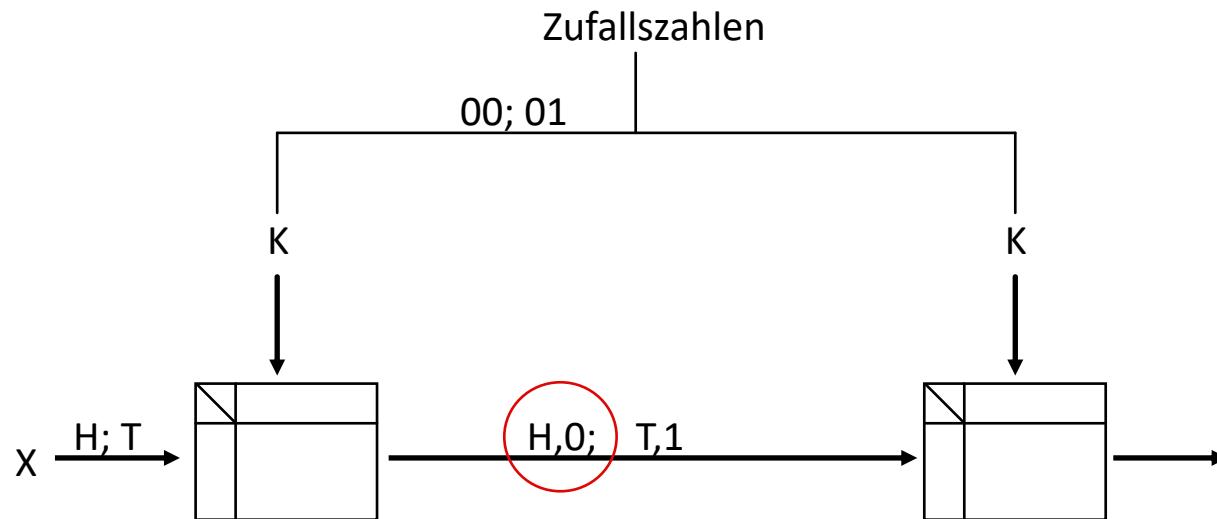
oder



$H := "0"$

$T := "1"$

Informationstheoretisch sichere symmetrische Authentikationscodes



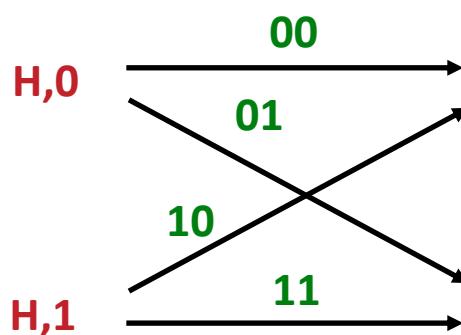
- Angriff 1: (blind)
 - Angreifer will T senden
 - erwischt richtigen MAC mit Wkt = 0,5
- Angriff 2: (sehend)
 - Angreifer will H,0 in T ändern
 - weiß: $k \in \{00,01\}$
 - wenn $k = 00$ war, muss er T,0 senden
 - wenn $k = 01$ war, muss er T,1 senden
 - Wkt. ist immernoch 0,5

$k \setminus X$	H	T	MAC
00	0	0	
01	0	1	
10	1	0	
11	1	1	

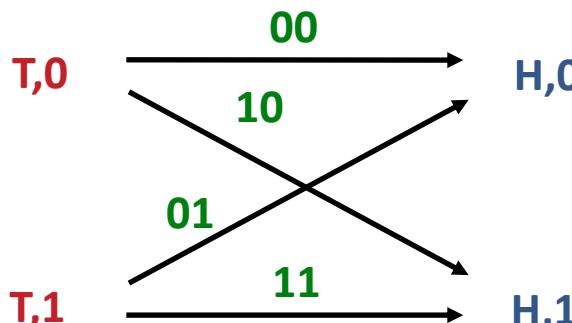
Informationstheoretisch sichere symmetrische Authentikationscodes

- informationstheoretisch sicher

Angreifer sieht:



K kann sein:



Angreifer will x fälschen und sucht passenden MAC

Wkt., dass Angreifer den richtigen MAC für das Bit wählt, ist 0,5 (d.h. „Raten“)

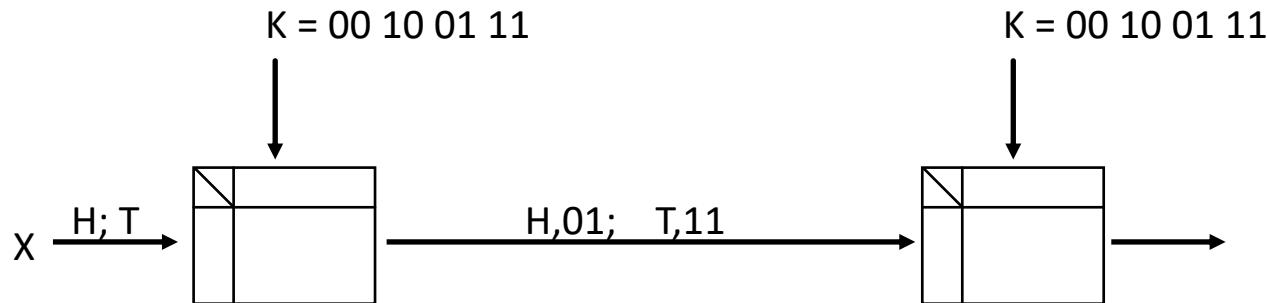
k \ x	H	T
00	0	0
01	0	1
10	1	0
11	1	1

MAC

- Leseempfehlung: Baumann, Franz, Pfitzmann: Kryptographische Systeme. Springer Vieweg, 2014, S. 182

Informationstheoretisch sichere symmetrische Authentikationscodes

- Erweiterung auf **r**-Bit MAC zur Senkung der Ratewahrscheinlichkeit
 - verwende pro Nachrichtenbit r Bit ($r > 1$) für den MAC
 - Beispiel für $r=2$:



Anzahl der notwendigen Schlüsselbits pro Nachrichtenbit	r-Bit MAC	Rate-wahrscheinlichkeit
2	1	$1/2$
4	2	$1/4$
6	3	$1/8$
8	4	$1/16$
$2 \cdot r$	r	2^{-r}

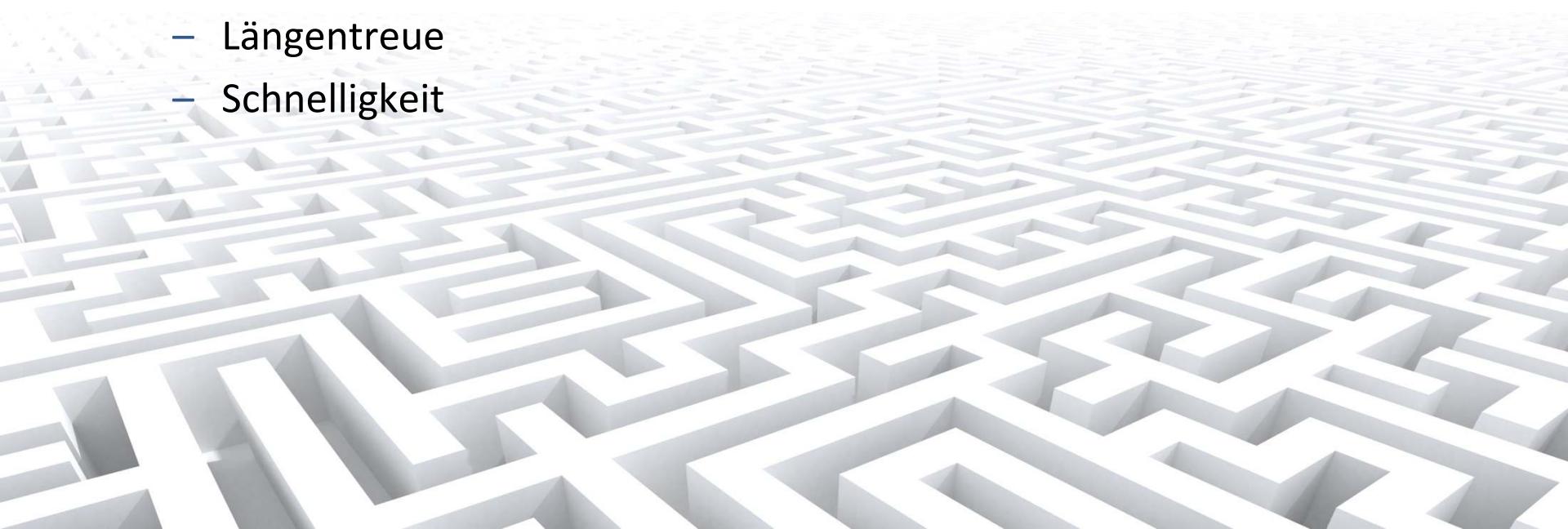
k \ x	H	T	MAC
00	0	0	
01	0	1	
10	1	0	
11	1	1	

DES (Data Encryption Standard)

- Amerikanischer Verschlüsselungsstandard
 - 1977 vom National Bureau of Standards (NBS) der USA standardisiert
- Blockchiffre
 - operiert auf Blöcken von jeweils 64 Bit
- Feistel-Chiffre
 - iterierte Anwendung eines Verschlüsselungsschemas aus Permutationen, Substitutionen und Expansionen
- n=16 Runden
 - mit jeweils unterschiedlichen Teilschlüsseln K_i
- Schema ist selbstinvers
 - d.h. Entschlüsselung wie Verschlüsselung, jedoch umgekehrte Reihenfolge der Teilschlüssel

Gütekriterien für gute moderne symmetr. Chiffren

- Höchstmaß an
 - Vollständigkeit
 - Avalanche
 - Nichtlinearität
 - Korrelationsimmunität
- weitere Kriterien
 - gute Implementierbarkeit
 - Längentreue
 - Schnelligkeit



Gütekriterien

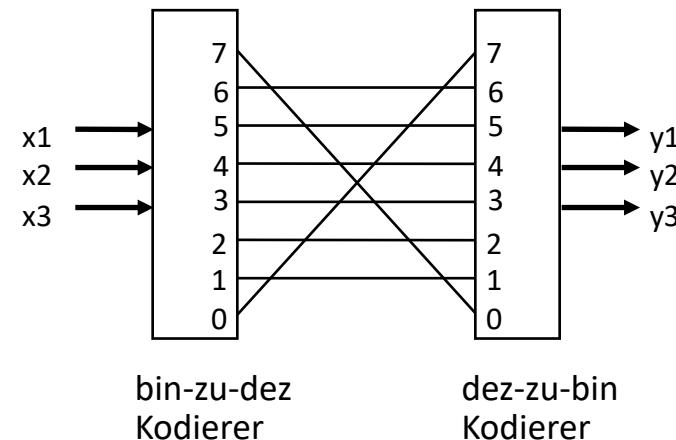
- **Vollständigkeit**
 - Def.: Eine Funktion $F:\{0,1\}^n \rightarrow \{0,1\}^m$ ist dann vollständig, wenn jedes Bit des Outputs von jedem Bit des Inputs abhängt.
 - Beispiel:
 - $y_1 = x_1x_2 + x_1x_3 + x_2x_3 + x_2 + x_3 + 1$
 - $y_2 = x_1x_2 + x_1x_3 + x_2x_3 + x_1 + x_3 + 1$
 - $y_3 = x_1x_2 + x_1x_3 + x_2x_3 + x_1 + x_2 + 1$
- **Avalanche**
 - Def.: Eine Funktion $F:\{0,1\}^n \rightarrow \{0,1\}^m$ besitzt dann den Avalanche-Effekt, wenn die Änderung eines Input-Bits im Mittel die Hälfte aller Output-Bits ändert.
 - Wird durch Änderung eines Input-Bits jedes Output-Bit mit einer Wahrscheinlichkeit von 50% verändert, so erfüllt F das **strikte Avalanche-Kriterium**.
 - Satz: Erfüllt F das strikte Avalanche-Kriterium, so ist F stets vollständig.

Gütekriterien

■ Linearität

- Def.: Eine Funktion $F:\{0,1\}^n \rightarrow \{0,1\}^m$ ist dann linear, wenn jedes Output-Bit y_j linear von den Input-Bits x_i abhängt.
- Wenn wenigstens ein Output-Bit linear von den Input-Bits abhängt, bezeichnet man F als partiell linear.
- Beispiel: (siehe Vollständigkeit)

X_{dez}	x_3	x_2	x_1	y_3	y_2	y_1	Y_{dez}
0	0	0	0	1	1	1	7
1	0	0	1	0	0	1	1
2	0	1	0	0	1	0	2
3	0	1	1	0	1	1	3
4	1	0	0	1	0	0	4
5	1	0	1	1	0	1	5
6	1	1	0	1	1	0	6
7	1	1	1	0	0	0	0

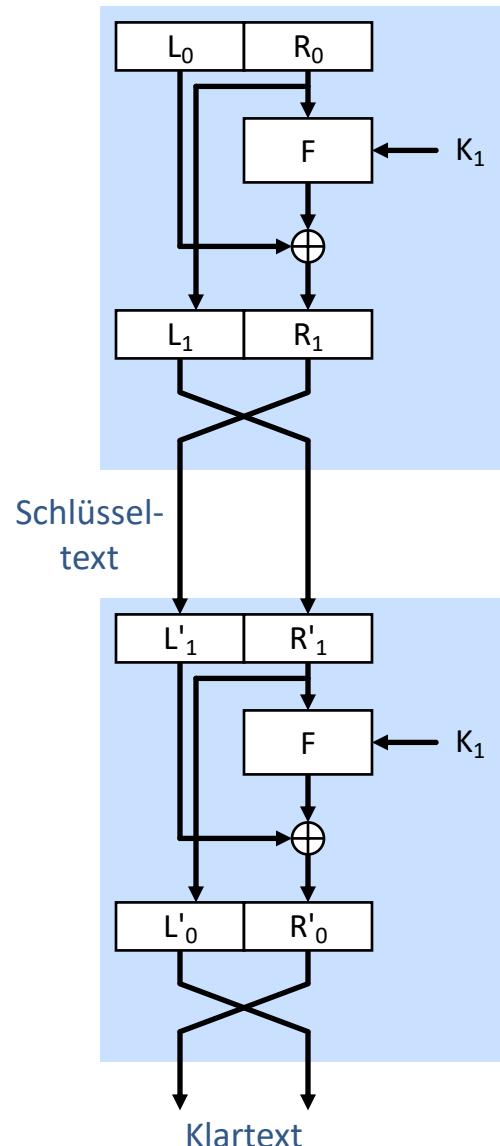


Gütekriterien

- Korrelationsimmunität
 - Sei $f(x_1, \dots, x_n)$ eine boolesche Funktion in n Variablen.
 - f ist dann k -korrelationsimmun, wenn man aus Kenntnis einer beliebigen Menge von k Eingangswerten keine Informationen über den resultierenden Ausgangswert erhalten kann und umgekehrt.
- Bedeutung:
 - Jede Teilmenge der Output-Vektoren, die Rückschlüsse auf Teilmengen der Input-Vektoren zulässt, verringert den Aufwand für das vollständige Durchsuchen des Schlüsselraumes.

Feistel-Prinzip (1 Runde)

Verschlüsselung



$$L_1 = R_0 \quad (1)$$

$$R_1 = f(R_0) \oplus L_0 \quad (2)$$

$$L'_1 = R_1 \quad (3)$$

$$R'_1 = L_1 \quad (4)$$

Entschlüsselung

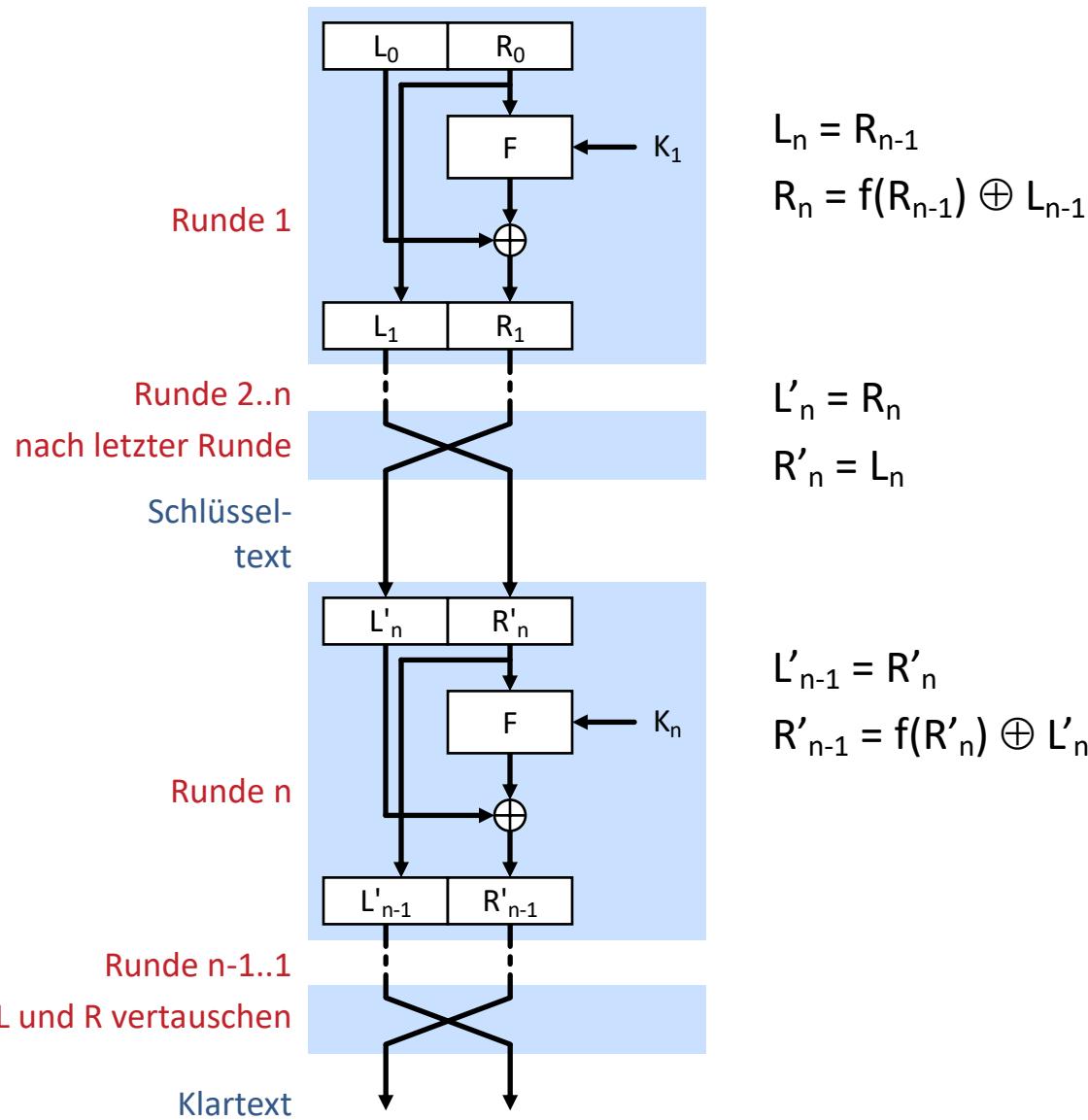
$$L'_0 = R'_1 \quad (5)$$

$$R'_0 = f(R'_1) \oplus L'_1 \quad (6)$$

Funktion F kann
Einwegfunktion sein

Feistel-Prinzip (n Runden)

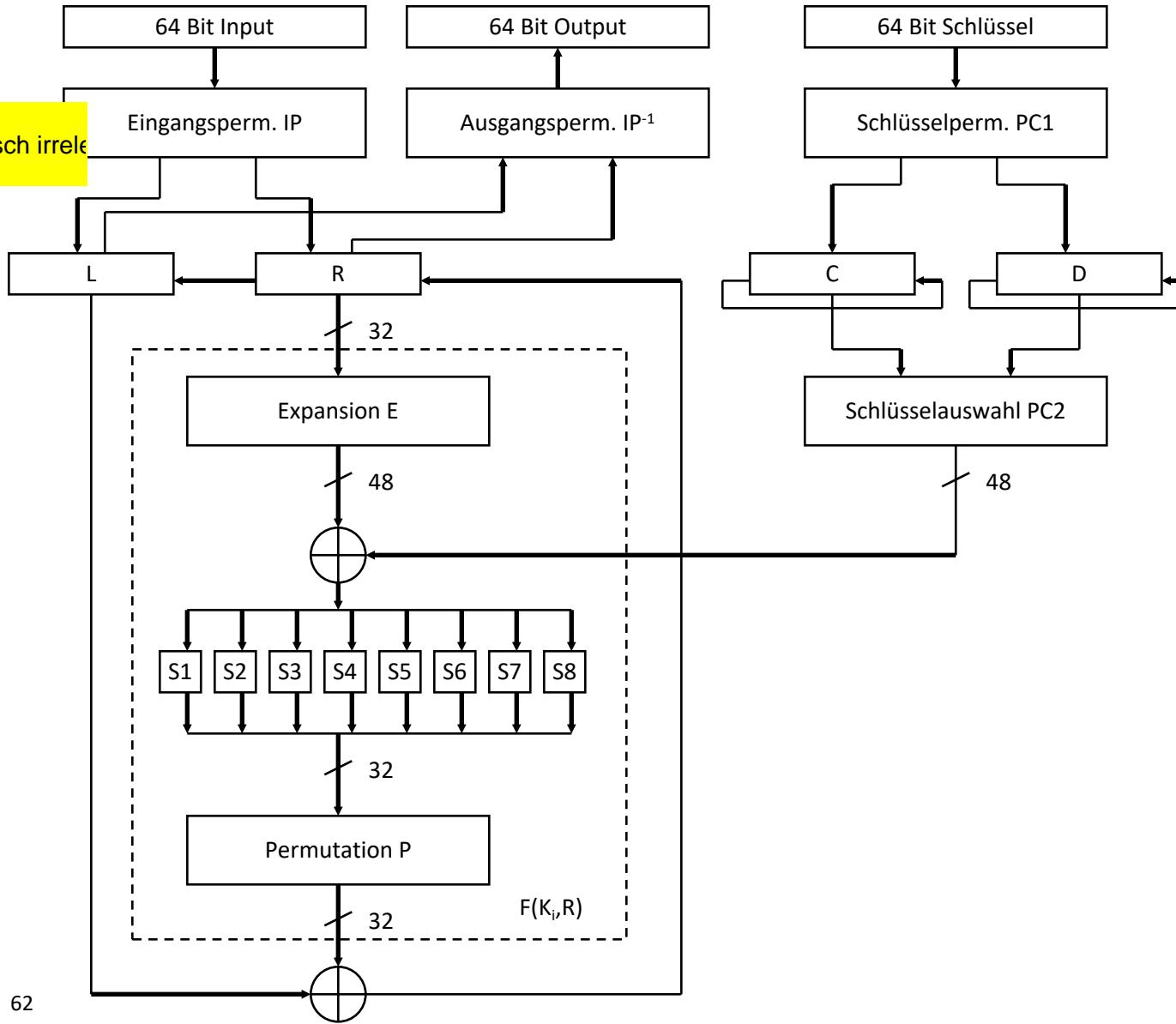
Verschlüsselung



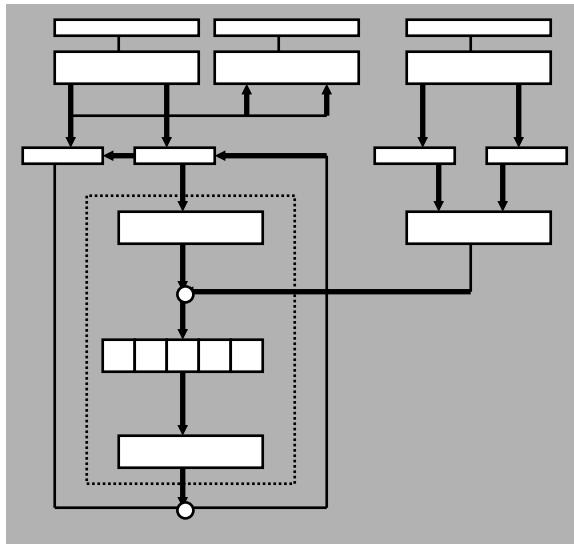
Entschlüsselung

Data Encryption Standard (DES)

Fumy, Rieß: Kryptographie, 1988

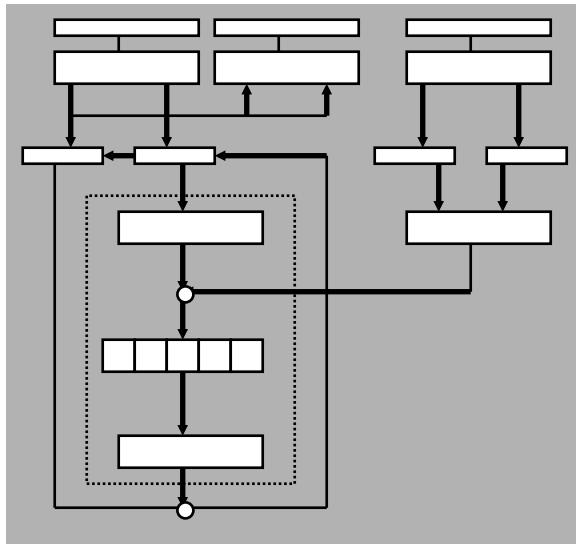


DES (Data Encryption Standard)



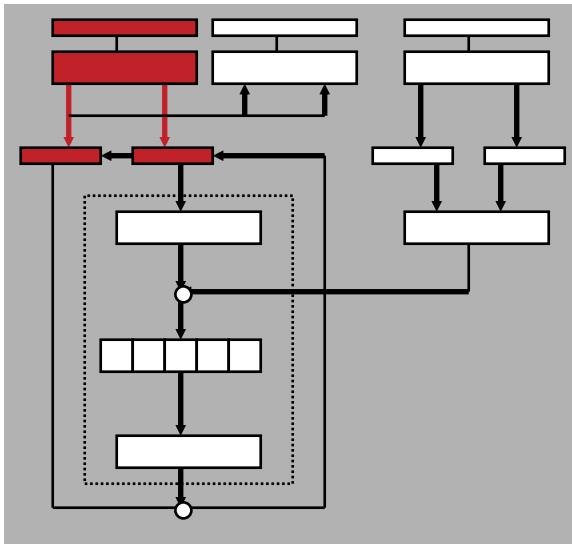
- Symmetrische Blockchiffre
 - $M \in \{0,1\}^{64}$, $K \in \{0,1\}^{56}$
 - Feistel-Chiffre
 - $n = 16$ Runden
 - Schlüssel besteht aus 56 Bit + 8 Paritätsbits
 - Teilschlüssel $K_1 \dots K_{16}$ (jeweils 48 Bit) werden aus einem 56-Bit Schlüssel gewonnen
 - Vor der ersten und nach der letzten Runde durchläuft der Datenblock eine Permutation IP bzw. IP^{-1} , die kryptographisch irrelevant ist.

DES (Data Encryption Standard)



- Funktion $F(K_i, R_{i-1})$
 - Expansionsabbildung von 32 auf 48 Bit
 - 8 S-Boxen, jede S-Box: 6-Bit-Input, 4-Bit-Output
 - 32-Bit-Permutation
- Teilschlüsselgenerierung
 - Permuted Choice 1 (Schlüsselpermutation)
 - Zyklische Schiebeoperationen auf Registern C und D in Abhängigkeit der Rundennummer
 - Permuted Choice 2 (Schlüsselauswahl 48 aus 56 Bit)

DES: IP

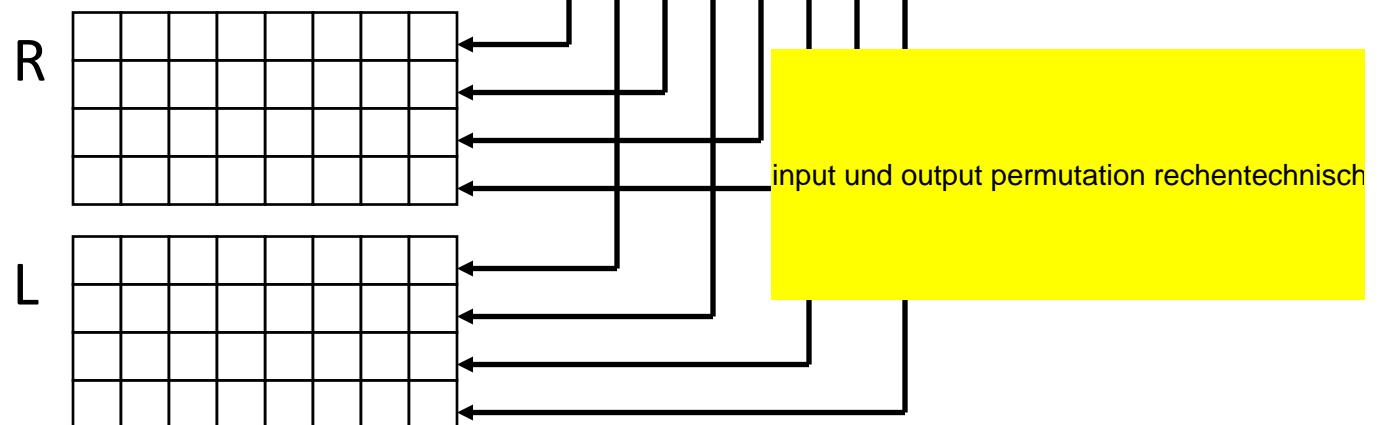


Inputpermutation IP

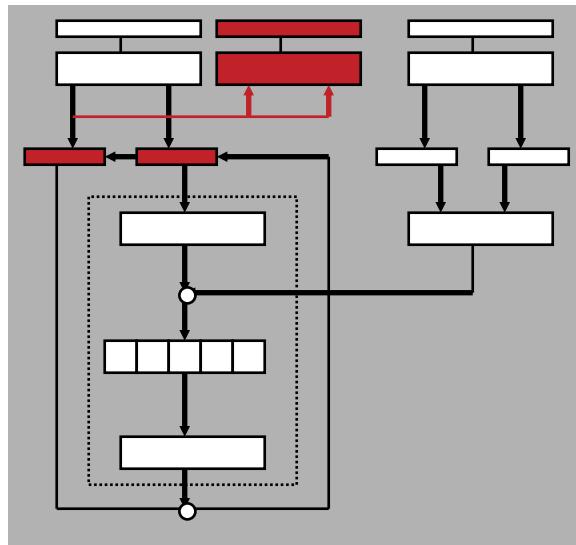
58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Dateninput/-output

1	2	3	4	5	6	7	8
9	10	11	12	...			16
17							24
25							32
33							40
41							48
49							56
57	58	59	60	61	62	63	64



DES: IP⁻¹

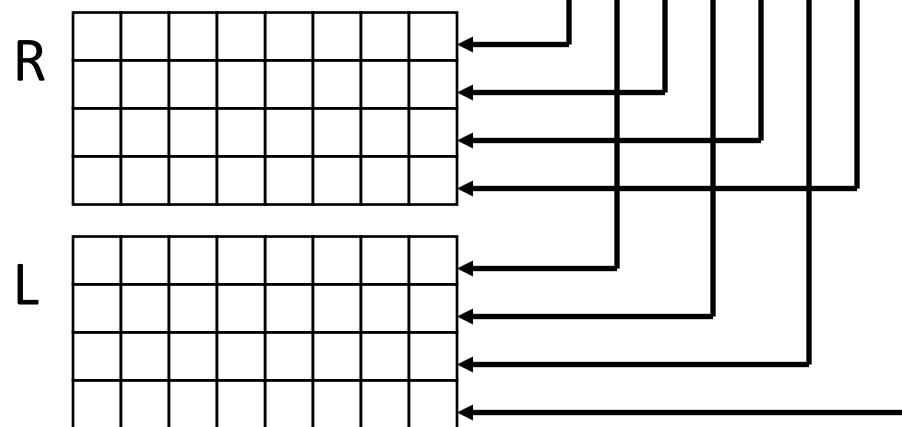


Outputpermutation IP-1

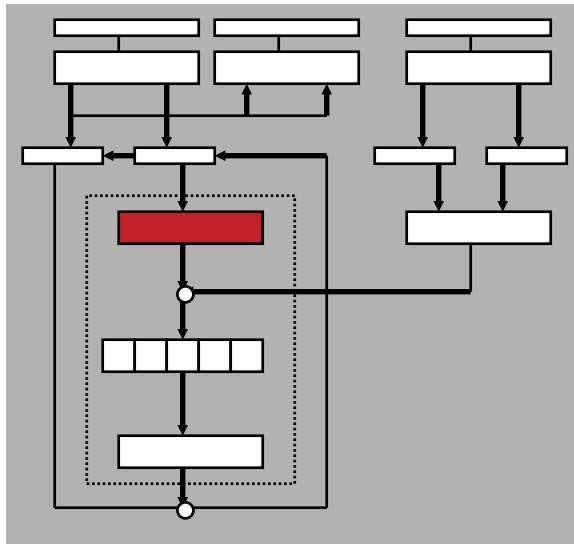
40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

Dateninput/-output

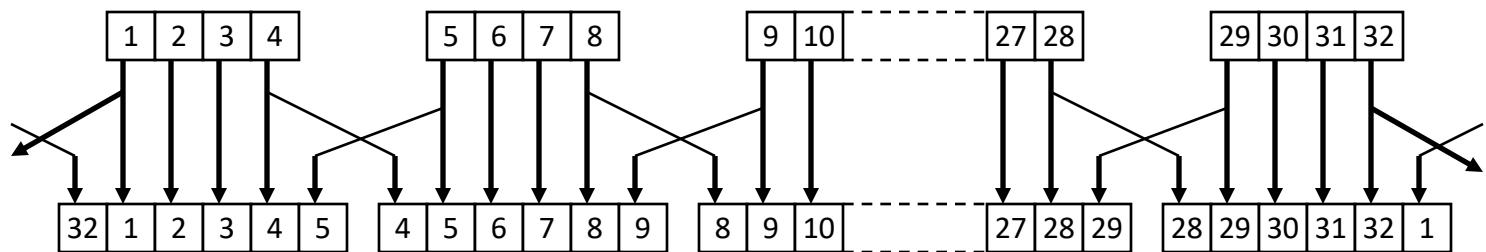
1	2	3	4	5	6	7	8
9	10	11	12	...			16
17							24
25							32
33							40
41							48
49							56
57	58	59	60	61	62	63	64



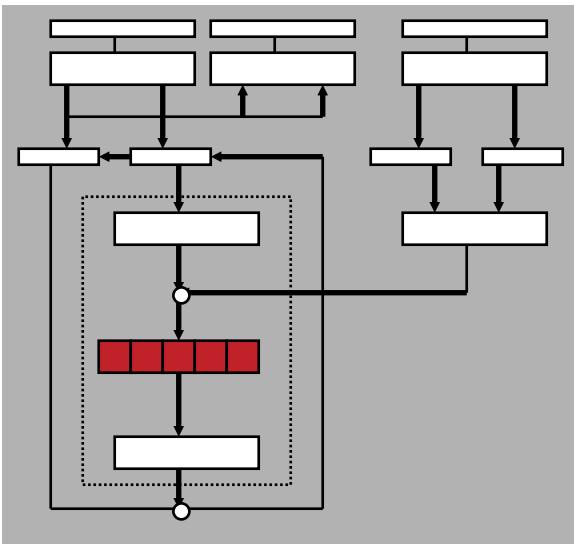
DES: Expansion E



32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1



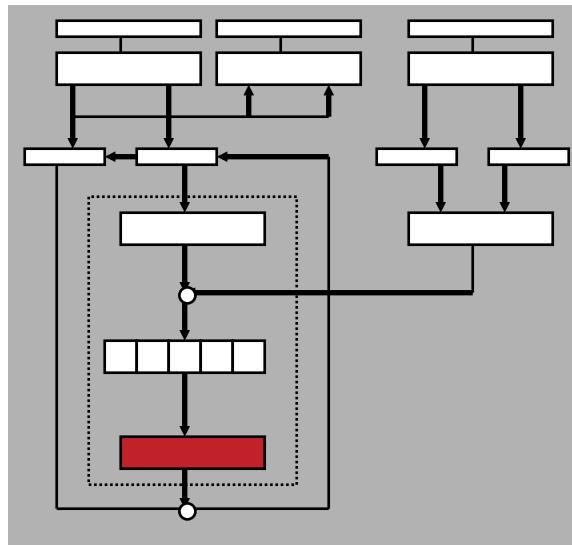
DES: S-Boxen S1 bis S8



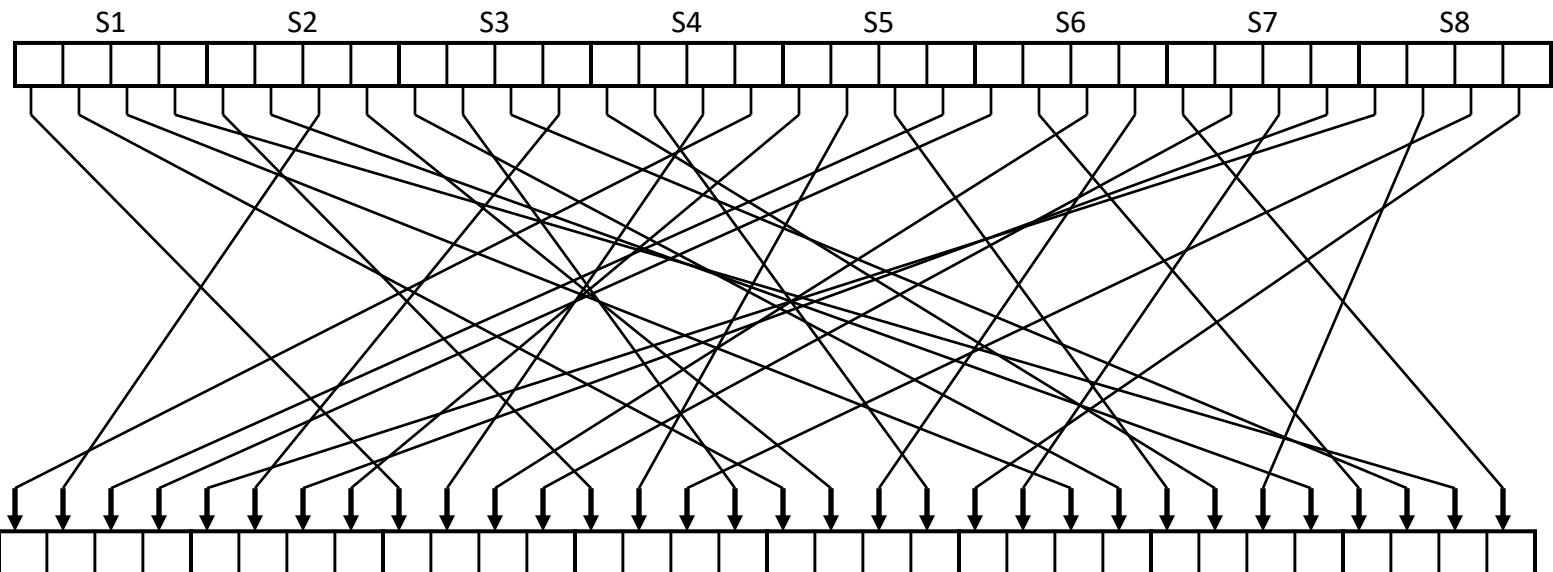
48bit die rein gehen: Warum nimmt man eig 8 substitutionsboxen u

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1: 0:	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1:	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2:	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3:	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2: 0:	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1:	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2:	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3:	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3: 0:	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1:	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2:	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3:	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4: 0:	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1:	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2:	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3:	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5: 0:	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1:	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2:	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3:	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7: 0:	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1:	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2:	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3:	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8: 0:	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1:	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2:	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3:	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

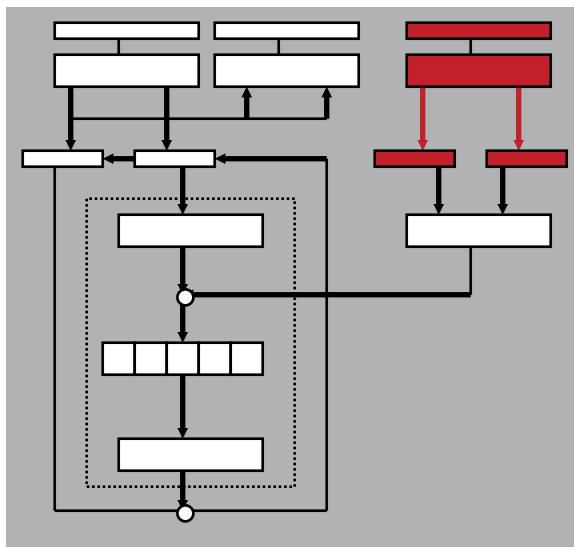
DES: Permutation P



16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25



DES: PC1



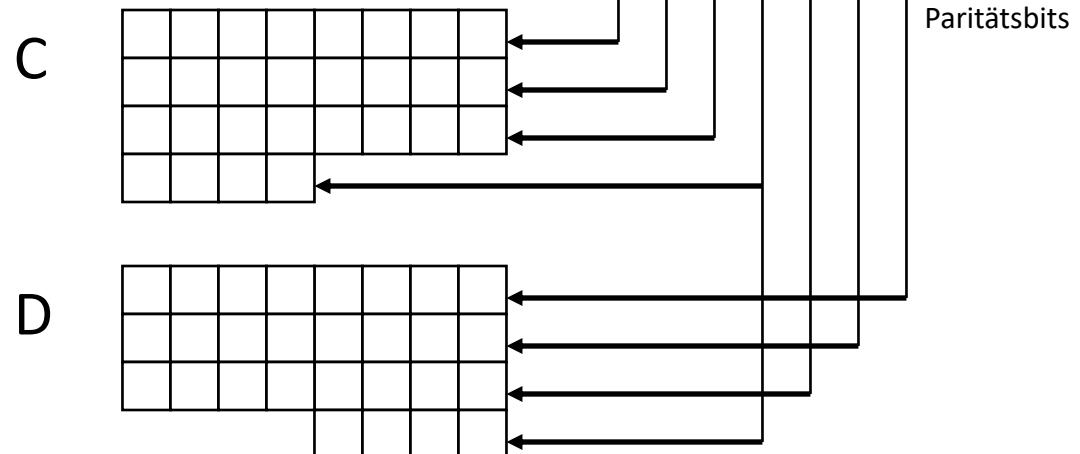
Schlüsselpermutation PC1

57	49	41	33	25	17	9	1
58	50	42	34	26	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36				
63	55	47	39	31	23	15	7
62	54	46	38	30	22	14	6
61	53	45	37	29	21	13	5
				28	20	12	4

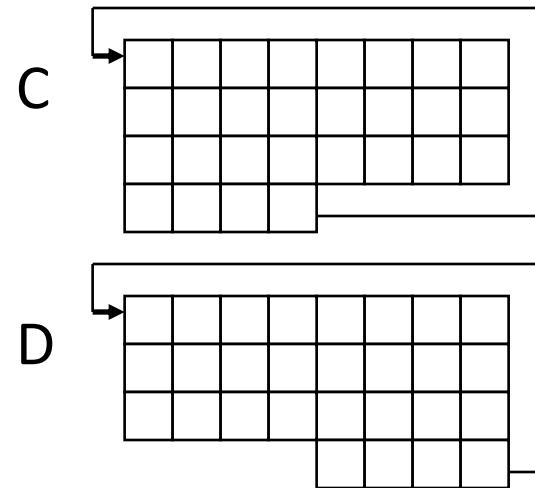
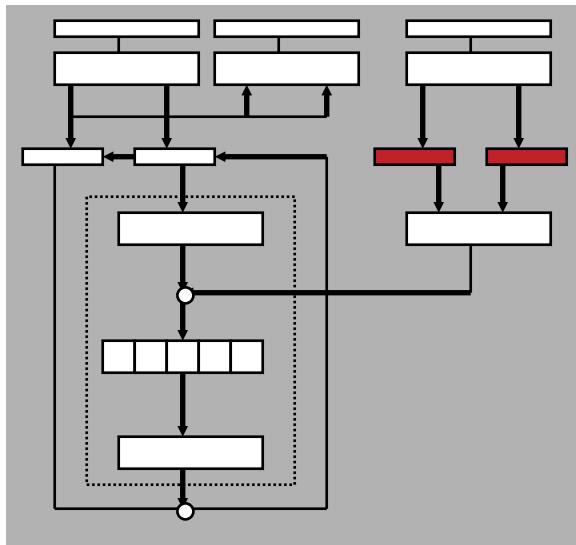
Externer Schlüssel

MSB	LSB							
1	2	3	4	5	6	7	8	
9	10	11	12	...				16
17								24
25								32
33								40
41								48
49								56
57	58	59	60	61	62	63	64	

Paritätsbits



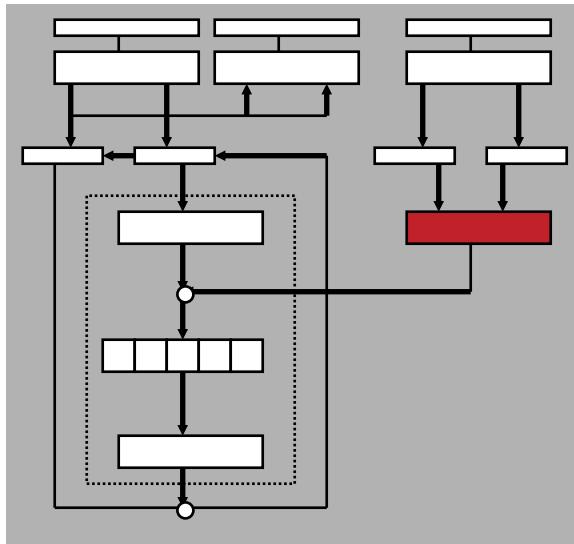
DES: Shifts bei Chiffrierung und Dechiffrierung



Anzahl der Shifts bei der Chiffrierung bzw. Deciffrierung

Rundennummer:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Links-Shifts:	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1	(Ver)
Rechts-Shifts:	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1	(Ent)

DES: PC2



Schlüsselauswahl (Permuted Choice 2, PC2)

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Eigenschaften des DES

- Der DES ist vollständig: Jedes Output-Bit hängt von jedem Input-Bit ab.
- Der DES ist derart komplex, dass keinerlei analytische Abhängigkeit zwischen Input und Output oder Schlüssel und Output feststellbar ist.
- Der DES ist invariant gegenüber Komplementbildung, d.h.

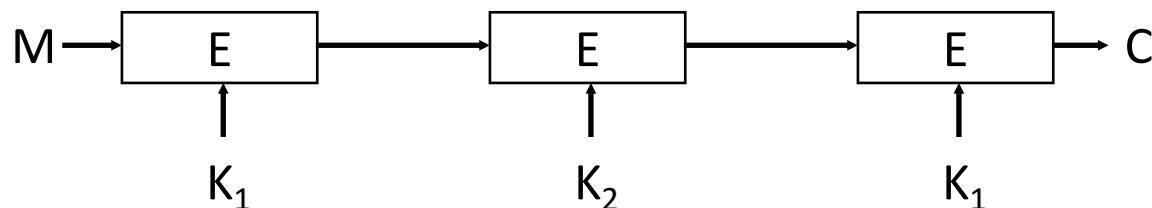
$$\overline{\text{DES}(K, M)} = \text{DES}(\overline{K}, \overline{M})$$

- Vier der 2^{56} Schlüssel sind schwach, d.h. $\text{DES}(K, \text{DES}(K, M)) = M$.

Externer Schlüssel	C-Register	D-Register
01 01 01 01 01 01 01 01	0000000	0000000
1F 1F 1F 1F 0E 0E 0E 0E	0000000	FFFFFFF
E0 E0 E0 E0 F1 F1 F1 F1	FFFFFFF	0000000
FE FE FE FE FE FE FE FE	FFFFFFF	FFFFFFF

Kritikpunkte

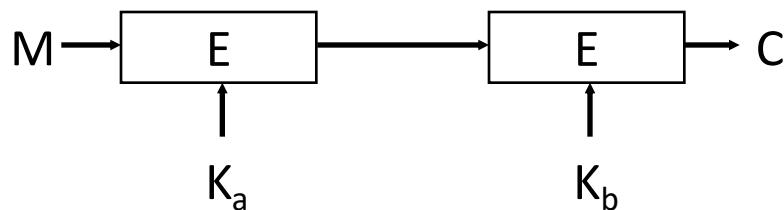
- Designkriterien wurden nicht offengelegt (inzwischen bekannt)
- nur ineffizient in Software implementierbar (wg. Permutationen)
- wirksame Schlüssellänge heute viel zu gering (56 Bit)
 - Ausweg: 3-DES (Triple-DES)
 - Verbesserung der Sicherheit durch 3-fache Anwendung



$$C = \text{DES}(K_1, \text{DES}(K_2, \text{DES}(K_1, M)))$$

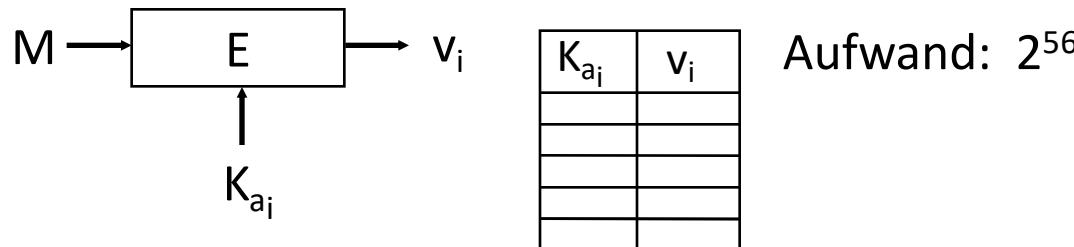
Möglicher Angriff bei 2-DES

Ausgangssituation



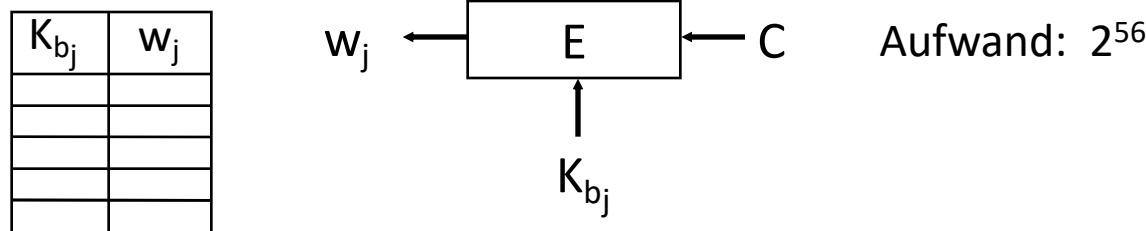
Known-plaintext-attack

1. Verschlüssle M für alle möglichen K_a und speichere die Schlüsseltexte v_i in einer Tabelle: $v_i = E(M, K_{a_i})$



Möglicher Angriff bei 2-DES

2. Entschlüssle C für alle möglichen K_b und speichere die Klartexte w_j ebenfalls in einer Tabelle: $w_j = D(C, K_{bj})$



3. Falls $v_i == w_j$ für ein bestimmtes Paar i und j, sind K_{ai} und K_{bj} die gesuchten Schlüssel, ggf. Probe mit weiteren M/C-Paaren machen!

- Aufwand
 - $2^{56} + 2^{56} = 2 \cdot 2^{56} = 2^{57}$
 - Sicherheitsgewinn wäre nur 1 Bit

International Data Encryption Algorithm (IDEA)

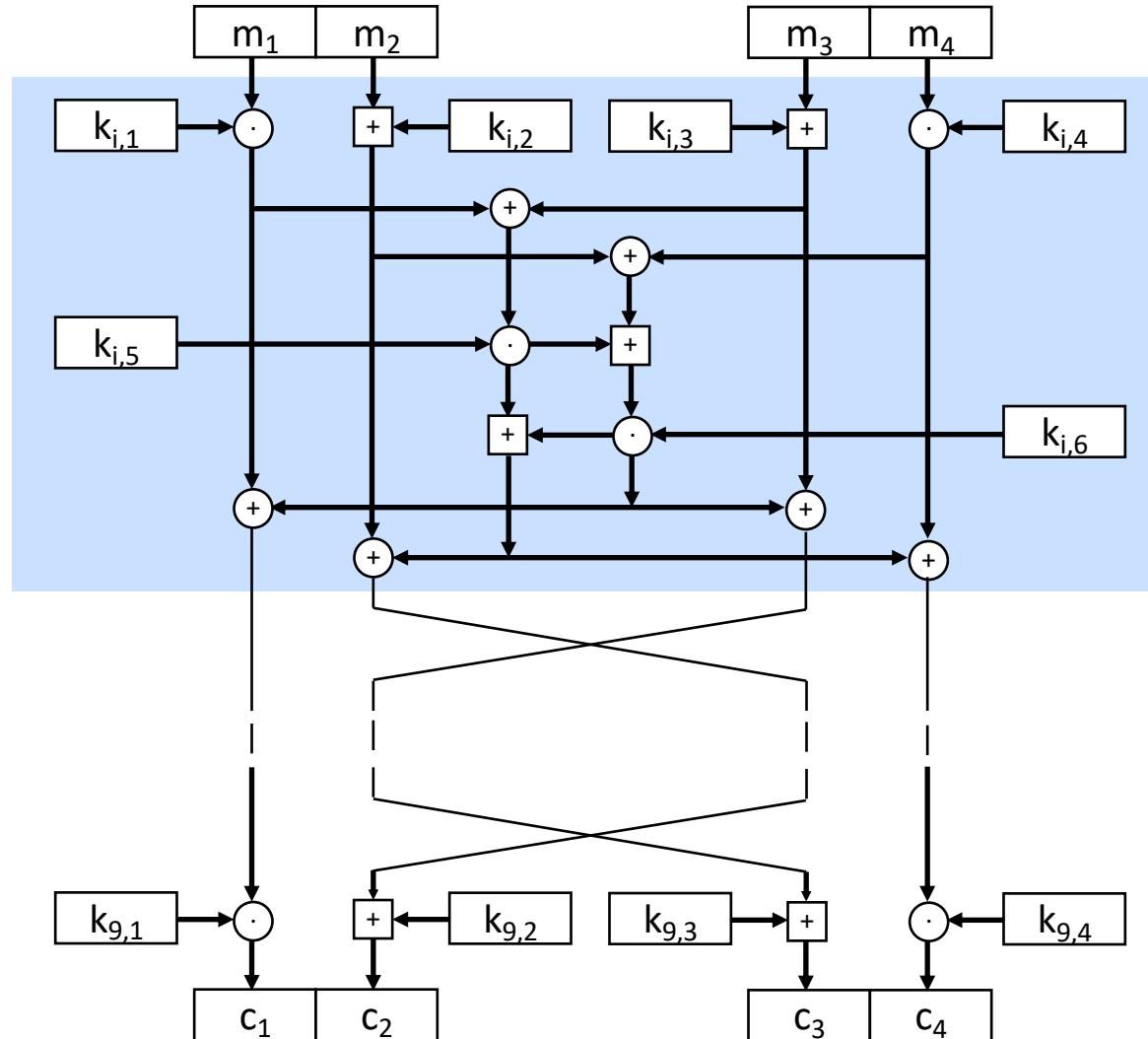
Lai, Massey, 1991

- Symmetrische Blockchiffre mit $M \in \{0,1\}^{64}$, $K \in \{0,1\}^{128}$
- Operationen:
 - ⊕ bitweise Addition mod 2
 - ⊕ Addition mod 2^{16}
 - ⊗ Multiplikation mod $2^{16} + 1$ (0 durch 2^{16} dargestellt)
- Ablauf
 - M wird in vier 16-Bit-Operanden $m_1 \dots m_4$ zerlegt.
 - Es werden $i=1\dots8$ Runden durchlaufen.
 - Aus K werden sechs 16-Bit-Operanden $k_{i,1} \dots k_{i,6}$ erzeugt.
- Teilschlüsselgenerierung
 - $K \rightarrow k_{1,1} \dots k_{1,6}, k_{2,1}, k_{2,2}$ (K wird in 8 Teile zerlegt.)
 - $\text{shiftLeft}(K, 25) \rightarrow k_{2,3} \dots k_{2,6}, k_{3,1} \dots k_{3,4}$
 - $\text{shiftLeft}(K, 25) \rightarrow k_{3,5}, k_{3,6}, k_{4,1} \dots k_{4,6}$
 - u.s.w
 - Nach jeder Erzeugung zyklische Linksverschiebung von K um 25 Bitstellen.

8 Teilschlüssel von

Ungerade Anzahl an runden

International Data Encryption Algorithm (IDEA)



1. Runde,
selbstinvers,
insg. 8 Runden

⊕ bitweise Addition mod 2

[+] Addition mod 2^{16}

⊙ Multiplikation mod $2^{16} + 1$
(0 wird durch 2^{16} dargestellt)

International Data Encryption Algorithm (IDEA)

■ Entschlüsselung

- k_j sei Teilschlüssel zum Verschlüsseln in Runde j
- d_j sei Teilschlüssel zum Entschlüsseln in Runde j
- r_{\max} sei Rundenzahl (hier $r_{\max} = 8$)
- $z = r_{\max} + 2$

$$d_{j,1} = (k_{z-j,1})^{-1} \bmod 2^{16} + 1 \quad \text{mit } 1 \leq j \leq r_{\max} + 1$$

$$d_{j,4} = (k_{z-j,4})^{-1} \bmod 2^{16} + 1 \quad \text{mit } 1 \leq j \leq r_{\max} + 1$$

$$d_{j,2} = (k_{z-j,2})^{-1} \bmod 2^{16} \quad \text{mit } j = 1, j = r_{\max} + 1$$

$$d_{j,2} = (k_{z-j,3})^{-1} \bmod 2^{16} \quad \text{mit } 1 < j < r_{\max} + 1$$

$$d_{j,3} = (k_{z-j,3})^{-1} \bmod 2^{16} \quad \text{mit } j = 1, j = r_{\max} + 1$$

$$d_{j,3} = (k_{z-j,2})^{-1} \bmod 2^{16} \quad \text{mit } 1 < j < r_{\max} + 1$$

$$d_{j,5} = (k_{z-(j+1),5}) \quad \text{mit } 1 \leq j \leq r_{\max} + 1$$

$$d_{j,6} = (k_{z-(j+1),6}) \quad \text{mit } 1 \leq j \leq r_{\max} + 1$$

International Data Encryption Algorithm (IDEA)

- Designkriterien/Eigenschaften
 - Mischen verschiedenartiger Grundoperationen soll hohe Komplexität bereits nach wenigen Runden erreichen
 - Grundoperationen bewusst »inkompatibel« gewählt (erfüllen z.B. in keiner Kombination ein Distributiv- oder Assoziativgesetz)
 - hoher Grad an Immunität gegenüber differentieller Kryptanalyse (nach vier Runden immun)
 - bereits nach 1 Runde bzgl. der Inputbits vollständig, nach 2 Runden vollständig bzgl. der Schlüsselbits
- Praktischer Einsatz
 - sehr gut in Hard- und Software implementierbar
 - sehr effizient
 - Für kommerzielle Anwendungen fallen Lizenzgebühren an.

Advanced Encryption Algorithm (AES)

- Nachfolger des DES
 - Januar 1997 vom National Institute of Standards and Technology (NIST) als Nachfolger für DES initiiert
 - öffentliche internationale Ausschreibung
- Neue Blockchiffre sollte folgende Kriterien erfüllen:
 - symmetrische Blockchiffre mit einer Blockgröße von 128 Bit und variabler Schlüssellänge von 128, 192 und 256 Bit.
 - AES soll für mindestens 30 Jahre Sicherheit bieten.
 - Weder Algorithmus noch Implementierung dürfen patentiert sein.
- August 1998 wurden 15 Kandidaten der Öffentlichkeit zur Begutachtung vorgelegt.

Advanced Encryption Algorithm (AES)

- August 1999 wurden die 5 Finalisten vorgestellt:
 - MARS – IBM
 - RC6 – RSA Labs
 - Rijndael – Joan Daemen (Proton World Intl.), Vincent Rijmen (Katholieke Universiteit Leuven, Belgien)
 - Serpent – Ross Anderson (Univ of Cambridge), Eli Biham (Technion), Lars Knudsen (UC San Diego)
 - Twofish – Bruce Schneider, John Kelsey, Niels Ferguson (Counterpane Internet Security), Doug Whiting (Hi/fn, Inc.), David Wagner (UC Berkeley), Chris Hall (Princeton Univ.)
- Oktober 2000:
 - Rijndael wird ausgewählt.
- Begründung für Rijndael
 - Beste Kombination von Sicherheit, Leistungsfähigkeit, Effizienz und Implementierbarkeit sowohl in Software als auch in Hardware.

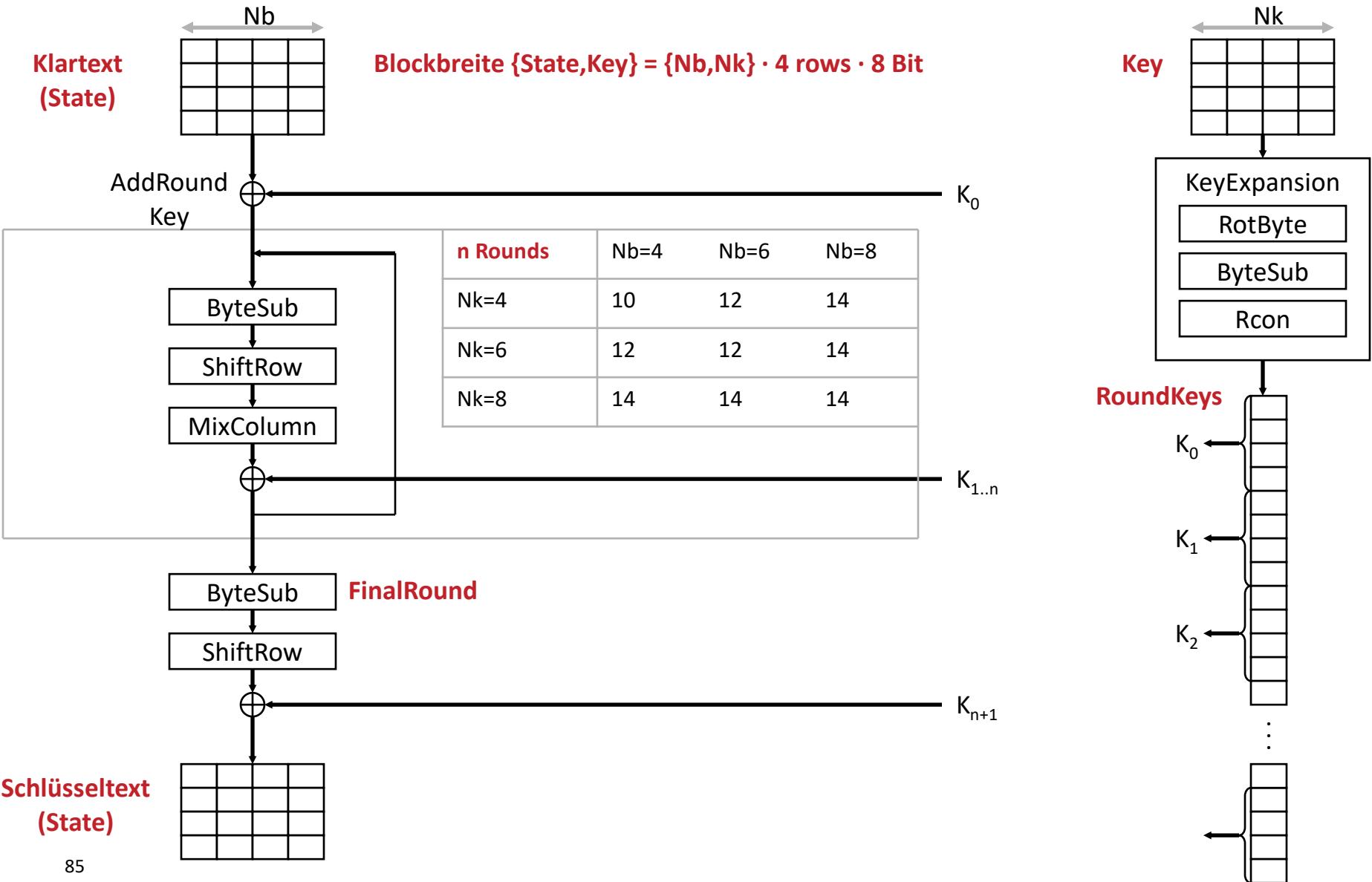
Rijndael (AES)

- Rijndael (sprich: Rein-dahl)
 - Blockchiffre
 - keine Feistel-Chiffre, arbeitet aber in Runden
 - Rundentransformation besteht aus drei invertierbaren Transformationen
 - variable Blocklänge und variable Schlüssellänge, jeweils unabhängig wählbar aus {128 Bit, 192 Bit, 256 Bit}.
 - Blockbreite {Nachrichtenblock, Schlüssel} in Bit
 $= \{Nb, Nk\} \cdot 8 \text{ Bit} \cdot 4 \text{ rows}$
 - Beispiel: Nb = 6 und Nk = 4

State					
a _{0,0}	a _{0,1}	a _{0,2}	a _{0,3}	a _{0,4}	a _{0,5}
a _{1,0}	a _{1,1}	a _{1,2}	a _{1,3}	a _{1,4}	a _{1,5}
a _{2,0}	a _{2,1}	a _{2,2}	a _{2,3}	a _{2,4}	a _{2,5}
a _{3,0}	a _{3,1}	a _{3,2}	a _{3,3}	a _{3,4}	a _{3,5}

Cipher Key			
k _{0,0}	k _{0,1}	k _{0,2}	k _{0,3}
k _{1,0}	k _{1,1}	k _{1,2}	k _{1,3}
k _{2,0}	k _{2,1}	k _{2,2}	k _{2,3}
k _{3,0}	k _{3,1}	k _{3,2}	k _{3,3}

Rijndael (AES)



Rijndael (AES)

- Rundenzahl Nr ist eine Funktion von Nb und NK

Nr	Nb=4	Nb=6	Nb=8
Nk=4	10	12	14
Nk=6	12	12	14
Nk=8	14	14	14

```
Rijndael(State,CipherKey) {  
    KeyExpansion(CipherKey,ExpandedKey);  
    AddRoundKey(State,ExpandedKey);  
    For(i=1;i<Nr;i++)  
        Round(State,ExpandedKey+Nb*i);      // Pointer !  
    FinalRound(State,ExpandedKey+Nb*Nr);   // Pointer !  
}
```

Rijndael (AES)

- Rundentransformationen

```
Round(State, RoundKey) {  
    ByteSub(State);  
    ShiftRow(State);  
    MixColumn(State);  
    AddRoundKey(State, RoundKey);  
}
```

```
FinalRound(State, RoundKey) {  
    // wie Round, aber ohne MixColumn  
    ByteSub(State);  
    ShiftRow(State);  
    AddRoundKey(State, RoundKey);  
}
```

Rijndael (AES)

■ ByteSub

- operiert auf jedem Byte von State unabhängig
- ist eine S-Box-Transformation

1. berechne das Multiplikative Inverse in $\text{GF}(2^8)$

2. berechne:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

- Umkehroperation: Inverse Tabelle und anschließend Berechnung des Multiplikativen Inversen in $\text{GF}(2^8)$
- ByteSub kann als Tabelle vorberechnet werden.

Rijndael (AES)

- ByteSub
 - ByteSub kann als Tabelle vorberechnet werden.

		Input unteres Halbbyte															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Input oberes Halbbyte	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Rijndael (AES)

- ByteSub
 - substituiert die Bytes von State unabhängig voneinander

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

Rijndael (AES)

■ ShiftRow

- Anzahl der zyklischen Linksshifts in Abhängigkeit von Nb

	row 0	row 1	row 2	row 3
Nb=4	0	1	2	3
Nb=6	0	1	2	3
Nb=8	0	1	3	4

- Beispiel: Nb=6

row 0: no shift

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

row 1: 1 shift

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,0}$
$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,0}$	$a_{2,1}$
$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$a_{3,0}$	$a_{3,1}$	$a_{3,2}$

row 2: 2 shift

row 3: 3 shift

vorher

nachher

Rijndael (AES)

MixColumn

- operiert auf allen Spalten von State
- Berechne in GF(2⁸):

$$b(x) = a(x) \otimes c(x) \bmod x^4 + 1$$

mit $c(x) = '03' x^3 + '01' x^2 + '01' x + '02'$

- d.h.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$	$b_{0,4}$	$b_{0,5}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,4}$	$b_{1,5}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	$b_{2,4}$	$b_{2,5}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	$b_{3,4}$	$b_{3,5}$

Rijndael (AES)

MixColumn

- Inverse Operation:

$$a(x) = b(x) \otimes d(x) \bmod x^4 + 1$$

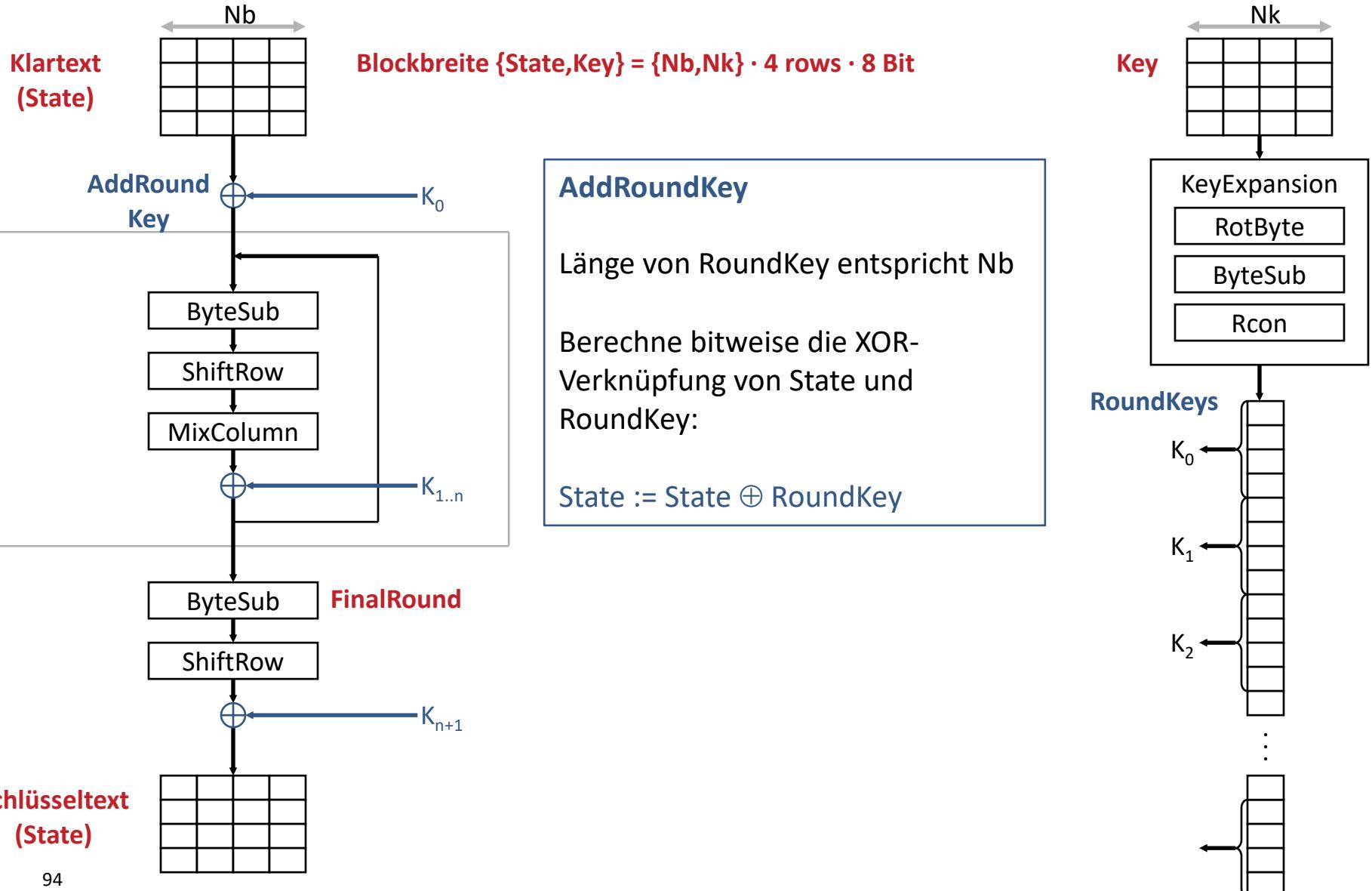
mit $d(x) = '0B' x^3 + '0D' x^2 + '09' x + '0E'$,

da $('03' x^3 + '01' x^2 + '01' x + '02') \otimes d(x) = '01'$
(neutrales Element bzgl. Multiplikation)

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$	$b_{0,4}$	$b_{0,5}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,4}$	$b_{1,5}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	$b_{2,4}$	$b_{2,5}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	$b_{3,4}$	$b_{3,5}$

Rijndael (AES)



Rijndael (AES)

- KeyExpansion

- für Nk<=6:

```
KeyExpansion(byte Key[4*Nk] word W[Nb*(Nr+1)]){  
    for(i = 0; i < Nk; i++)  
        W[i] = (Key[4*i],Key[4*i+1],Key[4*i+2],Key[4*i+3]);  
    for(i = Nk; i < Nb * (Nr + 1); i++) {  
        temp = W[i - 1];  
        if (i % Nk == 0)  
            temp = ByteSub(RotByte(temp)) ^ Rcon[i / Nk];  
        W[i] = W[i - Nk] ^ temp;  
    }  
}
```

- für Nk >6:

```
KeyExpansion(byte Key[4*Nk] word W[Nb*(Nr+1)]) {  
    for(i = 0; i < Nk; i++)  
        W[i] = (key[4*i],key[4*i+1],key[4*i+2],key[4*i+3]);  
    for(i = Nk; i < Nb * (Nr + 1); i++) {  
        temp = W[i - 1];  
        if (i % Nk == 0)  
            temp = ByteSub(RotByte(temp)) ^ Rcon[i / Nk];  
        else if (i % Nk == 4)  
            temp = ByteSub(temp);  
        W[i] = W[i - Nk] ^ temp;  
    }  
}
```

Rijndael (AES)

- RotByte: zyklische Schiebeoperation (byteweise links)
- **ByteSub** (wie bei Rundentransformation)
- $Rcon[i] = (RC[i], 0x00, 0x00, 0x00)$ mit
 - $RC[1] = 1$
 - $RC[i] = 2 \cdot RC[i-1]$ für $i > 1$ und $RC[i-1] < 0x80$
 - $RC[i] = 2 \cdot RC[i-1] \text{ XOR } 0x11$ für $i > 1$ und $RC[i-1] \geq 0x80$

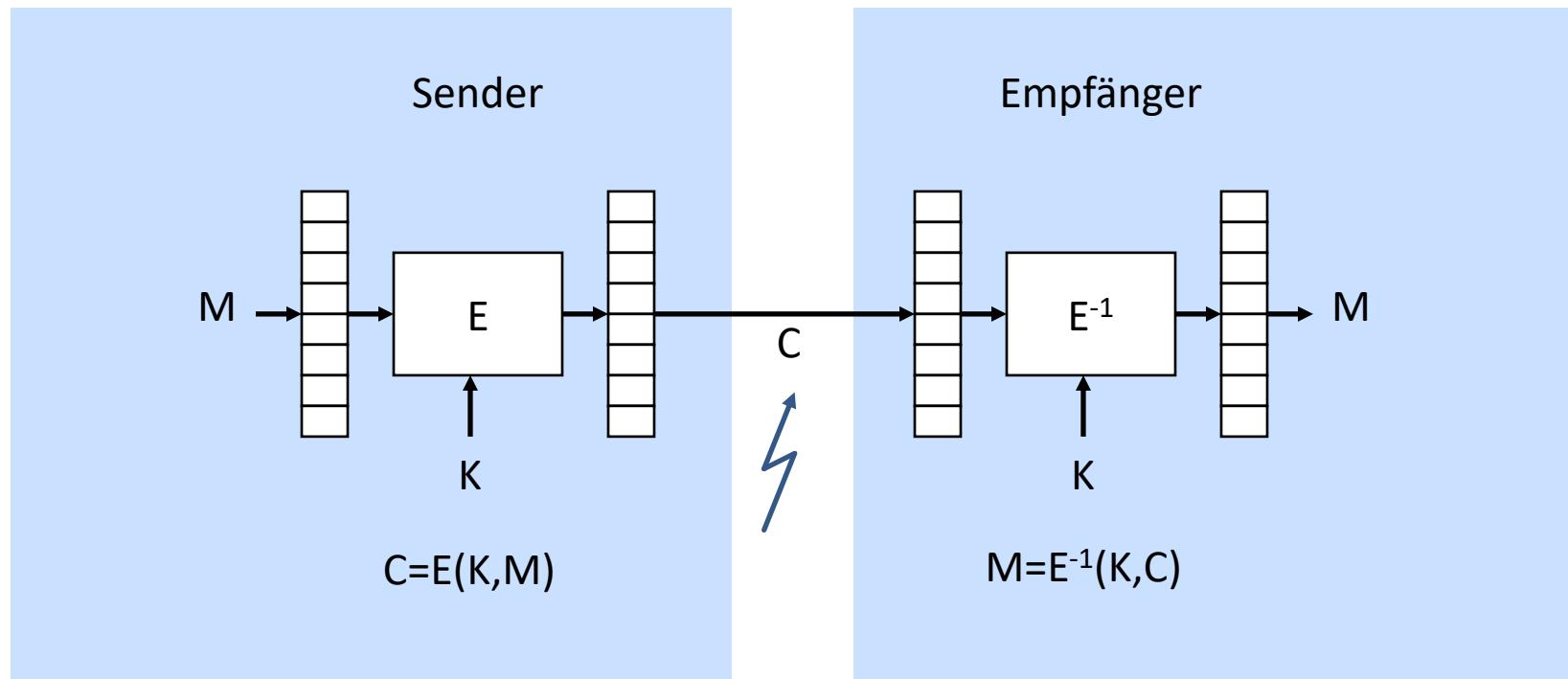
i	1	2	3	4	5	6	7	8	9	10
$RC[i]$	01	02	04	08	10	20	40	80	1B	36

- RoundKey Selection
 - fortlaufende Auswahl
 - Beispiel für $Nb = 6$ und $Nk = 4$:

W0	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	...
Round Key 0						Round Key 1						...			

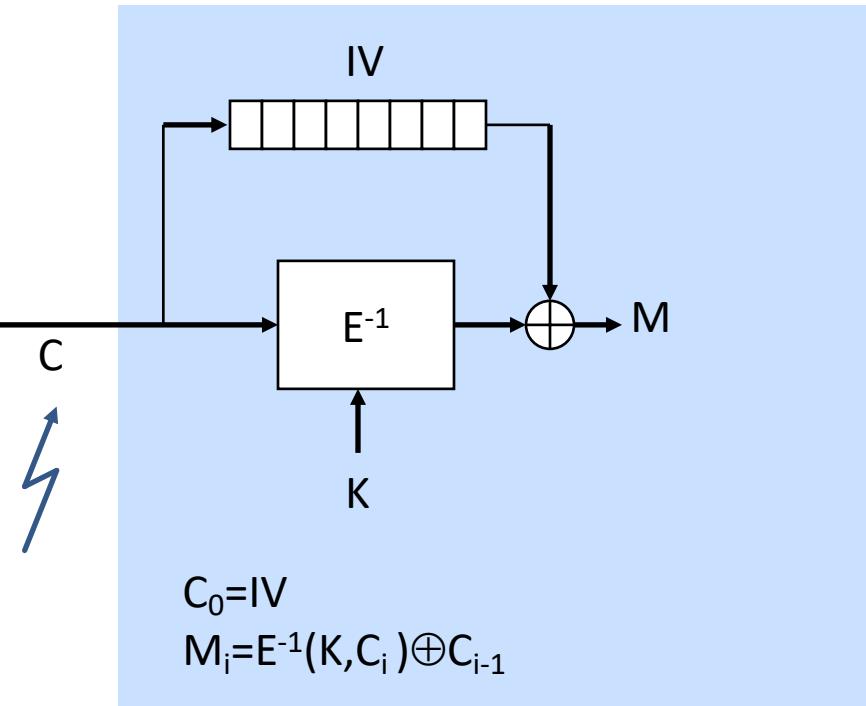
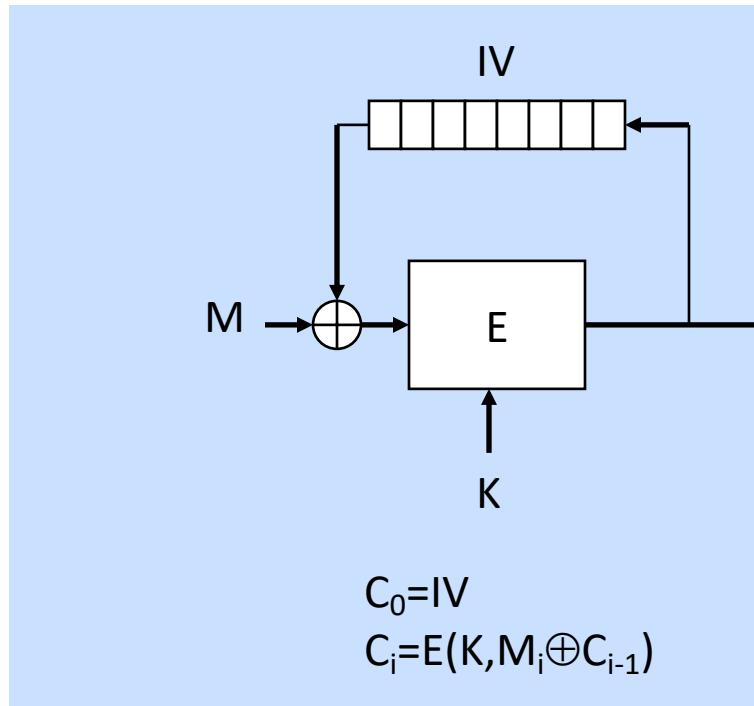
Betriebsarten von Blockchiffren

- Electronic Code Book (ECB)

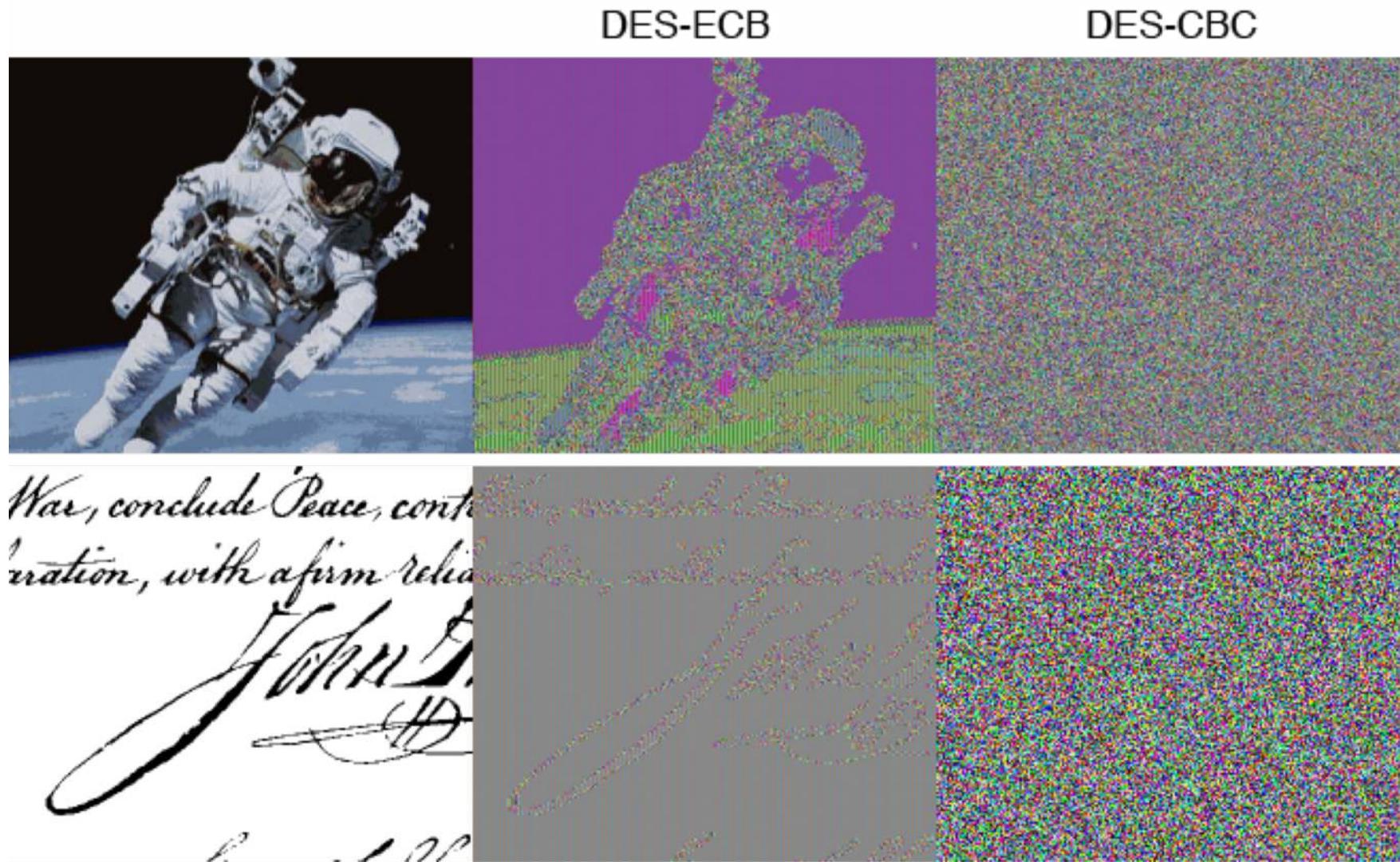


Betriebsarten von Blockchiffren

- Cipher Block Chaining (CBC)



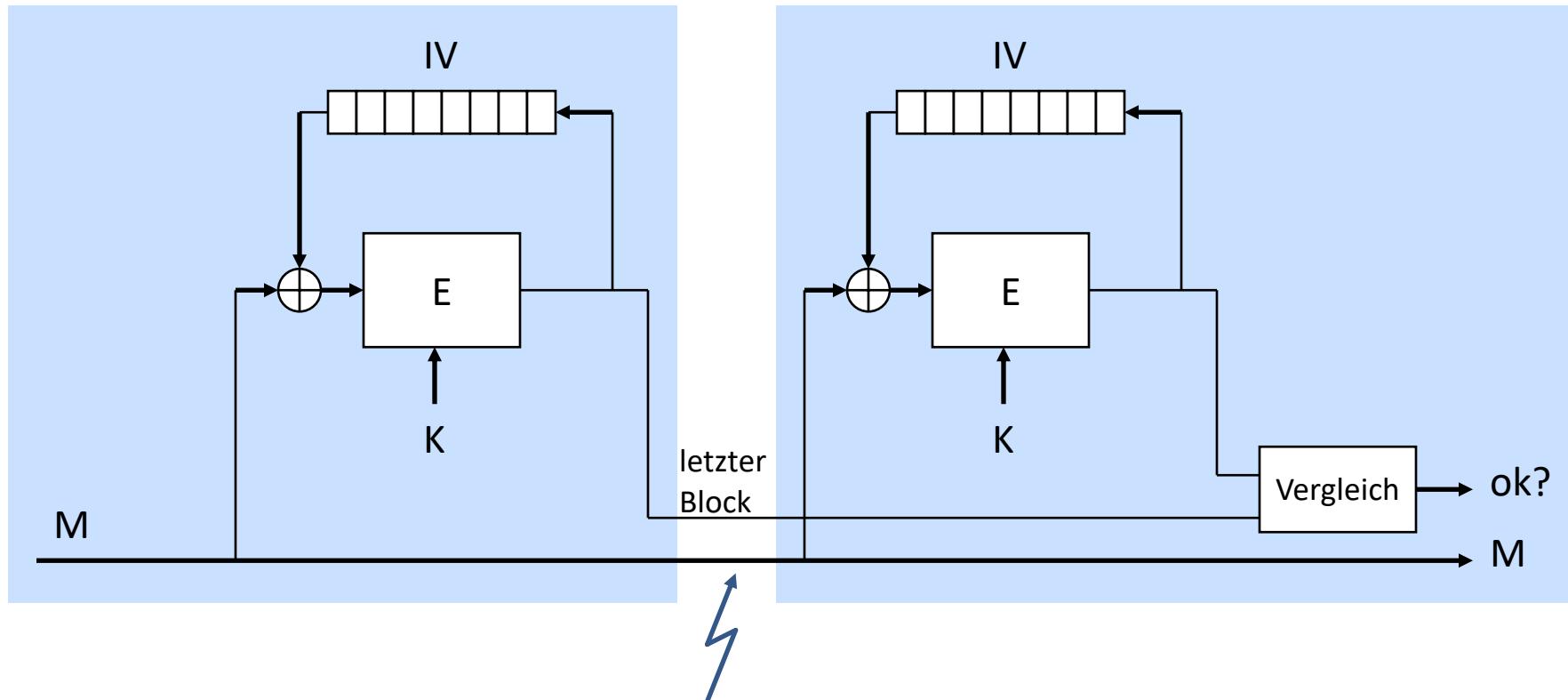
ECB und CBC im visuellen Vergleich



<http://gustlik.wordpress.com/2008/10/15/importance-of-block-cipher-modes/>

Betriebsarten von Blockchiffren

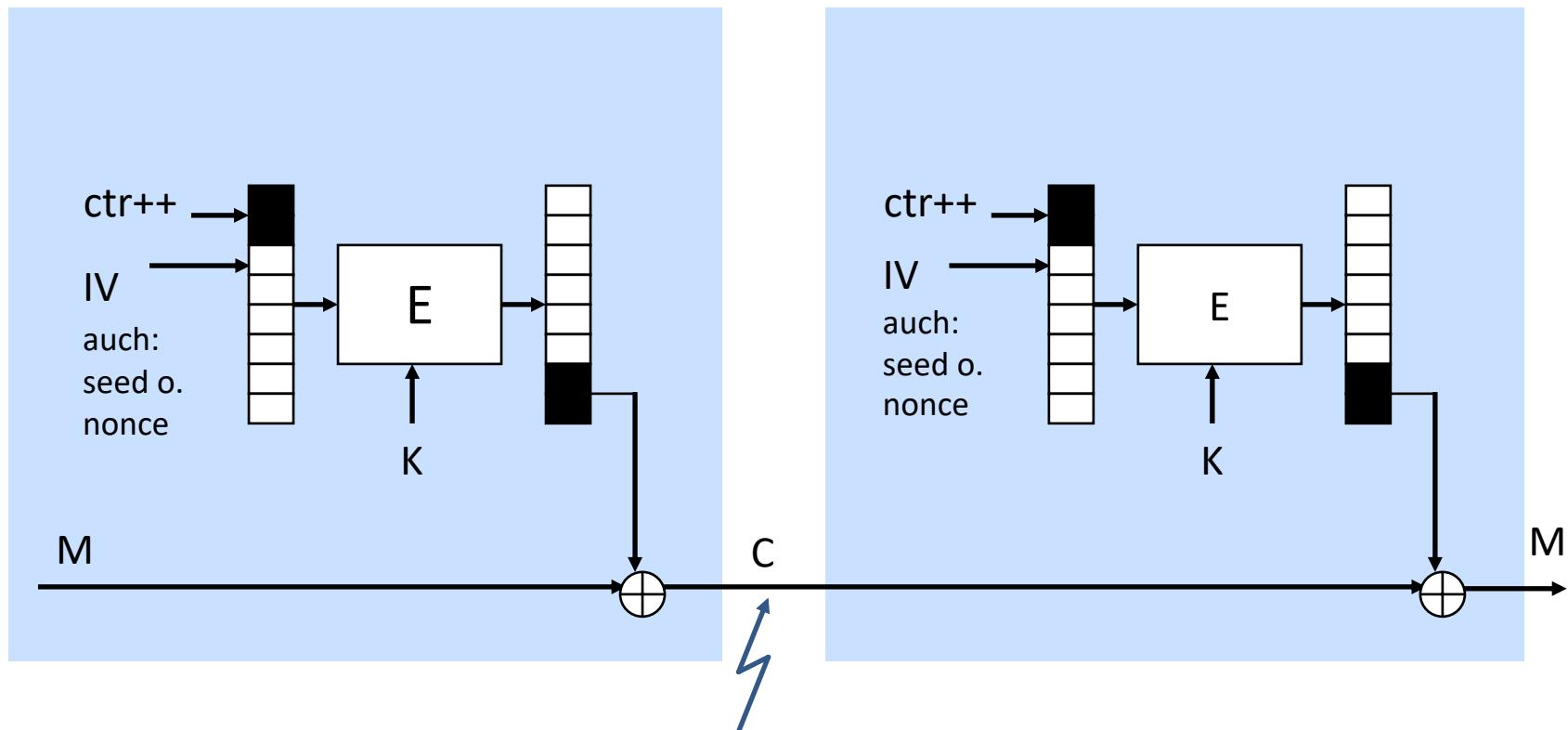
- CBC zur Authentikation (auch: CBCAuth, CBC-MAC)



- Nur anwenden auf Nachrichten fester Länge, da ansonsten Length Extension Attack möglich

Betriebsarten von Blockchiffren

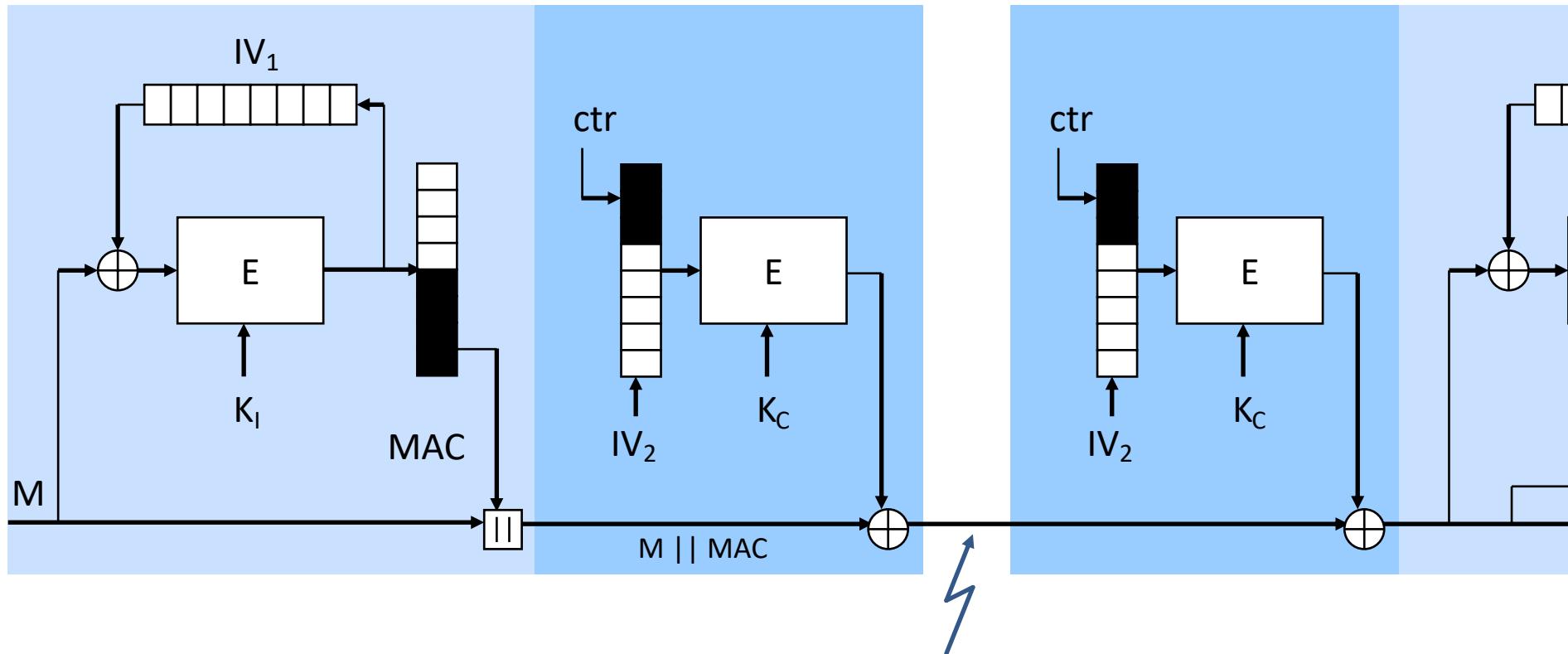
■ Counter Mode



- IV stets zufällig wählen!

Betriebsarten von Blockchiffren

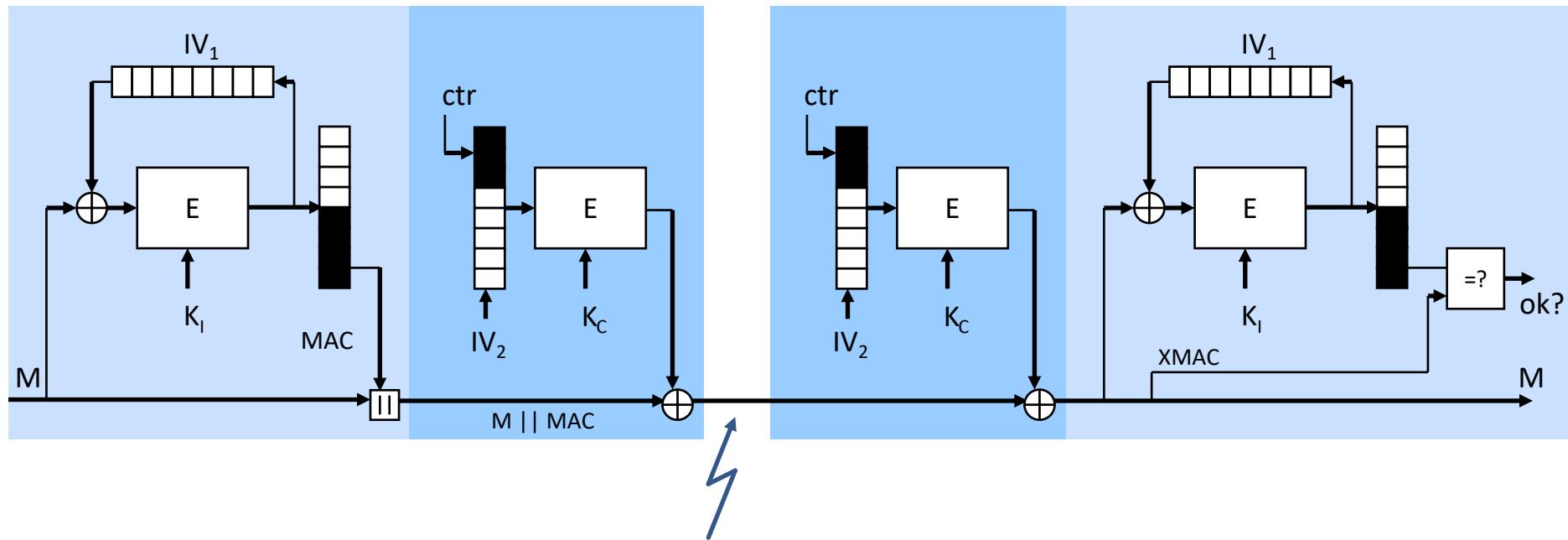
- CCM (Counter with CBC-MAC)



- CCM als Kombination von CBC-MAC und Counter Mode
- angewendet bei WPA2 mit AES als Blockchiffre und $|MAC|=64\text{Bit}$

Betriebsarten von Blockchiffren

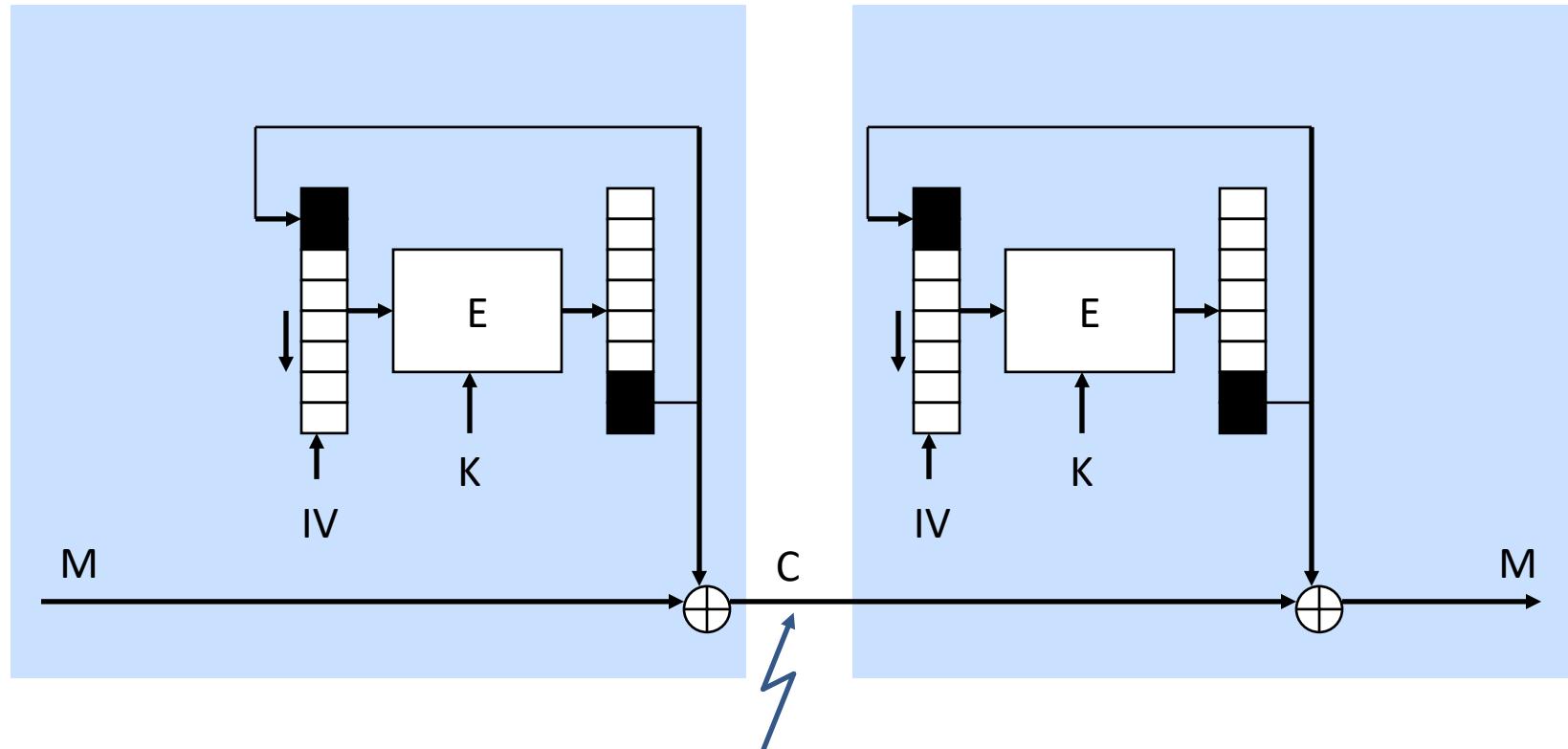
- CCM (Counter with CBC-MAC)



- CCM als Kombination von CBC-MAC und Counter Mode
- angewendet bei WPA2 mit AES als Blockchiffre und $|MAC|=64\text{Bit}$

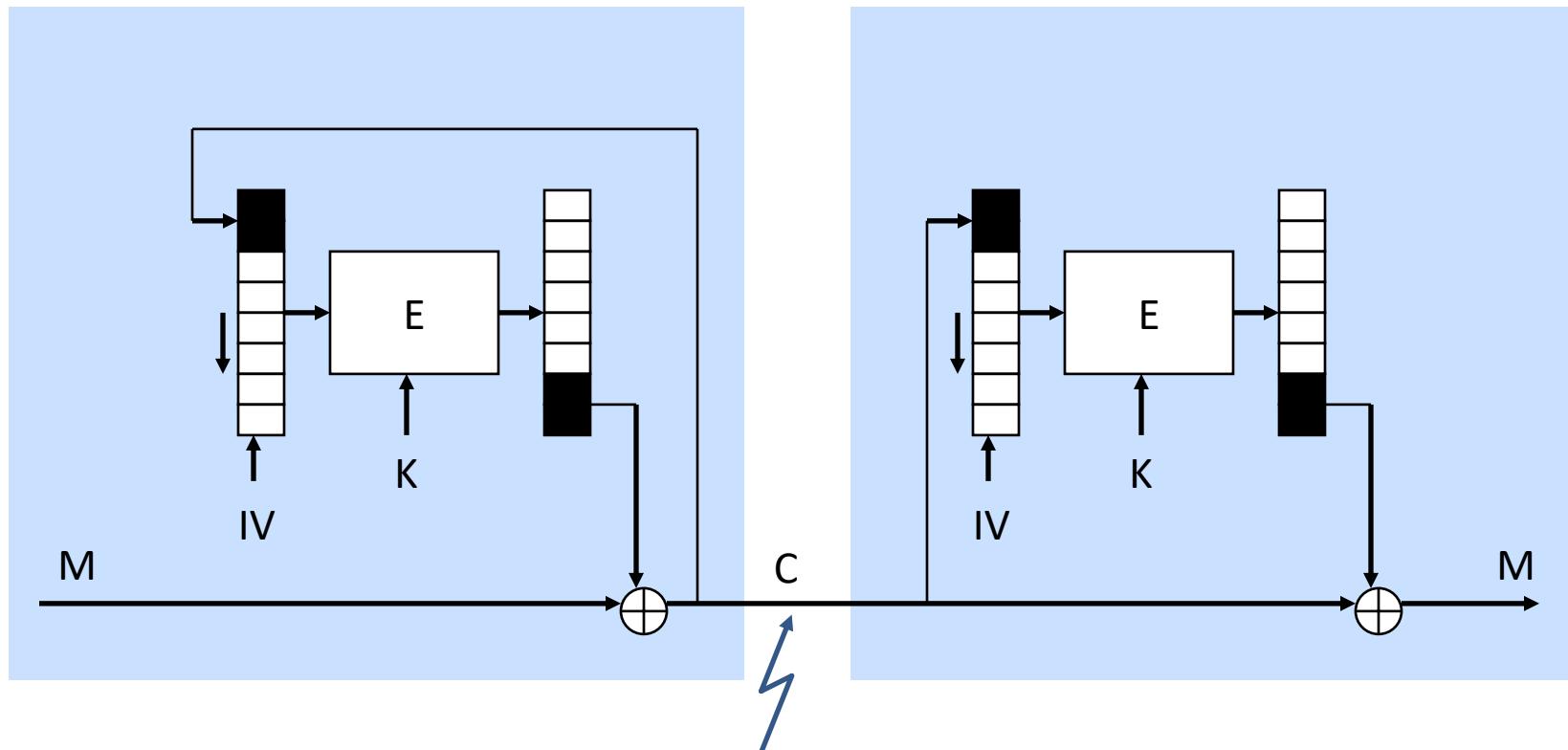
Betriebsarten von Blockchiffren

- Output Feedback (OFB)



Betriebsarten von Blockchiffren

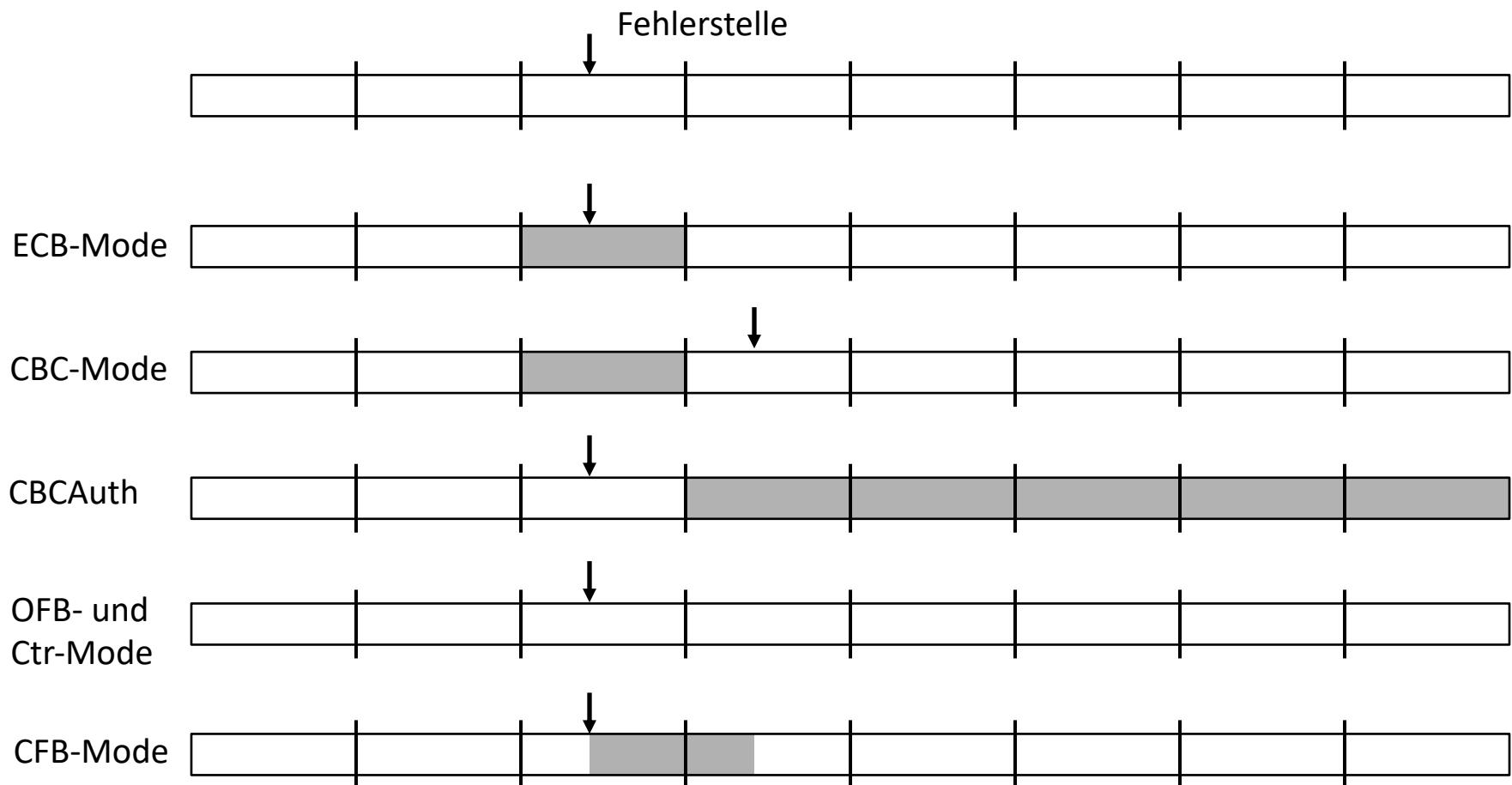
- Cipher Feedback (CFB)



- CFB hat heute eher historische Bedeutung.

Betriebsarten von Blockchiffren

■ Fehlerfortpflanzung

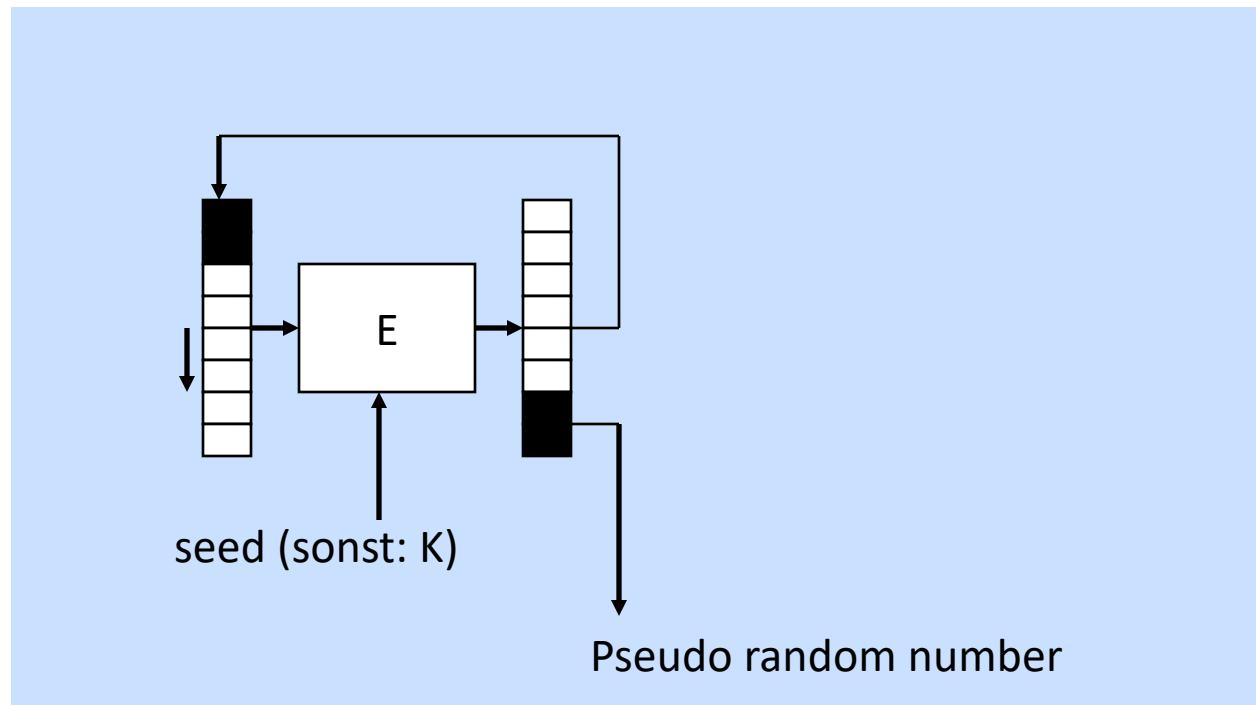


Betriebsarten von Blockchiffren

Modus	Vorteile	Nachteile
ECB	<ul style="list-style-type: none"> • Direktzugriff möglich • keine Fehlerfortpflanzung bei additiven Fehlern 	<ul style="list-style-type: none"> • Fehlerfortpflanzung in alle nachfolgenden Blöcke bei Synchronisationsfehlern • unerkennbare additive Veränderungen möglich • gezieltes Einfügen und Entfernen von Blöcken möglich • gleiche Klartextblöcke liefern gleiche Chiffretextblöcke • Codebuchanalyse möglich
CBC	<ul style="list-style-type: none"> • gleiche Klartextblöcke liefern unterschiedliche Chiffretextblöcke • Manipulationen sind erkennbar • Kryptoanalyse erschwert gegenüber ECB-Modus 	<ul style="list-style-type: none"> • Fehlerfortpflanzung in alle nachfolgenden Blöcke bei Synchronisationsfehlern • Fehlerfortpflanzung in den Folgeblock bei additiven Fehlern • kein Direktzugriff möglich
OFB, Counter	<ul style="list-style-type: none"> • keine Fehlerfortpflanzung bei additiven Fehlern 	<ul style="list-style-type: none"> • Fehlerfortpflanzung in alle nachfolgenden Bits bei Synchronisationsfehlern • unerkennbare additive Veränderungen möglich • geringere Verschlüsselungsrate pro DES-Aufruf als ECB- und CBC-Modus (abh. von Bitbreite) • kein Direktzugriff möglich
CFB	<ul style="list-style-type: none"> • Schlüsselstrom abhängig von Klartextstrom • Kryptoanalyse erschwert gegenüber OFB-Modus • Manipulationen sind erkennbar • selbstsynchronisierender Modus 	<ul style="list-style-type: none"> • Fehlerfortpflanzung in den Folgeblock bei additiven Fehlern • geringere Verschlüsselungsrate pro DES-Aufruf als ECB- und CBC-Modus (abh. von Bitbreite) • kein Direktzugriff möglich

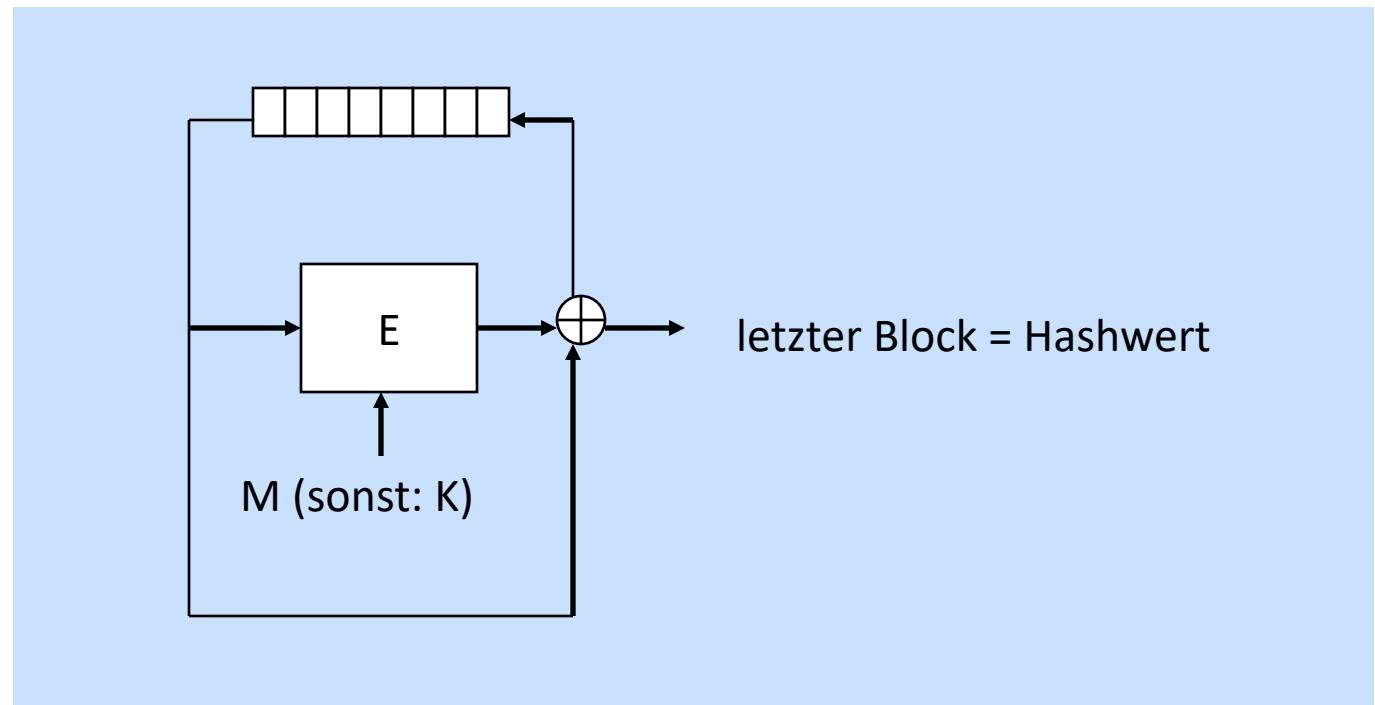
Konstruktionen aus einer symmetrischen Blockchiffre

- Pseudozufallszahlengenerator



Konstruktionen aus einer symmetrischen Blockchiffre

- Hashfunktion
 - Aus Sicherheitsgründen sollte die Schlüssellänge nicht wesentlich länger sein als die Blocklänge

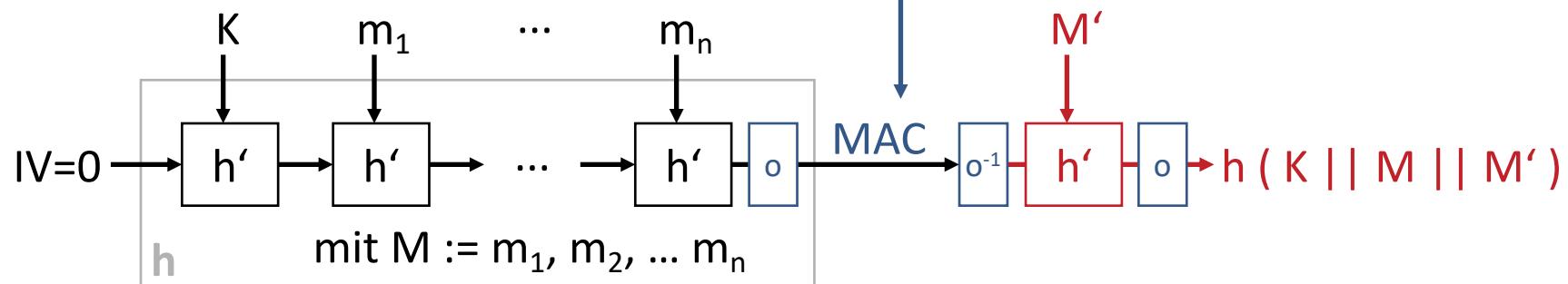


Unsichere Konstruktion von MACs aus Hashfunktionen

- Häufig verwendet, aber bei Verwendung von iterierten Hashfunktionen wie MD5, SHA-1, SHA-256/512 unsicher: $\text{MAC} = h(K || M)$

- Length Extension Attack (Skizze)

- existenzieller Angriff
 - Gegeben sind MAC, M.
 - Angreifer berechnet ohne Kenntnis von K: $h(K || M || M')$



Aufgrund der iterierten Anwendung einer (internen) Struktur von h lässt sich mit einem M' »weiterrechnen« und so ein gültiger MAC erzeugen.

(1) https://github.com/iagox86/hash_extender und (2) <https://github.com/bwall/HashPump> sind Beispielimplementierungen für Length Extension Attack.

(Un)sichere Konstruktion von MACs aus Hashfunktionen

- Naive, aber u.U. bereits sichere Abhilfe:
 - K nicht nur M voranstellen, sondern auch M nachstellen:
$$\text{MAC} = h(K \parallel M \parallel K)$$
 - Length Extension Attack sowohl am Anfang als auch am Ende von M wird erschwert
- Restproblem:

Zumindest für $\text{MAC} = h(M \parallel K)$ wurde gezeigt, dass, wenn ein Angreifer eine Kollision für zwei ungleiche Nachrichten M_1 und M_2 mit $h(M_1)=h(M_2)$ findet, dann auch für $h(M_1 \parallel K) = h(M_2 \parallel K)$ leicht eine Kollision konstruiert werden kann.

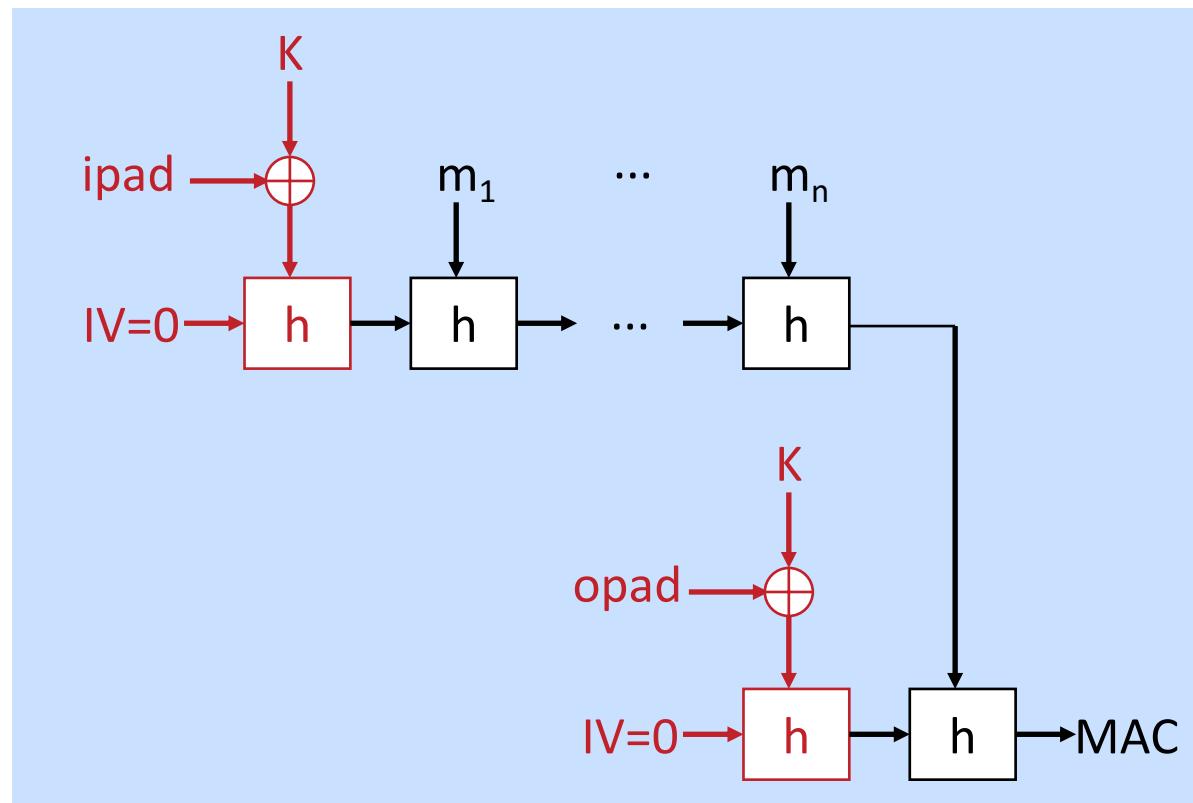
M. Bellare, R. Canetti, H. Krawczyk: Keying hash functions for message authentication. Proc. Crypto 96, LNCS 1109, Springer, 1996, 1-15

Sichere Konstruktion von MACs aus Hashfunktionen: HMAC

- (Keyed)-Hash MAC (HMAC):

$$\text{MAC} = h((K \oplus \text{opad}) || h((K \oplus \text{ipad}) || M))$$

mit $\text{ipad}=0x36\ldots0x36$ und $\text{opad}=0x5c\ldots5c$ nach RFC 2104



Mathematische Grundlagen asymmetrischer Systeme

- Modulo-Rechnung
 - Grundlagen
 - Erweiterter Euklidscher Algorithmus
- Systeme auf der Basis des diskreten Logarithmus
 - primitive Wurzel
 - diskreter Logarithmus Problem
- Systeme auf der Basis der Faktorisierungsannahme
 - Faktorisierungsannahme
 - Primzahlzerlegung



Erweiterter Euklidscher Algorithmus

- Anwendung
 - Bestimmung von $\text{ggT}(a,b)$ und Ermittlung des multiplikativen Inversen von b im Restklassenring modulo a , d. h. zur Berechnung von b^{-1} aus der Beziehung $b \cdot b^{-1} = 1 \bmod a$
- Algorithmus
 - Seien $a, b \in \mathbb{N}+1$, $a > b$
 - Setze $r_{-2} = a$ $s_{-2} = 0$ $t_{-2} = 1$
 $r_{-1} = b$ $s_{-1} = 1$ $t_{-1} = 0$
 - Berechne c_k, r_k, s_k und t_k nach folgenden Beziehungen:
 - $c_k = r_{k-2} \text{ DIV } r_{k-1}$
 - $r_k = r_{k-2} \text{ MOD } r_{k-1}$
 - $s_k = c_k s_{k-1} + s_{k-2}$
 - $t_k = c_k t_{k-1} + t_{k-2}$
 - Abbruchbedingung : $r_k = 0$
 - Es gilt: $b \cdot s_{k-1} - a \cdot t_{k-1} = (-1)^k \cdot r_{k-1}$
 - Ergebnisse: $r_{k-1} = \text{ggT}(a,b)$
 $s_{k-1} \cdot b = (-1)^k \bmod a$, falls $\text{ggT}(a,b) = 1$

Erweiterter Euklidscher Algorithmus

$$c_k = r_{k-2} \text{ DIV } r_{k-1} \quad r_k = r_{k-2} \text{ MOD } r_{k-1} \quad s_k = c_k s_{k-1} + s_{k-2} \quad t_k = c_k t_{k-1} + t_{k-2}$$

■ Beispiel 1

- Gegeben: $a=10 \ b=4$
- Gesucht: $\text{ggt}(a,b)$

k	c_k	r_k	s_k	t_k
-2		$10 = a$	0	1
-1		$4 = b$	1	0
0	2	$2 = \text{ggt}$	2	1
1	2	0 (Abbruch)		

Erweiterter Euklidscher Algorithmus

$$c_k = r_{k-2} \text{ DIV } r_{k-1} \quad r_k = r_{k-2} \text{ MOD } r_{k-1} \quad s_k = c_k s_{k-1} + s_{k-2} \quad t_k = c_k t_{k-1} + t_{k-2}$$

■ Beispiel 2

- Gegeben:
 $p=53 \quad q=61 \quad n=p \cdot q=3233 \quad \Phi(n)=(p-1) \cdot (q-1)=3120 \quad c=523$
- Gesucht:
 $c \cdot c^{-1} = 1 \pmod{\Phi(n)}$, d.h. Multiplikatives Inverses zu $c \pmod{\Phi(n)}$

k	c_k	r_k	s_k	t_k
-2		$3120 = a = \Phi$	0	1
-1		$523 = b = c$	1	0
0	5	505	5	1
1	1	18	6	1
2	28	1 = ggt	173	29
3	18	0 (Abbruch)		

Erweiterter Euklidscher Algorithmus

■ Beispiel 2 (Forts.)

- Es gilt: $s_{k-1} \cdot b = (-1)^k \pmod{a}$
 $173 \cdot 523 = (-1)^3 \pmod{3120}$
 $90479 = -1 \pmod{3120}$
 $3119 = -1 \pmod{3120}$
 $-1 = -1 \pmod{3120}$

- Da $(-1)^k = (-1)^3 = -1$, muss noch mit -1 multipliziert werden.

$$\begin{aligned}-s_{k-1} \cdot b &= -(-1)^k \pmod{a} \\ c &= -s_{k-1} = -173 = -173 + a \\ c &= 2947\end{aligned}$$

Eulersche- Φ -Funktion

- Def. Eulersche Φ -Funktion
 - Für eine beliebige ganze Zahl n bildet die Menge \mathbf{Z}_n^* der ganzen Zahlen modulo n , die zu n teilerfremd sind, eine multiplikative Gruppe. Die Ordnung dieser Gruppe ist $\Phi(n)$.
 - Beispiel: $\Phi(9) = 6 \quad \mathbf{Z}_9^* = \{1, 2, 4, 5, 7, 8\}$
 - Für den Sonderfall $n = p \in \mathbf{P}$ gilt $\Phi(p) = p-1$.
- Satz von Euler
 - Für ein beliebiges x mit $(1 \leq x < n)$, das zu n teilerfremd ist bzw. $x \in \mathbf{Z}_n^*$, gilt $x^{\Phi(n)} = 1 \pmod{n}$.
- Euler-Fermat-Identität (kleiner Satz von Fermat)
 - Mit $\Phi(p) = p-1$ ($p \in \mathbf{P}$) folgt aus dem Satz von Euler:
$$x^{p-1} = 1 \pmod{p} \quad (1 \leq x \leq p-1).$$
- Wenn n das Produkt zweier Primzahlen $n = p \cdot q$ ist, gilt
$$x^{\Phi(p \cdot q)} = x^{(p-1) \cdot (q-1)} \pmod{p \cdot q}.$$

Primitive Wurzel

- **Definition**
 - Eine beliebige ganze Zahl im Bereich $1 \leq a < n$ heißt primitive Wurzel, wenn gilt
 - $\text{ggT}(a,n) = 1$ *und*
 - $a^d \neq 1 \pmod{n}$ für alle d mit der Bedingung $1 \leq d < \Phi(n)$
- **Theorem**
 - Die ganze Zahl n hat genau dann eine primitive Wurzel, wenn $n = 1, 2$ oder 4 ist oder die Form p^k oder $2p^k$ hat, wobei p eine ungerade Primzahl ist.
 - Wenn n eine primitive Wurzel hat, dann hat n genau $\Phi(\Phi(n))$ primitive Wurzeln.
- **Vermutung**
 - Jede ganze Zahl, die keine Quadratzahl ist, ist die primitive Wurzel einer Primzahl.

Diskreter Logarithmus

- **Definition diskreter Logarithmus**
 - Sei p eine beliebige ganze Zahl, die eine primitive Wurzel a hat. Wenn für ein beliebiges c mit $0 \leq c < \Phi(p)$
$$b = a^c \bmod p$$
gilt, dann ist c der diskrete Logarithmus zur Basis a modulo p oder auch
$$c = \log_a b \bmod p.$$
- **Problemstellung für den Angreifer**
 - Öffentlich bekannt sind a, b, p .
 - Geheim ist c . Erfährt ein Angreifer c , ist das System gebrochen.
 - Folglich möchte ein Angreifer c ermitteln.
- **Beispiel**
 - $p = 3137$ und $a = 577$ öffentlich; $c = 1374$ geheim
 - $b = a^c \bmod p = 858$ öffentlich
 - $c = \log_a b \bmod p = \log_{577} 858 \bmod 3137 = ?$ (Angreifersicht)

Diskreter Logarithmus

- Algorithmen zur Berechnung des diskreten Logarithmus
 - Baby-Step, Giant-Step, Index-Calculus-Alg., Adleman-Alg.
 - Laufzeit zur Berechnung des diskreten Log. mod p mit $p \in P$
 $e^{(1+O(1))(\log p \cdot \log(\log p))^{1/2}}$
 - Rechenzeiten bei 10^8 Operationen pro sek. für verschiedene Größenordnungen von p :

p	Anzahl Ops.	benötigte Zeit in Jahren
$\approx 10^{100}$	$7,9 \cdot 10^{22}$	$2,5 \cdot 10^7$
$\approx 10^{200}$	$1,8 \cdot 10^{34}$	$5,7 \cdot 10^{19}$
$\approx 10^{300}$	$1,8 \cdot 10^{43}$	$5,7 \cdot 10^{28}$
$\approx 10^{400}$	$9,5 \cdot 10^{50}$	$4,8 \cdot 10^{36}$
$\approx 10^{500}$	$7,4 \cdot 10^{57}$	$4,8 \cdot 10^{43}$

Vergleich: Logarithmus

$\log_a b$ ($a>0; a\neq 1; b>0$) ist diejenige reelle Zahl c , für die gilt $a^c=b$.

$c = \log_a b$ wird z.B. gelöst mit
$$\log_a b = \frac{\log_x b}{\log_x a}$$

- **Beispiel 1:** Wieviel Bit benötigt man, um die Zahlen zwischen 0 und 255 binär zu kodieren?

$$\log_2 256 = \frac{\ln 256}{\ln 2} = 8 \text{ Bit}$$

- **Beispiel 2:** Wieviel Bit benötigt man, um die Zahlen bis 10^{200} im Binärcode darzustellen?

$$10^{200} \leq 2^x \quad x \geq \log_2 10^{200} = \frac{\log_{10} 10^{200}}{\log_{10} 2} = \frac{200}{\log_{10} 2} \quad x \geq 665 \text{ Bit}$$

- **Beispiel 3:** $a = 577; b = 858; c = ?$

$$c = \log_a b = \log_{577} 858 = \frac{\ln 858}{\ln 577} = 1,0624$$

Faktorisierungsannahme

- Seien p und q zwei große ($|p| \approx |q| \approx 500 \dots 1500$ Bit), unabhängig und zufällig gewählte Primzahlen. p und q werden geheim gehalten. Das Produkt n aus p und q wird veröffentlicht: $n=p \cdot q$.
- Annahme: Für jeden schnellen Faktorisierungsalgorithmus $F(n)$ wird die Wahrscheinlichkeit, dass $F(n)$ eine Zahl $n=p \cdot q$ tatsächlich faktorisieren kann, schnell kleiner, je größer die Länge $|p|$ und $|q|$ der Faktoren ist.
 - Es ist zwar mit vernünftigem Aufwand möglich, Primzahlen p und q zu finden und diese zu multiplizieren. Es ist aber nicht mit vernünftigem Aufwand möglich, die Primfaktoren von n zu finden.
 - Die öffentlich ausführbare Funktion (Verschlüsseln bzw. Signaturtest) kommt mit dem öffentlichen n aus. Die private Funktion (Entschlüsseln bzw. Erzeugen einer Signatur) nutzt p und q .
 - Dass Faktorisierung schwer ist, ist bisher nicht bewiesen.

Diffie-Hellmann-Key-Exchange

A will B die Nachricht m
schicken.

B:

$p_B \in P$ und a primitive Wurzel von p_B wählen

x_B mit $1 \leq x_B \leq p_B-1$ wählen

$y_B = a^{x_B} \text{ mod } p_B$ berechnen

x_B bleibt
geheim!

a, p_B und y_B sind auf key server veröffentlicht

A:

liest Eintrag für B: a, p_B und y_B

x_A mit $1 \leq x_A \leq p_B-1$ geheim wählen

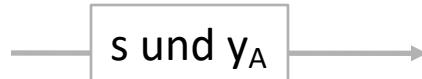
$y_A = a^{x_A} \text{ mod } p_B$ berechnen

Key Agreement:

$k_{AB} = y_B^{x_A} \text{ mod } p_B$ berechnen

Verschlüsselung:

$s = E(k_{AB}, m)$



B:

$k_{BA} = y_A^{x_B} \text{ mod } p_B$ berechnen

$m = E^{-1}(k_{BA}, s)$ entschlüsseln

Diffie-Hellmann-Key-Exchange mit Forward Secrecy

A will B die Nachricht m schicken.

B:

$p_B \in P$ und a primitive Wurzel von p_B wählen

x_B mit $1 \leq x_B \leq p_B-1$ wählen

$y_B = a^{x_B} \text{ mod } p_B$ berechnen

x_B bleibt geheim!

a, p_B und y_B

Wenigstens y_B wird je Sitzung neu erzeugt.

A:

liest Eintrag für B: a, p_B und y_B

x_A mit $1 \leq x_A \leq p_B-1$ geheim wählen

$y_A = a^{x_A} \text{ mod } p_B$ berechnen

Key Agreement:

$k_{AB} = y_B^{x_A} \text{ mod } p_B$ berechnen

Verschlüsselung:

$s = E(k_{AB}, m)$



B:

$k_{BA} = y_A^{x_B} \text{ mod } p_B$ berechnen

$m = E^{-1}(k_{BA}, s)$ entschlüsseln

Diffie-Hellmann-Key-Exchange

- Berechnung des Kommunikationsschlüssels

k_{AB} bzw. k_{BA} erfolgt durch

$$k_{AB} = y_B^{x_A} \bmod p_B \text{ bei A und}$$

$$k_{BA} = y_A^{x_B} \bmod p_B \text{ bei B.}$$

- Nachweis

$$k_{AB} = y_B^{x_A} = (a^{x_B})^{x_A} = (a^{x_A})^{x_B} = y_A^{x_B} = k_{BA} \pmod{p_B}$$

- Angreifer muss zum Brechen x_A oder x_B ermitteln, d.h. er muss berechnen:

$$x_A = \log_a y_A \bmod p_B \quad \text{oder}$$

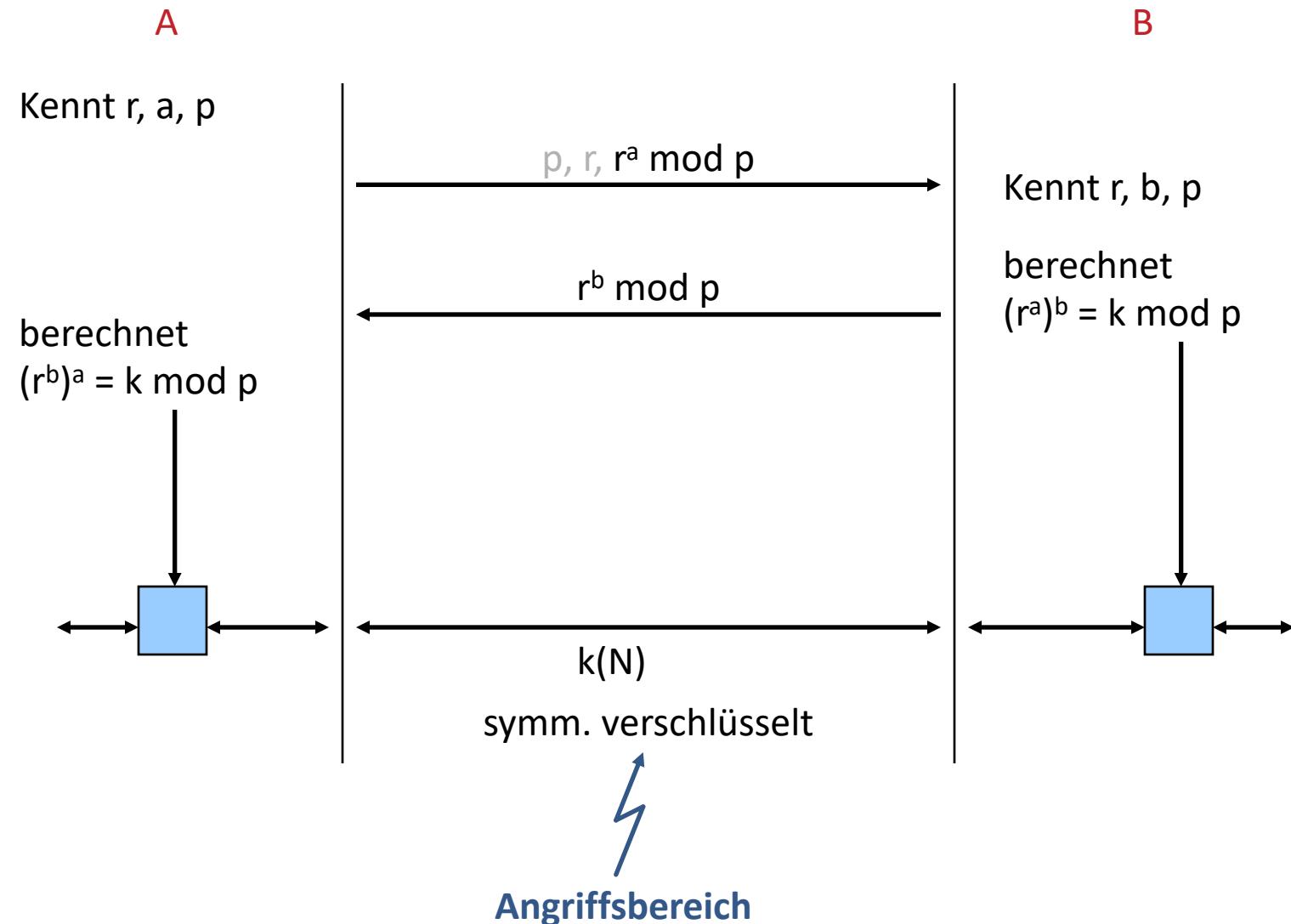
$$x_B = \log_a y_B \bmod p_B$$

- Sicherheit

- Verfahren ist sicher gegen einen passiven Angreifer.
- Verfahren ist unsicher gegen einen aktiven Angreifer (Maskerade).

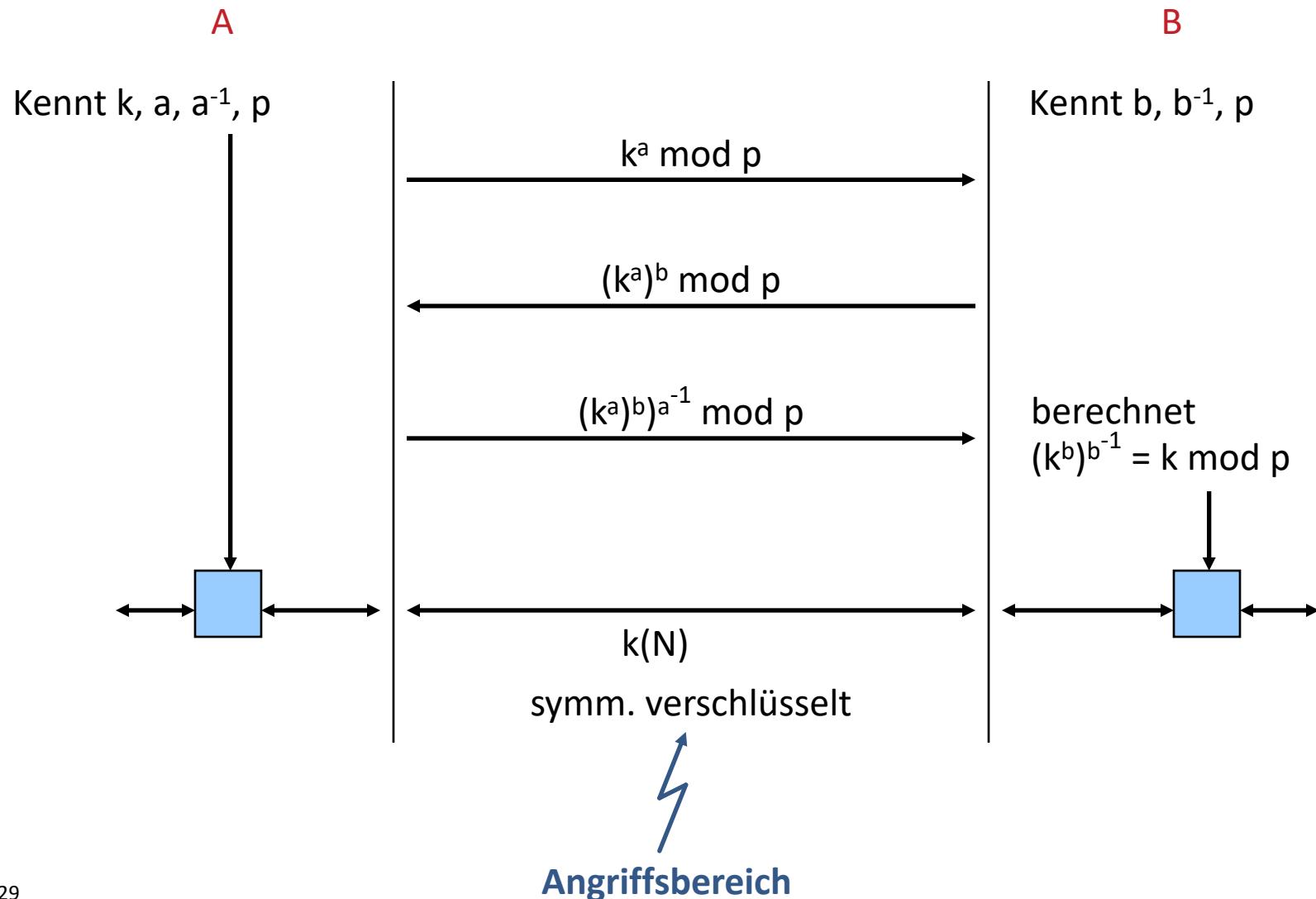
Asymmetrische Schlüsselvereinbarung (v1)

- Wert von k kann nicht festgelegt werden (ist zufällig)



Asymmetrische Schlüsselvereinbarung (v2)

- Teilnehmer A kann k festlegen

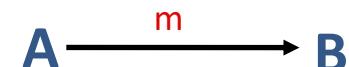


Konzelationssystem nach ElGamal

- **Schlüsselgenerierung**
 - wähle global:
 - $p \in \mathbb{P}$ öffentlich
 - a primitive Wurzel von p öffentlich
 - jeder TIn. wählt:
 - geheimen Schlüssel k_i ($k_i < p-1$) geheim
 - berechnet $-k_i \bmod (p-1)$ geheim
 - $y_i = a^{-k_i} \bmod p$ (*) öffentlich

Verschlüsselung

- A will Nachricht m ($m < p$) an B schicken
- A besorgt sich p, a, y_B
- A wählt Zufallszahl z ($z < p$)
- A berechnet $c = y_B^z \cdot m \bmod p$
- A sendet an B: $a^z \bmod p, c$



Entschlüsselung

- B berechnet $m^* = (a^z)^{k_B} \cdot c \bmod p$

ElGamal basiert auf der Schwierigkeit der Berechnung des diskreten Log

Konzelationssystem nach ElGamal: Beispiel

- **Schlüsselgenerierung**
 - Global öffentlich: $p = 3137$ und $a = 577$
 - Teilnehmer B: $k_B = 1762$ geheim
 - $-k_B \bmod (p-1) = -1762 \bmod 3136 = -1762 + 3136 = 1374$ geheim
 - $y_B = a^{-k_B} \bmod p = 577^{1374} \bmod 3137 = 858$
- **Verschlüsselung**
 - A will B vertraulich die Nachricht $m = 2115$ schicken.
 - A wählt $z = 932$ geheim.
 - berechnet $a^z \bmod p = 577^{932} \bmod 3137 = 1852$
 - berechnet $y_B^z \bmod p = 858^{932} \bmod 3137 = 749$
 - berechnet $c = y_B^z \cdot m \bmod p = 749 \cdot 2115 \bmod 3137 = 3087$
 - schickt $a^z = 1852$ und $c = 3087$ an B
- **Entschlüsselung**
 - B berechnet
$$(a^z)^{k_B} \cdot c \bmod p = 1852^{1762} \cdot 3087 \bmod 3137 = 2115$$

Signatursystem nach ElGamal

- Schlüsselgenerierung
 - wähle global:
 - $p \in P$ öffentlich
 - a primitive Wurzel von p öffentlich
 - jeder Tln. wählt:
 - $x_i \in \mathbb{Z}_{p-1}^*$ geheim
 - berechnet $y_i = a^{x_i} \bmod p$ öffentlich
- Signatur
 - A wählt:
 - Zufallszahl k mit $k \in \mathbb{Z}_{p-1}^*$
bzw. k relativ prim zu p-1, d.h. $\text{ggt}(k,p-1)=1$
 - $k^{-1} \bmod (p-1)$
 - $r = a^k \bmod p$
 - $h(m)$ (Hash-Wert von m; $h(m) < p$)
 - löst die Kongruenz
$$h(m) = (x_A \cdot r + k \cdot s) \bmod (p-1)$$
 nach s auf:
$$s = k^{-1}(h(m) - x_A \cdot r) \bmod (p-1)$$
 - A bildet $\text{sig} = (s, r)$
 - A sendet m, sig
- Test
 - B berechnet:
 - $t_1 = a^{h(m)} \bmod p$
 - $t_2 = y_A^r \cdot r^s \bmod p$
 - B vergleicht:
 - $t_1 = t_2 \rightarrow$ gültige Signatur
 - $t_1 \neq t_2 \rightarrow$ ungültige Signatur



- Schlüsselgenerierung

- wähle unabh. und zufällig $p, q \in \mathbb{P}$ mit $|p| \approx |q|$ und $p \neq q$
- berechne $n = p \cdot q$
- wähle c mit $3 \leq c < \Phi(n)$ und
 $\text{ggf}(c, \Phi(n))=1$ mit $\Phi(n) = (p-1)(q-1)$
- berechne d mittels p, q, c als multiplikatives Inverses von $c \bmod \Phi(n)$
 $c \cdot d \equiv 1 \bmod \Phi(n)$

	Konzelationssystem	Signatursystem
öffentl.	c, n	d (hier meist t genannt), n
geheim	d, p, q	c (hier meist s genannt), p, q

RSA basiert auf der
Faktorisierungsannahme

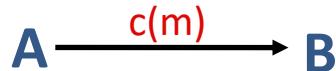
RSA-Verfahren

Rivest, Shamir, Adleman, 1978

- Ein Sicherheitsbeweis von RSA ist bisher nicht bekannt.

Verschlüsselung:

A will Nachricht m ($1 < m < n$) an B
schicken



Signatur:

A will Signatur einer Nachricht m
($1 < m < n$) von B testen



A besorgt sich öffentliche Parameter von B: c bzw. t , sowie n

naiv: $c(m) := m^c \bmod n$

$\text{sig}_s(m) := m^s \bmod n$

sicher: $c(m) := (z, m, h(z, m))^c \bmod n$

$\text{sig}_s(m) := (h(m))^s \bmod n$

Entschlüsselung:

naiv: $m^* = (m^c)^d \bmod n$

Signaturtest:

$m^* = (m^s)^t \bmod n$

sicher: $(z^*, m^*, y) = c(m)^d \bmod n$
 $y =? h(z^*, m^*) \rightarrow \text{out}(m)$

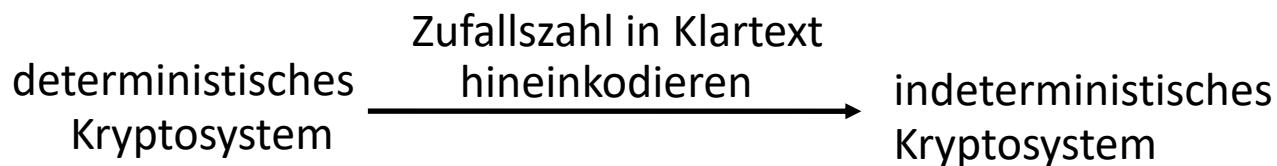
$m^* = ? m' \rightarrow \text{out(ok)}$

$h(m)^* = ((h(m))^s)^t \bmod n$
 $h(m)^* = ? h(m') \rightarrow \text{out(ok)}$

RSA-Verfahren: Angriffe

- **Raten von Klartextblöcken**
 - Angreifer kann wahrscheinliche Klartextblöcke raten, mit c verschlüsseln und mit abgefangenen Schlüsseltextrten vergleichen.

- **Verhinderung**
 - Zufallszahl in Klartext hineinkodieren



RSA besitzt multiplikative Struktur

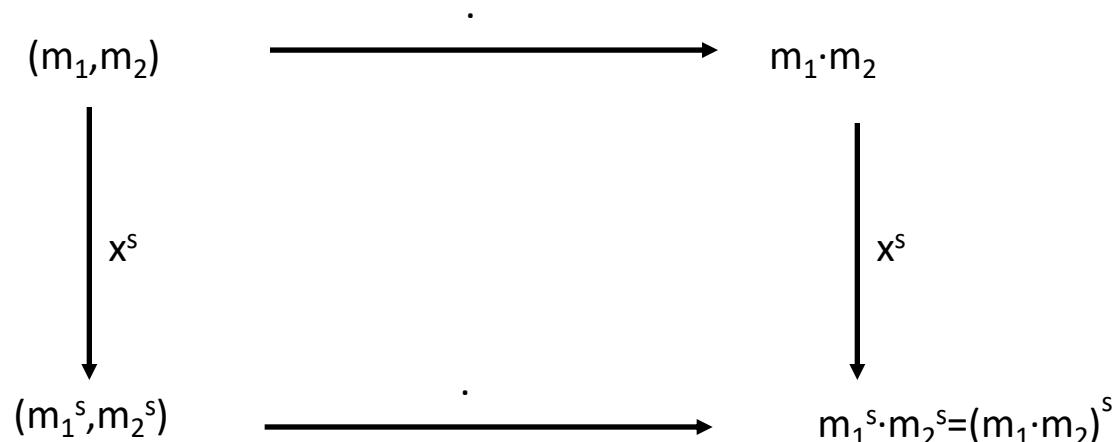
- Passiver Angriff auf naives Signatursystem (David)
- Angenommen, Angreifer kennt zwei Signaturen m_1^s und m_2^s sowie die Nachrichten m_1 und m_2 und kann eine dritte Signatur m_3^s bilden. m_3 ist jedoch nicht beliebig wählbar.

$$m_3^s = m_1^s \cdot m_2^s \bmod n$$

$$m_3 = m_1 \cdot m_2 \bmod n$$

$$\text{gilt, da } m_1^s \cdot m_2^s = (m_1 \cdot m_2)^s$$

- Homomorphismus bezüglich Multiplikation



RSA-Verfahren: Angriffe

- Aktiver Angriff zum selektiven Brechen von RSA nach Judy Moore
 - Angreifer möchte Schlüsseltextblock s_3 entschlüsselt haben
 - wählt Zufallszahl r mit $1 \leq r < n$
 - berechnet multiplikatives Inverses mod n : r^{-1}
 - berechnet $s_2 := s_3 \cdot r^c \pmod{n}$
 - lässt s_2 entschlüsseln, d.h. Angreifer erhält s_2^d
 - weiß $s_2^d \equiv (s_3 \cdot r^c)^d \equiv s_3^d \cdot r^{c \cdot d} \equiv s_3^d \cdot r \pmod{n}$
 - berechnet $s_3^d \equiv s_2^d \cdot r^{-1} \pmod{n}$
- Verhinderung der Angriffe (aktiv und passiv)
 - Konzelation: Hinzunahme eines Redundanzprädikates, z.B. einer Zufallszahl, so dass das Multiplizieren zweier Klartextblöcke keinen dritten Klartextblock mit passender Redundanz ergibt; alternativ: vor Verschlüsselung Hashwert an Nachricht anhängen
 - Signatur: Signatur des Hashwertes $h(m)$ der Nachricht m . h ist eine kollisionsfreie Hashfunktion. Das Finden einer Kollision, d.h. $h(m) = h(m^*)$ mit $m \neq m^*$ ist ein schwieriges Problem.

Angriff wird nutzbar gemacht für blinde Signaturen (Chaum 1985)

Blinde Signatur mit RSA-Verfahren

- Teilnehmer möchte Nachricht m signiert haben, ohne dass der Signierer die Nachricht m selbst zur Kenntnis bekommt
 - wählt Zufallszahl r mit $1 \leq r < n$
 - berechnet multiplikatives Inverses mod n : r^{-1}
 - Blendet die Nachricht m , d.h. berechnet
$$w := m \cdot r^t \pmod{n}$$
 - lässt w signieren, d.h. erhält w^s
 - Er weiß
$$w^s = (m \cdot r^t)^s = m^s \cdot r^{t \cdot s} = m^s \cdot r \pmod{n}$$
 - Entblendet die Nachricht, d.h. berechnet
$$\text{sig}(m) = m^s \cdot r \cdot r^{-1} \pmod{n}$$
- Anwendungsbeispiel: Anonyme digitale Zahlungssysteme

Blinde Signatur mit RSA-Verfahren

- Anwendungsbeispiel: Anonyme digitale Zahlungssysteme
 - Bank erfährt nichts über Zahlungsflüsse ähnlich Bargeld
 - Signierer = Bank
-
- Geld abheben:
 - Kunde schickt geblendete digitale Banknote w an Bank
 - Bank belastet Konto des Kunden mit Gegenwert
 - Bank signiert w und schickt w^s zurück an Kunden
 - Kunde entblendet Banknote und erhält $\text{sig}(m)$
 - Bezahlen:
 - Kunde kauft bei Händler ein und bezahlt mit Banknote $\text{sig}(m)$
 - Händler löst $\text{sig}(m)$ bei Bank ein
 - Bank prüft $\text{sig}(m)$ auf Gültigkeit (korrekte Signatur und nicht bereits eingelöst)
 - Bank schreibt Händler Gegenwert auf seinem Konto gut

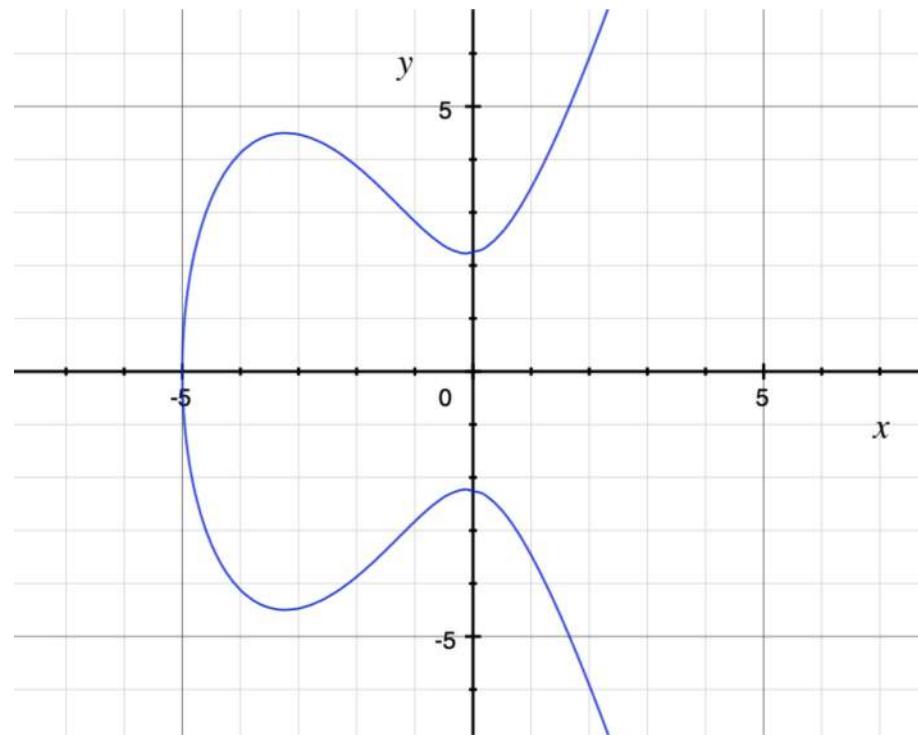
Elliptic Curve Cryptography (ECC)

- Elliptische Kurve
 - Eine elliptische Kurve $E(K)$ über einem Körper K ist die Menge aller Punkte der Ebene $K \times K$, deren Koordinaten x und y einer kubischen Gleichung der folgenden Form genügen:

$$F(x,y): y^2 = x^3 + ax^2 + bx + c \text{ mit } a,b,c \in \mathbb{Z}$$

■ Beispiel

$$y^2 = x^3 + 5x^2 + x + 5$$



Elliptic Curve Cryptography (ECC)

- Spezielle Form für Kryptosysteme
 - $F(x,y): y^2 = x^3 + ax + b \text{ mod } p$ mit $a,b \in \mathbb{Z}$ und $p \in \mathbb{P}$
 - Beispiel: $E(\text{GF}(p))$ mit $a=2$, $b=4$, $p=5$ besitzt folgende Punkte:
 $\{(0,2), (0,3), (2,1), (2,4), (4,1), (4,4), o\}$
 - Gruppenstruktur über den Punkten $P_i = (x_i, y_i)$ definiert:
 - Abgeschlossenheit: $P_1 + P_2 = P_3$
 - Assoziativgesetz: $(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$
 - Neutrales Element: $P_i + o = P_i$ mit o definiert als $o = (\infty, \infty)$
 - Inverses Element $P_i^{-1} = A$: $P_i + A = o$
 - Kommutativgesetz (nur abelsche Gruppe): $P_1 + P_2 = P_2 + P_1$
- Beispiele für Algorithmen
 - ECDSA: Standardisiertes Signatursystem auf Basis elliptischer Kurven
 - ECEIGamal: ElGamal auf Basis elliptischer Kurven

Elliptic Curve Cryptography (ECC)

- Anwendung von elliptischen Kurven für Kryptographie
 - Nicht alle elliptischen Kurven sind gleich gut für ECC geeignet.
 - Ordnung (Punkteanzahl) von $E(K)$ ist u.a. wichtig für Sicherheit.
 - Methoden zum Finden geeigneter elliptischer Kurven:
 - Wähle a, b, p , berechne die Gruppenordnung und überprüfe Sicherheit
 - Wähle p und Gruppenordnung und bestimme a und b (schnellere Methode)
- Vorteile
 - bei vergleichbarem Sicherheitsniveau kürzere Schlüssel (≥ 200 Bit)
 - erzeugen deutlich kürzere Signaturen
 - weniger Rechenaufwand
 - Hoffnung: Alternative Kryptoalgorithmen, wenn z.B. Faktorisierungsannahme nicht halten sollte

Public Key Infrastructures

Signierte Nachricht und Zertifikat

-----BEGIN SIGNED MESSAGE-----

Hiermit bestelle ich folgende Waren:

10 Eier	Euro 2,00
1 Flasche Milch	Euro 1,50
1 Kasten Bier	Euro 15,00

Gesamtbetrag Euro 18,50

Die Zahlung erfolgt bei Lieferung.

Hannes Federrath

-----BEGIN SIGNATURE-----

iQAAwUBOi9VLoBzbJXQK0fCYo1rMKCKrdhAn2Rs
amogkkm+Off90L0W5RxUubfVuUFSXuv=

-----END SIGNED MESSAGE-----

-----BEGIN CERTIFICATE-----

Name: Hannes Federrath

Public key:

h833hd38dddajscbicme098k236egfkw74h5445
84hdbscldmrtpofjrkt0jedagaszw12geb3u4b=

Valid from: 19.11.2014

Valid until: 18.11.2017

Issuer: Einwohnermeldeamt Dresden

-----BEGIN SIGNATURE OF ISSUER-----

23j423vdsaz345kj435ekj4z2983734ijo23i72
kj867wdbez2o074j51kdmcd1237t3rgbdbwdj=

-----END CERTIFICATE-----

Digitales Signatursystem

Schutzziel:

Zurechenbarkeit

geheimer Bereich

öffentlicher Bereich

Text mit
Signatur
und Test-
ergebnis

»ok« oder
»falsch«,
 $\text{sig}(x)$

Test

Schlüssel zum Testen
der Signatur,
öffentlich bekannt

Text mit
Signatur

Angriffsbereich

$x, \text{sig}(x)$

Zufallszahl

Schlüssel-
generie-
rung

Schlüssel zum
Signieren, geheim
gehalten

Sig

Text

Vertrauensbereich des
Signierers,
Schlüsselgenerierung in
Signierkomponente für
optimalen Schutz von s

Drei Arten von Signaturen nach SigG

- **Signaturgesetz (SigG) vom 16. Mai 2001:** schafft rechtliche Rahmenbedingungen für den Beweiswert digitaler Signaturen

Elektronische Signatur Daten in elektronischer Form, die	Fortgeschrittene Signatur Daten in elektronischer Form, die	Qualifizierte Signatur Daten in elektronischer Form, die
<ul style="list-style-type: none">■ anderen elektronischen Daten beigefügt oder logisch mit ihnen verknüpft sind und die zur Authentifizierung dienen	<ul style="list-style-type: none">■ ausschließlich dem Signaturschlüssel-Inhaber zugeordnet sind■ die Identifizierung des Signaturschlüssel-Inhabers ermöglichen■ mit Mitteln erzeugt werden, die der Signaturschlüssel-Inhaber unter seiner alleinigen Kontrolle halten kann■ mit den Daten, auf die sie sich beziehen, so verknüpft sind, dass eine nachträgliche Veränderung der Daten erkannt werden kann	<ul style="list-style-type: none">■ die Anforderungen an eine fortgeschrittene Signatur erfüllen■ auf einem zum Zeitpunkt ihrer Erzeugung gültigen qualifizierten Zertifikat beruhen■ mit einer sicheren Signaturerstellungseinheit erzeugt werden <p>Sicherheit</p>

Drei Arten von Signaturen nach SigG

- Signaturgesetz (SigG) vom 16. Mai 2001: schafft rechtliche Rahmenbedingungen für den Beweiswert digitaler Signaturen

Elektronische Signatur Beispiel: E-Mail mit "Signatur"	Fortgeschrittene Signatur Beispiel: PGP-signierte E-Mail	Qualifizierte Signatur
<p>From: Hannes Federrath Subject: Beispiel</p> <p>Das ist der Text.</p> <p>--</p> <p>Hannes Federrath FB Informatik Uni Hamburg</p>	<p>-----BEGIN PGP SIGNED MESSAGE----- Hash: SHA1</p> <p>Das ist der Text.</p> <p>-----BEGIN PGP SIGNATURE----- Version: PGP 8.0.2</p> <p>iQA/AwUBP6wDdOFAIGFJ7x2EEQK9VgCg2Q4 eQAztVIHP0HNFO10eaXte96sAnR2p 53T/SdevjXIuX6WOF5IXA44S =K3TO -----END PGP SIGNATURE-----</p>	<p>Zertifikatausstellung nach Identitätsüberprüfung sichere Signaturerstellungseinheit</p> <p>Sicherheit</p> 

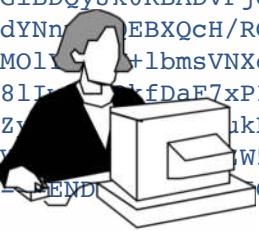
Zweck der Schlüsselzertifizierung

- Betrifft öffentlichen Testschlüssel für die digitale Signatur
- Zertifikat:
 - bestätigt die Zusammengehörigkeit von Testschlüssel und Benutzeridentität bzw. Testschlüssel und Pseudonym.
 - Enthält selbst die **Signatur des Zertifizierers**
- Ohne Zertifikate:
 - Angreifer kann ein Schlüsselpaar generieren und einfach behaupten, dass dieser Schlüssel jmd. gehört.
 - Testschlüssel sind wertlos ohne Zertifikat (zumindest in einer offenen Welt)

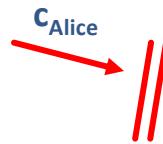
Maskerade-Angriff 1/2

Alice hat Schlüsselpaar generiert und will ihn veröffentlichen.

Alice <alice@abc.de>
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQGiBDQyJk0RBADVPjcdvwmyOtqsZBt6z4/5M9MYDB
i+dYNn+EBXQcH/RGe2i30LRvRk4asX++JStylku
8LMOL+lbmsVNxeQSdmbSAUfd3d9bI/+fGwQcz
6W8lT+fDaF7xPI7ovZUY1I7cqEfTvic003bgL
SUZYukKj01O66wVmqlnXcbi2XUebka
LOVW59gf5I0eUBevSmydIaliH9Pm
---END PGP PUBLIC KEY BLOCK----



c_{Alice}



//

Angreifer

- hält c_{Alice} zurück (blockiert Verteilung)
- generiert selbst ein Schlüsselpaar $c_{\text{Mask}}, d_{\text{Mask}}$ unter falschem Namen
- schickt c_{Mask} an Bert

c_{Mask}



Bert besitzt jetzt nicht authentischen Schlüssel von Alice.

Alice <alice@abc.de>

-----BEGIN PGP PUBLIC KEY BLOCK-----
OTUAoLncfli6Yit0Kqgp/N9h37uopJHbiQCVAw
xBBPLRdmalP22ij0dARxbJL07u7XOrnyV3b4m0
14ydps/ruj9yaY62BwQNMEoGjAnZGA5t3MM0
7ZLp1dmFYVVYPL4xRfOJ+MF5ifb8RXaDAL+1
CwMBAgAKCRDhQCBhSe8dhOYYAJseI
u64hbO2wuFQlwwq1yb+JAD8DBRA00
-----END PGP PUBLIC KEY BLOCK-----



Maskerade-Angriff 2/2

Ohne die Gewissheit über die Echtheit eines öffentlichen Schlüssels funktioniert keine sichere asymmetrische Kryptographie. Deshalb:
Schlüsselzertifizierung

Bert will Alice eine Nachricht N schicken.



Angreifer:

- Weiterleitung verhindern
- entschlüsseln von $c_{\text{Mask}}(N)$ mit d_{Mask}
- verschlüsseln von N mit c_{Alice}

$c_{\text{Alice}}(N)$

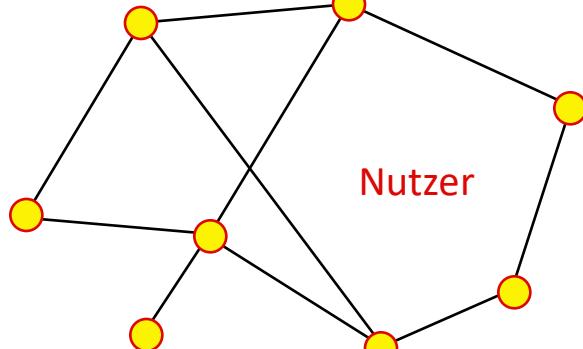
Alice erhält die Nachricht N.
N ist verschlüsselt mit ihrem
öffentlichen Schlüssel.



Zertifizierungsmodelle

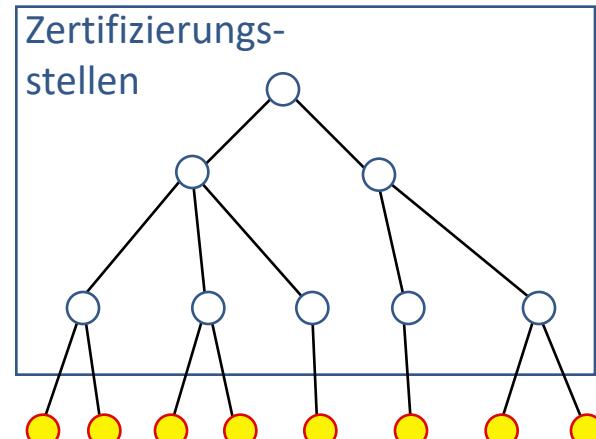
- Zur Verschlüsselung und Signierung wird asymmetrische Kryptographie verwendet
 - Zwei Schlüssel: Private und Public Key
 - Zwei Ansätze zur Zuordnung des Public Keys zu einer Person
 - PGP und GnuPG: »Web of Trust«
 - TLS und S/MIME: Hierarchische Zertifizierung

Web of Trust



Vertreter: PGP, GnuPG

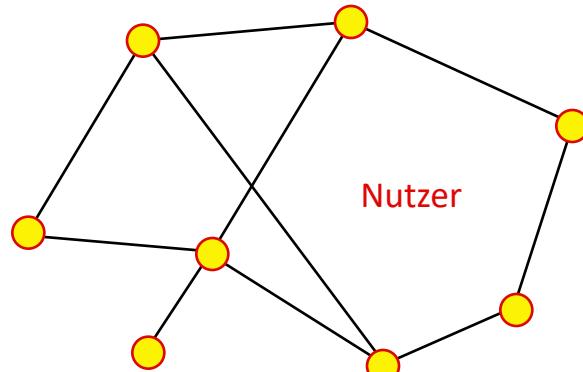
Hierarchische Zertifizierung



Vertreter: TLS, S/MIME

Zertifizierungsmodelle

■ Web of Trust



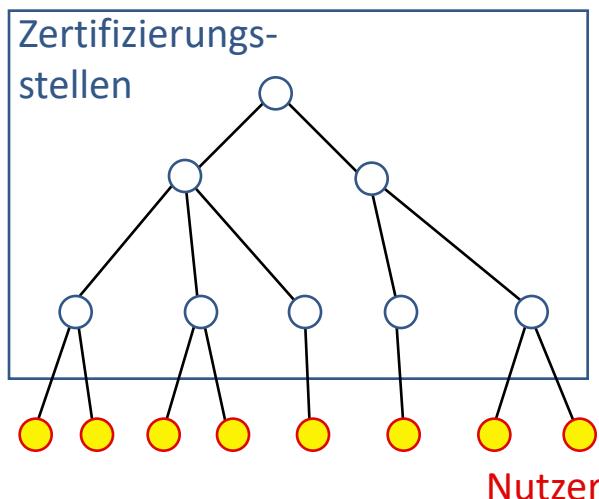
Vorteile:

- einfache, flexible Nutzung
- viele potentielle Zertifikatsketten

Nachteile:

- keine oder nur schwer erreichbare Beweisführung im Streitfall
- finden eines vertrauenswürdigen Pfades aufwendiger

■ Hierarchische Zertifizierung



Vorteile:

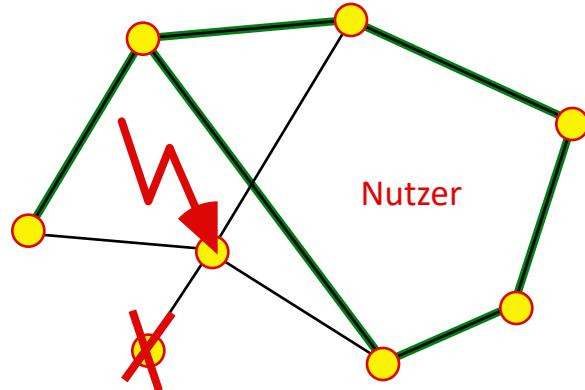
- klare Strukturen und Zurechenbarkeiten (wichtig im Streitfall)

Nachteile:

- Overhead durch Organisationsstruktur

Zertifizierungsmodelle

■ Web of Trust



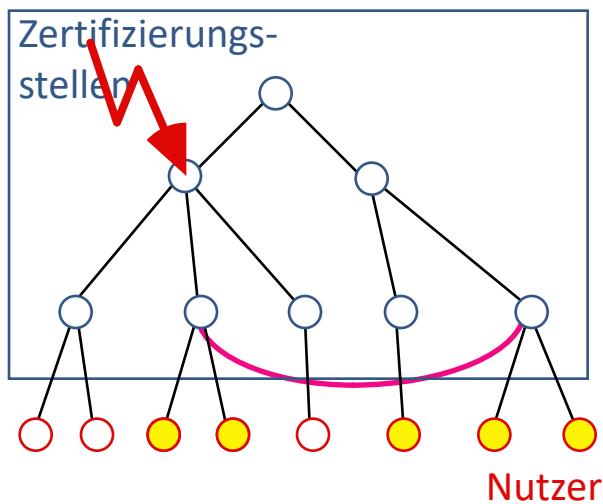
Vorteile:

- einfache, flexible Nutzung
- viele potentielle Zertifikatsketten

Nachteile:

- keine oder nur schwer erreichbare Beweisführung im Streitfall
- finden eines vertrauenswürdigen Pfades aufwendiger

■ Hierarchische Zertifizierung



Vorteile:

- klare Strukturen und Zurechenbarkeiten (wichtig im Streitfall)

Nachteile:

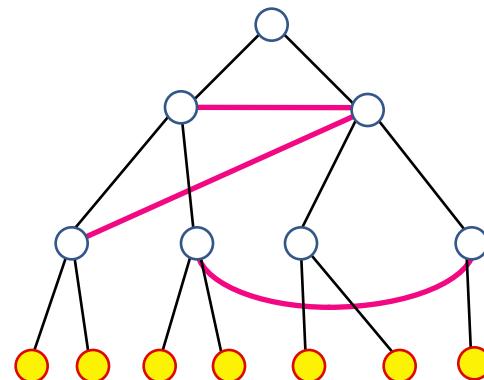
- Overhead durch Organisationsstruktur
- anfällig gegen Fehlverhalten
- Cross Certification reduziert Fehlermöglichkeiten

Zusammenfassung: Zertifizierungsmodelle

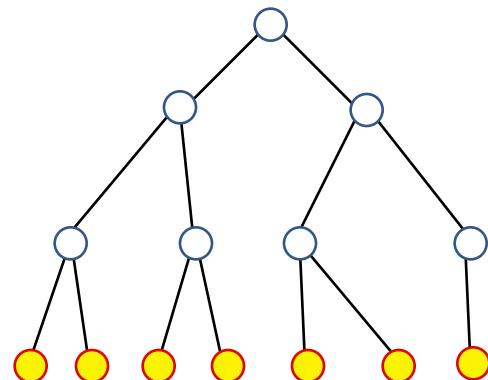
Haftung des
Zertifizierers

Ja

Vermaschter
Graph

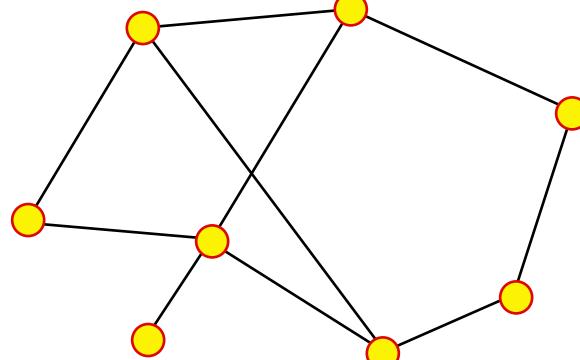


Baum
(minimal zusammenhängender Graph)



Reduzieren der Fehleranfälligkeit
durch Cross Certification

Nein



-

X.509-Zertifikate

ITU X.509 Zertifikate

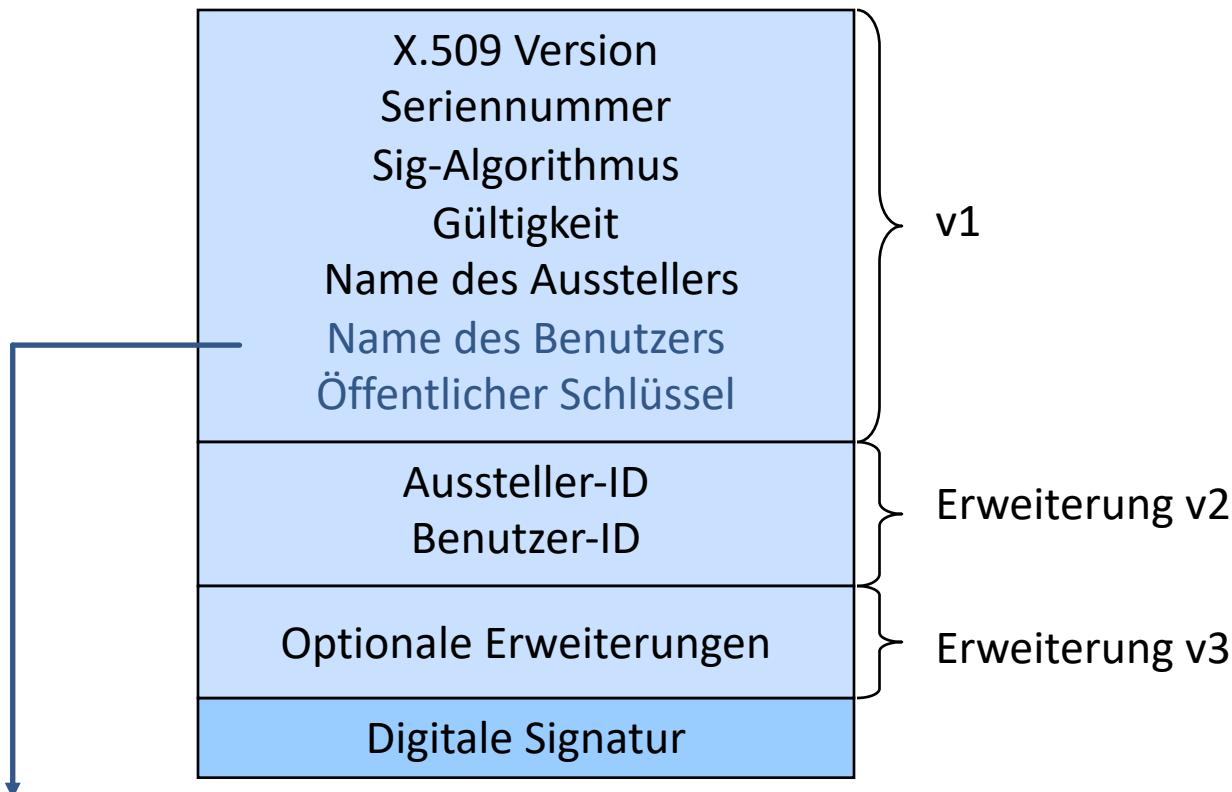
... werden insb. angewendet bei TLS und S/MIME.

- **S/MIME (Secure Multipurpose Internet Mail Extensions)**
 - Ursprünglich von RSA Data Security Inc.
 - S/MIME v3 im Juli 1999 als IETF-Standard verabschiedet
 - Internet Standards RFCs 2632-2634 (und weitere)
 - In die meisten E-Mail-Clients integriert
- **TLS Transport Layer Security**
 - vormals SSL (Secure Sockets Layer)
 - Verschlüsselung von TCP-Verbindungen
 - ursprünglich von Netscape für Browser entwickelt
 - heute in jedes moderne Betriebssystem integriert

Schutz der Vertraulichkeit und Integrität

ITU X.509 Zertifikate

- Festlegung eines standardisierten Formats für Zertifikate



Hierarchisch aufgebauter »distinguished name«, z.B.:

cn = Hannes Federrath, ou = Informatik, o = Uni Hamburg, c = DE

ITU X.509 Zertifikate

- Erweiterungen in X.509v3
 - Art des Schlüssels, Anwendungsbereich
 - Alternative Namen für Inhaber und Aussteller
 - Einschränkungen bzgl. cross certification
 - Informationen bzgl. Sperrlisten (URL)
 - private ausstellerspezifische Erweiterungen
- Zertifizierungsprozess
 - Antrag bei der Registration Authority (RA)
 - Identitätsprüfung durch die RA
 - Zertifizierung durch die Certificate Authority (CA)
 - Ausgabe des Zertifikats an den Antragsteller
- Widerruf der Zertifikate
 - Certificate Revocation Lists (CRLs)
 - Online Certificate Status Protocol (OCSP)

Optionen für die Zertifikatserstellung

- OpenCA
 - <http://www.openca.org>
 - Beschreibung:
»Open Source out-of-the-box Certification Authority implementing the most used protocols with full-strength cryptography world-wide.«
 - Benutzt OpenLDAP, OpenSSL, Apache, mod_ssl, Perl und lässt sich per Browser bedienen/konfigurieren
 - Antragstellung per Web-Browser bei der RA, nach Zertifizierung durch die CA Download des Zertifikats
 - Kosten:
 - Potentiell niedrig
 - Administrationsaufwand:
 - Potentiell mittel
 - allerdings hoher Einrichtungsaufwand

Optionen für die Zertifikatserstellung

■ FlexiTrust

- <http://www.flexsecure.de> (jetzt Kobil)
- Umfassende Lösung zur Erstellung einer PKI / eines Trustcenters
- Entwicklung der Technischen Universität Darmstadt
- Nach dem Signaturgesetz zertifiziert
- Plattformunabhängige Java-Anwendung

- Kosten:
 - Sehr hoch
- Administrationsaufwand:
 - Potentiell niedrig



The screenshot shows a Microsoft Internet Explorer window displaying the FlexiTrust software interface. The title bar reads "Personen Daten - Microsoft Internet Explorer". The main content area is titled "FlexiTrust Trustcenter Software".
The form is divided into sections:

- Person:** Fields include Name, Vorname, CN-Erweiterung (set to "SIGN"), eMail, Strasse, Ort, CA erzeugt PIN (checkbox checked), and CA erzeugt Revokationspasswort (checkbox checked).
- Organisation:** Fields include Organisationseinheit, Organisation, Ort, and Land (set to "DE").
- Zertifikat:** Fields include Gültig von (set to 25.09.2001) and Gültig bis (set to 25.09.2002).
At the bottom are buttons for "Absenden" and "Rücksetzen".

Bezug von X.509-Zertifikaten von einem kommerziellen Anbieter

- Verschiedene Zertifikatsklassen
 - Class 0: Demo-Zertifikat
 - Class 1: Existenz der E-Mail-Adresse wird geprüft
 - Class 2: Schriftlicher Auftrag
 - Class 3: Antragsteller muss sich persönlich identifizieren
 - Class 4: »Online business transactions between companies«
 - Class 5: »for private organizations or governmental security«
- Vorteile:
 - Wurzelzertifikat ist in den meisten Clients schon enthalten
 - keine eigene CA nötig
 - kein Administrationsaufwand
- Kosten: 0-130 EUR pro Jahr und Zertifikat

TLS-Handshake

Client



Server-Zertifikat prüfen

- Common Name ist gleich Domain Name
- Gültigkeit
- Signatur des Issuers

Zertifikatskette prüfen

- CA-Attribut
- Gültigkeit
- Signatur des Issuers

Server
server.com



ClientHello

ServerHello

Certificate

ServerKeyExchange

CertificateRequest

ServerHelloDone

Certificate

ClientKeyExchange

CertificateVerify

ChangeCipherSpec

Finished

ChangeCipherSpec

Finished

Application Data

ab hier symmetrisch verschlüsselt

ab hier symmetrisch verschlüsselt

TLS-Handshake (vereinfacht)

Client ruft auf
<https://server.com/>

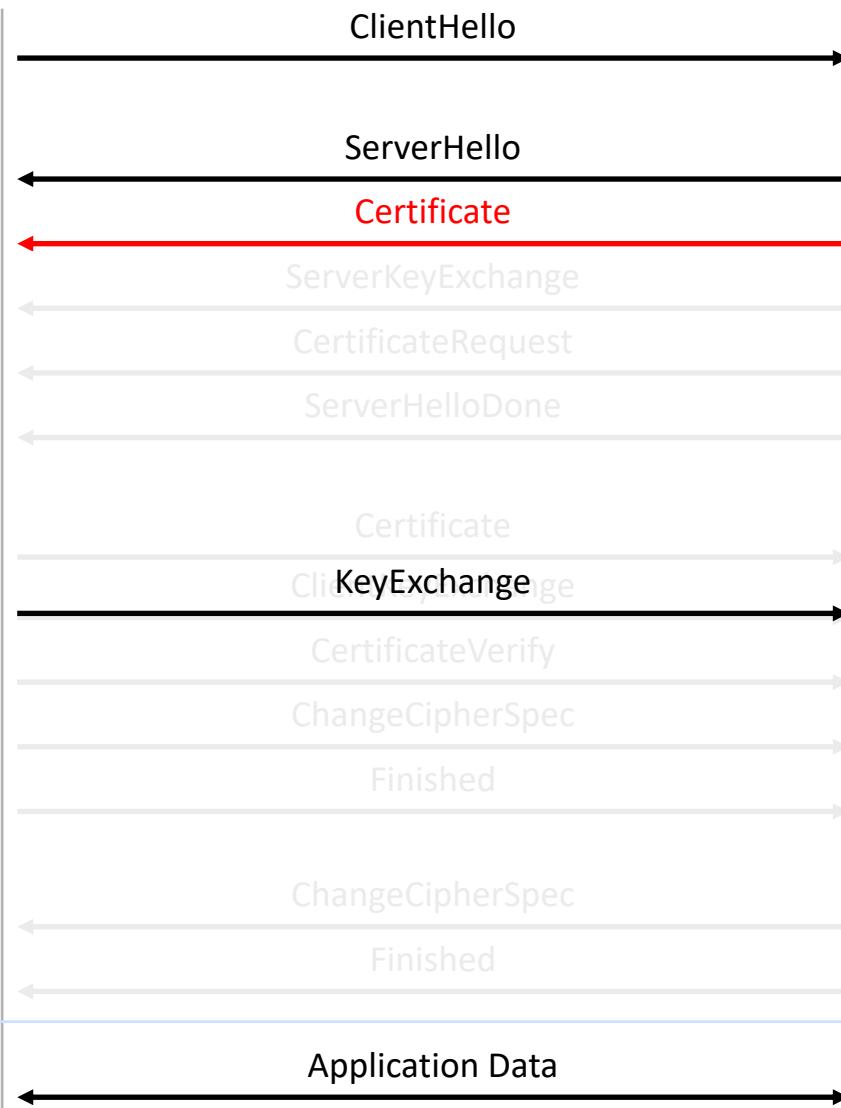


Server-Zertifikat prüfen

- Common Name ist gleich Domain Name
- Gültigkeit
- Signatur des Issuers

Zertifikatskette prüfen

- CA-Attribut
- Gültigkeit
- Signatur des Issuers



Server
server.com



Client teilt dem Server den symmetrischen Schlüssel mit (verschlüsselt mit Public Key des Servers)

ab hier symmetrisch verschlüsselt

ab hier symmetrisch verschlüsselt

Benutzeransicht eines Zertifikats (Firefox)

This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN) svs.informatik.uni-hamburg.de
Organization (O) Universitaet Hamburg
Organizational Unit (OU) MIN-Fakultaet
Serial Number 12:68:6D:71:1C:BA:E6

Issued By

Common Name (CN) UHH CA - G02
Organization (O) Universitaet Hamburg
Organizational Unit (OU) Regionales Rechenzentrum

Validity

Issued On 15.08.11
Expires On 13.08.16

Fingerprints

SHA1 Fingerprint 4F:C6:A7:CD:EC:F5:B6:42:24:F7:76:57:04:A3:6C:41:D2:7E:75:0C
MD5 Fingerprint 6C:5E:74:BD:2F:F5:D6:29:B8:3F:B9:15:B5:C5:65:50

Normales Serverzertifikat

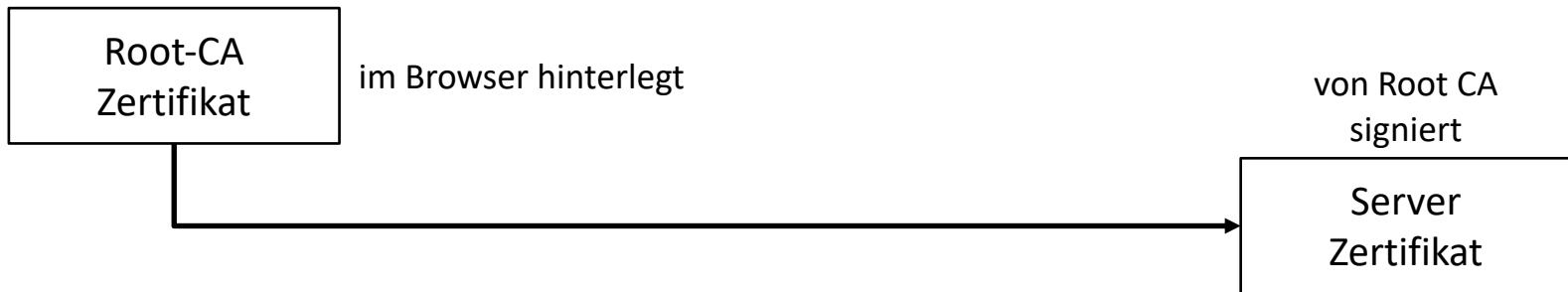


Extended Validation Certificate

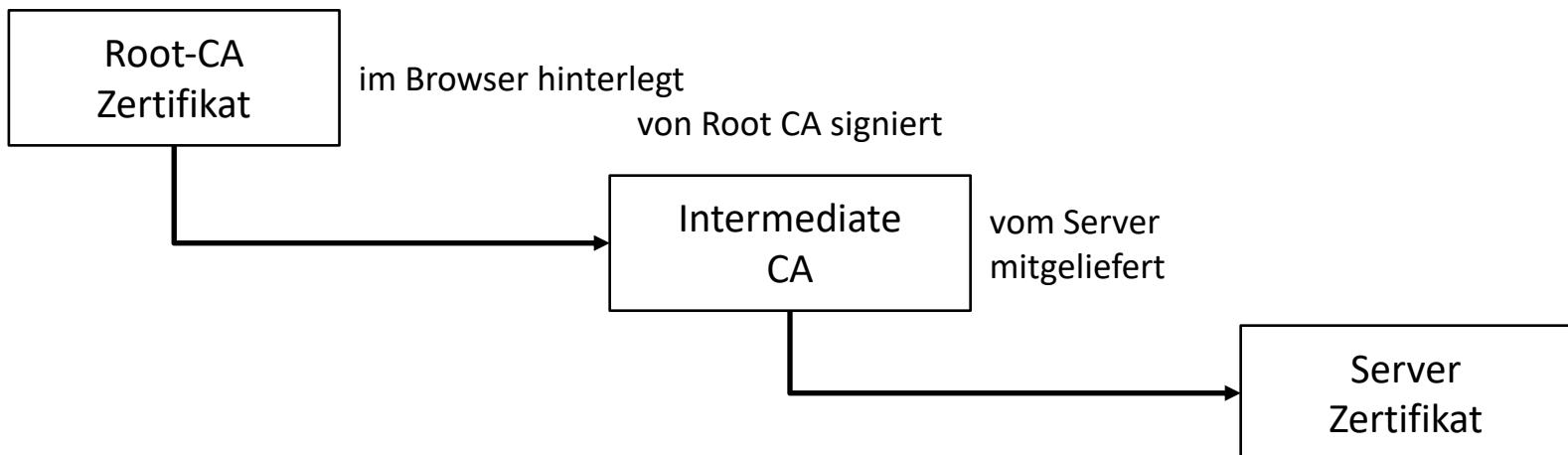


Zertifikatsvalidierung: Root CA vs. Intermediate CA

- Fall 1: Root CA stellt direkt ein Server-Zertifikat aus



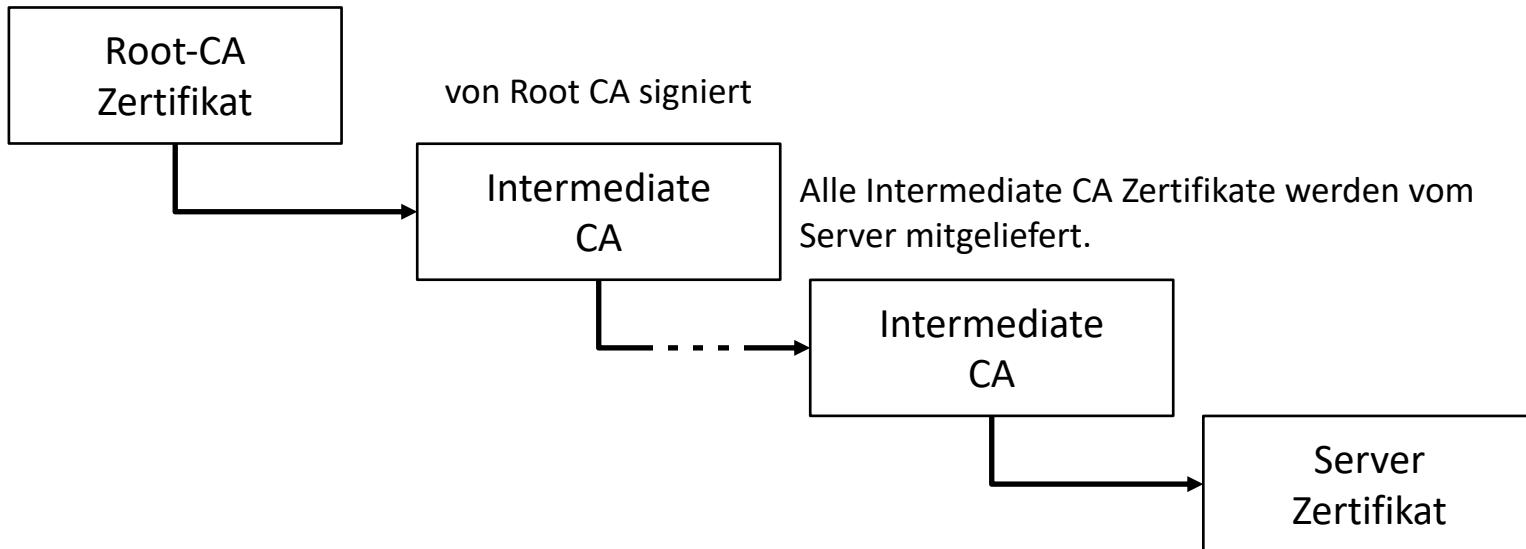
- Fall 2: Root CA zertifiziert Intermediate CA, diese stellt Server-Zertifikat aus



Rekursive Zertifikatsvalidierung

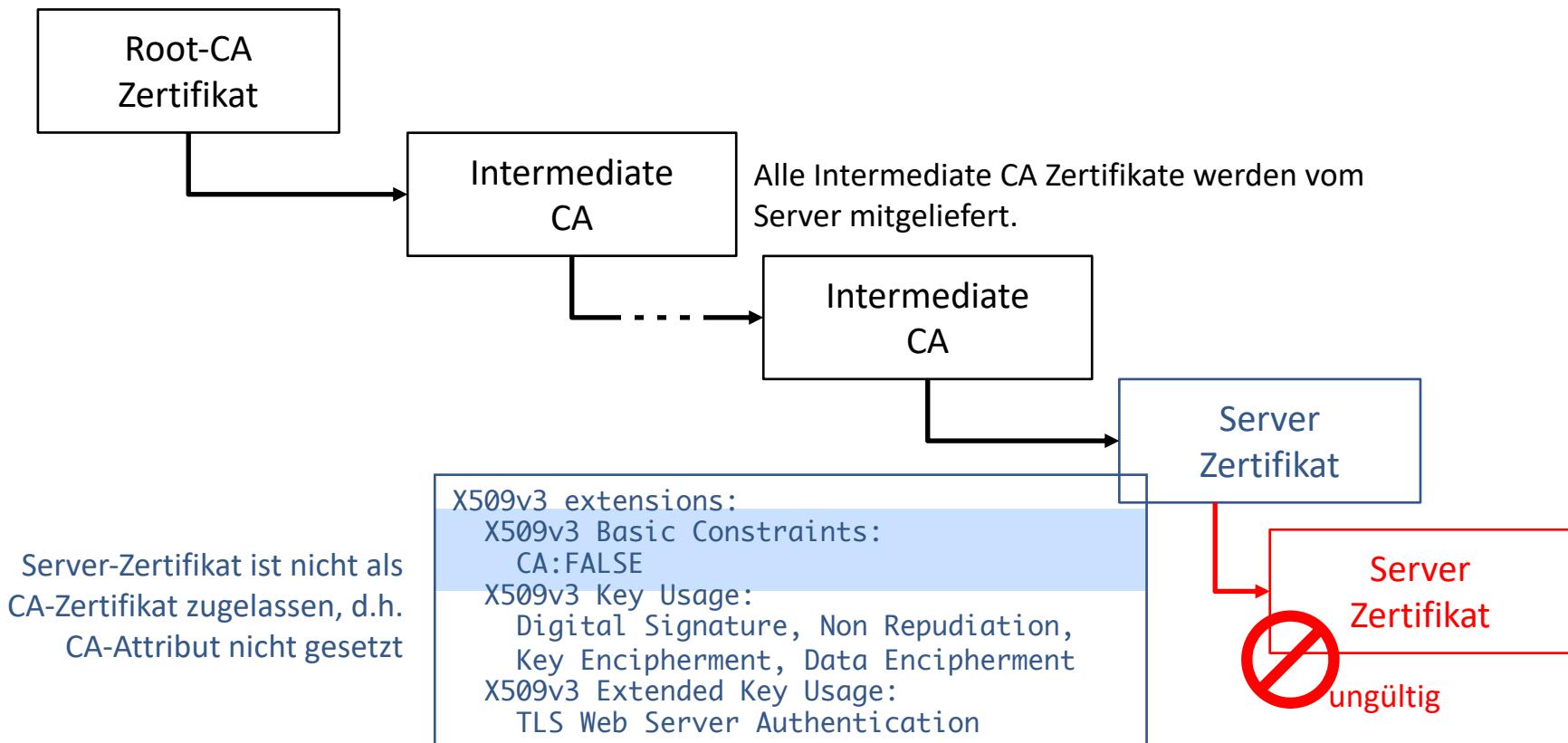
■ Rekursives Verfahren:

- Root CA zertifiziert Intermediate CA, Intermediate CA zertifiziert Intermediate CA, u.s.w, Intermediate CA stellt ein Server-Zertifikat aus



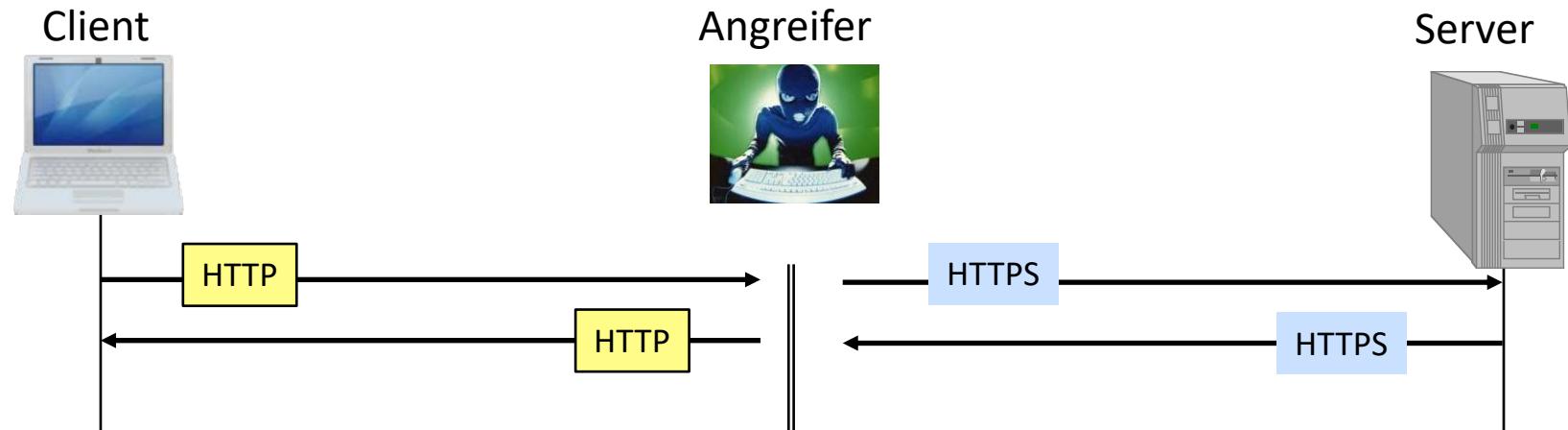
Einschränkungen bzgl. Cross Certification

Wie wird verhindert, dass vom Inhaber des Server-Zertifikats ein weiteres gültiges **Server-Zertifikat** erzeugt wird?



Man-in-the-Middle-Angriffe auf HTTPS: sslstrip

- Ziel:
 - Angreifer möchte Kommunikation mitlesen und/oder verändern
- Angreifer verhindert Umleitung zu HTTPS



Nutzer geben www.bank.de ein —
ohne https://

Angreifer verhindert Umleitung zu HTTPS

Server liefert normalerweise:

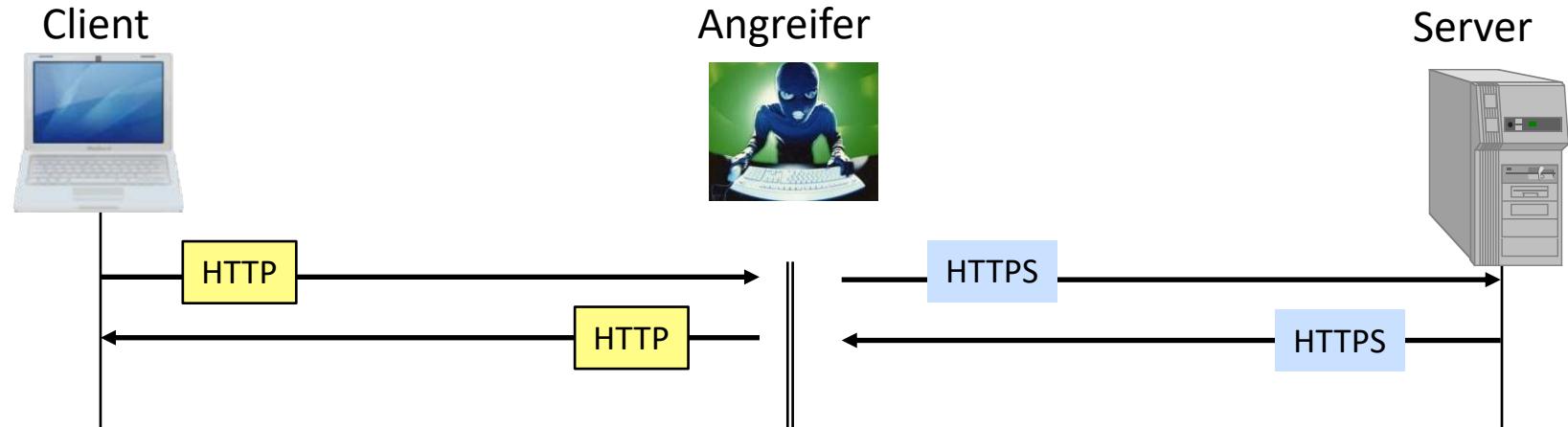
```
> GET / HTTP/1.1  
> Host: www.bank.de  
  
< HTTP/1.1 301 Moved Permanently  
< Location: https://www.bank.de/
```

HTTP Strict Transport Security (HSTS): HTTP-Header liefert bei Erstkontakt Verfallsdatum für https-Zwang

```
Strict-Transport-Security: max-age=<sek>
```

Man-in-the-Middle-Angriffe auf HTTPS: sslstrip

- Ersetzen aller Umleitungen und Links zu HTTPS
- Verbindung zum Server per HTTPS, zum Client per HTTP
- Server merkt nicht, dass Client kein HTTPS verwendet



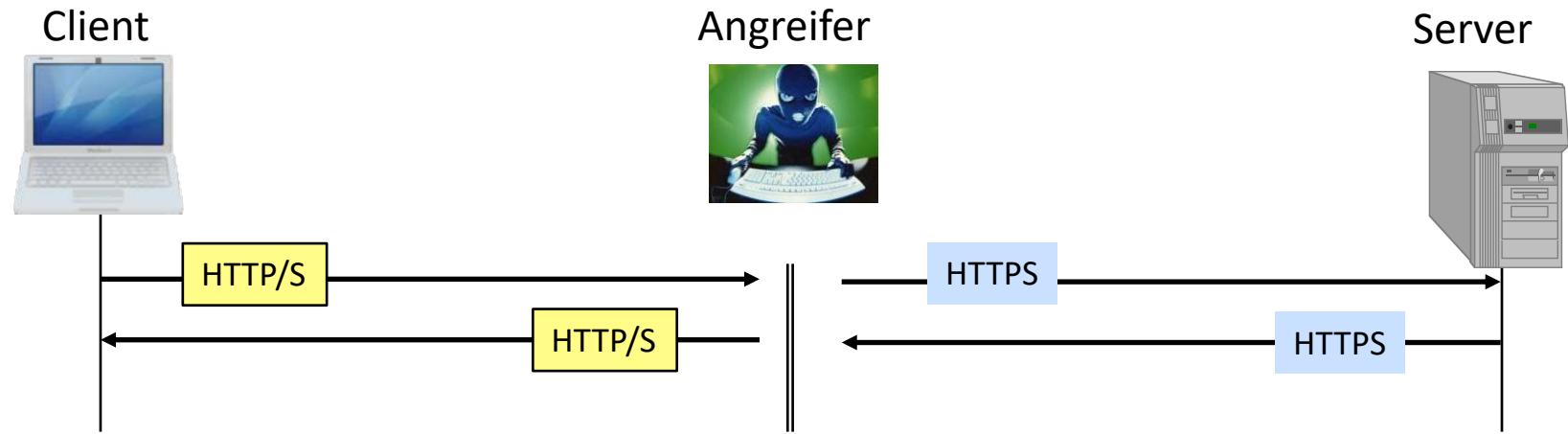
```
<html> [...]
<body> [...]
<a href="http://www.shop.de/login">
Zum sicheren Login-Formular</a>[...]
</body></html>
```

wird vom
Angreifer
ersetzt
durch

```
<html> [...]
<body> [...]
<a href="https://www.shop.de/login">
Zum sicheren Login-Formular</a>[...]
</body></html>
```

Man-in-the-Middle-Angriffe auf HTTPS: burp proxy, mitmproxy

- Ziel: Angreifer möchte Kommunikation mitlesen und/oder verändern
- Angreifer stellt »on the fly« gültige Zertifikate aus



Nutzer haben etwa Firmen-CA-Zertifikat importiert

»On the fly« vom Angreifer ausgestellte Zertifikate

HTTP Public Key Pinning (HPKP): HTTP-Header liefert Hash des korrekten öffentlichen Schlüssels

```
Public-Key-Pins: pin-sha256="cUPcTAZWK..."; max-age=<sek>
```

Fehlende oder fehlerhafte Zertifikatsvalidierung in Clients

The image shows a laptop screen on the left and a smartphone screen on the right.

Laptop Screen (Left):

- Terminal window title: bash
- Log entry: 2012-11-05 19:25:18 POST http://54.246.136.213:443/androidapp/DE/shopgate/api/shopgate/ice_id=a3584c59d3edd1e1.a48cbef35222c90a0112131e7bb4ecd&lang=de&ver=2.2.2
- Response status: ← 200 text/html 85B
- Request section:
 - Content-Length: 150
 - Content-Type: application/x-www-form-urlencoded
 - Host: api.shopgate.com
 - Connection: Keep-Alive
 - User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)
 - Cookie: SHOPGATE=643208576098896d411df975f55ea9df; login=domi%
 - Cookie2: \$Version=1
- URLEncoded form:
 - users_address_id: 213084
 - card_type: visa
 - card_number: 5490019409001363
 - cvc: 866
 - expiry_year: 12
 - card_holder: Adgkl Yky
 - expiry_month: 11
- Terminal prompt: [665/665]
- Root command: root@grml | (load: 0.07 0.07 0.12 | cpu: [cpufreq n/a] | net: 134.100
0* ./mitmproxy | 0* ./mitmproxy 1- zsh)

Smartphone Screen (Right):

The smartphone displays a mobile application for "Neue Kreditkarte". The card details shown are:

- Name: Adgkl Yky
- Card Number: 5490 0194 0900 1363
- CVV: 866
- Expiry Date: Gültig 11 / 2012

Below the card information, there is a section titled "Ihre Daten sind sicher!" containing the following text:

Alle Daten werden verschlüsselt übertragen. Ihre Kreditkarte wird nicht bei Snipes GmbH, sondern bei Shopgate gespeichert. Shopgate ist einer der weltweit führenden Mobile-Shopping-Provider und hat die PCI-DSS-Zertifizierung erhalten. Dies bestätigt, dass die höchsten technischen Anforderungen an die Datensicherheit im elektronischen

PCI DSS approved FEB-2012
MasterCard Security Verified

Sicher einkaufen SSL-Verschlüsselung 256 BIT

Nach welchen Kriterien kommen die Root-Zertifikate in Anwendungen wie Firefox oder die Betriebssysteme?

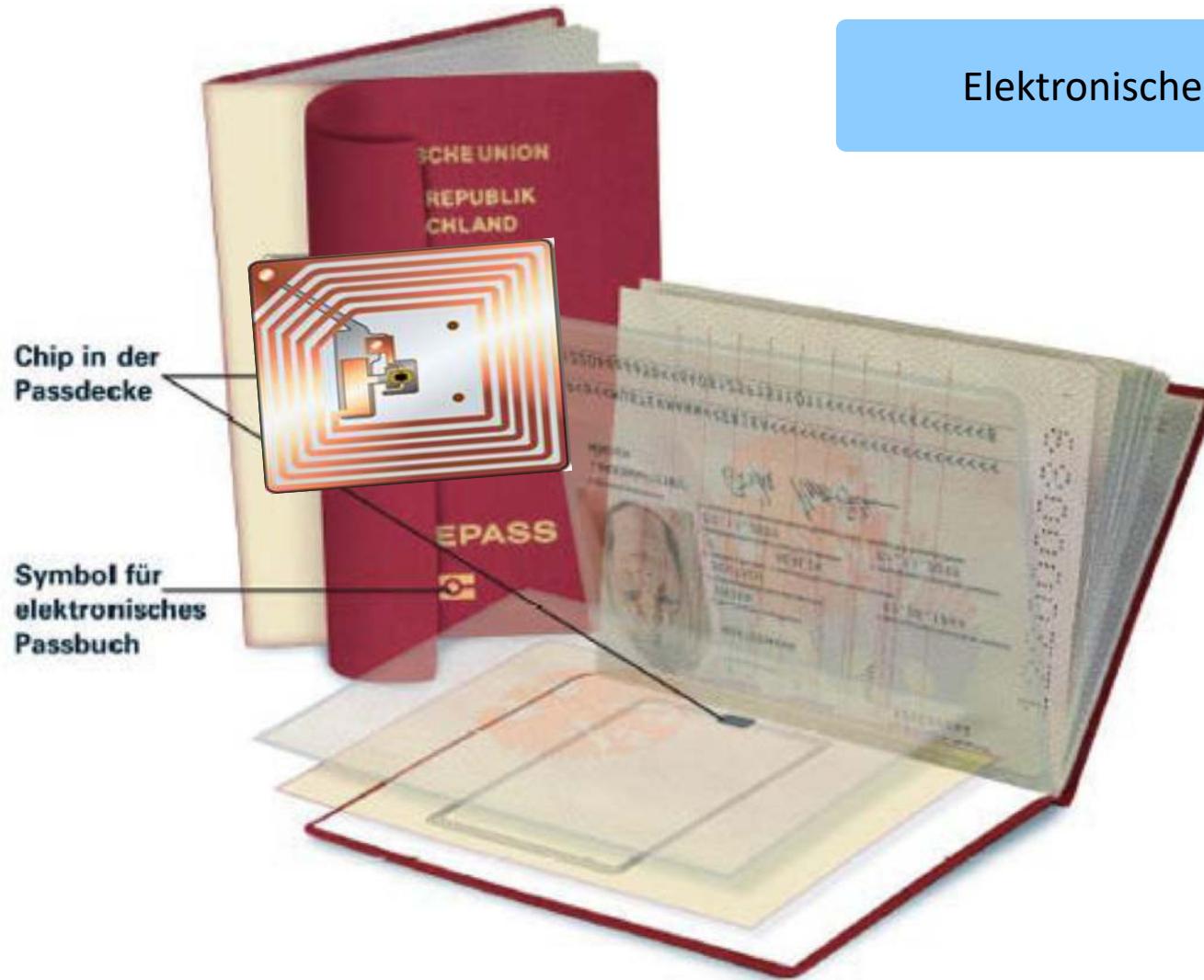
- Audit-Standards
 - Überprüfung der Steuerungs- und Kontrollprozesse von (Root)-CAs
 - WebTrust-Standard der Canadian Institute of Chartered Accountants
 - ETSI TS 102 042 – Policy for requirements for certification authorities – Issuing public key certificates
 - ETSI TS 101 456 – Policy for requirements for certification authorities – Issuing qualified certificates
 - ISO 21128 – Public key infrastructure for financial services – Practices and policy framework
- | Plattform | Anzahl Zertifikate |
|-----------------------------------|--------------------|
| Apple iOS, OS X | 150 |
| Microsoft Windows 10 | 230 |
| Mozilla Network Security Services | 130 |

Quelle: Hicken 2017

Biometrischer Reisepass

Ein Beispiel für den internationalen Aufbau einer PKI

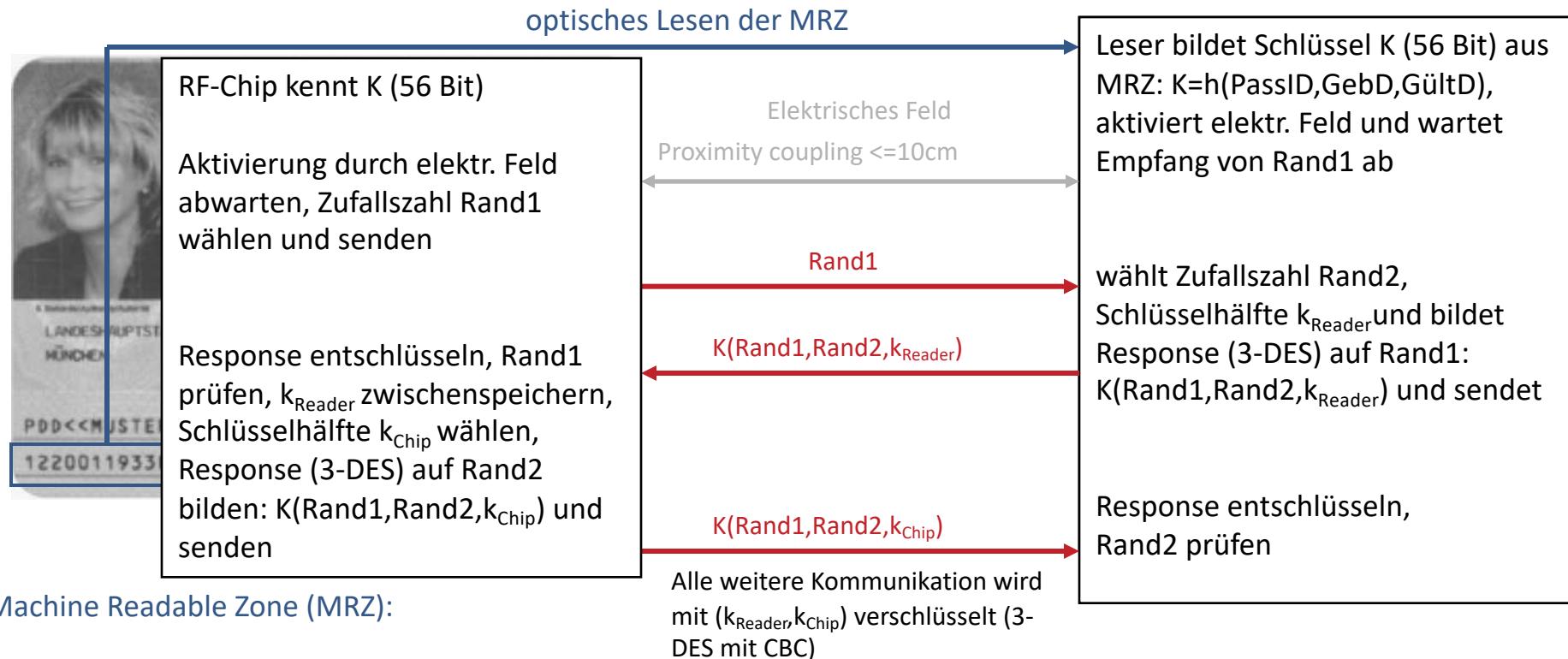
RFID zur drahtlosen Kommunikation



Elektronischer Reisepass

Wh.: Biometrische Daten in Reisepässen

■ Basic Access Control und Active Authentication



Passnummer Geburtsdatum Gültigkeitsdatum
jeweils mit Prüfziffer versehen

nach: Dr. Dennis Kügler: Risiko Reisepass?
Schutz der biometrischen Daten im RF-Chip.
ct 5 (2005) 88

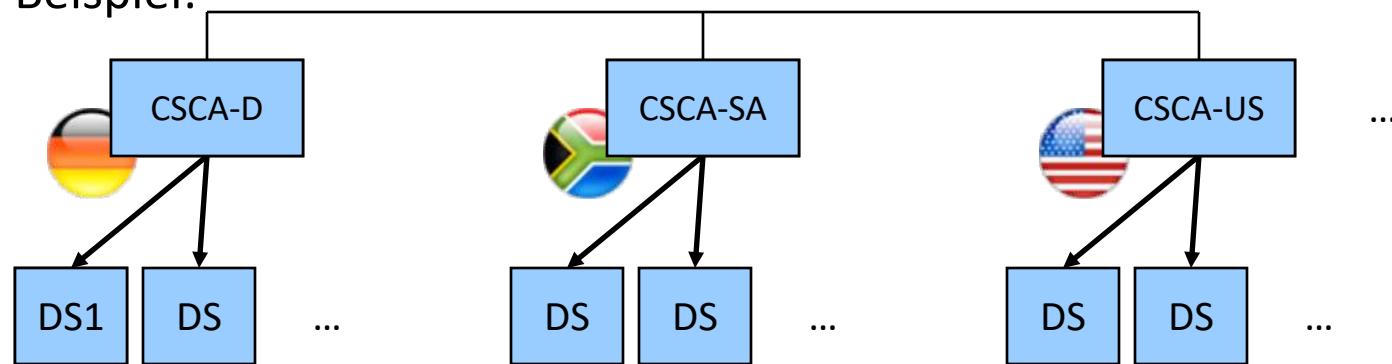
Sicherheitsfunktionen in elektronischen Reisepässen

- Basic Access Control
 - Auslesen der biometrischen Daten benötigt optische Daten der maschinenlesbaren Zone
 - **Schutz des digitalen Fotos**
- Active Authentication
 - Soll 1:1-Kopien authentischer Daten auf gefälschten Pässen (Chips) verhindern
 - Authentikation eines Originalchips mittels Challenge-Response
- Symmetrisch verschlüsselte Kommunikation
 - zwischen Pass und Lesegerät
- Passive Authentication
 - Digitale Signatur der gespeicherten Biometriedaten
 - Aufbau der PKI: -->(nächste Folie)



Sicherheitsfunktionen in elektronischen Reisepässen

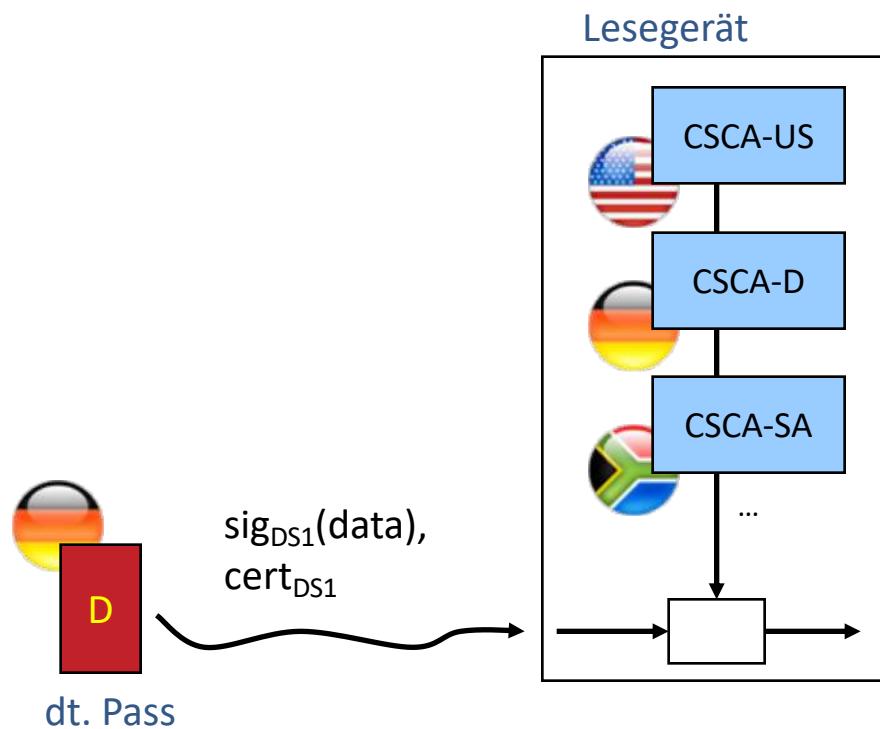
- Aufbau der PKI bei Passive Authentication der Biometriedaten
 - eine Country Signing Certification Authority (CSCA) pro Land
 - zertifiziert Testschlüssel mehrerer Document Signer (DS)
 - keine übergeordnete weltweite CA
 - alle Lesegeräte enthalten die Zertifikate der CSCAs des eigenen Landes und aller fremden Länder
 - Beispiel:



Quelle: Dennis Kügler, Ingo Naumann: Sicherheitsmechanismen für kontaktlose Chips im deutschen Reisepass. DuD 3 (2007)

Sicherheitsfunktionen in elektronischen Reisepässen

- Aufbau der PKI bei Passive Authentication der Biometriedaten
 - Beispiel: (ausländisches) Lesegerät prüft dt. Passdaten



Vorausgegangen:

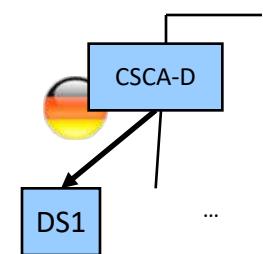
- Basic Access Control
- Active Authentication

Lesegerät bei Grenzkontrolle:

- kennt alle CSCAs anderer Länder

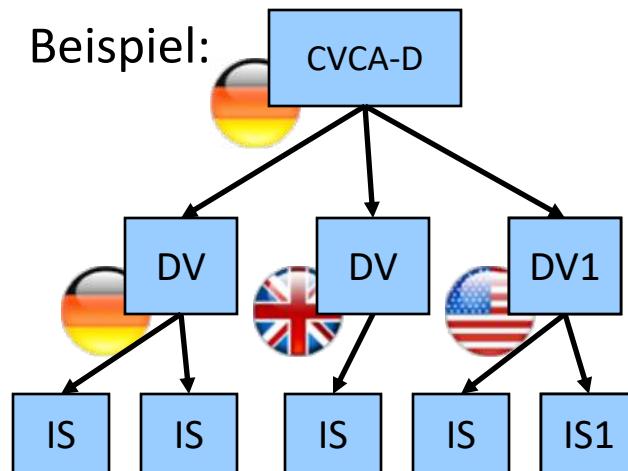
Ablauf der Zertifikatsprüfung:

- erhält vom Pass: $\text{sig}_{\text{DS}1}(\text{data})$, $\text{cert}_{\text{DS}1}$
- prüft: $\text{sig}_{\text{DS}1}$ mit Testschlüssel (in $\text{cert}_{\text{DS}1}$ enthalten)
- prüft: $\text{cert}_{\text{DS}1}$ mit Testschlüssel (in $\text{cert}_{\text{CSCA-D}}$)



Sicherheitsfunktionen in elektronischen Reisepässen

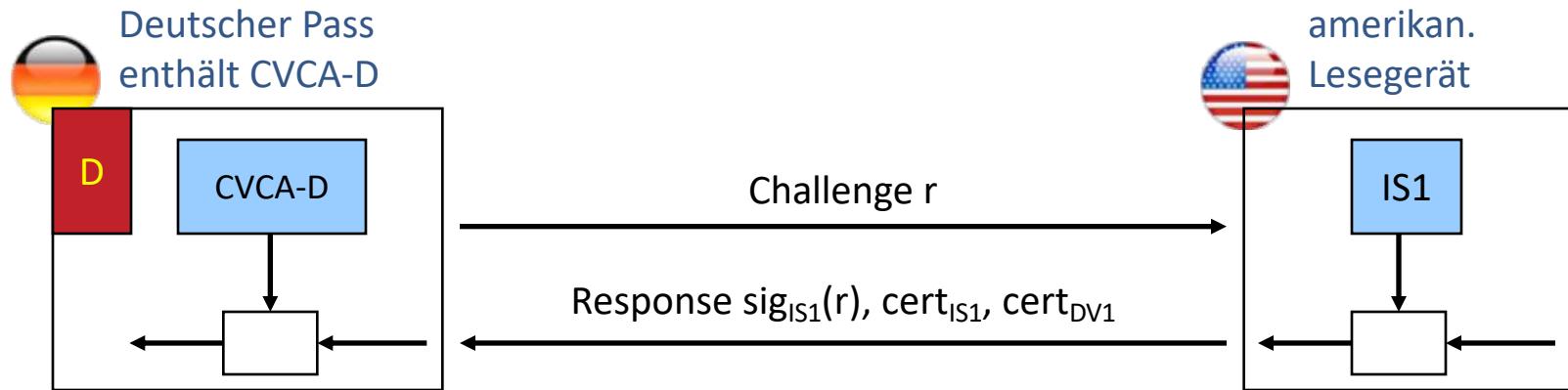
- Anstelle von Basic Access Control ab 2007: Extended Access Control
 - Schutz der Fingerabdrücke
 - Beschränkt den Zugriff auf autorisierte Lesegeräte
 - Authentikation des Lesers über Public Key Zertifikat:
 - Eine Country Verifying Certification Authority (CVCA) pro Land zertifiziert diejenigen
 - Document Verifier (DV) (fremder Länder), die (Fingerabdruck)-Daten auslesen dürfen.
 - Chips auf Pässen enthalten nationales CVCA-Zertifikat
 - Beispiel:



Dt. zertifiziert Public Keys nationaler und internationaler DVs (nur von solchen Ländern, die Fingerabdruckdaten lesen dürfen); DV zertifiziert Public Keys von (landeseigenen) Inspection Systems (IS)

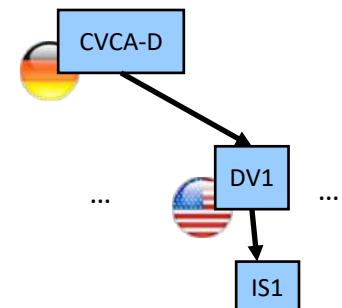
Sicherheitsfunktionen in elektronischen Reisepässen

- Authentikation des Lesers über Public Key Zertifikat
- Beispiel: dt. Pass prüft amerikan. Leser



Ablauf der Zertifikatsprüfung:

- Pass erhält vom Lesegerät: $\text{sig}_{IS1}(r)$, cert_{IS1} , cert_{DV1} als Response auf Challenge r (Zufallszahl)
- Pass prüft: sig_{IS1} mit Testschlüssel (in cert_{IS1} enthalten)
- Pass prüft: cert_{IS1} mit Testschlüssel (in cert_{DV1} enthalten)
- Pass prüft: cert_{DV1} mit Testschlüssel (in cert_{CVCA-D} enthalten)

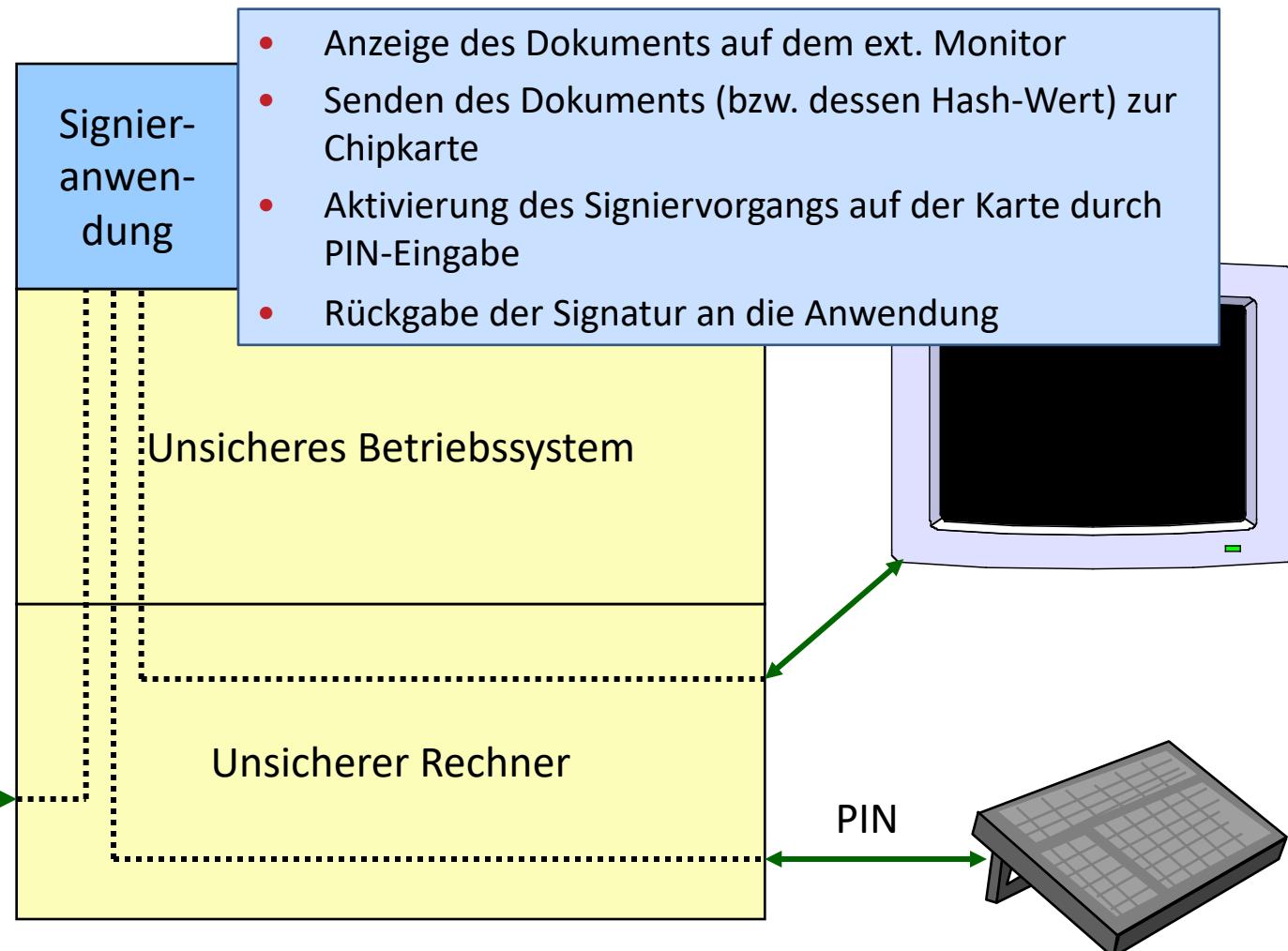


Zur Notwendigkeit sicherer Signaturerstellungseinheiten

Ablauf auf Standard-PC mit Chipkarte

- Sichere Geräte sind eine Voraussetzung für sichere Signaturen

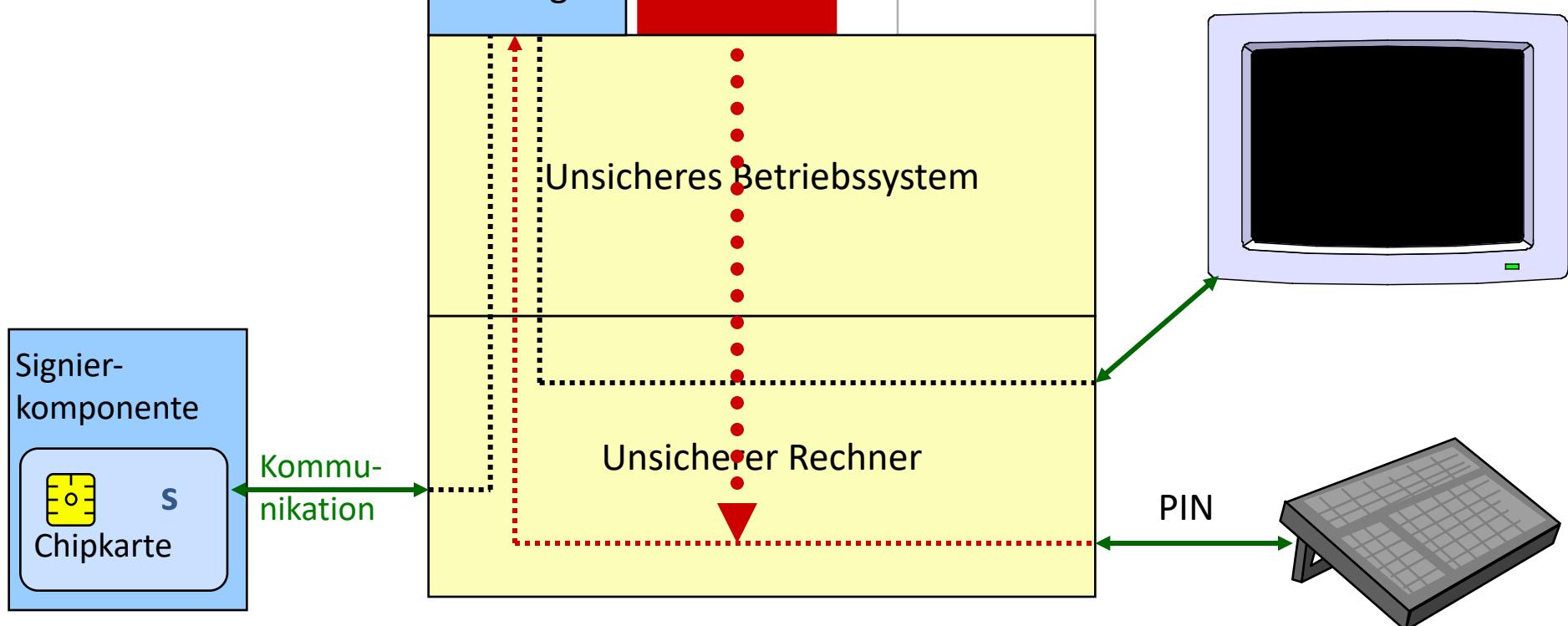
UNSICHER



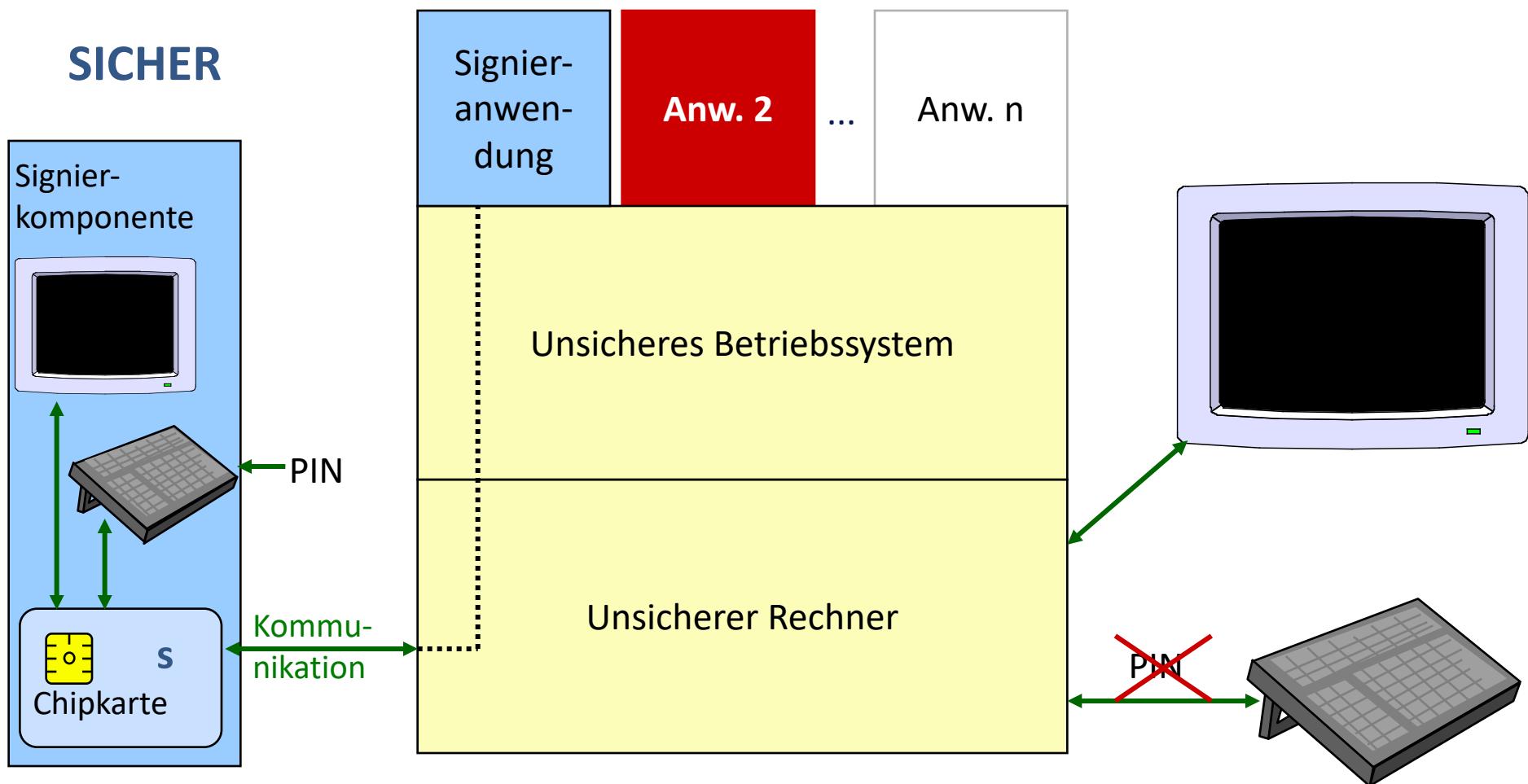
Standard-PC mit Chipkarte

Bösartige Anwendung könnte z.B. PIN abfangen oder Text nach Anschauen und vor Senden an Signierkomponente heimlich ersetzen

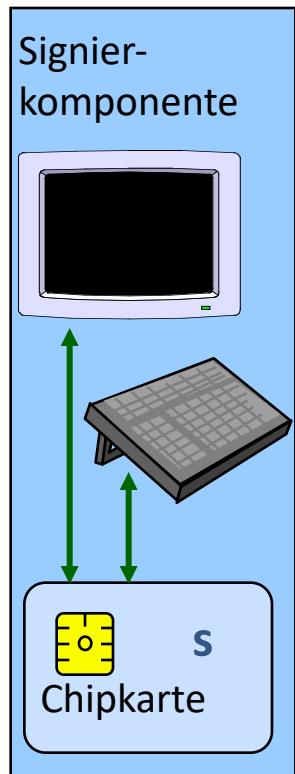
UNSICHER



Sichere Signierkomponente mit Standard-PC



Sichere Signierkomponente



=



- Display
- Tastatur
- Physischer Schutz:
Manipulationserkennung
- Entwurf offen gelegt (keine
versteckten Trojanischen Pferde)

Chipkartenleser: Sicherheitsklassen

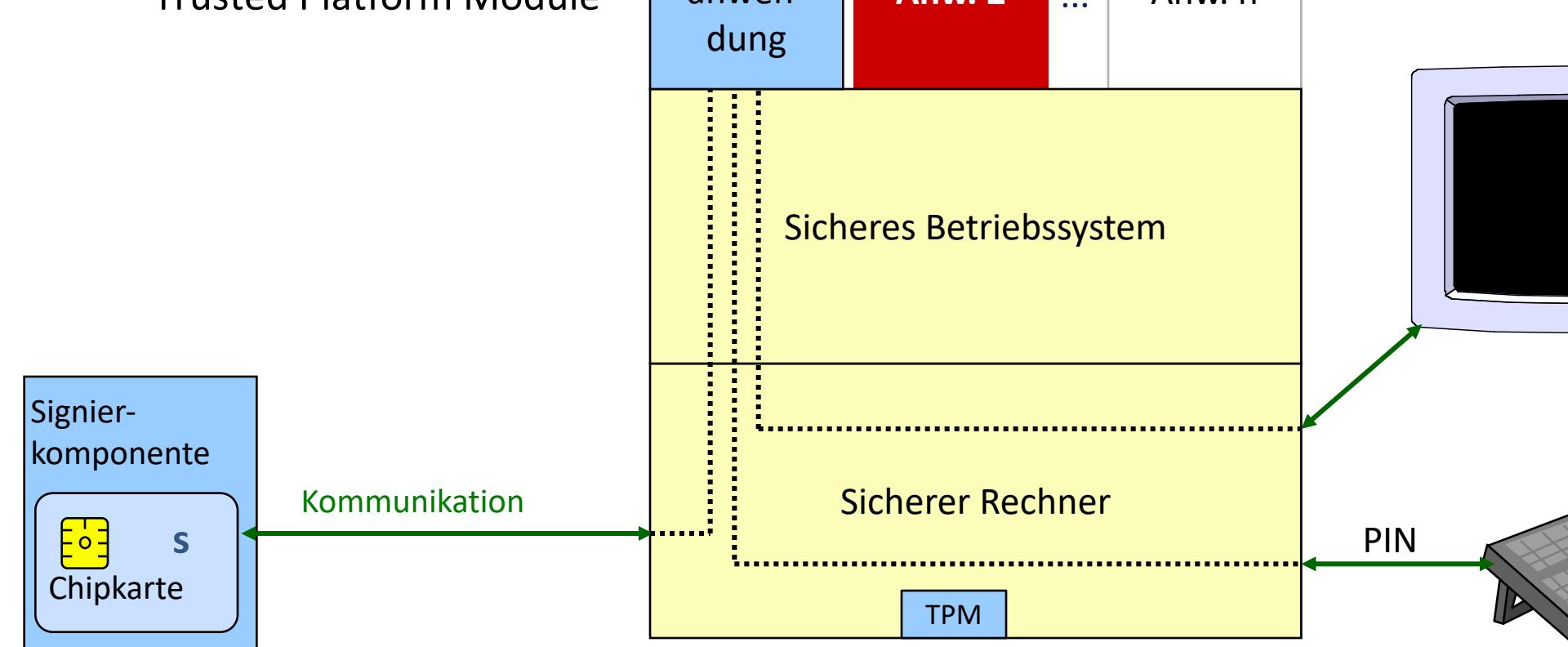
- **Klasse 1**
 - Keine Sicherheitsfunktionen
 - realisieren nur Kommunikation zwischen PC und Leser
- **Klasse 2**
 - PIN-Eingabe kann nicht vom PC mitgelogggt werden
 - Variante 1: PC-Tastatur ist direkt mit Leser verbunden, Verbindung zu PC wird während PIN-Eingabe (physisch) unterbrochen
 - Variante 2: Eigene Tastatur im Leser
- **Klasse 3**
 - eigene Tastatur und eigene Anzeige
 - PC ist nicht an der Kommunikation zwischen Karte, Tastatur und Anzeige beteiligt
- **Klasse 4**
 - eigener Signaturschlüssel
 - kann später ermittelt werden, in welchem Lesegerät die Signatur geleistet wurde



Physisch sichere Geräte und sichere Betriebssysteme

SICHER, wenn

- Physisch sichere Geräte
- sichere Betriebssysteme
- Trusted Platform Module



Detailaspekte zu PKI

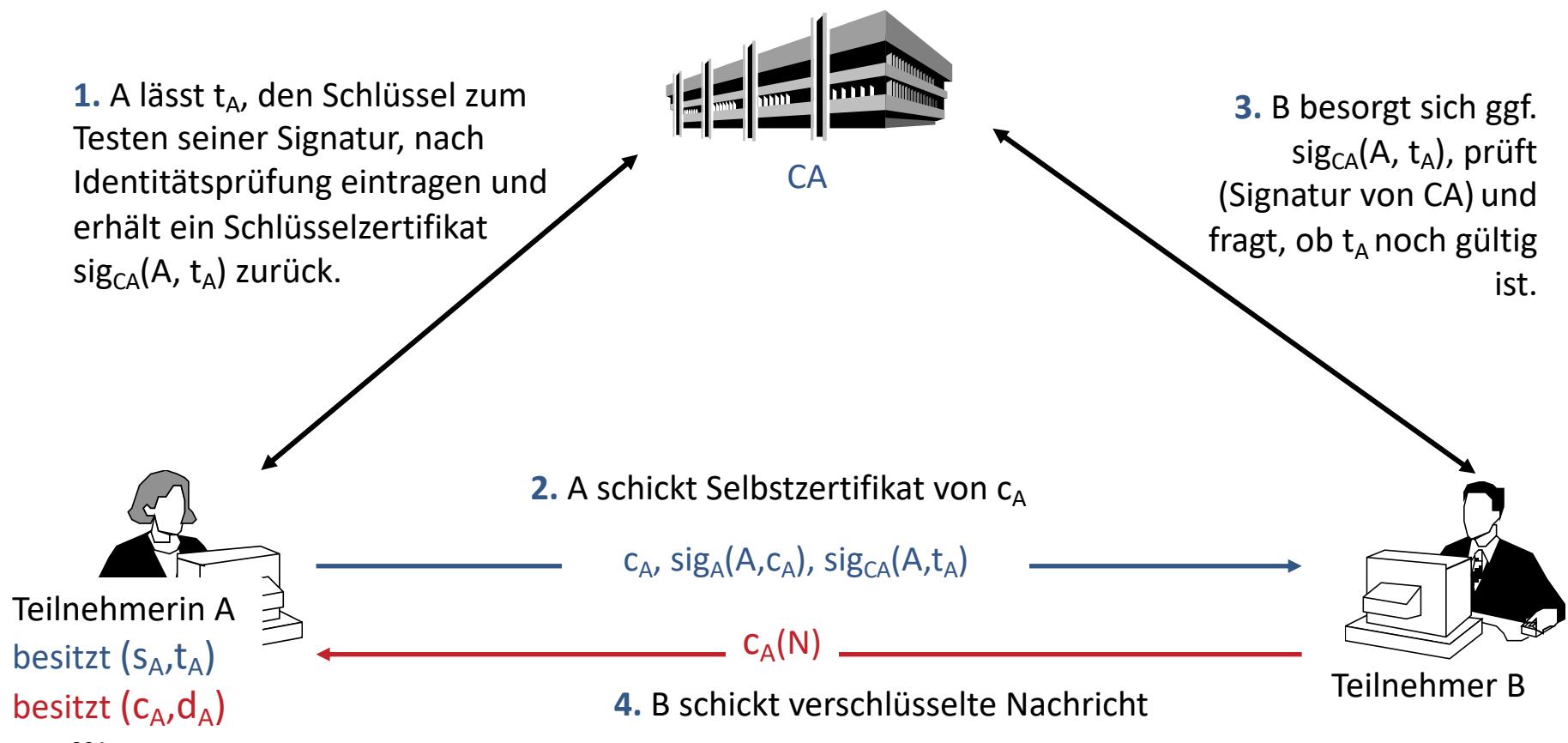
Zeitstempeldienste und Selbstzertifizierung
Gültigkeitsmodelle für Zertifikate und Signaturen

Zeitstempeldienste und Digitale Signatur

- Früher geleistete Signaturen können auch nach Kompromittierung des **Signierschlüssels** noch getestet werden.
- Frage: Wie verhindert man, dass rückwirkend gültige Signaturen erzeugt werden können? (Nach Kompromittierung des Signierschlüssels)
- Nachricht x erhält einen Zeitstempel (fest mit der Nachricht verbunden):
 - Teilnehmer S (Signierer)
 1. bildet **Hash-Wert** (Fingerabdruck) der Nachricht: $h(x)$
 2. fordert **Zeitstempel** von Zeitstempeldienst TS an: $\text{sig}_{TS}(\text{time}, h(x))$
 3. signiert Nachricht und Zeitstempel: $\text{sig}_S(x, \text{sig}_{TS}(\text{time}, h(x)))$
 - Solange der Zeitstempeldienst keine in der Vergangenheit (oder Zukunft) liegende Zeit signiert, kann eine bestimmte Nachricht nicht nachträglich signiert werden
 - Zeitstempel muss einen Fingerabdruck der Nachricht enthalten, damit nicht einfach frühere Zeitstempel der Form $\text{sig}_{TS}(\text{time})$ wiederverwendet werden können.

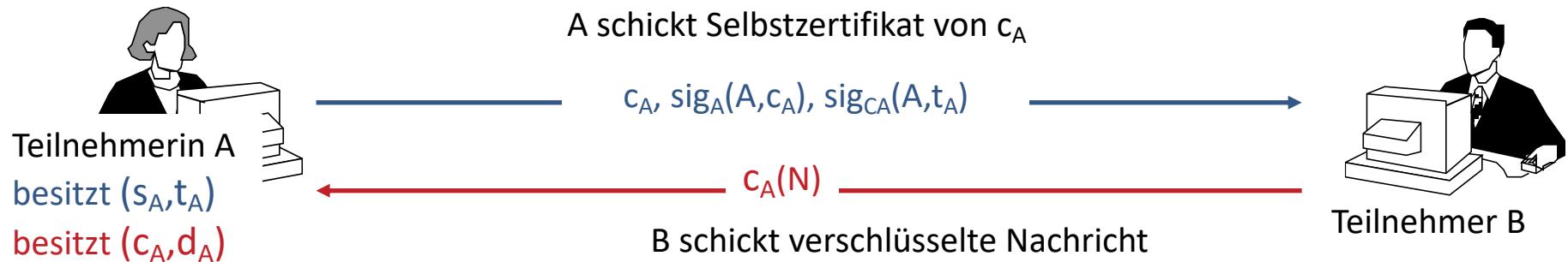
Selbstzertifizierung öffentlicher Verschlüsselungsschlüssel

Unter der Voraussetzung, dass eine funktionierende Infrastruktur für Digitale Signaturen existiert, wird keine zusätzliche für Verschlüsselung benötigt. Vorhandene Infrastruktur kann zur Sicherung der Authentizität der öffentlichen Verschlüsselungsschlüssel verwendet werden.



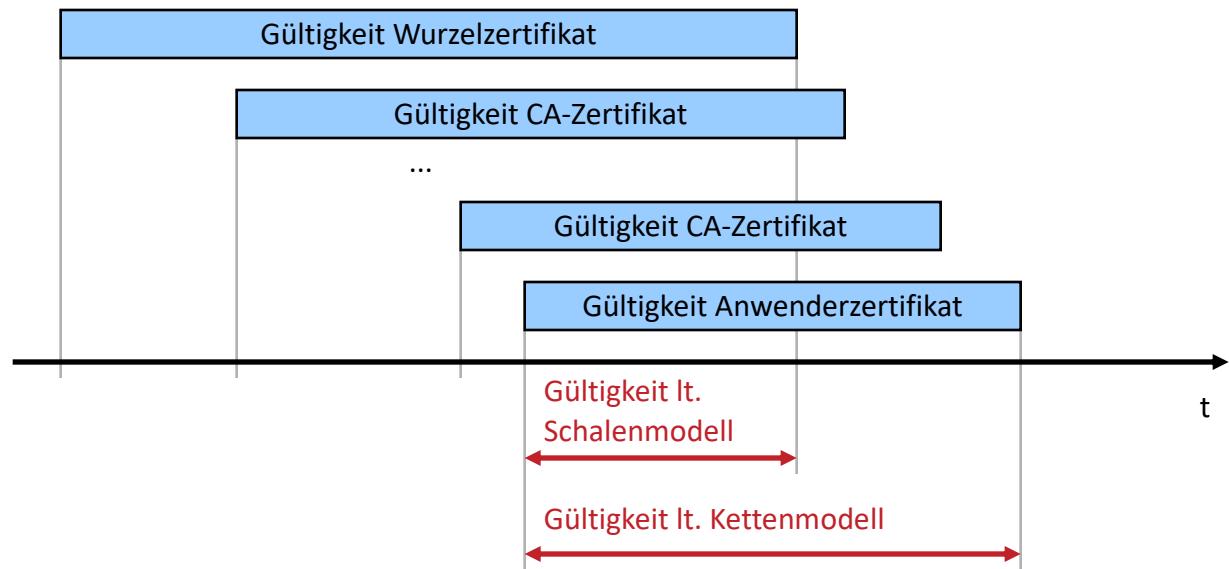
Selbstzertifizierung öffentlicher Verschlüsselungsschlüssel

- Trusted Third Parties werden gebraucht, wenn etwas bewiesen werden soll und nur dort.
 - Digitale Signatur: Beweisbarkeit erforderlich
 - Fremdzertifizierung des Testschlüssels durch Zertifizierungsstellen
- Bei asymmetrischer Verschlüsselung genügt es, sicher zu sein, dass der öffentliche Verschlüsselungsschlüssel authentisch ist.
 - Echtheit von Verschlüsselungsschlüsseln kann über vorhandene Infrastruktur für Digitale Signatur gesichert werden
 - Selbstzertifizierung des öffentlichen Verschlüsselungsschlüssels, jedoch keine Fremdzertifizierung



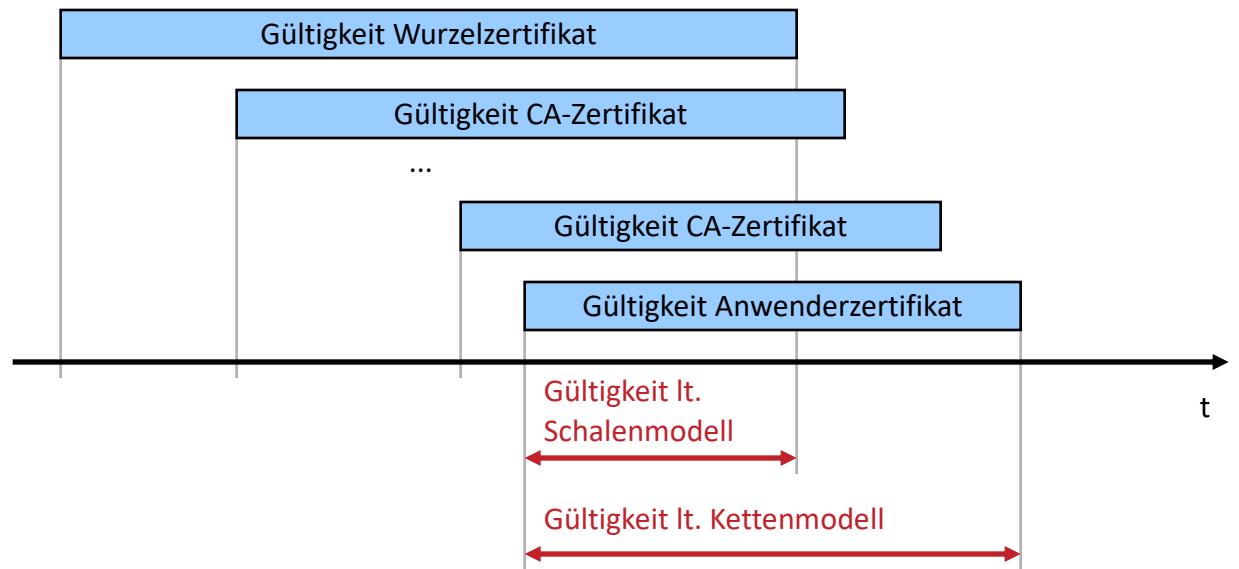
Gültigkeitsmodelle für Zertifikate und Signaturen

- Schalenmodell
 - Eine Signatur unter einem Dokument oder Zertifikat ist dann gültig, wenn zum Zeitpunkt ihrer Erstellung alle zugrunde liegenden Zertifikate gültig waren.
- Kettenmodell
 - Eine Signatur unter einem Dokument oder Zertifikat ist dann gültig, wenn zum Zeitpunkt ihrer Erstellung das zugehörige Schlüsselzertifikat gültig war.



Schalenmodell

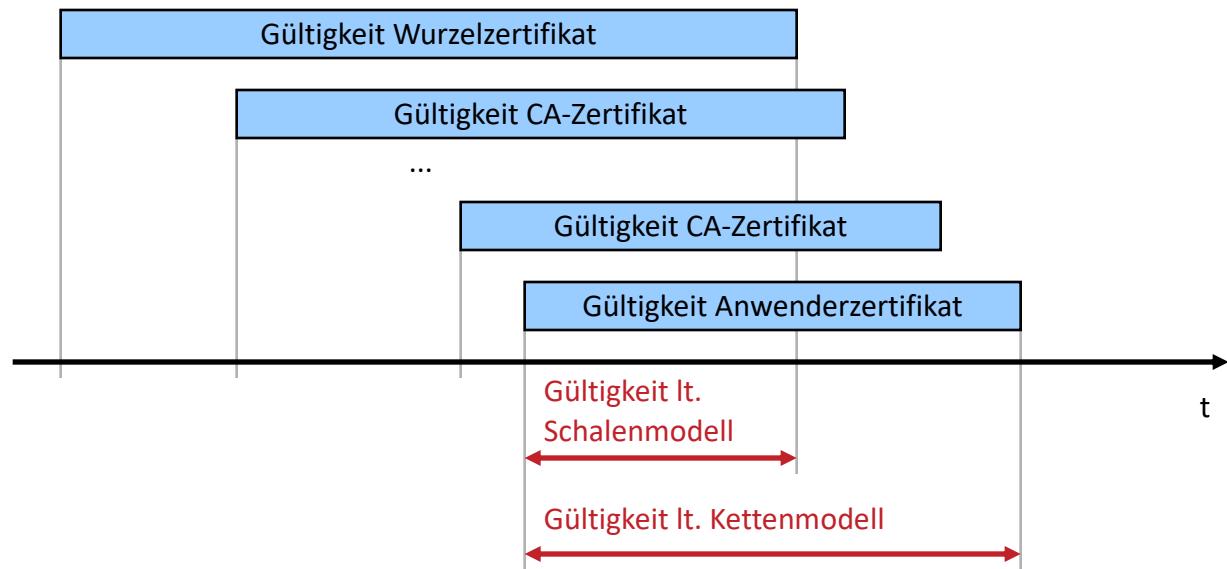
- Bei Ablauf oder Widerruf eines Zertifikats
 - Abgeleitete Zertifikate werden ungültig.
 - Beachte: Signaturen bleiben auch nach Ablauf der Zertifikate gültig, wenn Zertifikate zum Signierzeitpunkt gültig waren.
- Einfache praktische Umsetzung
 - Bei Überprüfung ist für alle Zertifikate derselbe Zeitpunkt maßgeblich.
 - int. Standard, basiert auf X.509 und RFC 3280 (PKIX-AG)



Kettenmodell

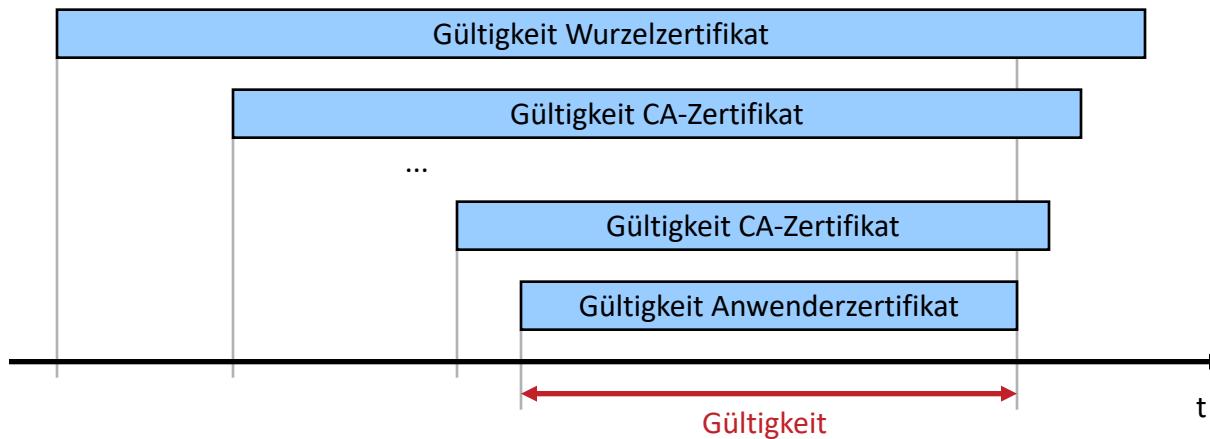
- Bei Ablauf oder Widerruf eines Zertifikats
 - Abgeleitete Zertifikate trotzdem bleiben gültig.
 - rekursive Anwendung auf alle zugrunde liegenden Zertifikate
- Basiert auf einer Forderung des § 19 (5) SigG:

»Die Gültigkeit der von einem Zertifizierungsdiensteanbieter ausgestellten qualifizierten Zertifikate bleibt von der Untersagung des Betriebes und der Einstellung der Tätigkeit sowie der Rücknahme und dem Widerruf einer Akkreditierung unberührt.«



Gültigkeitsmodelle für Zertifikate und Signaturen

- Praxis
 - Übereinstimmende Gültigkeit von Schalenmodell und Kettenmodell bei folgender Situation



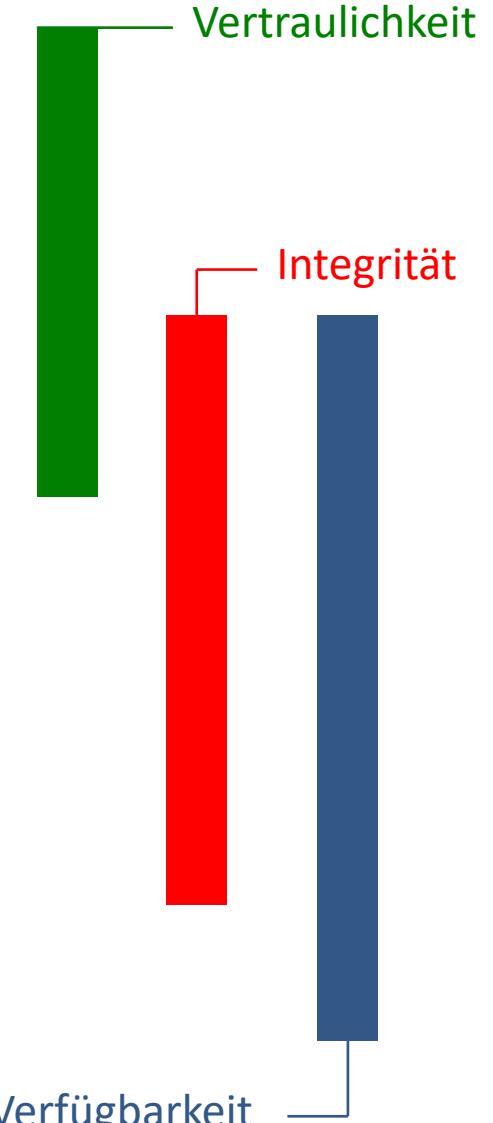
Zusammenfassung

- Schlüsselgenerierung durch Teilnehmergeräte
 - Erhöht Sicherheit
 - Erlaubt vielfältige Pseudonyme, dadurch weniger Profile
- Vertrauenswürdige Zertifizierungsinfrastruktur
 - Beglaubigung der öffentlichen Testschlüssel
 - Keine Hinterlegung von Schlüsseln
 - Kreuzzertifizierung
- Vertrauenswürdige Kommunikation mit Signier- und Anzeigekomponente
 - Sichere Kartenleser mit Display und Tastatur oder
 - Sichere Betriebssysteme

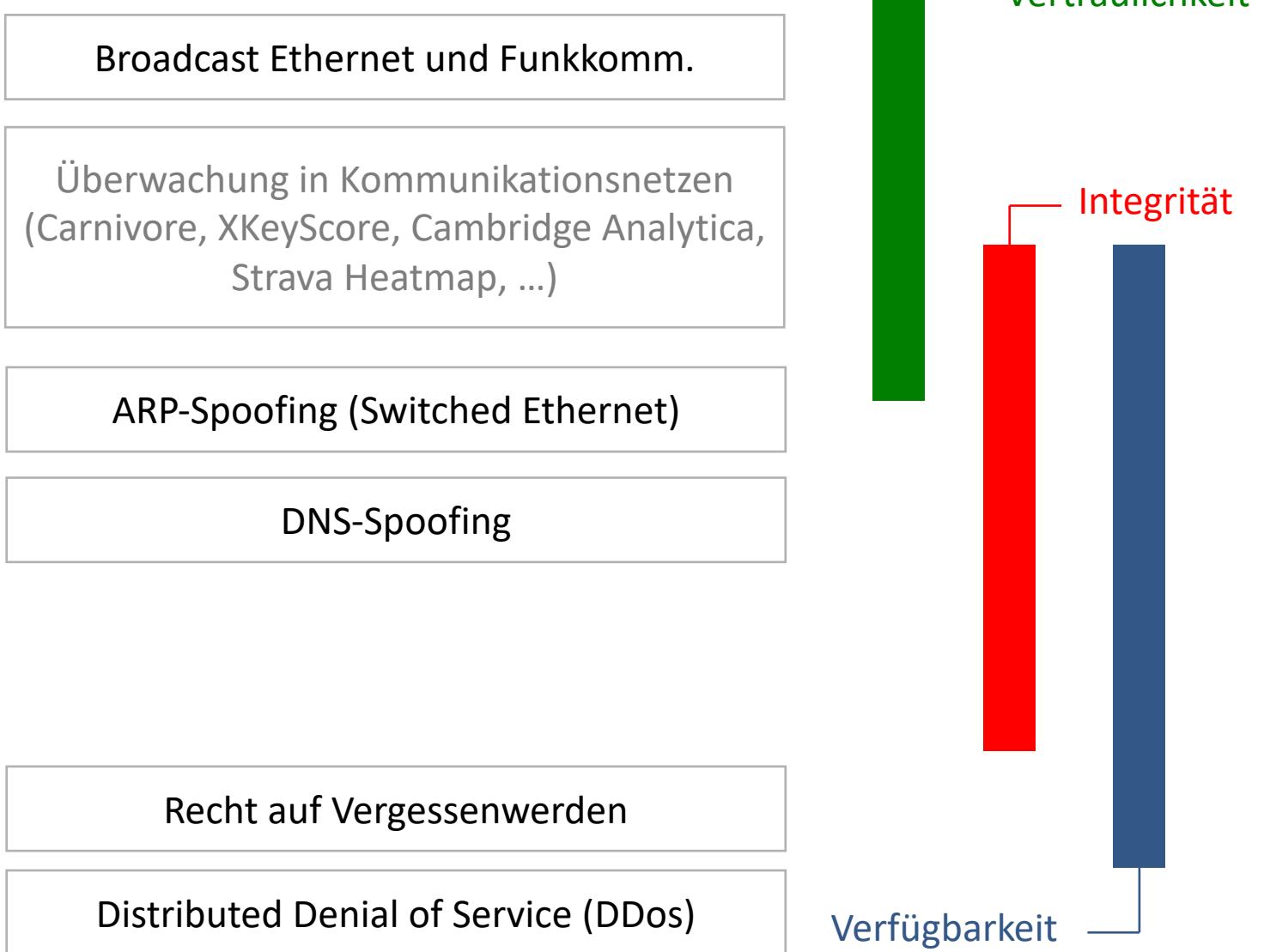
Sniffing, Spoofing, Denial of Service, Internet of Things and Security

Angriffsformen

- Passive Angriffe
 - Lauschangriff (eavesdropping, sniffing)
 - Verkehrsflussanalyse (traffic analysis)
- Aktive Angriffe
 - Maskerade (masquerading)
 - Man-in-the-middle attack
 - Verändern von Daten (modification)
 - Einfügen von Daten (injection, spoofing)
 - Wiederholen (replay)
 - Fluten (flooding, spamming)
 - Dienstverweigerung (denial of service)



Konkrete Beispiele

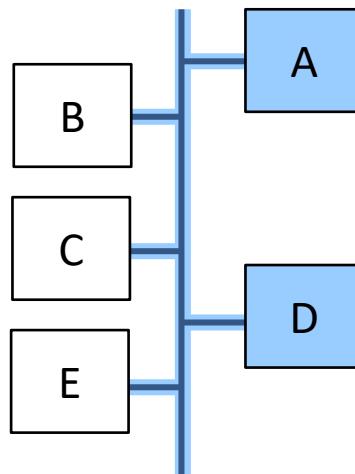


Sniffing-Angriffe: Funktionsweise (Ethernet)

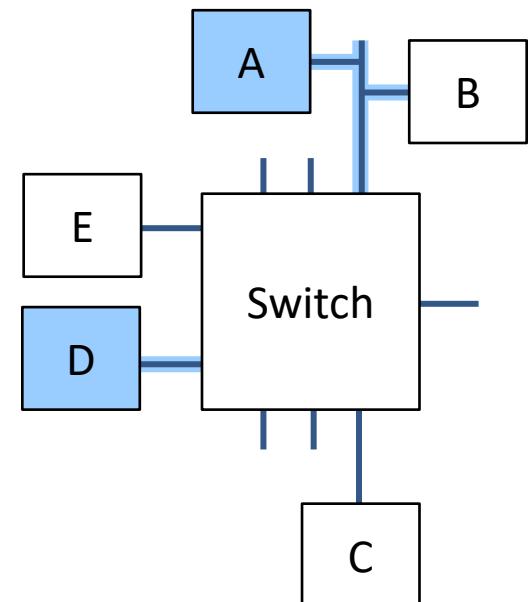
- alle Stationen erhalten alle Datenpakete (im Ethernet)
- lokale Filterfunktion
- Abschalten des Filters möglich: »promiscuous mode«
- Sniffing im Switched Ethernet erschwert

Rechner A und D kommunizieren miteinander

a) im Ethernet



b) im Switched Ethernet



— Ausbreitung der übertragenen Daten

Sniffing-Angriffe: Vorgehen

- 1. Schritt – Beschaffung der Daten
 - Konfiguration der Netzwerkschnittstelle (promiscuous mode)
 - Auslesen sämtlicher Datenpakete
- 2. Schritt – Informationsgewinnung
 - Auswahl der »interessanten« Pakete anhand der Protokoll-Informationen (Sender- bzw. Empfängeradresse, TCP-Port etc.)
- 3. Schritt – Auswertung des Datenteils



```
/usr/bin/login (tty1)
SourceName=
WARNING: Short packet. Try increasing the snap length

11:46:50.885110 arp who-has 160.45.110.189 tell router-110.inf
11:46:51.099430 titanus.inf.fu-berlin.de.49156 > fubinf.inf.fu
11:46:51.100215 fubinf.inf.fu-berlin.de.domain > titanus.inf.f
11:46:51.214719 arp who-has 160.45.110.189 tell router-110.inf
11:46:52.112502 titanus.inf.fu-berlin.de.49156 > fubinf.inf.fu
11:46:52.113040 fubinf.inf.fu-berlin.de.domain > titanus.inf.f
11:46:52.113293 titanus.inf.fu-berlin.de.49156 > fubinf.inf.fu
11:46:52.113706 fubinf.inf.fu-berlin.de.domain > titanus.inf.f
11:46:52.885123 arp who-has 160.45.110.189 tell router-110.inf
11:46:57.010498 jefe.inf.fu-berlin.de > dvmrp.mcast.net: igmp
11:46:58.363997 arp who-has 160.45.110.189 tell router-110.inf
11:46:59.884553 arp who-has 160.45.110.189 tell router-110.inf
11:47:01.884507 arp who-has 160.45.110.189 tell router-110.inf
11:47:03.734152 silver.inf.fu-berlin.de.2611 > 255.255.255.255
11:47:03.884505 arp who-has 160.45.110.189 tell router-110.inf
11:47:05.884498 arp who-has 160.45.110.189 tell router-110.inf
```

Sniffing-Angriffe: Vorgehen

■ 3. Schritt – Auswertung des Datenteils

- Im Beispiel ASCII-Textdarstellung eines Ethernet-Datenpaketes gewählt (Punkte stehen für Steuerzeichen)

```
....Ih..OyB..OyB...E...S'@.....QP\..G<..C.H.M../(~.P.....>.*....  
..E.....w.R$..6..f%A....4.6.f%A.....  
.....U.....MailSaveOptions...O.U.....SECUREMAIL..  
U.....tmpReview...U.....Form MemoU.....Type..  
MemoU.....DeletionPeriod.....>@U.....HoldPeriod..  
.....U.....ReturnReceipts..OnU.....DeliveryReport  
--B=U.....Sign..liU.....DefaultMailSaveOptions..lrU.  
D.....ReplyToa..U.....Body.....Hallo,...  
.....,.....das ist ein Test f.r unsere Sneaker.....  
.....THE MAGIC WORDS ARE FEEBLE GIBBERISH.....  
.....Gru.,.....Matthias  
Mueller.....U.....ReminderDate..U.....Dele  
tionDate..U.....Encrypts..OtU.....$Folders..U.....  
...PreparedToSend..O U.....DeliveryPriority..NMU.....  
..$KeepPrivate..U.....Subject ..Testmail fuer SniffingU.E.  
..6.....SendTo..CN=Andreas Maier/OU=DuD/OU=Datenschutz/O=TUD@TU-Dresd  
enU.E.....CopyTo..U.D.....BlindCopyTo..U.E.../.....Fr  
om..CN=Matthias Mueller/OU=DuD/OU=Datenschutz/O=TUD.EU.....Po  
stedDate..}.6..f%AU.....i.....$Signature.....X6..f%A.....O...  
.....6...H.....j8..d%.....&....@.....$.  
.a%...$.t.%.....O=TUD.....O=TUD.....BV...l.0.BC...BA..0BL..v.NN  
P....w...%m...]i.u....;...ys}..}...4]..yl.)....c...|ohi<'..5L.r..B...  
BZ%;m<.....L...Q])..EN..D..MA..l...So;|..PURSAFO..d.YK.....<>3.....  
.#->k.....|..Jj//...R... |..U...ka..Ofz.....@@
```

Sniffing-Angriffe: Abwehr

schwacher
Angreifer

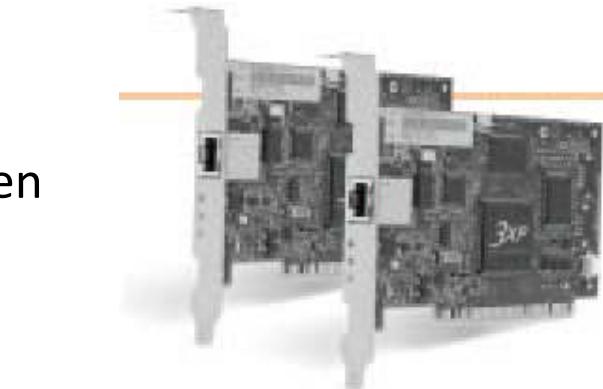


- Physischer Schutz
 - incl. physischer Schutz des Übertragungsmediums
- Netzwerkadapter
 - ohne »promiscuous mode«
 - Signalisierung des Umschaltens in »promiscuous mode«
 - switched networks
- Schutz gegen einen relativ starken Angreifer
 - kann Datentransfer über das Medium ablauschen
 - Einsatz von Verschlüsselungsverfahren

starker
Angreifer

Sniffing-Angriffe: Abwehr

- Hardwareverschlüsselung direkt auf Netzwerkkarte
 - Historisches Beispiel:
 - 3COM 10/100 Secure Network Interface Cards
 - IPSec-Verschlüsselung mit 3DES und DES
 - IPSec-Authentikation (RFC 2402 Authentication Header) mit SHA-1 und MD5
 - enthält Kryptoprozessor
 - Variante für Client-PCs und Server
 - Speichert bis zu 700 bzw. 1000 Security Associations (Schlüssel der Gegenstelle)
 - wird nicht mehr vertrieben



Quelle:

http://www.3com.com/products/en_US/detail.jsp?tab=features&pathtype=purchase&sku=3CR990-TX-97

Das Recht auf Vergessen im Internet

■ Kontext: Allgemeines Ziel

- Verbotenes und Unerwünschtes im Internet soll

nicht möglich

nicht mehr vorhanden

wenigstens nicht mehr erreichbar

(Untaugliche) Lösungsansätze

[Juristisch: Verbote]

Technisch: x-pire!

Technisch: DNS-Sperre

- sein.

Google saufbilder

Alle Bilder Shopping Videos News Mehr Einstellungen Tools SafeSearch ▾

alkohol alkohol exzesse partys peinliche mallorca ballermann betrunken verona pooth > mehr

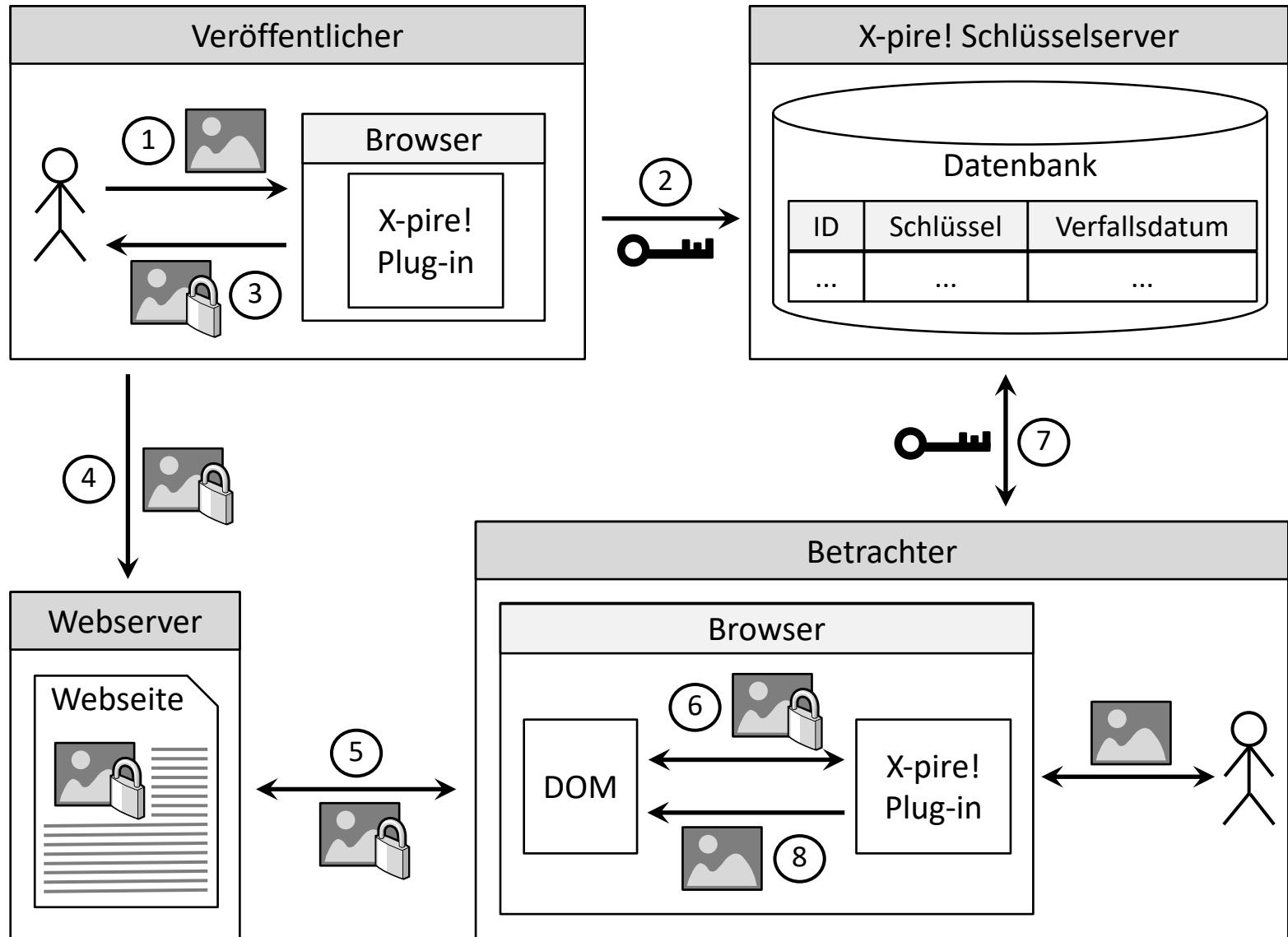
Das Internet vergisst nichts.

- Soziale Netze wissen viel über Menschen, ihre Entwicklungen und Fehlritte.
- Geschlossene Benutzergruppe wünschenswert

X-pire!



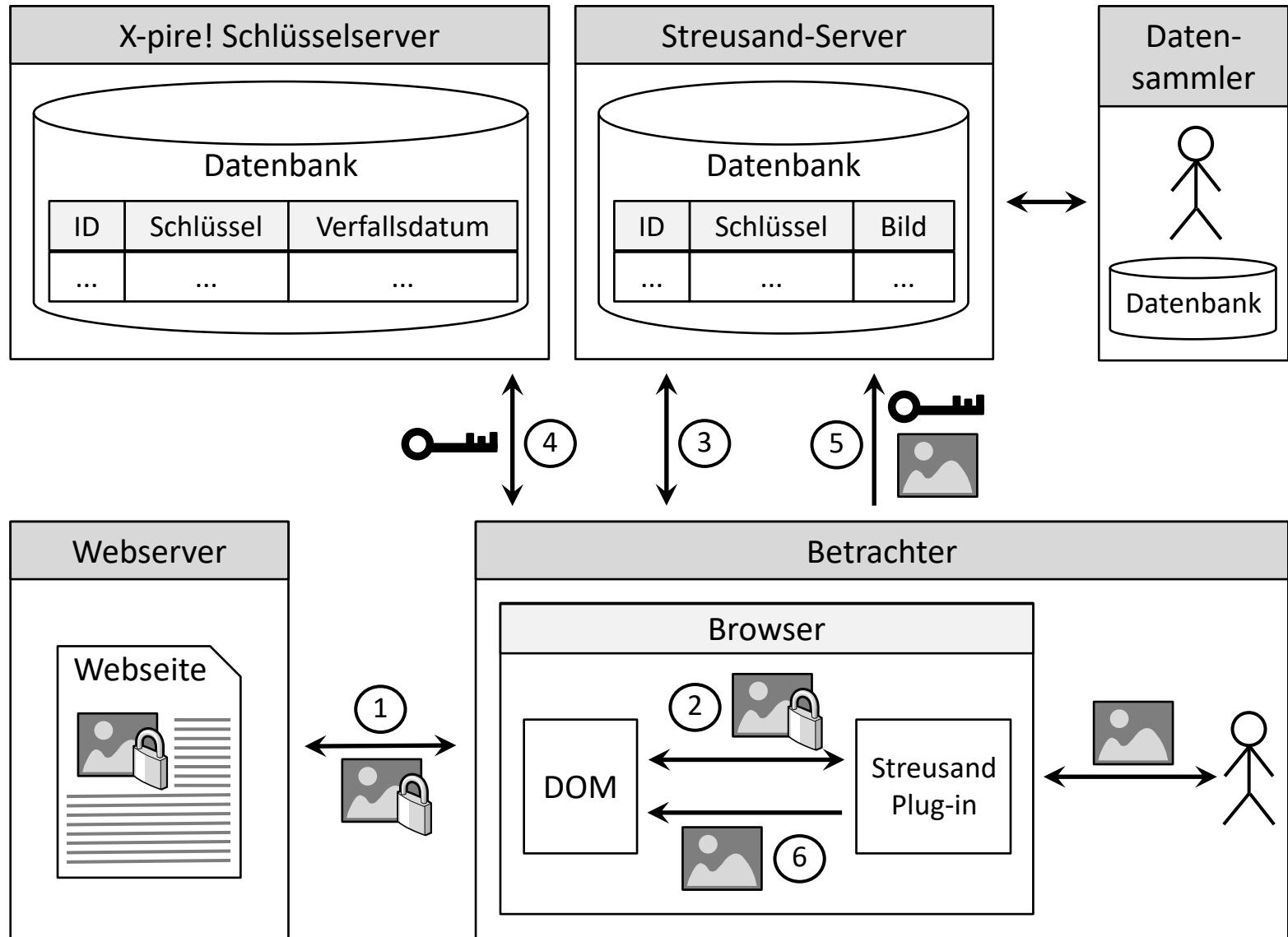
Funktionsweise X-pire!



Sicherheitsaspekte

- Zentraler Schlüsselserver
 - Verfügbarkeit: single-point-of-failure
 - Vertraulichkeit: Datenbank-Betreiber kennt alle Schlüssel
 - Erweiterungen denkbar:
 - Verteilte Datenbanken
 - Verwendung von Secret-Sharing-Verfahren und Anonymitätstechniken
- Kein Schutz gegen Angreifer in der Rolle »Betrachter«
 - Software im Verfügungsbereich des Betrachters (Browser) erhält Zugriff auf Schlüssel und unverschlüsselten Inhalt
 - Weder Verschlüsselung noch CAPTACHs helfen hier!
- Streisand-Effekt
 - Insbesondere Inhalte, die wieder aus dem Netz verschwinden sollen, halten sich möglicherweise besonders lange.

Funktionsweise Streusand-Erweiterung



Streusand-Erweiterung

- Nutzung von X-pire! kann sogar schädlich sein
 - Streusand-Galerie: längst verschwundene Bilder archiviert

01.02.2011 08:25:36	4d4d75d742863ab9656f3d5f76dff858	a7385c51a13dd53030ee2f18c7fcb689ad4094b06ff90c601c3abac722f1f5c	
31.01.2011 20:24:00	ab897fbdedfa502b2d839b6a56100887	eee65472de6234f647cf5c25d959e2f116707f76bcb7a5a5de2ad1a99e1d4628	
31.01.2011 20:23:12	ab897fbdedfa502b2d839b6a56100887	17150bb7b618f8e11358b5d8b7d6be438394213eb2a5e582703d8ee733c198e1	
31.01.2011 20:21:08	ab897fbdedfa502b2d839b6a56100887	2b4c6711793140ea5fa88c27f61354034f69dbdbaaae82f6c88490fc019bd09	
27.01.2011 18:29:03	e6f207509afa3908da116ce61a757695	fb1c038c912c46c41181c8cb32b39e396abacdb0abf1d0683b6ca3d12ee386ba	

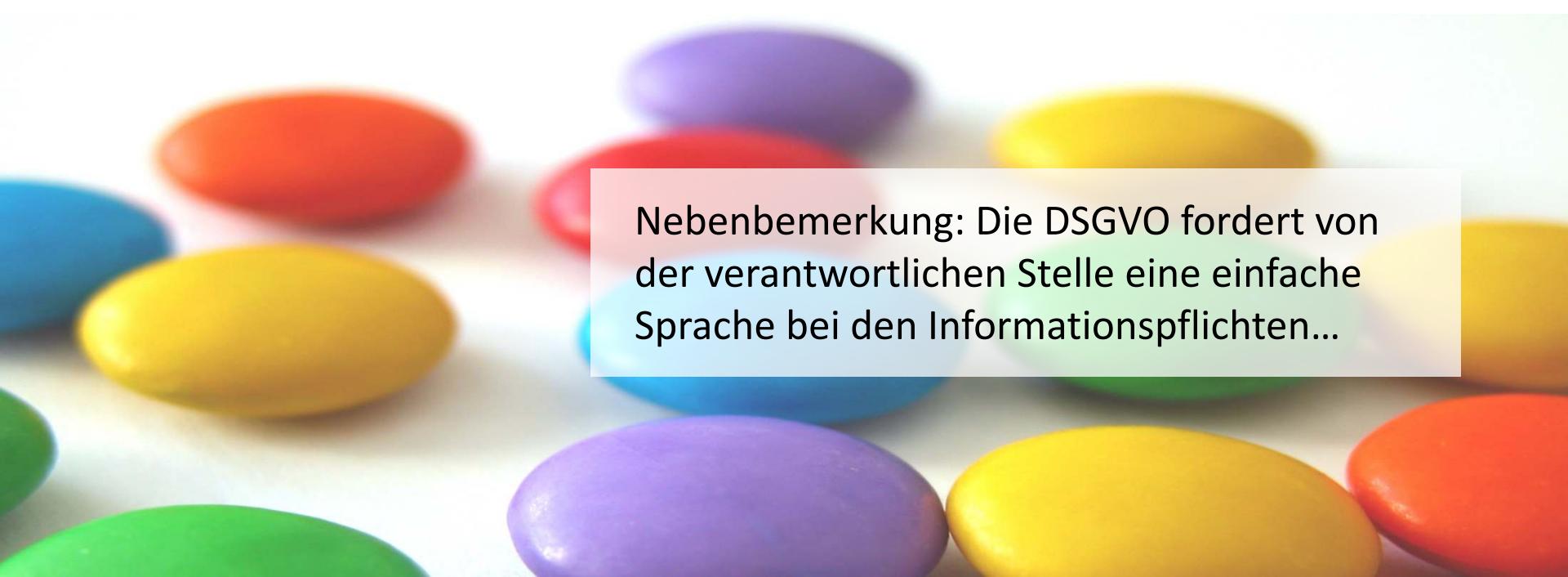
- Softwarelösungen zur Durchsetzung des Rechts auf Vergessen sind untauglich.

Auszug aus Artikel 17 DSGVO

Art. 17 DSGVO Recht auf Löschung (»Recht auf Vergessenwerden«)

(1) ...

(2) Hat der Verantwortliche die personenbezogenen Daten öffentlich gemacht und ist er gemäß Absatz 1 zu deren Löschung verpflichtet, so trifft er unter Berücksichtigung der verfügbaren Technologie und der Implementierungskosten angemessene Maßnahmen, auch technischer Art, um für die Datenverarbeitung Verantwortliche, die die personenbezogenen Daten verarbeiten, darüber zu informieren, dass eine betroffene Person von ihnen die Löschung aller Links zu diesen personenbezogenen Daten oder von Kopien oder Replikationen dieser personenbezogenen Daten verlangt hat.

A background image showing a pile of colorful, oval-shaped candies in various colors like orange, yellow, blue, green, and purple, scattered across a light surface.

Nebenbemerkung: Die DSGVO fordert von der verantwortlichen Stelle eine einfache Sprache bei den Informationspflichten...

support.google.com

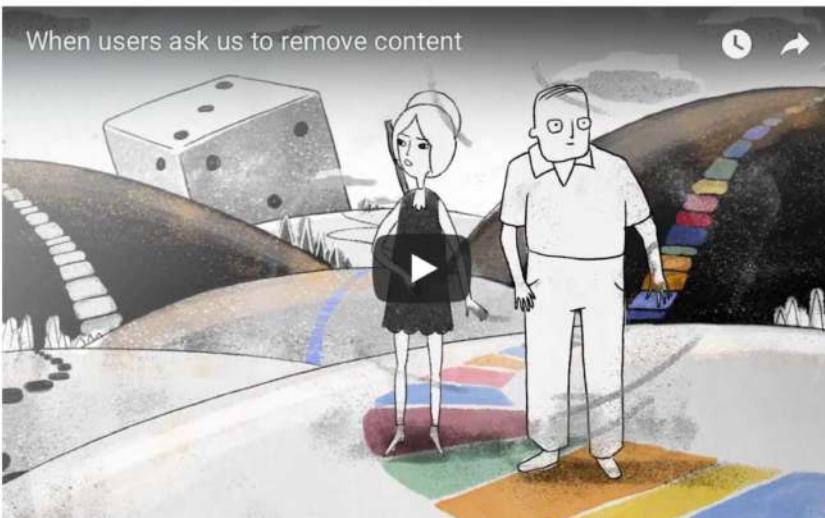
Anträge auf Entfernung von Inhalten - Hilfe für Rechtliche Hinweise Entfernen von Inhalten aus Google - Hilfe für Rechtliche Hinweise

Google Anmelden

Hilfe für Rechtliche Hinweise

Anträge auf Entfernung von Inhalten

When users ask us to remove content



Falls Sie auf Google Inhalte finden, die mutmaßlich rechtswidrig sind, informieren Sie uns. Wir werden das Material dann sorgfältig prüfen und gegebenenfalls den Zugriff darauf sperren oder beschränken bzw. das Material entfernen. Missbräuchliche Inhalte in den Google-Diensten verstößen möglicherweise auch gegen die [Produktrichtlinien von Google](#). In diesem Fall können Sie die betreffenden Inhalte melden, bevor Sie einen Antrag auf Entfernung der Inhalte einreichen. Die Inhalte werden dann von unseren Teams geprüft. Im Folgenden finden Sie weitere Informationen zu unseren Produktrichtlinien und Datenschutzerklärungen sowie unserem Bekenntnis zu Transparenz. Zudem können Sie hier nachlesen, wie Sie eine gültige rechtliche Mitteilung bei Google einreichen.

Problemspezifische Unterstützung finden

RECHTLICHE HINWEISE

Informationen zur Entfernung von Inhalten

Fehlerbehebung bei Datenschutzproblemen

Hilfe zum Urheberrecht

Häufig gestellte Fragen

Anträge auf Entfernung von Inhalten - Hilfe für Rechtliche Hinweise Entfernen von Inhalten aus Google - Hilfe für Rechtliche Hinweise

Google Anmelden

Hilfe für Rechtliche Hinweise

Entfernen von Inhalten aus Google

Auf dieser Seite finden Sie Hinweise dazu, wo Sie Inhalte melden können, die Sie gemäß geltendem Recht aus den Diensten von Google entfernen lassen möchten. Wir können Ihrer Anfrage am besten nachgehen, wenn Sie vollständige Angaben machen.

Bei nicht juristischen Problemen, die die [Nutzungsbedingungen](#) oder Produktrichtlinien von Google betreffen, rufen Sie bitte <http://support.google.com> auf.

Bitte reichen Sie eine separate Meldung für jeden Google-Dienst ein, bei dem der betreffende Inhalt zu sehen ist.

Auf welches Google-Produkt bezieht sich Ihre Anfrage?

-  Blogger/Blogspot
-  Google+
-  Google Websuche
-  Autocomplete und Verwandte Suchanfragen
-  Eine Google-Anzeige
-  Google My Business (Bewertungen, Fragen und Antworten (F&A) sowie Brancheneinträge)
-  Google Drive und Google Docs
-  Google Play – Musik
-  Google Play – Apps
-  Google Shopping
-  Google Bilder
-  Picasa
-  YouTube

Anträge auf Entfernung von Inhalten - Hilfe für Rechtliche Hinweise Entfernen von Inhalten aus Google - Hilfe für Rechtliche Hinweise

Google Anmelden

Hilfe für Rechtliche Hinweise

Entfernen von Inhalten aus Google

Auf dieser Seite finden Sie Hinweise dazu, wo Sie Inhalte melden können, die Sie gemäß geltendem Recht aus den Diensten von Google entfernen lassen möchten. Wir können Ihrer Anfrage am besten nachgehen, wenn Sie vollständige Angaben machen.

Bei nicht juristischen Problemen, die die [Nutzungsbedingungen](#) oder Produktrichtlinien von Google betreffen, rufen Sie bitte <http://support.google.com> auf.

Bitte reichen Sie eine separate Meldung für jeden Google-Dienst ein, bei dem der betreffende Inhalt zu sehen ist.

Auf welches Google-Produkt bezieht sich Ihre Anfrage? [Google Websuche](#)

Wobei können wir Ihnen helfen? [Ich möchte, dass meine personenbezogenen Daten aus den Google-Suchergebnissen entfernt werden.](#)

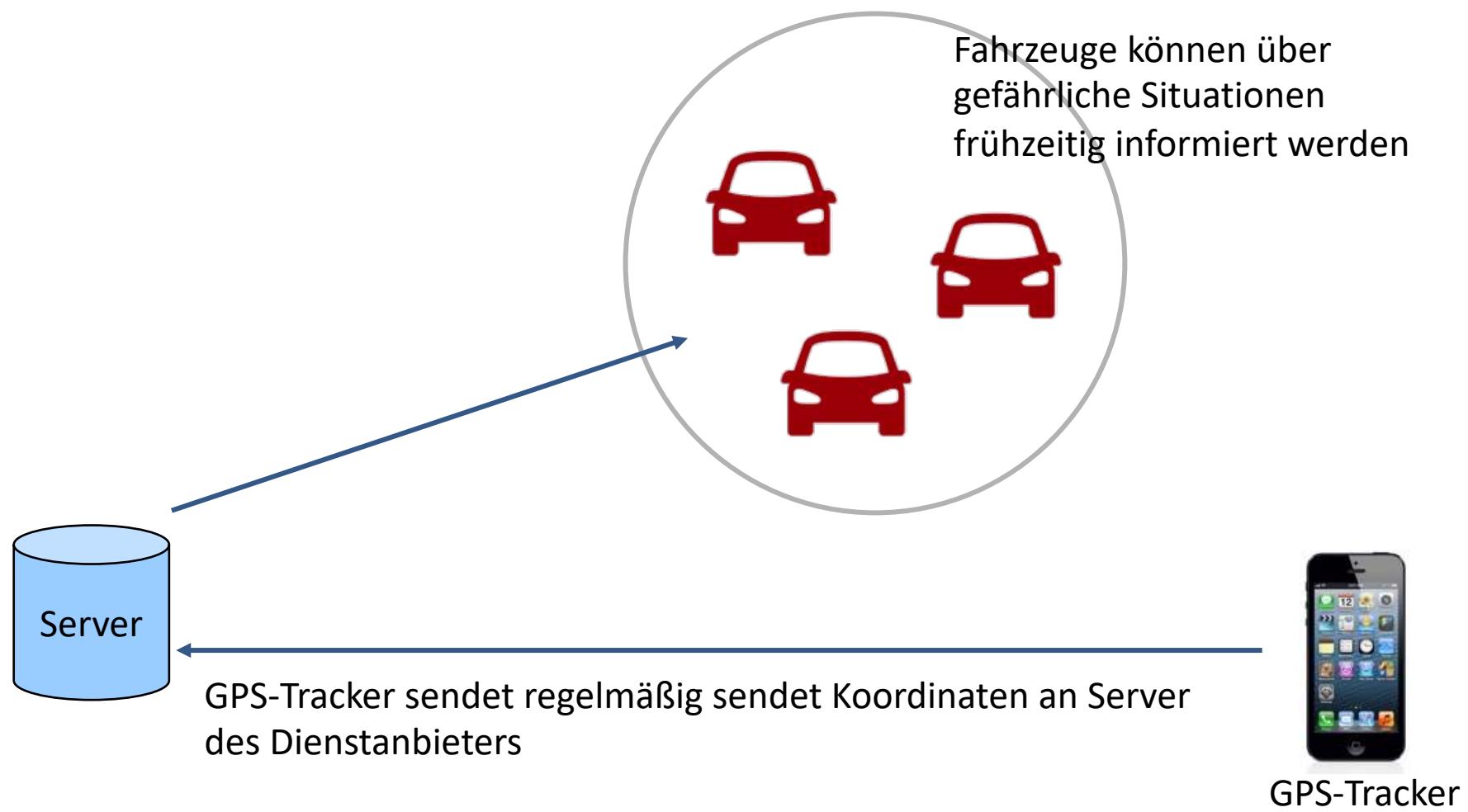
Beachten Sie, dass in unseren Suchergebnissen möglicherweise ein Hinweis angezeigt wird, dass einige Ergebnisse entfernt wurden.

Treffen Sie eine Auswahl aus folgenden Optionen.

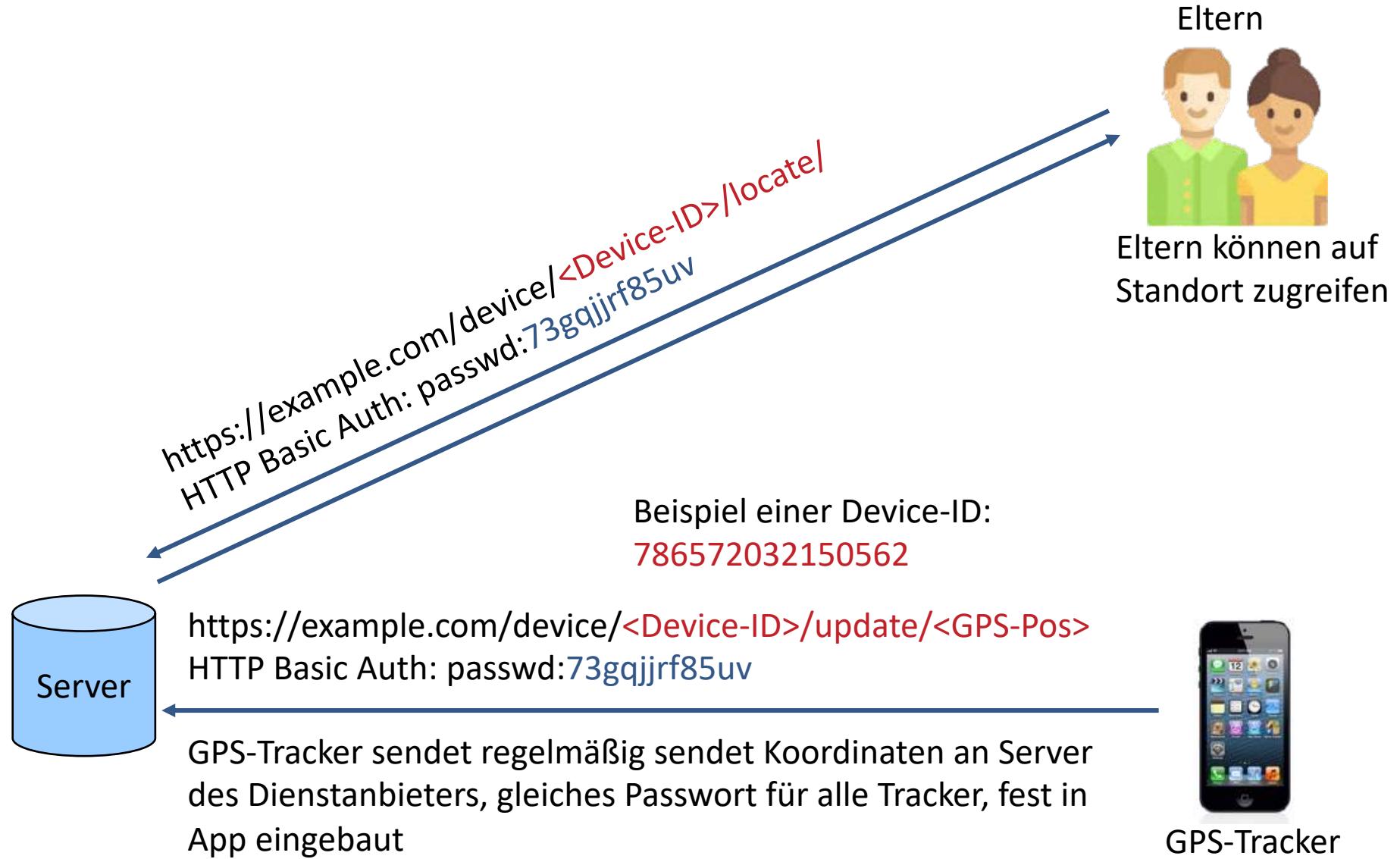
- Ich möchte gemäß den europäischen Datenschutzgesetzen bezüglich des "Rechts auf Vergessen" einen Antrag auf Entfernung von Informationen einreichen.
- Ich möchte, dass meine vertraulichen personenbezogenen Daten aus den Google-Suchergebnissen entfernt werden (z. B. Sozialversicherungs- oder Ausweisnummer, Konto- oder Kreditkartennummer oder ein Bild meiner handschriftlichen Unterschrift, ein Nacktbild oder -video bzw. sexuell explizite Bilder oder Videos von mir, die ohne mein Einverständnis veröffentlicht wurden).
- Ich möchte den Webmaster einer in den Suchergebnissen enthaltenen Seite, die falsche oder ungenaue Informationen enthält, auffordern, diese vollständig aus den Google-Suchergebnissen zu entfernen.

Unzureichende Implementierung von
Schutzmaßnahmen kann Vertrauen in IT
gefährden

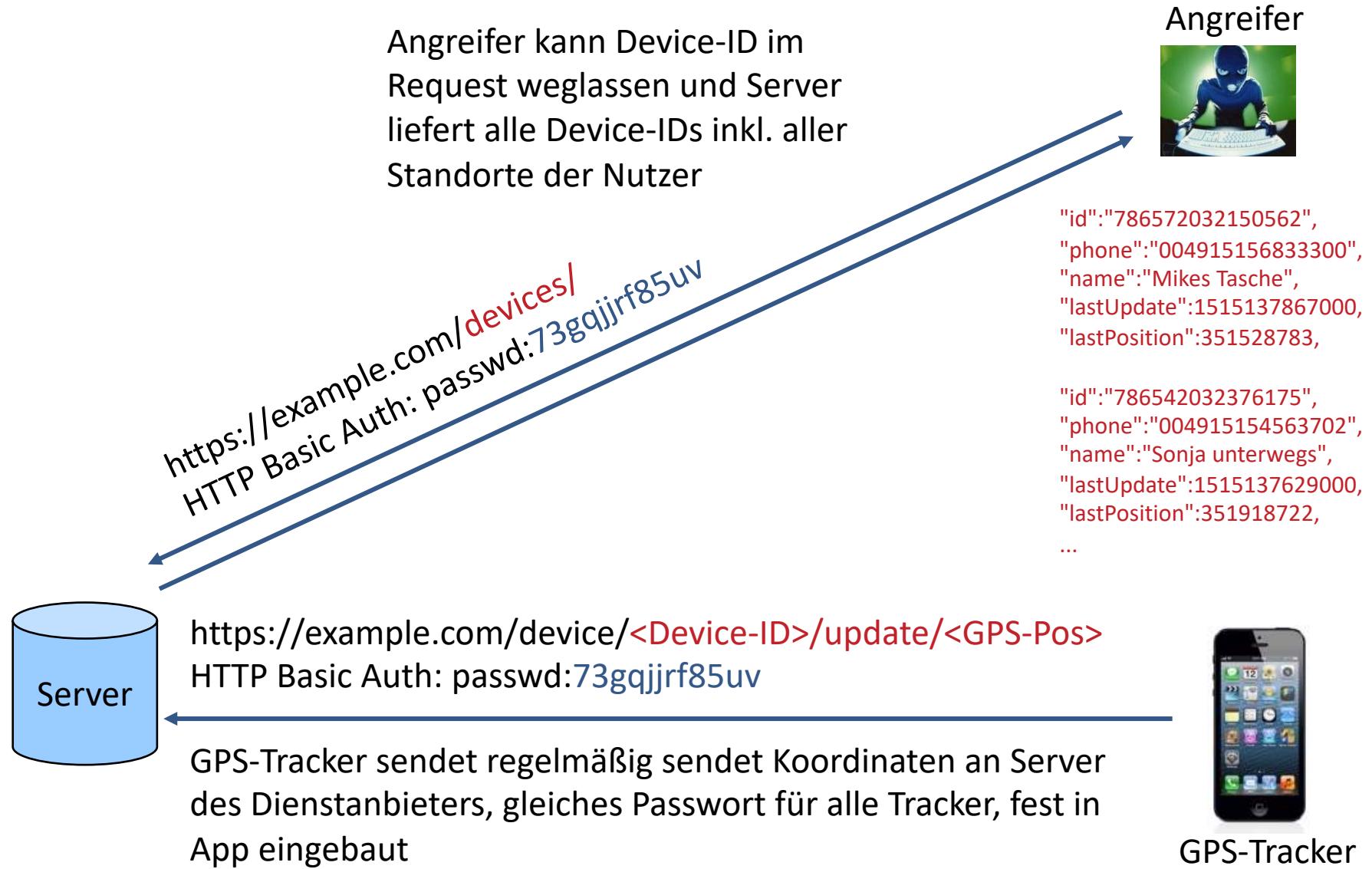
Unzureichende Schutzmaßnahmen gefährden Vertrauen in IT



Unzureichende Schutzmaßnahmen gefährden Vertrauen in IT



Unzureichende Schutzmaßnahmen gefährden Vertrauen in IT



Spoofing

Integrität: Spoofing

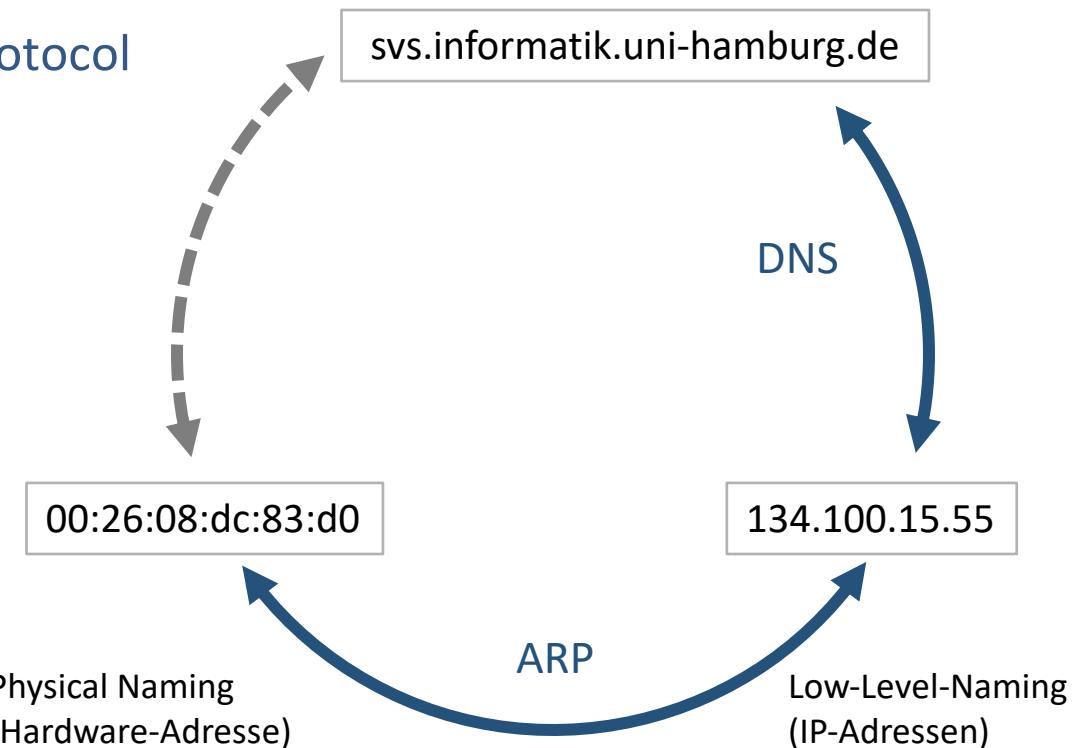
- Was ist Spoofing?
 - Vortäuschen falscher Information
 - Angriffe gegen die Integrität
 - auch mit dem Ziel, schließlich die Vertraulichkeit zu verletzen
- Arten von Spoofing
 - Mail-Spoofing
 - IP-Spoofing
 - DNS-Spoofing
 - ARP-Spoofing
 - SSID-Spoofing

- Szenario 1:
 - ISP greift an
 - DNS-Sperre als Beispiel
- Szenario 2:
 - Angriff im LAN
 - ARP- und DNS-Spoofing mit Tool Cain&Abel

Einordnung ARP, IP, DNS

- DNS: Domain Name System
 - Abbildung des Rechnernamens auf IP-Adresse
 - Anfrage an Nameserver
 - typischerweise in WANs

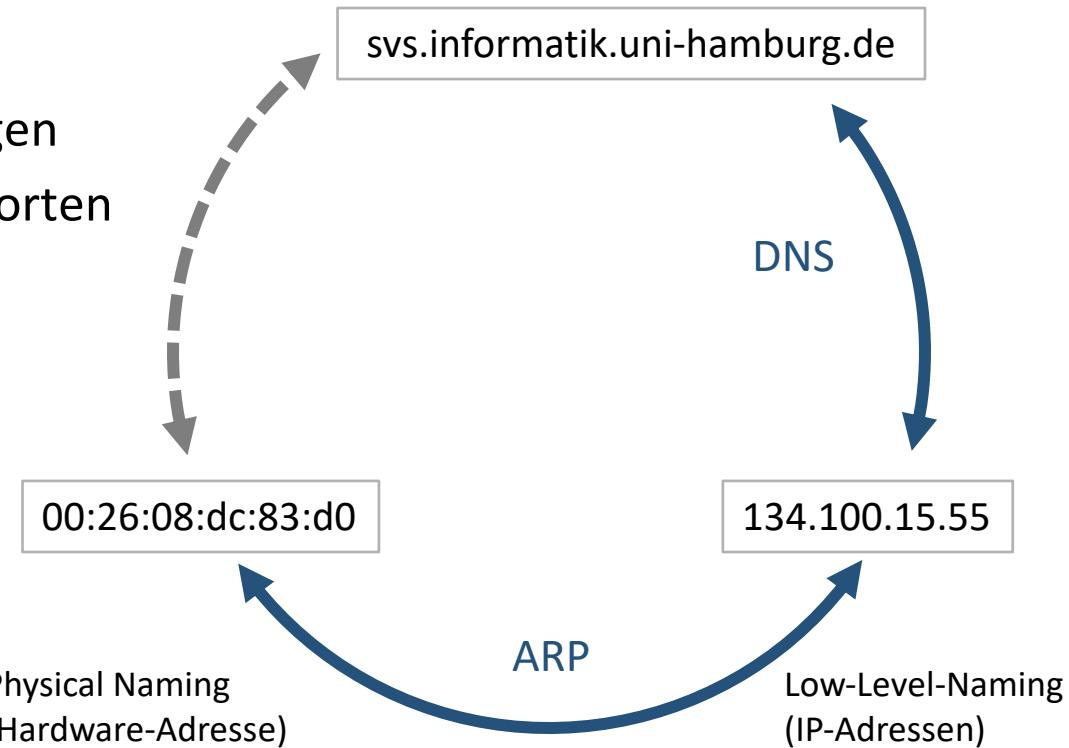
- ARP: Address Resolution Protocol
 - Abbildung von IP-Adresse auf Hardwareadresse
 - Anfrage an das lokale Netz (Broadcast)
 - nur in lokalen Netzen



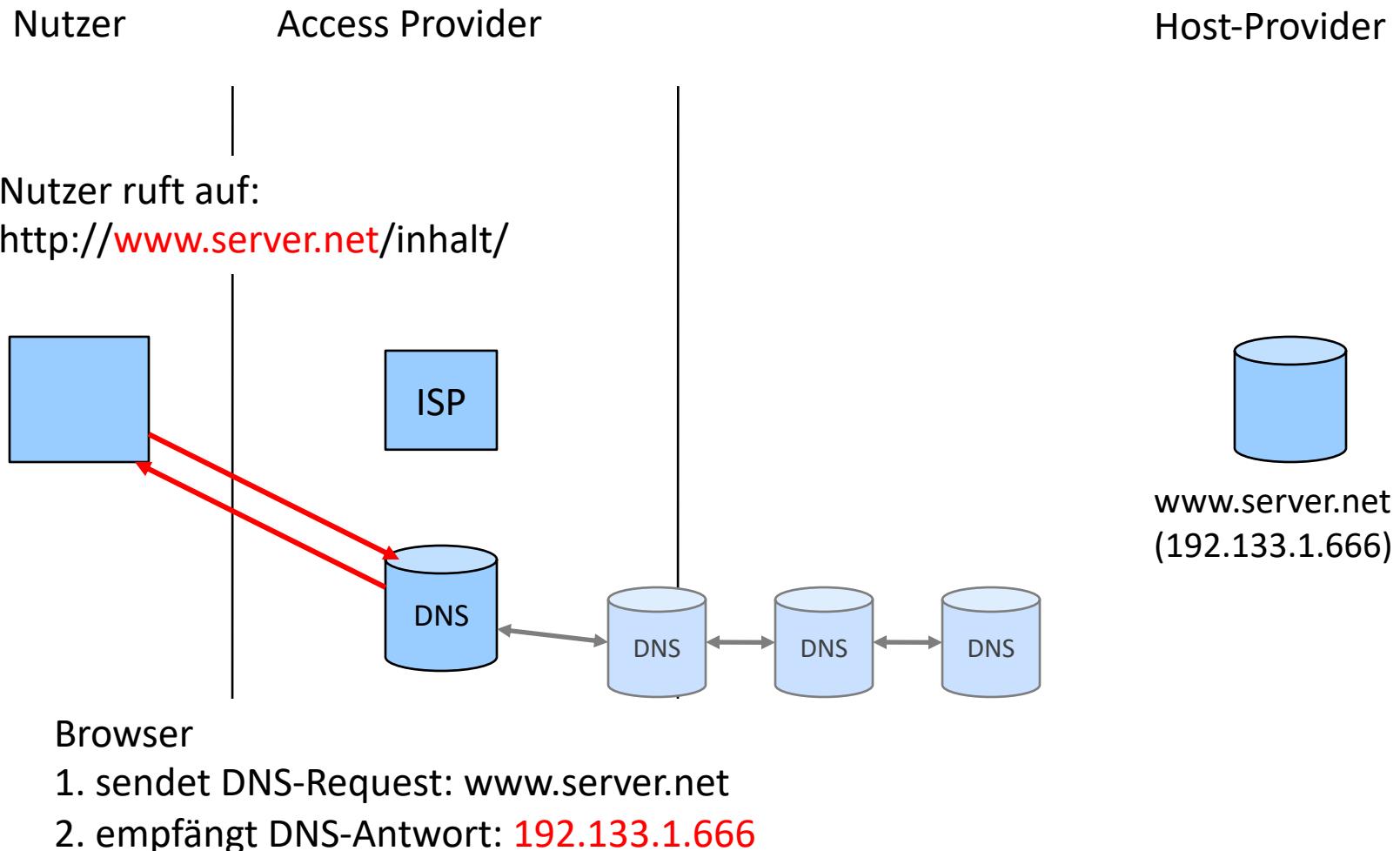
Sicherheit im Domain Name System (DNS)

- DNS: Domain Name System
 - Abbildung des Rechnernamens auf IP-Adresse
 - Anfrage an Nameserver
 - typischerweise in WANs

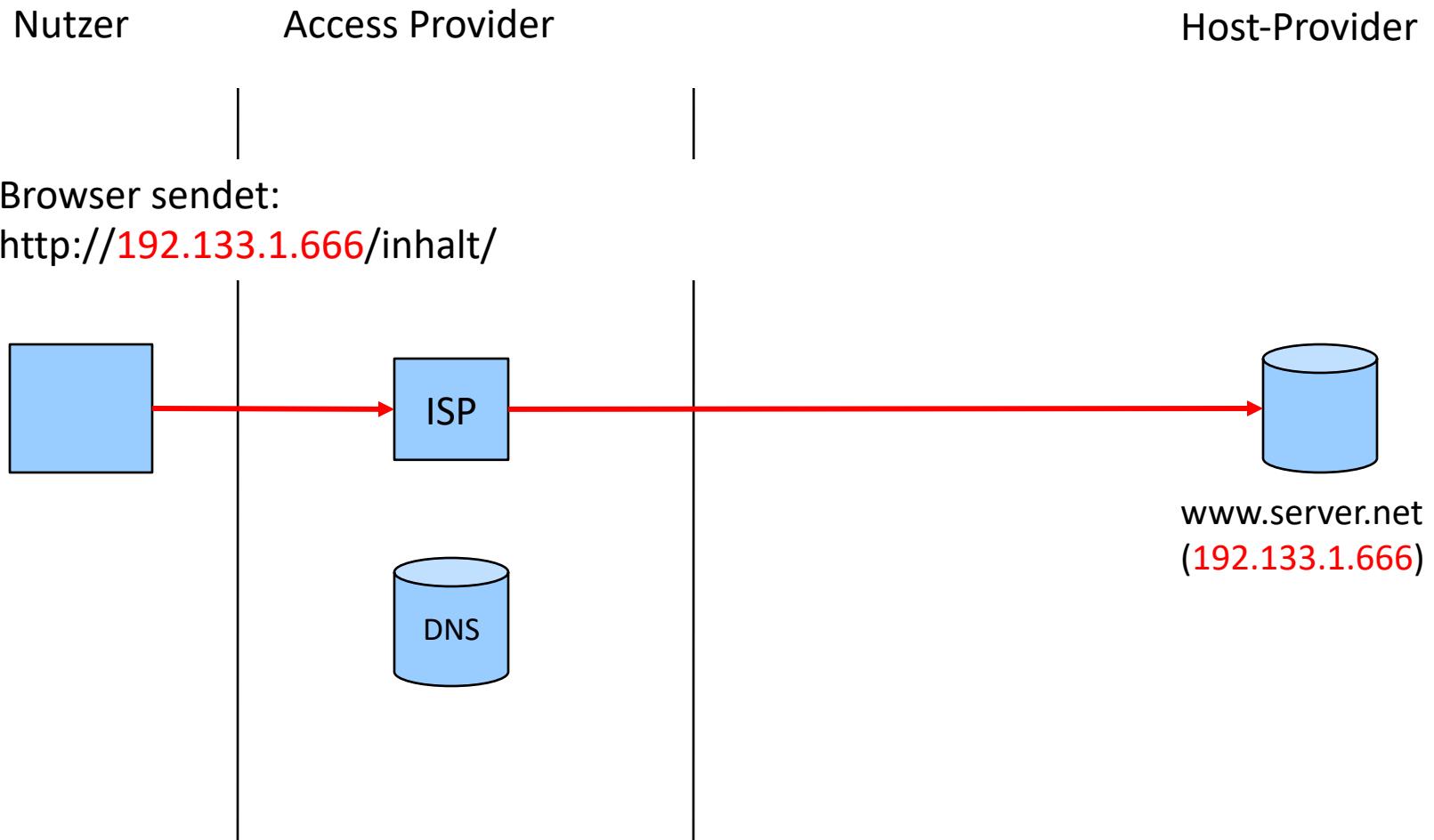
- Angriffe auf DNS
 - Sniffing von DNS-Anfragen
 - Fälschen der DNS-Antworten
 - Denial-of-Service



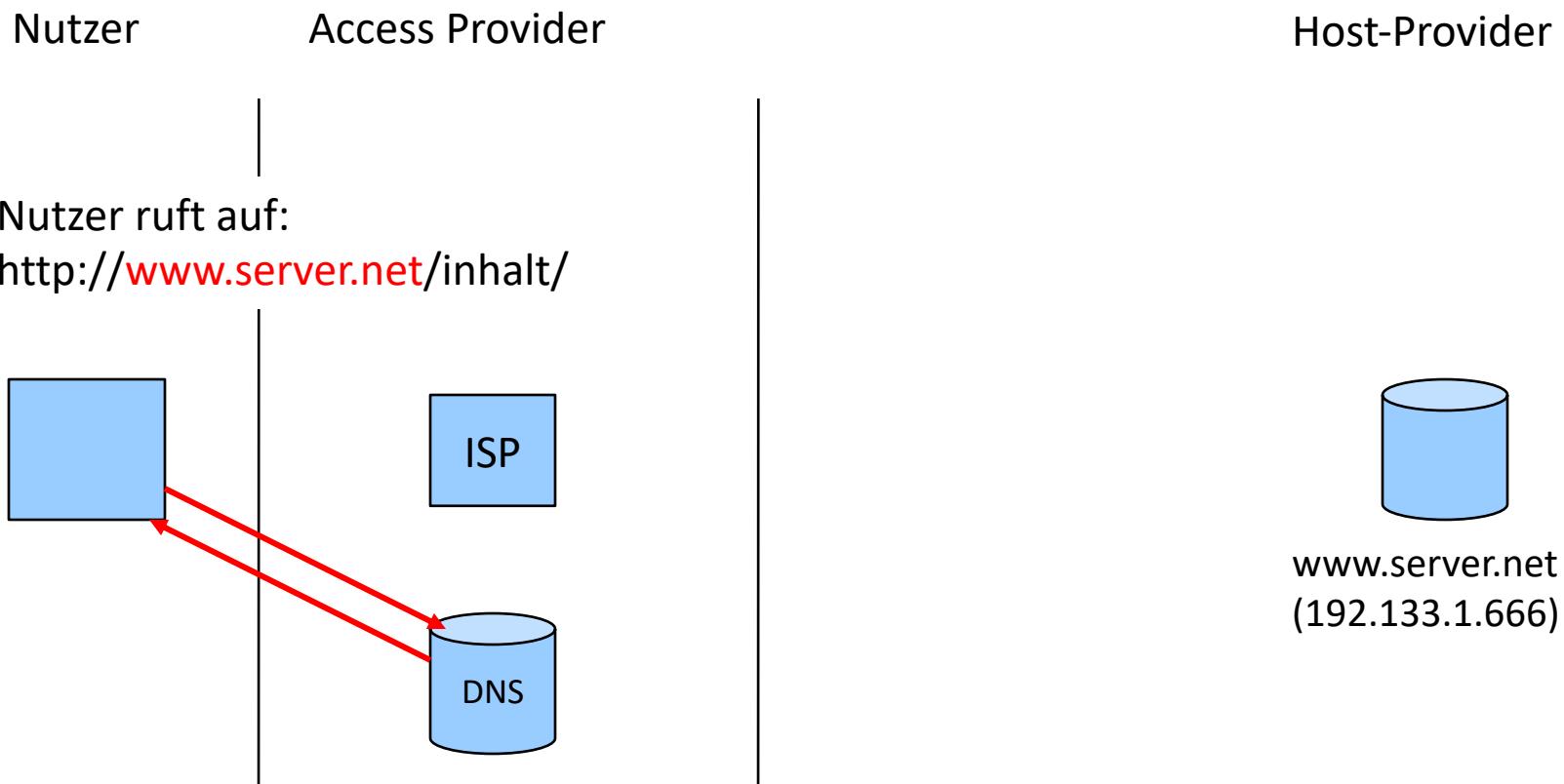
Zunächst wird DNS-Server angefragt



Anschließend wird Inhalt abgerufen



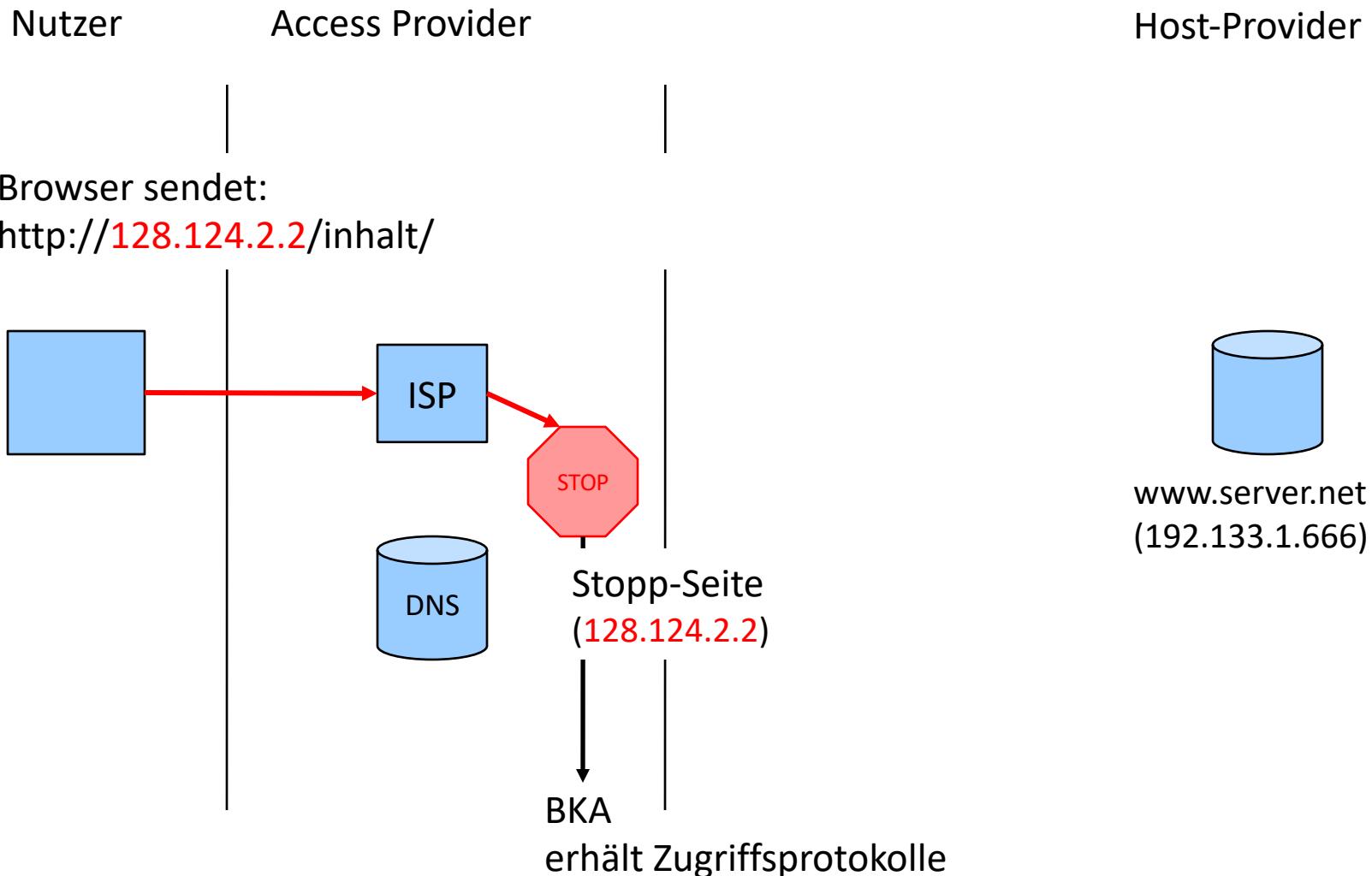
DNS-Sperre: DNS-Server sendet »falsche« Antwort



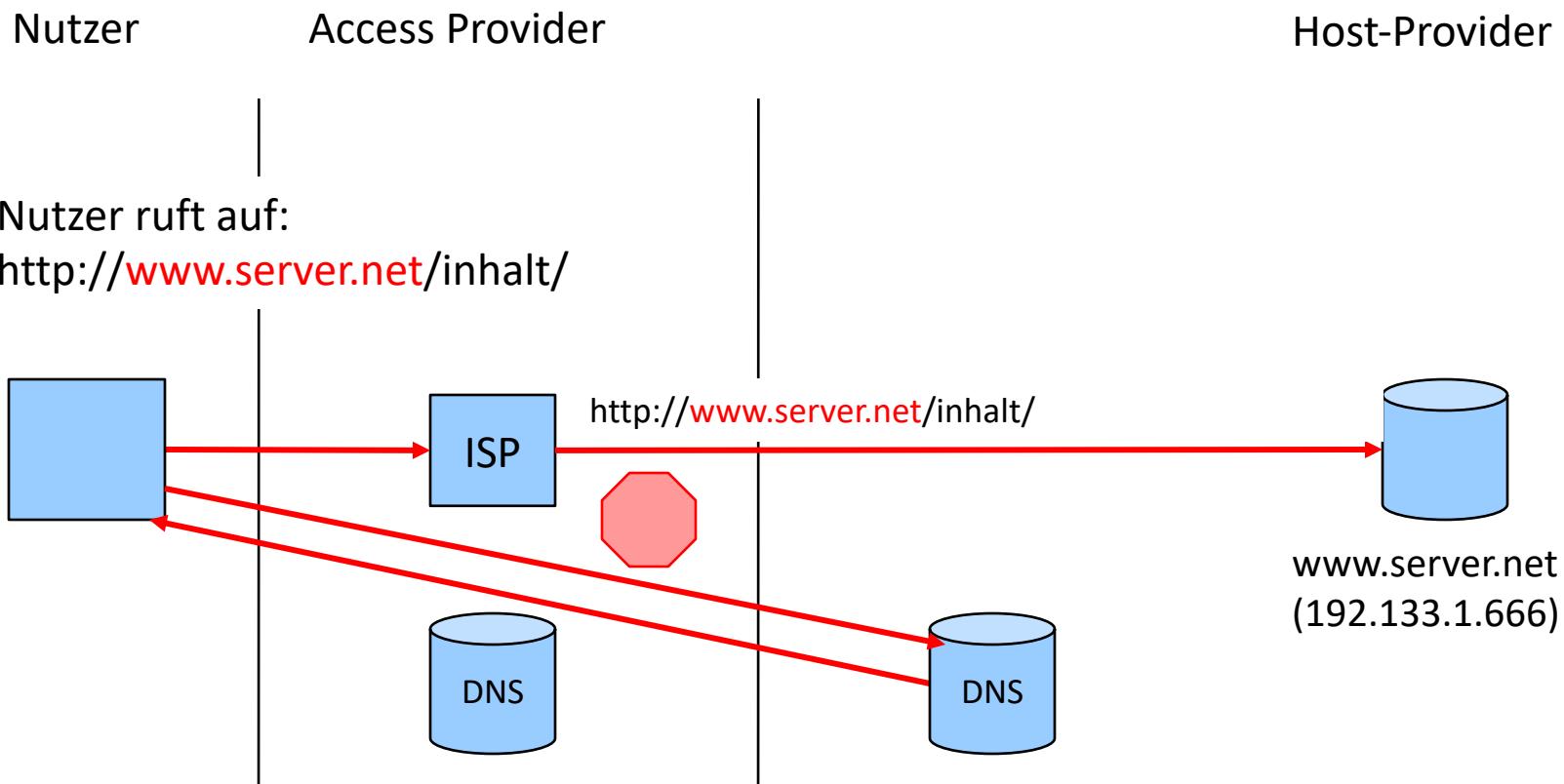
Browser

1. sendet DNS-Request: **www.server.net**
2. DNS-Server sieht Sperrliste durch (Treffer!)
3. empfängt DNS-Antwort: **128.124.2.2**

Mit DNS-Sperre landet der Nutzer im WWW auf Stopp-Seite



Mit DNS-Sperre und Open DNS



Open DNS

OpenDNS > Use OpenDNS

https://www.opendns.com/start/

open dns

OpenDNS.com Dashboard Community Sign In or Create account Your IP: 92.116.160.129

OpenDNS

HOME SOLUTIONS USE OPENDNS CUSTOMERS SUPPORT ABOUT US BLOG

Use OpenDNS (Step 1 of 3: Change DNS settings)

It only takes 2 minutes. Change DNS on your:

Computer OR Router OR DNS Server

Best for home users

Get instructions for Windows, Mac, mobile phones, and more.

Enable OpenDNS on your router so every computer benefits.

Learn how to use OpenDNS with your existing DNS servers.

1 Change your DNS settings

2 Create a free OpenDNS account (optional)

3 Manage settings in your Dashboard (optional)

Video Tutorial
Take a few minutes to watch our step-by-step [video](#) on getting started with OpenDNS.

Find out how OpenDNS complements your existing network setup
Read our IT Administrator [Best Practices](#).

The straight dope
Our nameservers are **208.67.222.222** and **208.67.220.220**.

Solutions
[For Home Network](#)
[For K-12 School](#)
[For Small/Medium Business](#)
[For Enterprise](#)

Use OpenDNS
[On your computer](#)
[On your router](#)
[On your DNS server](#)
[Best Practices](#)
[Create a free account](#)

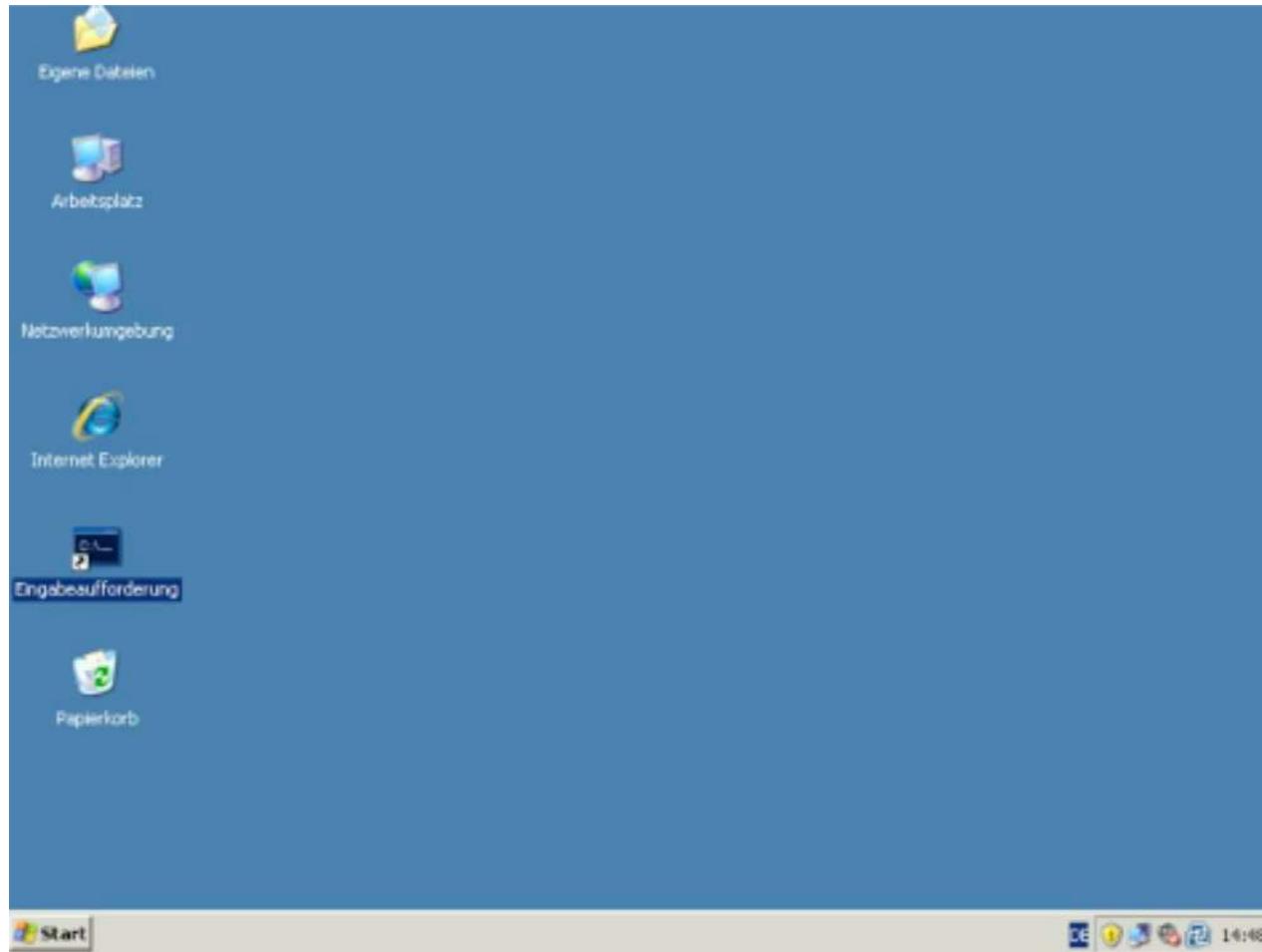
Support
[Knowledge Base](#)
[Forums](#)
[System Status](#)
[CacheCheck](#)
[Contact](#)

About Us
[Overview](#)
[Management](#)
[Press Center](#)
[Awards](#)
[Careers](#)

OpenDNS

208.67.222.222
208.67.220.220

DNS-Sperre und Windows (27 Sekunden)



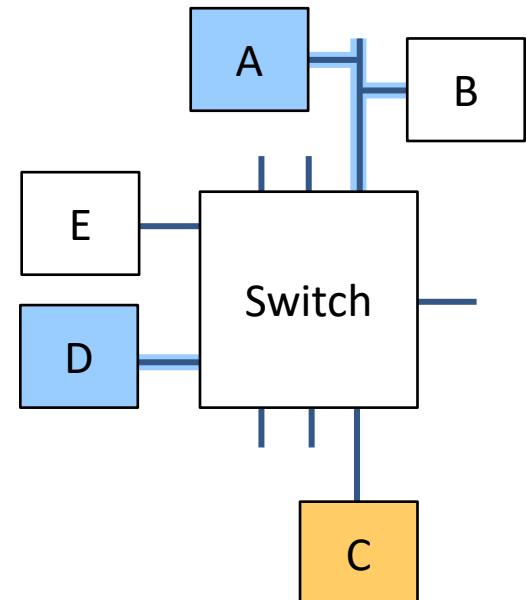
Quelle: <http://www.youtube.com/watch?v=1NNG5I6DBm0>

Spoofing-Angriffe: Funktionsweise (Ethernet)

- Switch verteilt Daten nur auf Netzabschnitt des Empfängers

Rechner **A** und **D** kommunizieren miteinander

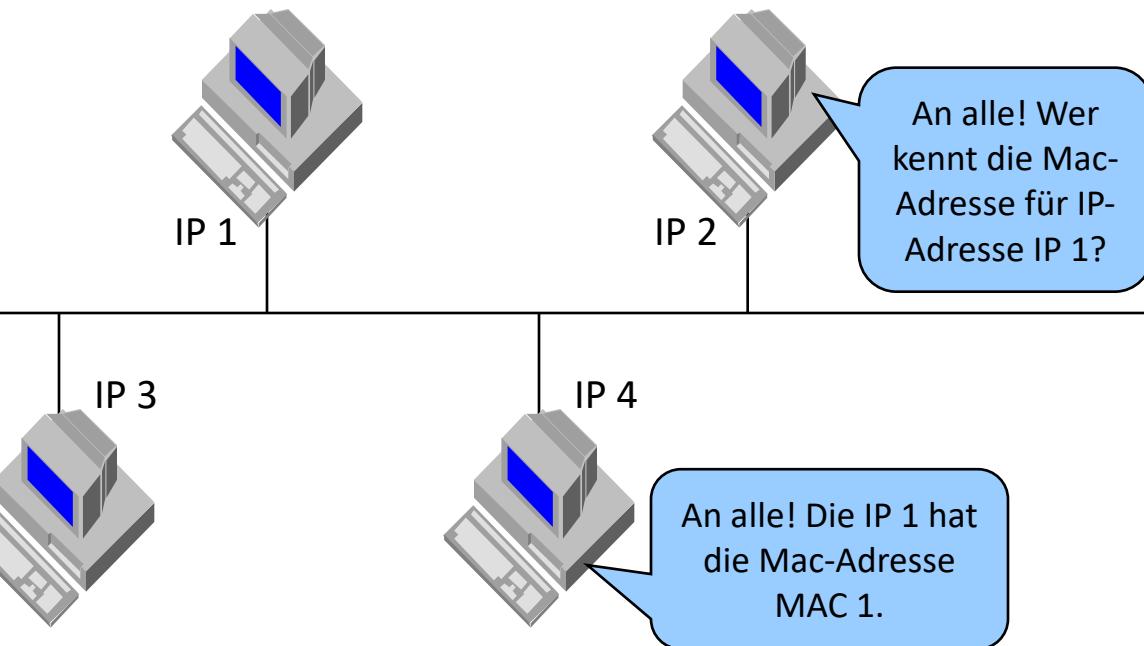
b) im Switched Ethernet



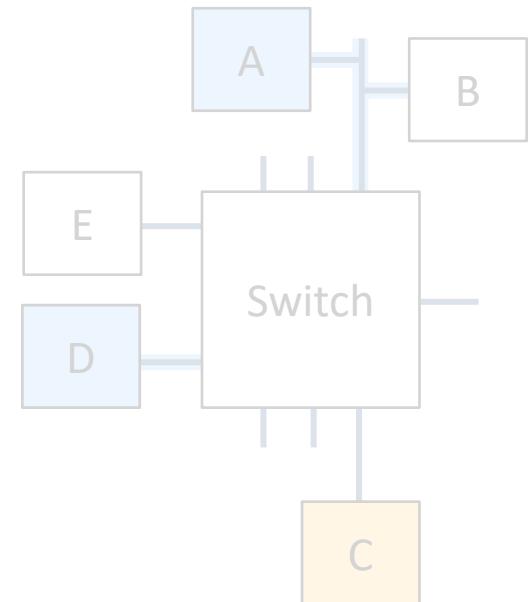
— Ausbreitung der übertragenen Daten

ARP: Address Resolution Protocol

- ARP-Anfrage
 - Anfrage wird an das gesamte lokale Netz gestellt (Broadcast)
 - Mitteilen der eigenen Adresse(n) in der Anfrage
- ARP-Antwort
 - Jeder Rechner kann antworten

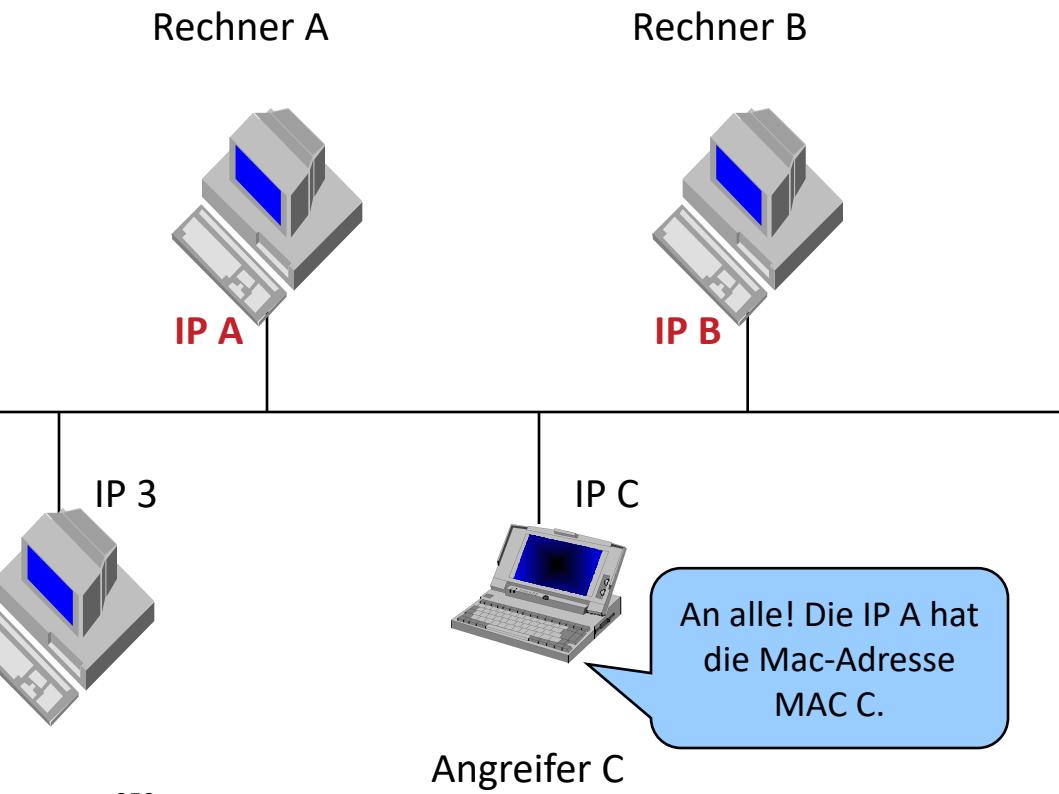


b) im Switched Ethernet

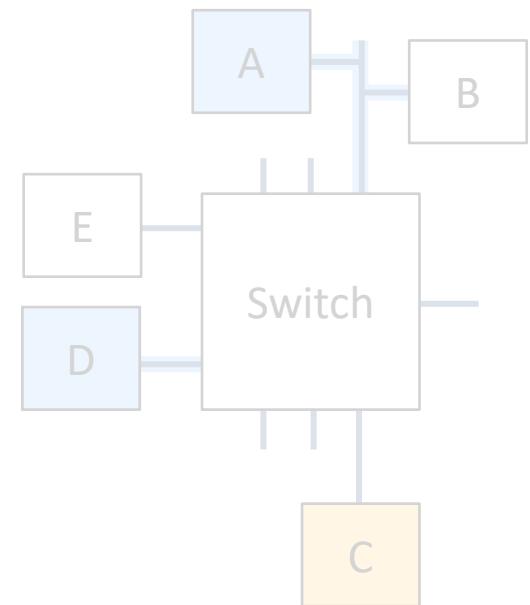


ARP Spoofing

- Angreifer C sendet gefälschte ARP-Response



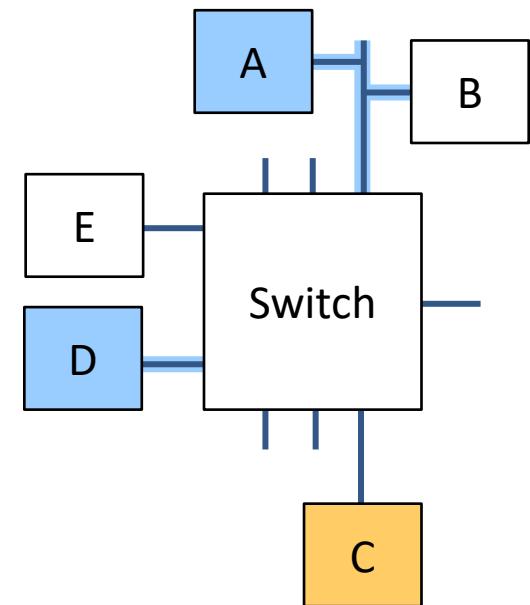
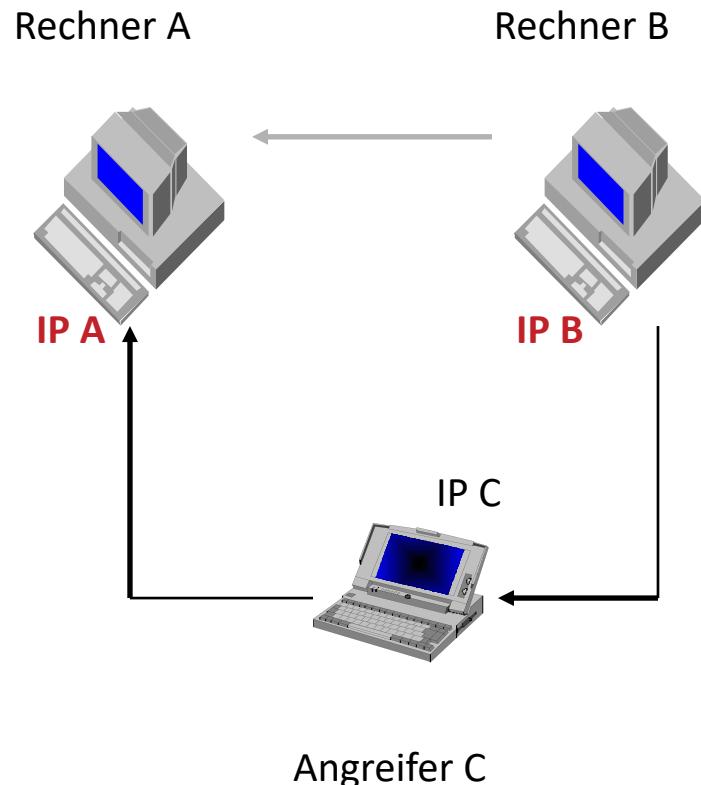
b) im Switched Ethernet



ARP Spoofing

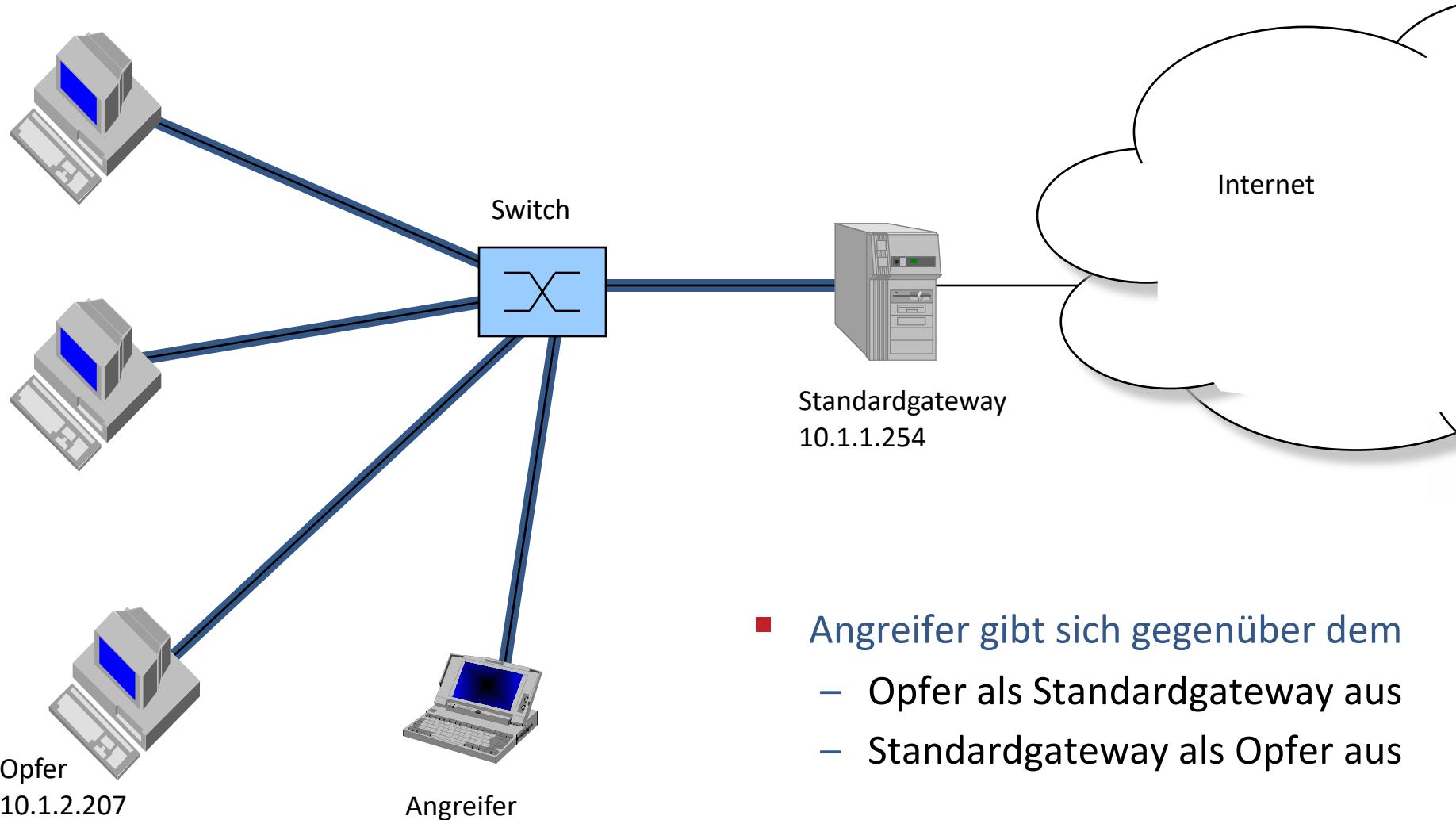
- B adressiert an IP A
- Ethernetkarte von Rechner B schickt die Daten jedoch an MAC C

b) im Switched Ethernet



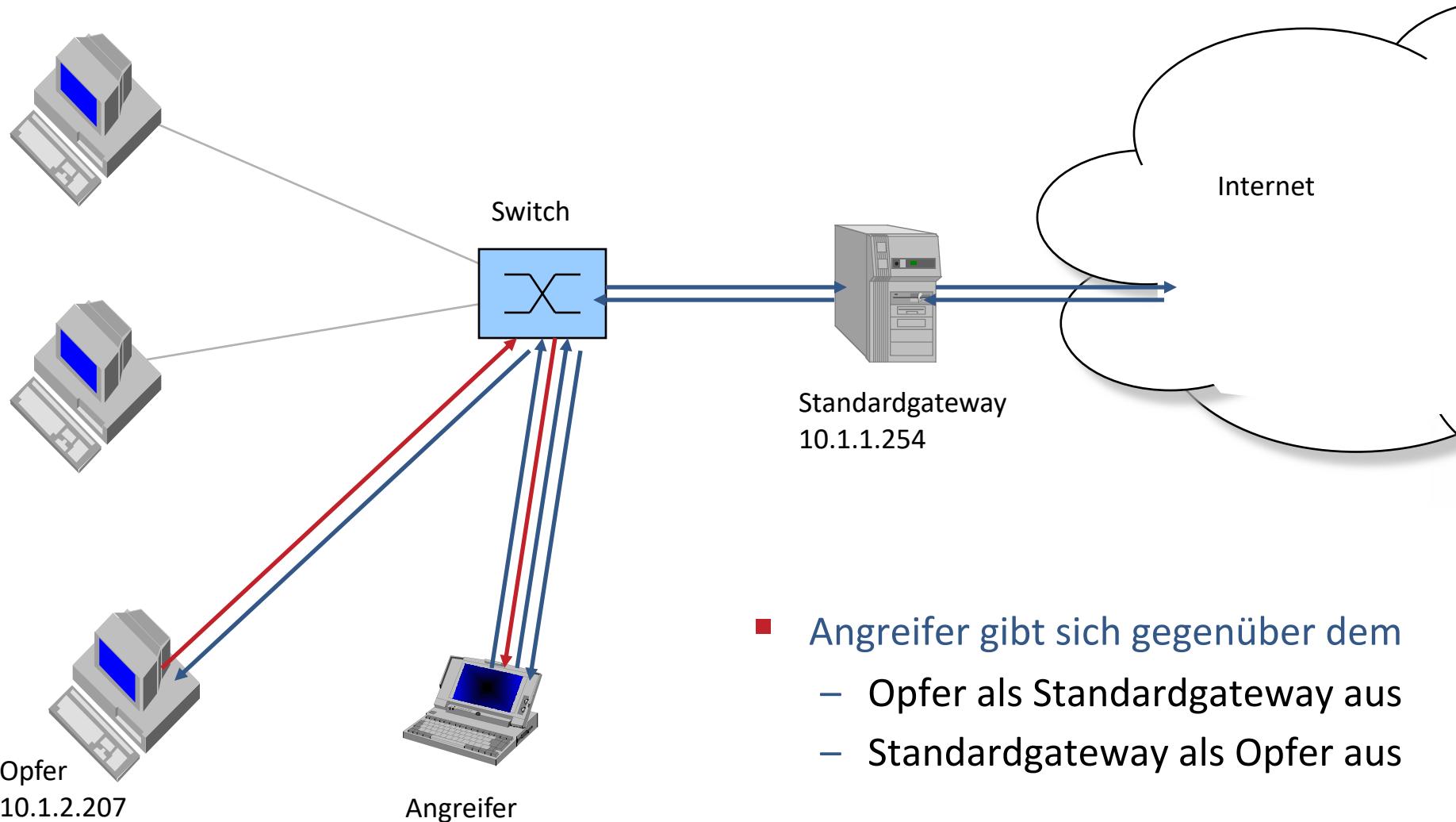
ARP-Spoofing-Demonstration: Vorbereitung

Weitere Rechner im Subnetz
10.1.0.0/255.255.252.0



ARP-Spoofing-Demonstration: Opfer sendet IP-Paket ins Internet

Weitere Rechner im Subnetz
10.1.0.0/255.255.252.0



- Angreifer gibt sich gegenüber dem
 - Opfer als Standardgateway aus
 - Standardgateway als Opfer aus

ARP-Spoofing

- Angreifer
 - empfängt den gesamten Netzwerkverkehr
 - vom Opfer zum Internet
 - vom Internet zum Opfer
 - kann diese Datenpakete beliebig manipulieren

- Demonstration:
 - Windows Tool »Cain & Abel«
 - <http://www.oxid.it/cain.html>
 - ARP-Spoofing:
 - Opfer: 10.1.2.207
 - Standardgateway: 10.1.1.254
 - DNS-Spoofing:
 - Umleitung von www.bsi.de nach jap.inf.tu-dresden.de

Rechner im Netzwerk identifizieren

The screenshot shows the Cain & Abel software interface. The title bar reads "Cain". The menu bar includes "File", "View", "Configure", "Tools", and "Help". The toolbar contains icons for Protected Storage, Network, Sniffer, LSA Secrets, Cracker, Traceroute, CCDU, and Wireless. The main window displays a table with columns: IP address, MAC address, OUI fingerprint, Host name, and several status indicators (B31, B16, B8, Gr, M0, M1, M3). The table is currently empty. A cursor arrow is visible in the center of the table area. The bottom navigation bar includes icons for Hosts, APR, Routing, Passwords, and VoIP.

IP address	MAC address	OUI fingerprint	Host name	B31	B16	B8	Gr	M0	M1	M3

Hosts APR Routing Passwords VoIP

Auswahl der Rechner für das ARP-Spoofing

Screenshot of Cain & Abel Network Sniffer tool interface.

The interface includes:

- Toolbar with icons for Protected Storage, Network, Sniffer, LSA Secrets, Cracker, Traceroute, CCDU, and Wireless.
- Menu bar: File, View, Configure, Tools, Help.
- Icon bar below the menu.
- Table view showing network hosts:

IP address	MAC address	oui fingerprint	Host name	B31	B16	B8	Gr	M0	M1	M3
10.1.1.1	0040CA190C2C	FIRST INTERNAT'L COMPUTE...								
10.1.1.2	0040CA18AAA6	FIRST INTERNAT'L COMPUTE...								
10.1.1.254	00E0B0E2906B	CISCO SYSTEMS, INC.								
10.1.2.206	000A95E8B4FC	Apple Computer, Inc.								
10.1.2.207	00D059D78A2F	AMBIT MICROSYSTEMS CORP.								
10.1.2.241	000D56C93866	Dell PCBA Test								
10.1.2.248	0001E697FD91	Hewlett-Packard Company								

- Bottom navigation bar with icons for Hosts, APR, Routing, Passwords, and VoIP.

Einrichten des DNS-Spoofing

The screenshot shows the Cain & Abel network security tool interface. The title bar reads "cain". The menu bar includes File, View, Configure, Tools, and Help. The toolbar contains various icons for file operations, network analysis, and cracking. The main window has tabs for Protected Storage, Network, Sniffer, LSA Secrets, Cracker, Traceroute, CCDU, and Wireless. On the left, a sidebar titled "APR" lists connections: APR-DNS, APR-SSH-1 (0), APR-HTTPS (0), and APR-RDP (0). The central area displays two tables of network interface information. The top table shows one entry: "Idle" with IP address 10.1.2.207, MAC address 00D059D78A2F, and MAC address 00E0B0E2906B with IP address 10.1.1.254. The bottom table is currently empty. At the bottom, a button labeled "Configuration / Routed Packets" is highlighted with a mouse cursor.

Status	IP address	MAC address	Packets ->	<- Packets	MAC address	IP address
Idle	10.1.2.207	00D059D78A2F			00E0B0E2906B	10.1.1.254

Status	IP address	MAC address	Packets ->	<- Packets	MAC address	IP address

Configuration / Routed Packets

Hosts APR Routing Passwords VoIP

Start des ARP- und DNS-Spoofings

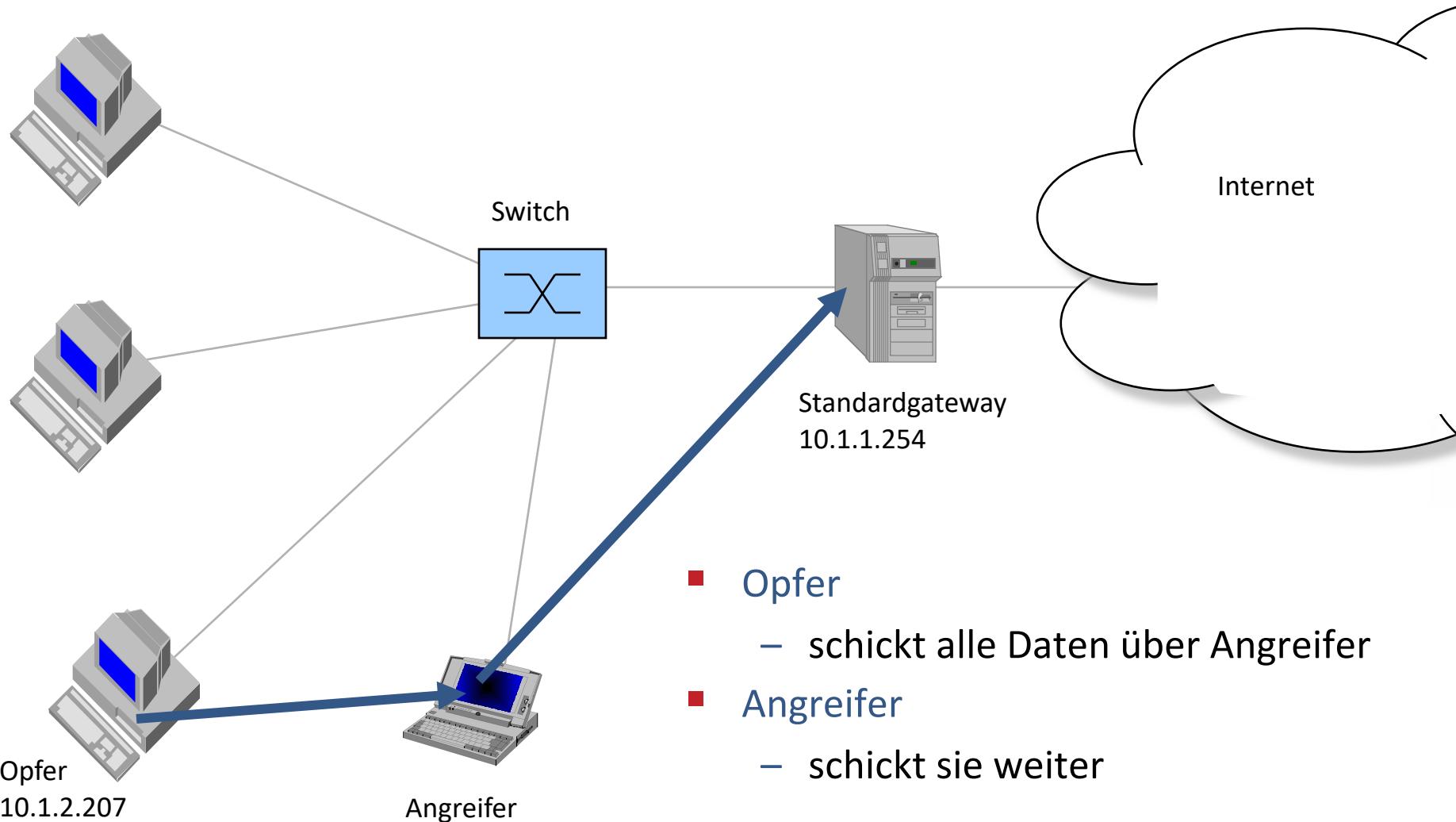
The screenshot shows the Cain & Abel network security tool interface. The title bar reads "Cain". The menu bar includes "File", "View", "Configure", "Tools", and "Help". The toolbar contains various icons for file operations, network analysis, and cracking. The main window has tabs for "Protected Storage", "Network", "Sniffer", "LSA Secrets", "Cracker", "Traceroute", "CCDU", and "Wireless". On the left, a sidebar titled "APR" lists "APR-DNS", "APR-SSH-1 (0)", "APR-HTTPS (0)", and "APR-RDP (0)". The central pane displays a table for "Spoofed DNS Requests". The table has columns: "Requested DNS name", "Spoofing IP", and "#Resp. Spoofed". A single entry is shown: "www.bsi.de" with "141.76.46.90" as the "Spoofing IP" and "0" as the "#Resp. Spoofed" count. A cursor arrow is visible in the center of the table area. At the bottom, a status bar shows "APR-DNS". The bottom navigation bar includes icons for "Hosts", "APR", "Routing", "Passwords", and "VoIP".

Requested DNS name	Spoofing IP	#Resp. Spoofed
<input checked="" type="checkbox"/> www.bsi.de	141.76.46.90	0

APR-DNS

Erreichte Situation

Weitere Rechner im Subnetz
10.1.0.0/255.255.252.0



Sicht des Opfers

The screenshot shows a Mozilla Firefox window displaying the "Lehrveranstaltungsangebote" (Teaching Offerings) page for the Chair of Management of Information Security at the University of Regensburg.

Page Headers:

- Firefox title bar: Lehrveranstaltungsangebote - Mozilla Firefox
- Menu bar: Datei, Bearbeiten, Ansicht, Gehe, Lesezeichen, Extras, Hilfe
- Toolbar: Back, Forward, Stop, Home, Refresh, Address bar: http://www-sec.uni-regensburg.de/teaching/
- Address bar: http://www-sec.uni-regensburg.de/teaching/
- Search bar: Go, G

Page Content:

The page features the University of Regensburg logo and navigation links for Contact, Teaching, Research, Publications, and Staff.

IT-Sicherheitsmanagement

Lehrstuhl Management der Informationssicherheit

Universität Regensburg > Wirtschaftswissenschaften > Wirtschaftsinformatik

Lehrveranstaltungsangebote

Lehrveranstaltungsangebote des Lehrstuhls Vorlesungsfolien in der VUR Themen für Diplomarbeiten Schwerpunkt Informationssicherheit Modellstudienplan Informationssicherheit

Wintersemester	SWS	Art
VL Informatik III (Algorithmen und Datenstrukturen)	2/2	Grundstudium
VL Allgemeine Wirtschaftsinformatik (Datenkommunikation)	2/1	Hauptstudium
Seminar IT-Sicherheit	2	Hauptstudium
Diplanden- und Doktorandenseminar	2	Hauptstudium
VL Sicherheitsmanagement	2/1	Schwerpunkt Informationssicherheit
VL Sicherheit mobiler Systeme	2/-	Schwerpunkt Informationssicherheit
VL Praxis der IT-Sicherheit (bedarfswise)	1/3	Schwerpunkt Informationssicherheit

Sommersemester

VL Informatik IV (Objektorientierte Programmierung)	2/1	Grundstudium
---	-----	--------------

Sicht des Angreifers

File View Configure Tools Help

Protected Storage Network Sniffer LSA Secrets Cracker Traceroute CCDU Wireless

Status	IP address	MAC address	Packets ->	<- Packets	MAC address	IP address
Poisoning	10.1.2.207	00D059D78A2F			00E0B0E2906B	10.1.1.254

Status	IP address	MAC address	Packets ->	<- Packets	MAC address	IP address

Configuration / Routed Packets

Hosts APR Routing Passwords VoIP

Schutz vor ARP-Spoofing

- Arpwatch
 - verfolgt Änderungen der Zuordnung von Ethernetadressen und IP-Adressen
 - Erstmaliges Erscheinen einer neuen Ethernetadresse
 - Wechseln der Zuordnung von der »üblichen« auf eine neue Zuordnung (Ethernetadresse–IP-Adresse)
 - Alarmiert Systemadministrator bei Auffälligkeiten per E-Mail
 - Manpage
 - http://linuxcommand.org/man_pages/arpwatch8.html
 - Package
 - <http://packages.debian.org/unstable/admin/arpwatch.html>

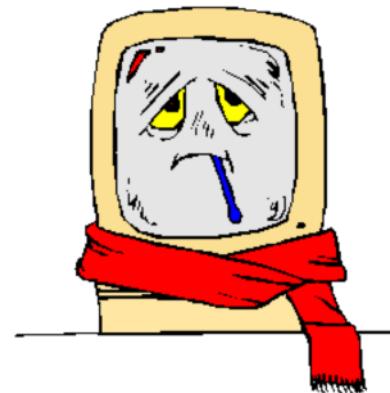
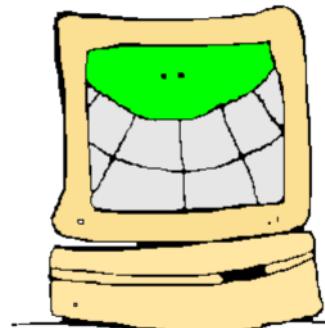
Schutz vor DNS-Spoofing

- DNSSEC (DNS Security Extensions)
 - vorgeschlagen im März 2005 als RFC 4034 (und weitere)
 - <http://www.dnssec.net/>
 - Kernidee:
 - Nutzung digitaler Signaturen zur Authentifizierung der DNS-Antwort
 - Schutzziele
 - Schutz der Integrität und Zurechenbarkeit
 - Kein Schutz der Vertraulichkeit und Verfügbarkeit

Denial-of-Service Angriffe

Verfügbarkeit: Denial-of-Service

- DoS-Angriffe auf Schwachstellen im Systemdesign (insb. Protokolle)
 - Mail-Bombing – Spamming
 - Broadcast-Storm
 - SYN-Flooding
- DoS-Angriffe auf Implementationsfehler
 - Ping of Death
 - WinNuke
 - Teardrop und Nachfahren

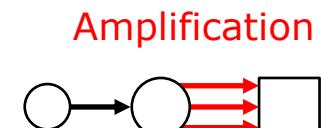


Distributed Denial-of-Service Angriffe im Internet

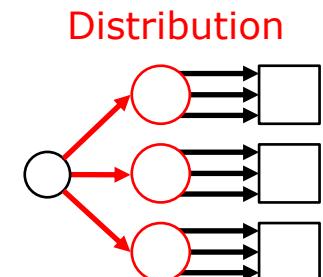
- Charakterisierung
 - Ziel wird von mehreren Quellen gleichzeitig angegriffen



- Typische Angriffsmuster
 - Reflexion, Spoofing
 - Amplification
 - Distribution (Botnets)



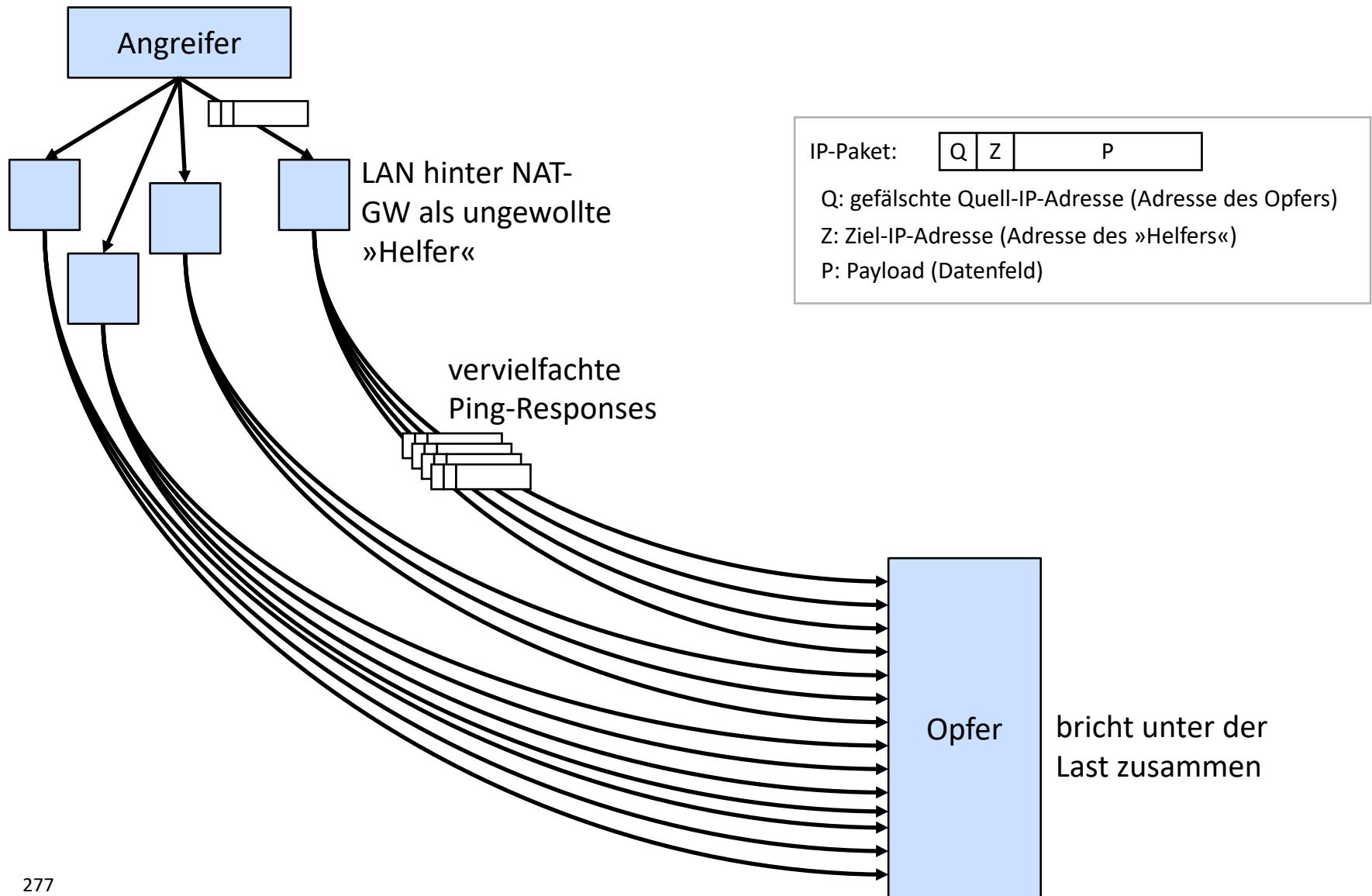
- Beispiele
 - Smurf IP Denial-of-Service Attack von 1998
 - Mirai-Botnet (2016)
 - Memcached-Angriff von 2017



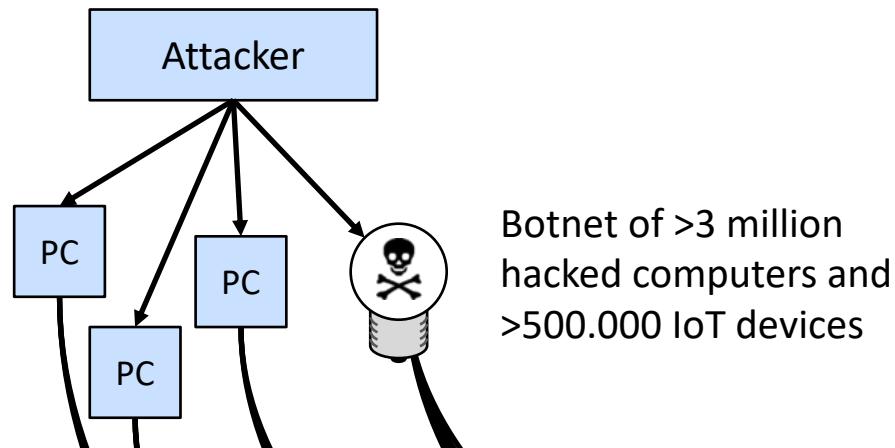
Smurf IP Denial-of-Service Attack (CERT Advisory CA-1998-01)

- DDos-Angriff basierend auf Flooding mit Ping-Paketen
 - Ping: Management-Service zur Überprüfung der Empfangsbereitschaft eines Rechners
 - Smurf IP DDos ist Beispiel für IP-Spoofing und Amplification
- Vorgehen
 - Angreifer schickt Ping-Pakete mit gefälschter Absender-Adresse an schlecht administriertes LAN/Intranet
 - Konfigurationsfehler im LAN vervielfacht Ping
 - Weiterleitung an alle Rechner des LAN hinter dem Gateway
 - Jeder Rechner des LAN antwortet mit Pong

Smurf IP Denial-of-Service Attack (1998)



Mirai Botnet (2016)

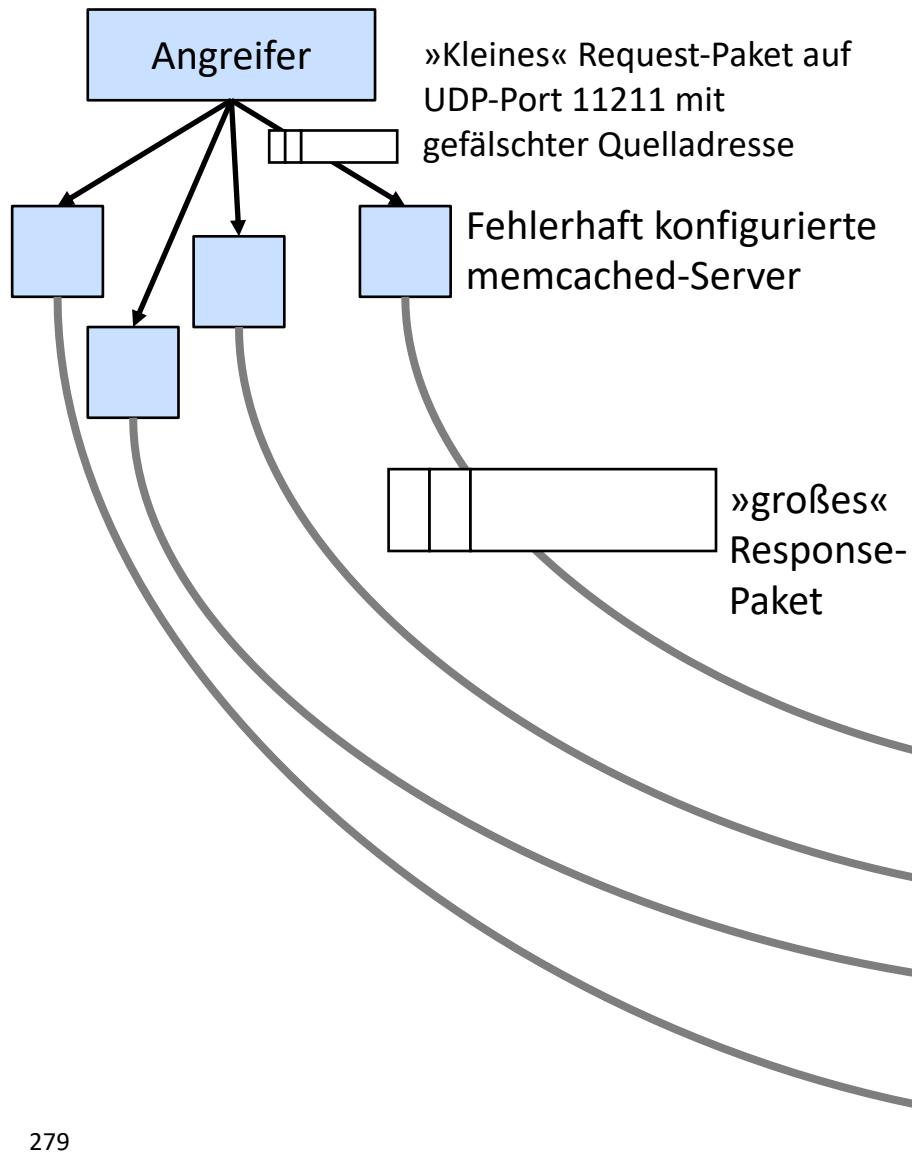


<https://www.youtube.com/watch?v=cQkQpbkhzIo>

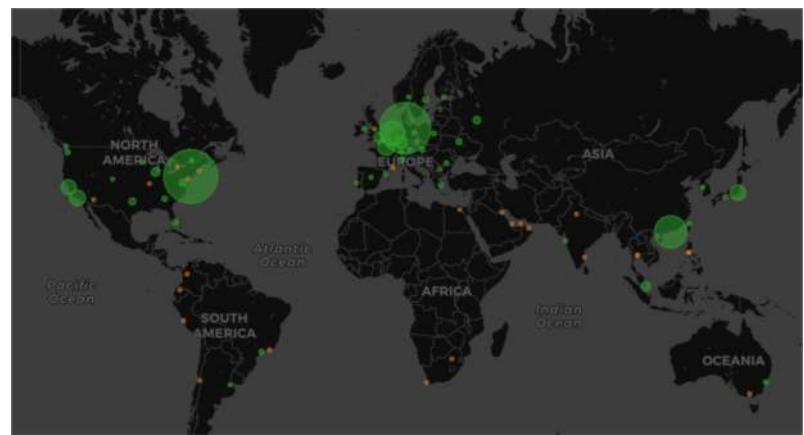


overloaded

Memcached-Angriff (2017)

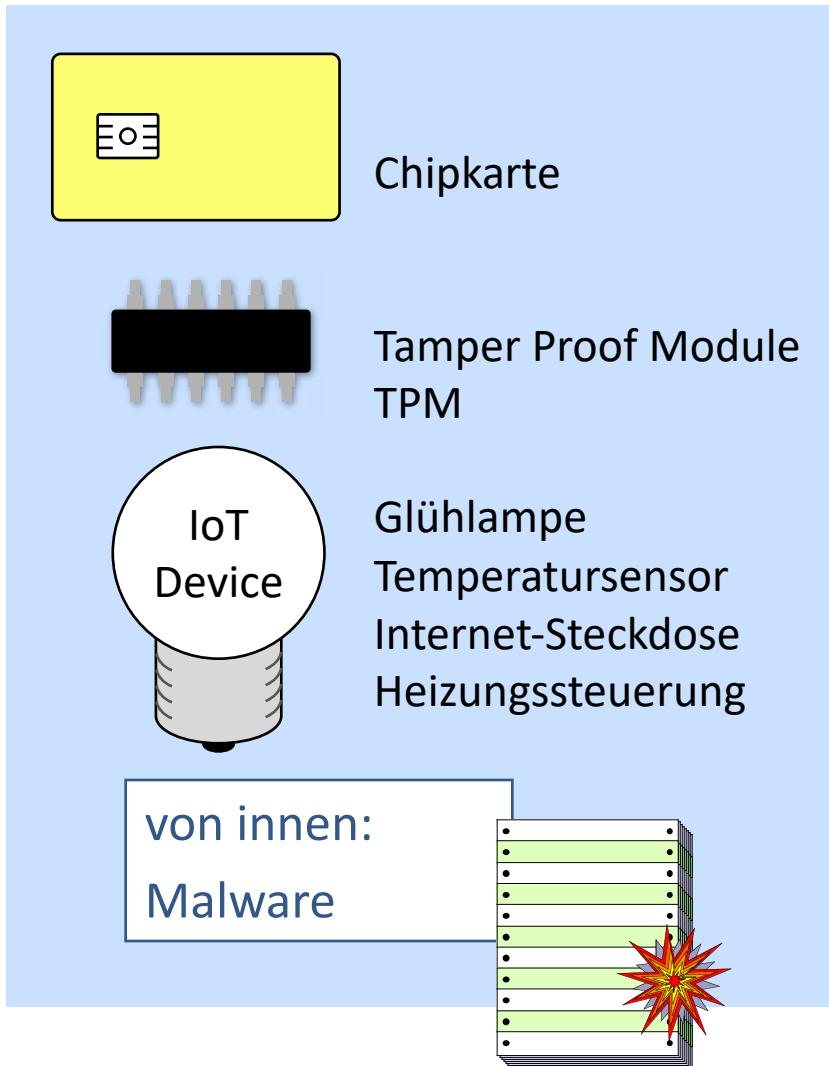


Herkunft der fehlerhaft konfigurierten Memcached-Server



<https://blog.cloudflare.com/memcrashed-major-amplification-attacks-from-port-11211/>

Internet of Things Security

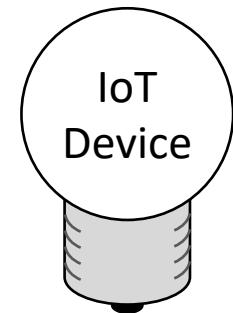


Angreifer kann alle drei Schutzziele verletzen:

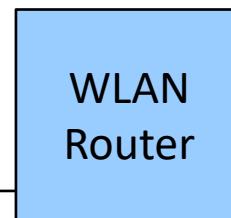
- Vertraulichkeit
- Integrität
- Verfügbarkeit

Internet of Things – im lokalen Netz

im lokalen Netz
erreichbar unter
192.168.2.10



Glühlampe
Temperatursensor
Internet-Steckdose
Heizungssteuerung



Einfach im Browser nutzbar:
<http://192.168.2.10/on>
<http://192.168.2.10/off>

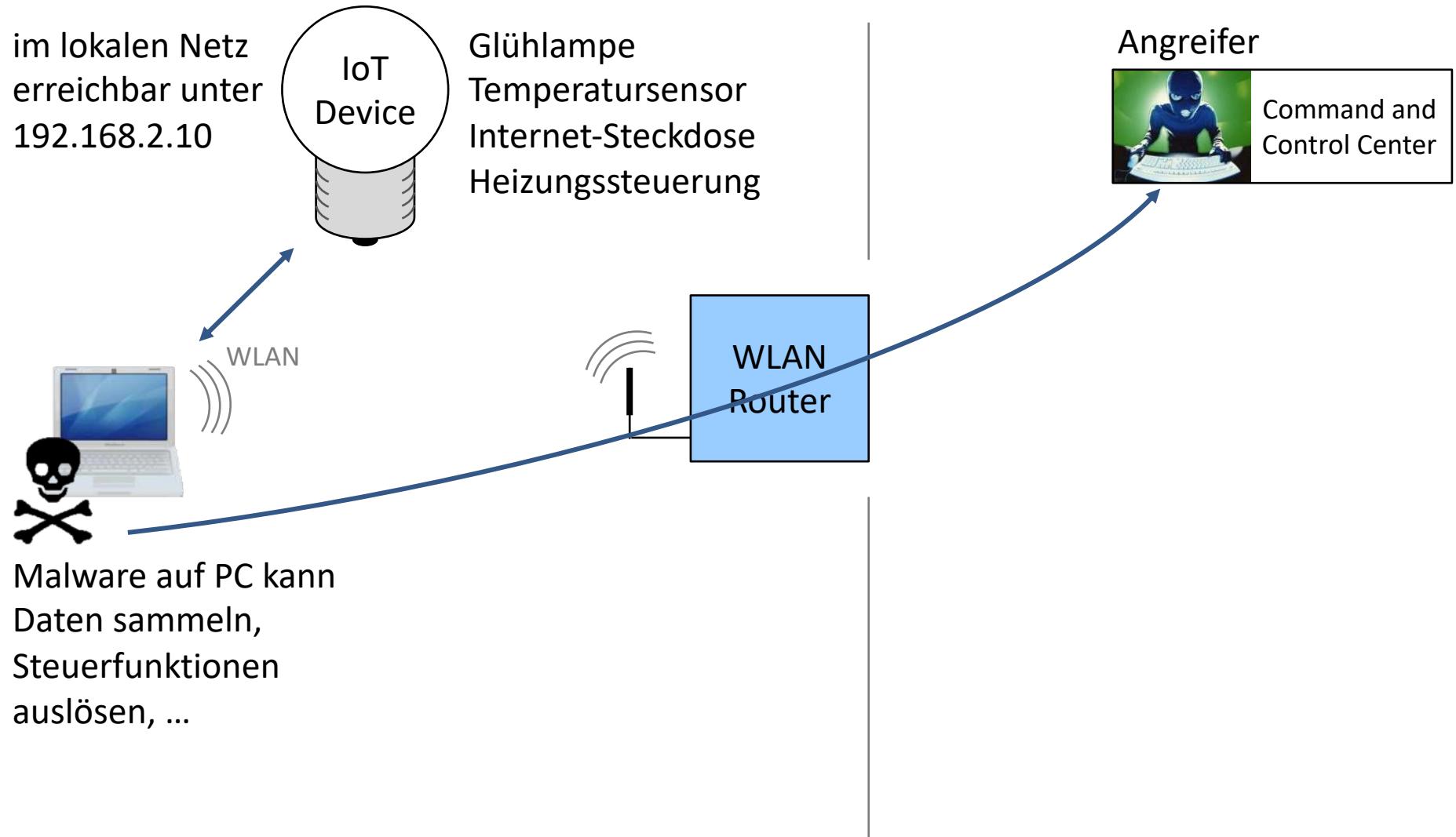


IoT App
Smartphone

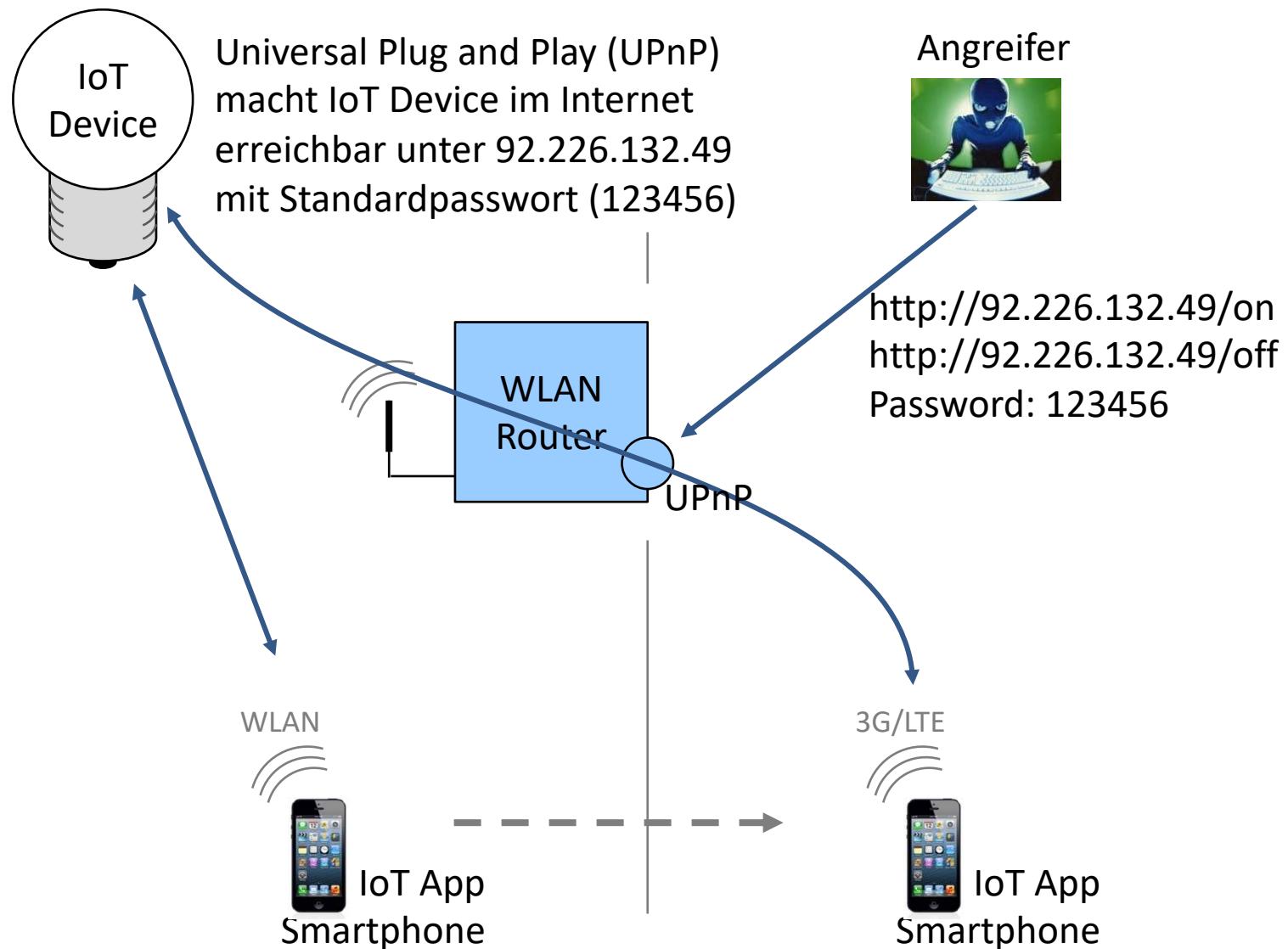


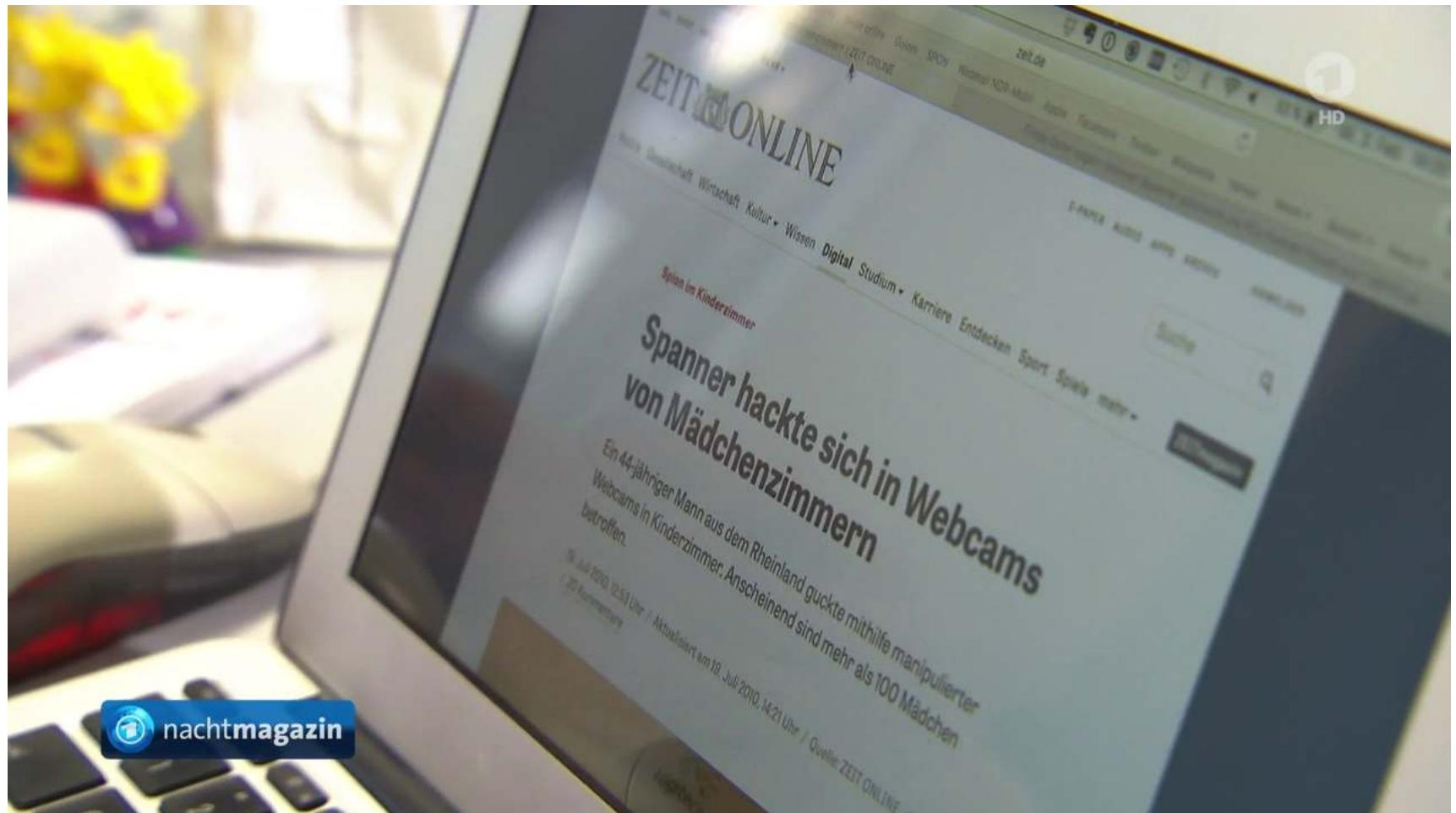
Internet

Internet of Things – im lokalen Netz angreifbar

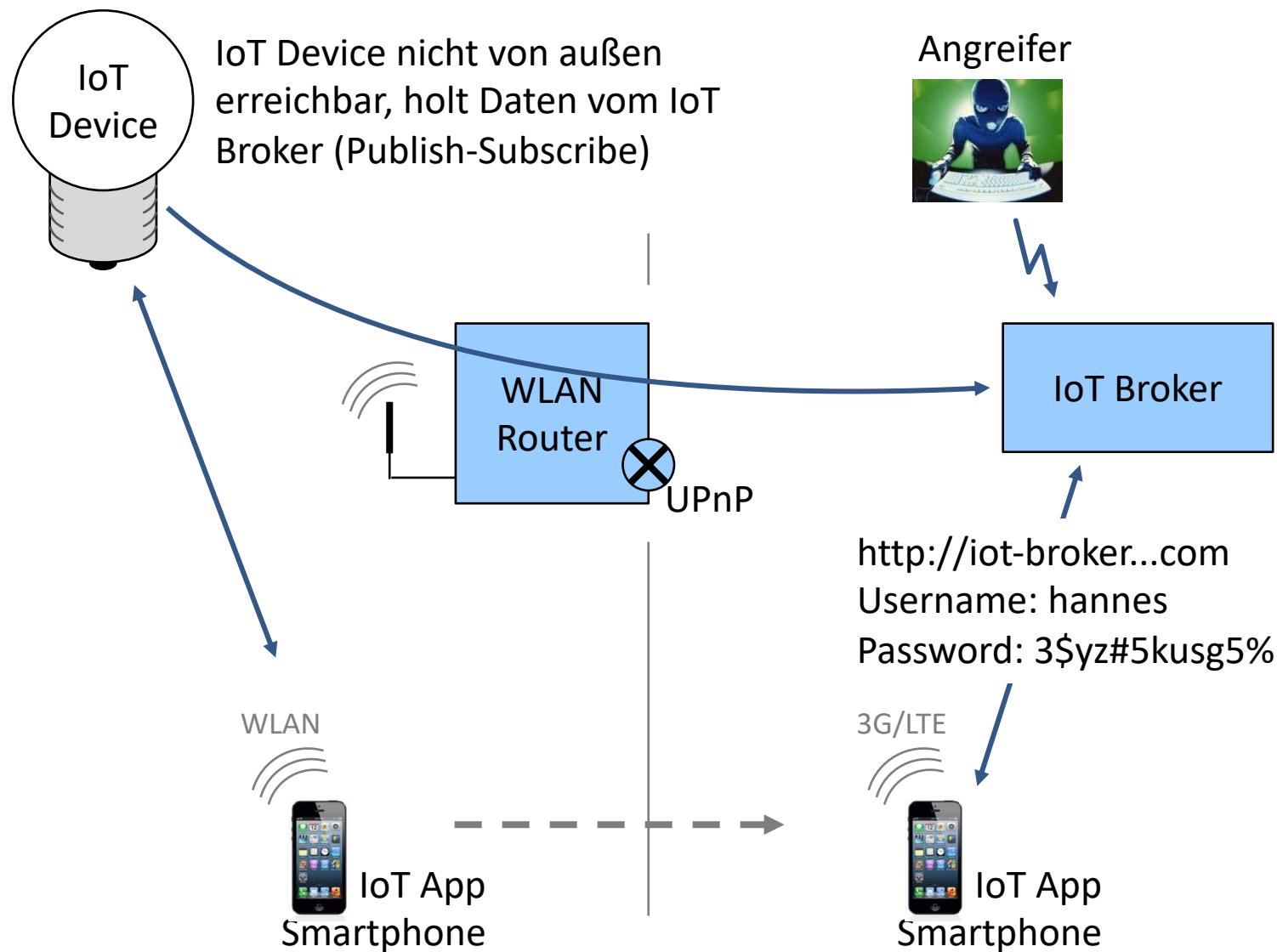


Internet of Things – Angriff über Universal Plug and Play (UPnP)

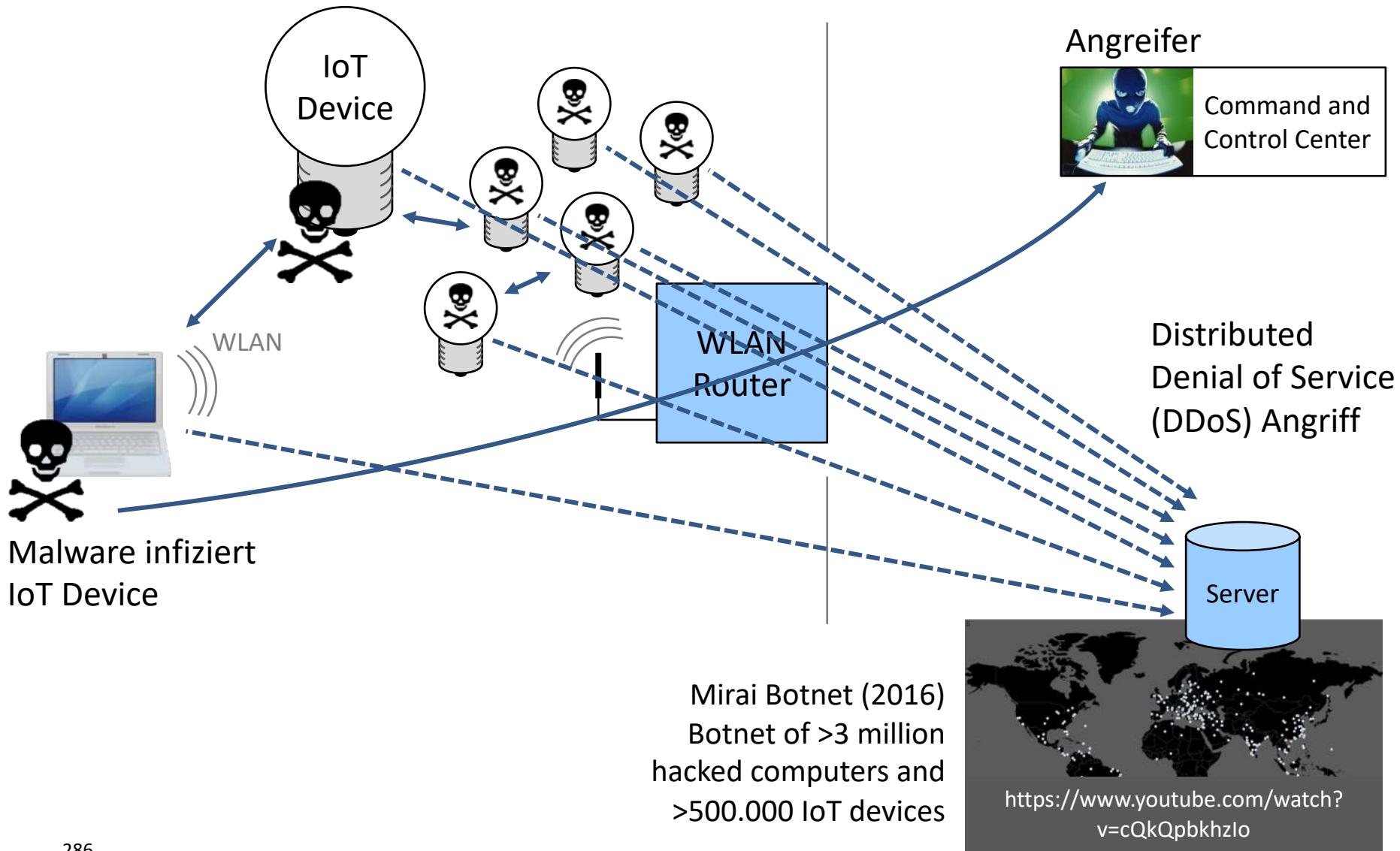




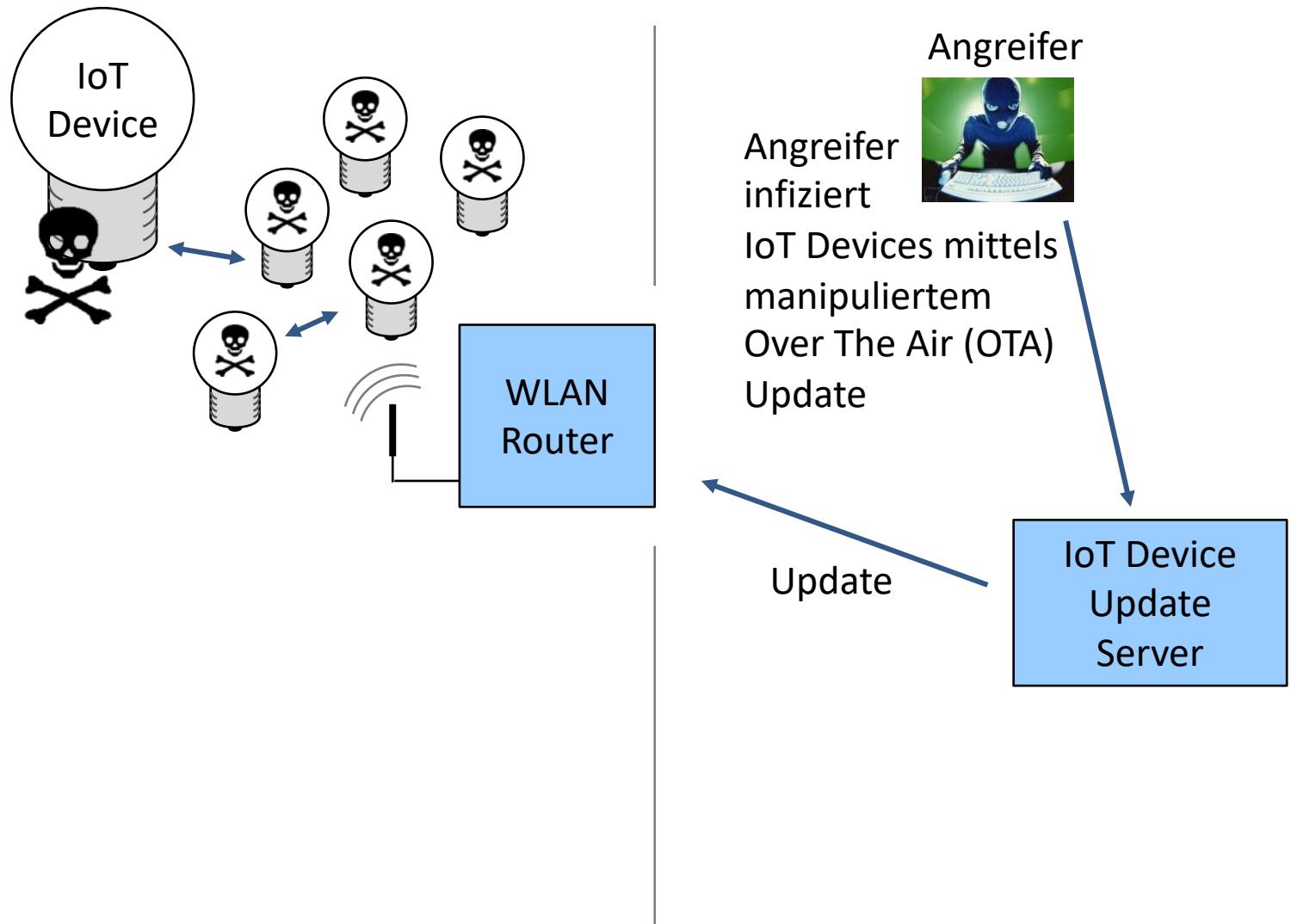
Internet of Things – Sichere Kommunikation über IoT Broker



Internet of Things – IoT Devices als Teil eines Botnetzes



Internet of Things – Over The Air (OTA) Update



Mobile Security

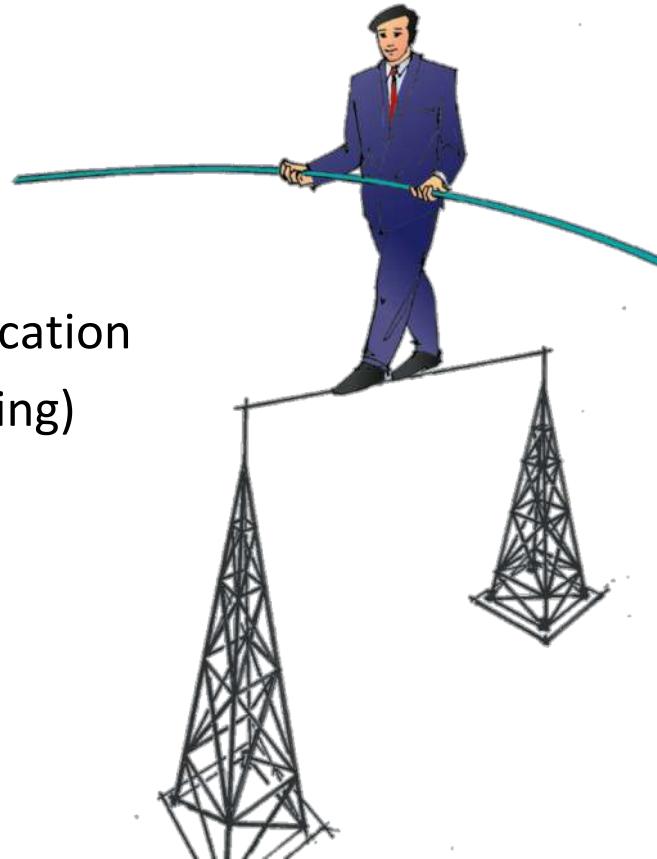
Contents

- Introduction
- Security functions of GSM
 - Basics and architecture of GSM
 - Security functions
 - Mobility management functions
 - Location based systems
 - Call management
- Security functions of further mobile Systems
 - UMTS
 - Bluetooth
 - WLAN



Mobile network communication vs. fixed networks

- Users are moving / roaming
- On air interface:
 - Limited bandwidth
 - Errors (bit failures, burst errors)
 - Communication breaks (lost connectivity)
- New threats
 - Sniffing / eavesdropping of wireless communication
 - Location finding (direction-finding, sense-finding)



Sensors make new Apps possible

»Appification«

■ Sensors in mobile devices

- GPS, Location
- WiFi, NFC
- Camera, Microphone
- Motion Sensor (Gyro)
- Compass
- Temperature
- Phonebook
- Internal Storage
- External Storage
- Screen distance
- Fingerprint Sensor



<http://blog.digifit.com/wp-content/uploads/2011/02/>

■ Adapters for more sensors

- Personal: heart rate monitors
- Cars and Houses: CAN bus adapters, smart meter, heater, alarm system

Sensors make new Apps possible

developer.android.com

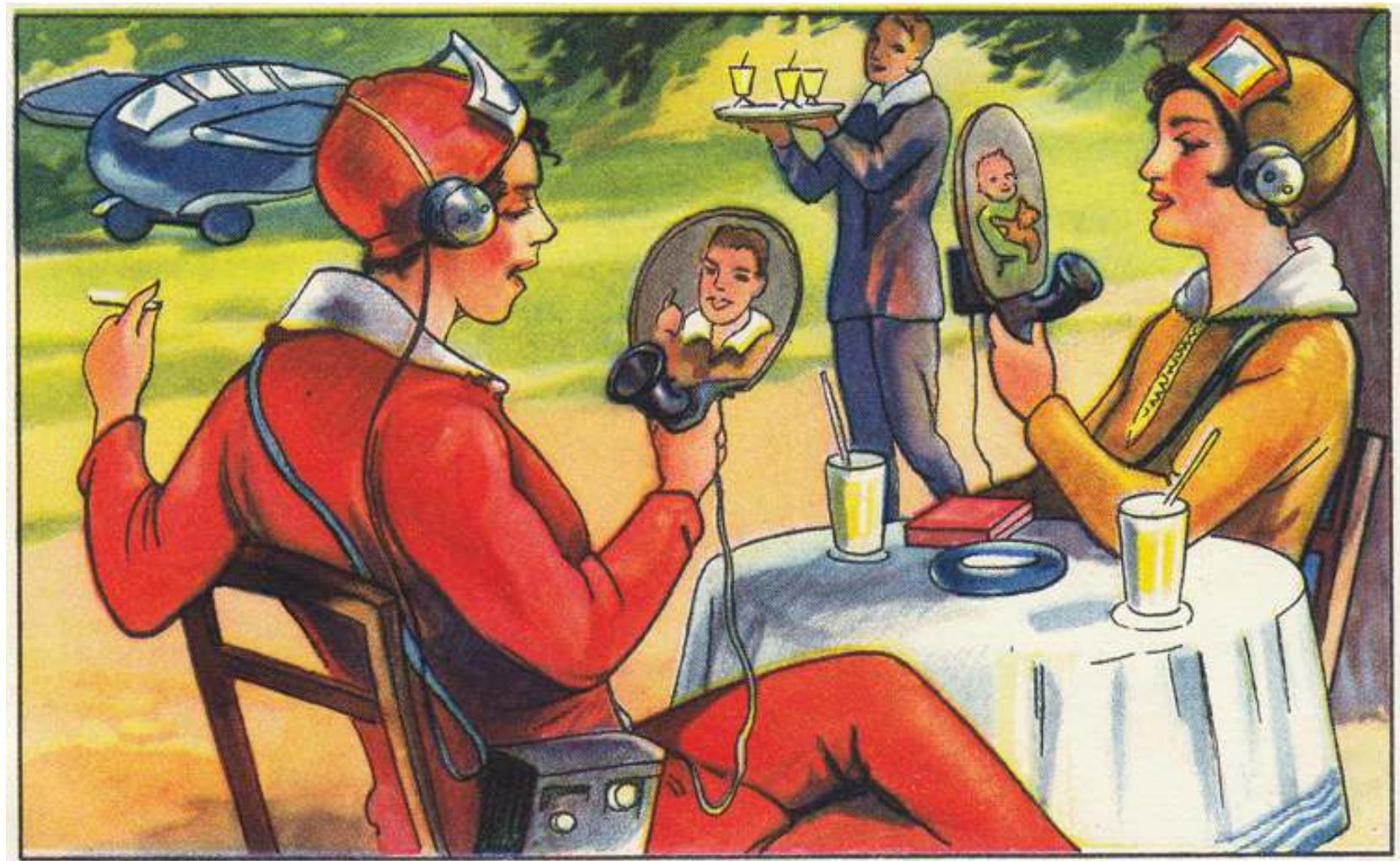
Sensors in mobile devices	Explicit	Implicit permissions
– GPS, Location	x	
– WiFi, NFC	x	
– Camera, Microphone	x	
– Motion Sensor (Gyro)		x
– Compass		x
– Temperature		
– Phonebook	x	
– Internal Storage		x
– External Storage	x	
– Screen distance		x
– Fingerprint Sensor	x	

All permissions to be found at

- <https://developer.android.com/reference/android/Manifest.permission.html>

Zukunftsphantasien

Echte Wagner Margarine, ca. 1930



Bildquellen: http://klausbuergle.de/sammelalben_zf.htm

https://monoskop.org/File:Echte_Wagner_Margarine_3_Serie_12_Zukunftsfantasien_Bild_4_c1930.jpg

Mobile communication – Classification

1. Types of Mobility

- **Terminal Mobility:**
 - Example: **Mobile Phone**
 - Wireless communication
 - Mobile device
- **Personal Mobility:**
 - Example: **Public Terminals**
 - Mobile user
 - Location-independent address
 - Special kind of personal mobility: **Session Mobility:**
 - **Session Freezing** and reactivation in other location and/or device

Mobile communication – Classification

2. Wave lengths

- Radio [waves] ($f = 100 \text{ MHz}$ up to several GHz)
- Light [waves] (infrared)
- Sonar [waves] (e.g. acoustic coupler)

3. Cell sizes

- Pico cells $d < 100 \text{ m}$
- Micro cells $d < 1 \text{ km}$
- Macro cells $d < 20 \text{ km}$
- Hyper cells $d < 60 \text{ km}$
- Overlay cells $d < 400 \text{ km}$

Further classifications

- Point-to-point communication, Broadcast (paging services)
- Analogue, Digital systems
- Simplex, Duplex communication channels

Examples for mobile Systems

- Speech communication = mass market
 - 1. Generation: analogue: C-Netz, Cordless Telephone, AMPS
 - 2. Generation: digital: GSM, DCS-1800, DECT
 - 3. Generation: service integration: UMTS/IMT-2000/FPLMTS
 - 4. Generation: LTE
- Satellite services
 - Iridium, Inmarsat, Globalstar, Odyssey
 - GPS (Global Positioning System), Galileo (European satellite navigation system), GLONASS
- Internet (Mobile IP)

Security deficits of existing mobile networks

- Example of security demands: Cooke, Brewster (1992)
 - protection of user data
 - protection of signaling information, incl. location
 - user authentication, equipment verification
 - fraud prevention (correct billing)
- General security demands
 - Confidentiality
 - Integrity
 - Availability
- Mobile network cannot be considered trustworthy



Attacker model

The attacker model defines the maximum strength of an adversary regarding a specific security mechanism

- Aspects of an attacker model
 - Roles of attacker (Outsider or Insider, ...)
 - combined roles also
 - Dissemination of attacker
 - Which stations or channels can be controlled?
 - Behavior of attacker
 - passive / active, observing / modifying
 - Computing power of attacker
 - unlimited: information theoretic
 - limited: complexity theoretic

Money

Time

Protection against an omnipotent attacker is impossible.

Attacker model (concrete)

- Outsiders
 - Passive attacks only (confidentiality)
- Insiders
 - Passive and active data modification attacks (integrity)
- Insiders and outsiders
 - Denial of Service attacks on air interface
- Mobile device
 - Trustwothy
- Network components
 - Safe against outsiders, but not against insiders
- Air interface
 - Location-finding (insiders and outsiders)

Global System for Mobile Communication (GSM)

■ Key features of Global System for Mobile Communication

- Very high international mobility
- Worldwide caller ID
- High geographic coverage
- High user capacity
- High speech quality
- Advanced error correction mechanisms
- Advanced resource allocation strategies (e.g. FDMA, OACSU)
- Priority emergency call service
- Built-in Security functions
 1. Subscriber Identity Module (SIM, smart card)
 2. Authentication (Mobile station → network)
 3. Pseudonymization of users on the air interface
 4. Link encryption on the air interface

GSM Timeline:

1989 Group Spécial Mobile (ETSI)

1990 GSM Standard

1991 GSM Network in Operation

2000 Transition to 3rd Generation

Architecture of GSM

Network Management



Call Management



Database Management



OMC: Operation and Maintenance Center

HLR: Home Location Register

AuC: Authentication Center

EIR: Equipment Identity Register

MSC: Mobile Switching Center

GMSC: Gateway MSC to fixed network

VLR: Visitor Location Register

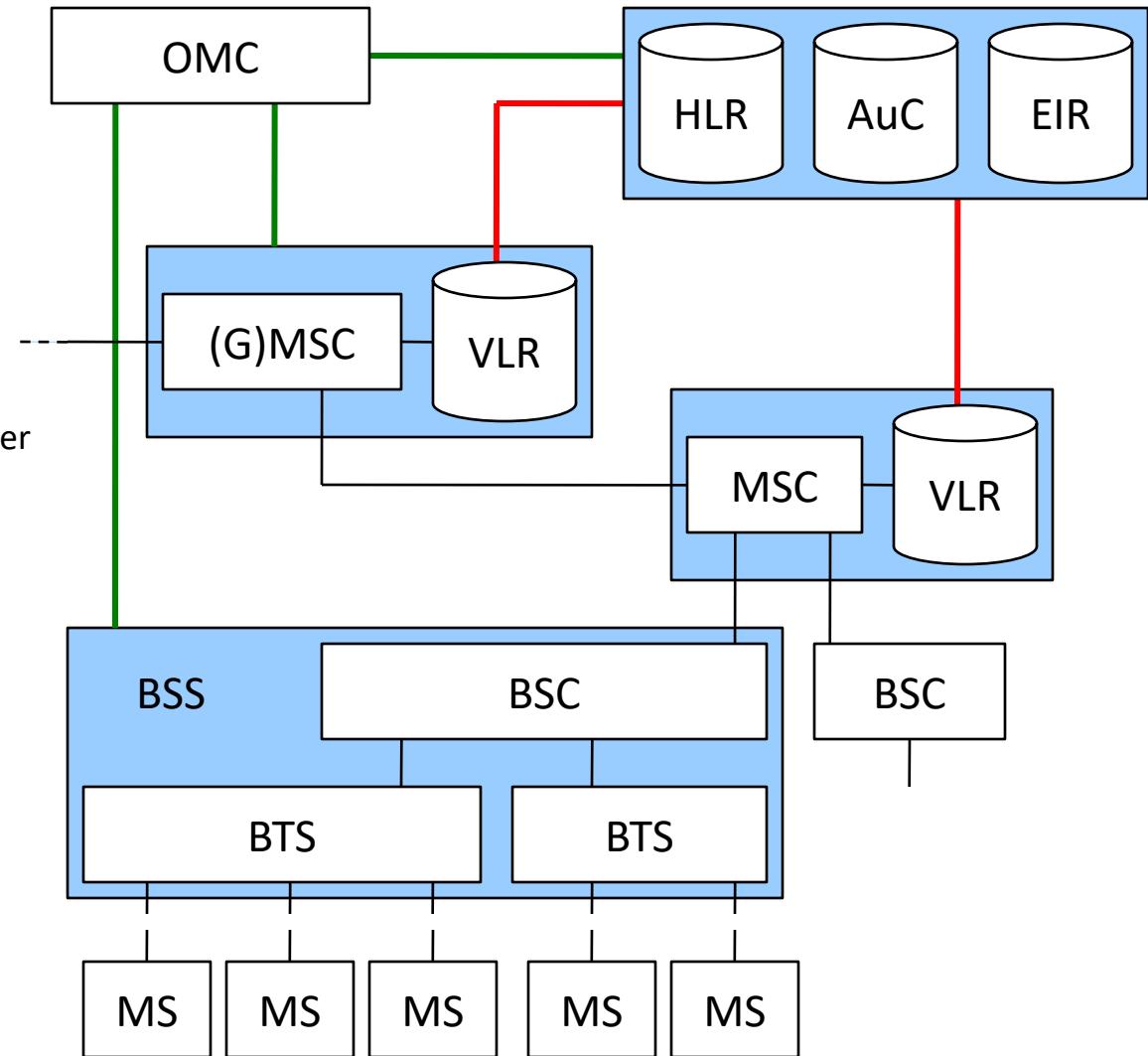
BSS: Base Station Subsystem

BSC: Base Station Controller

BTS: Base Transceiver Station

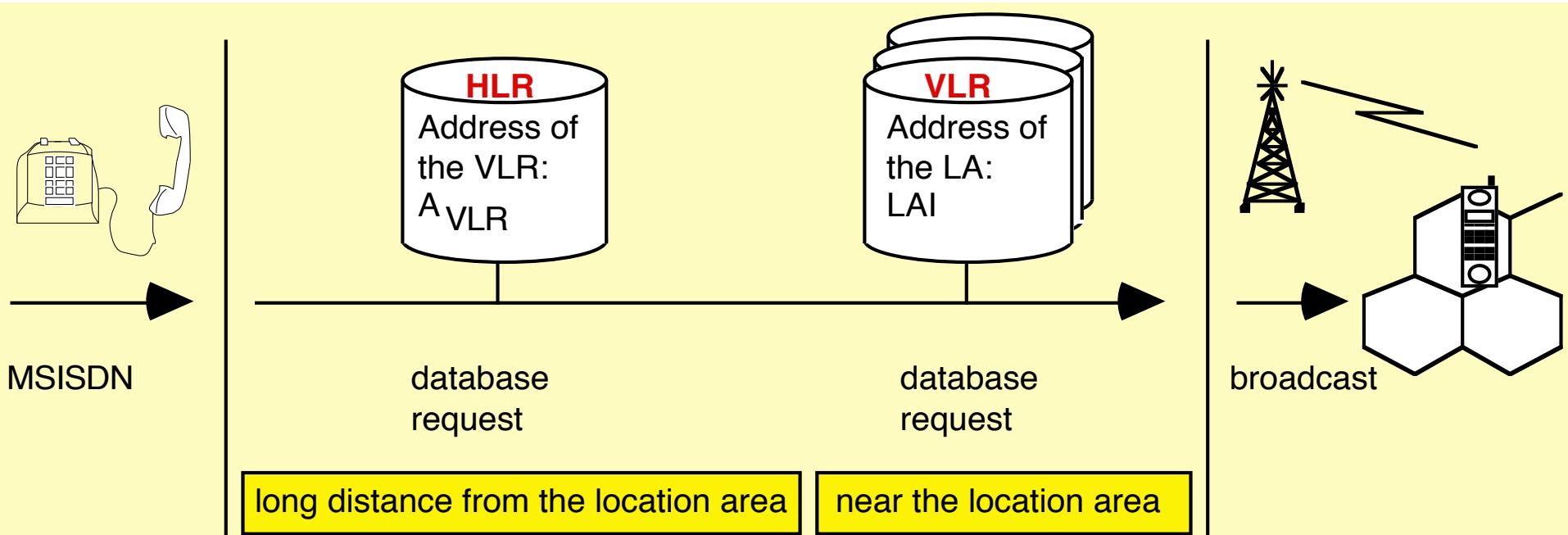
MS : Mobile Station

LA : Location Area



Location Management in GSM

- GSM (Global System for Mobile Communication)
 - Distributed storage at location registers
 - Home Location Register (HLR)
 - Visitor Location Register (VLR)
 - Network operator has global view on location information
- Tracking of mobile users is possible



Security deficits of existing mobile networks

- Example of security demands: Cooke, Brewster, 1992
 - protection of user data
 - protection of signaling information, incl. location
 - user authentication, equipment verification
 - fraud prevention (correct billing)
- Security deficits of GSM (selection)
 - Only symmetric cryptography (algorithms no officially published)
 - Weak protection of locations (against outsiders)
 - No protection against insider attacks (location, message content)
 - No end-to-end services (authentication, encryption)
- Summary
 - GSM provides protection against **external attacks** only.
 - »...the designers of GSM did not aim at a level of security much higher than that of the fixed trunk network.« Mouly, Pautet, 1992

Data bases (registers) in GSM

- Home Location Register (HLR): Semi permanent data
 - IMSI (International Mobile Subscriber Identity): max. 15 numbers
 - Mobile Country Code (MCC, 262) + Mobile Network Code (MNC, 01/02) + Mobile Subscriber Identification Number (MSIN)
 - MSISDN (Mobile Subscriber International ISDN Number): 15 numbers
 - Country Code (CC, 49) + National Destination Code (NDC, 171/172) + HLR Number + Subscriber Number (SN)
 - Number porting: translation table
 - Subscriber data (name, address, account etc.)
 - Service profile (priorities, call forwarding, service restrictions, e.g. roaming restrictions)

Data bases (registers) in GSM

- Home Location Register (HLR): Temporary data
 - VLR address, MSC address
 - MSRN (Mobile Subscriber Roaming Number)
 - CC + NDC + VLR number
$$\text{VLR number} = \text{MSC number} + \text{SN}$$
 - Authentication Set, consists of several Authentication Triplets:
 - RAND (128 Bit),
 - SRES (32 Bit) ,
 - Kc (64 Bit)
 - Billing data later on transferred to Billing Centres

Data bases (registers) in GSM

- Visitor Location Register (VLR)
 - TMSI (Temporary Mobile Subscriber Identity)
 - LAI (Location Area Identification)
 - MSRN
 - IMSI, MSISDN
 - MSC-address, HLR-address
 - Copy of Service profile
 - Billing data later on transferred to Billing Centres

Data bases (registers) in GSM

- Equipment Identity Register (EIR)
 - IMEI (International Mobile Station Equipment Identity): 15 numbers
= serial number of mobile station
 - white-lists (valid mobiles, shortened IMEI)
 - grey-lists (mobiles with failures are observed)
 - black-lists (blocked, stolen mobiles)
 - USSD (Unstructured Supplementary Service Data) code for showing IMEI: *#06#

Security functions of GSM

■ Overview

1. Subscriber Identity Module (SIM, smart card)
 - Admission control and crypto algorithms
2. Authentication (SIM → network)
 - Challenge-Response-Authentication (A3)
3. Pseudonymization of users on the air interface
 - Temporary Mobile Subscriber Identity (TMSI)
4. Link encryption on the air interface
 - Generation of session key: A8
 - Encryption: A5

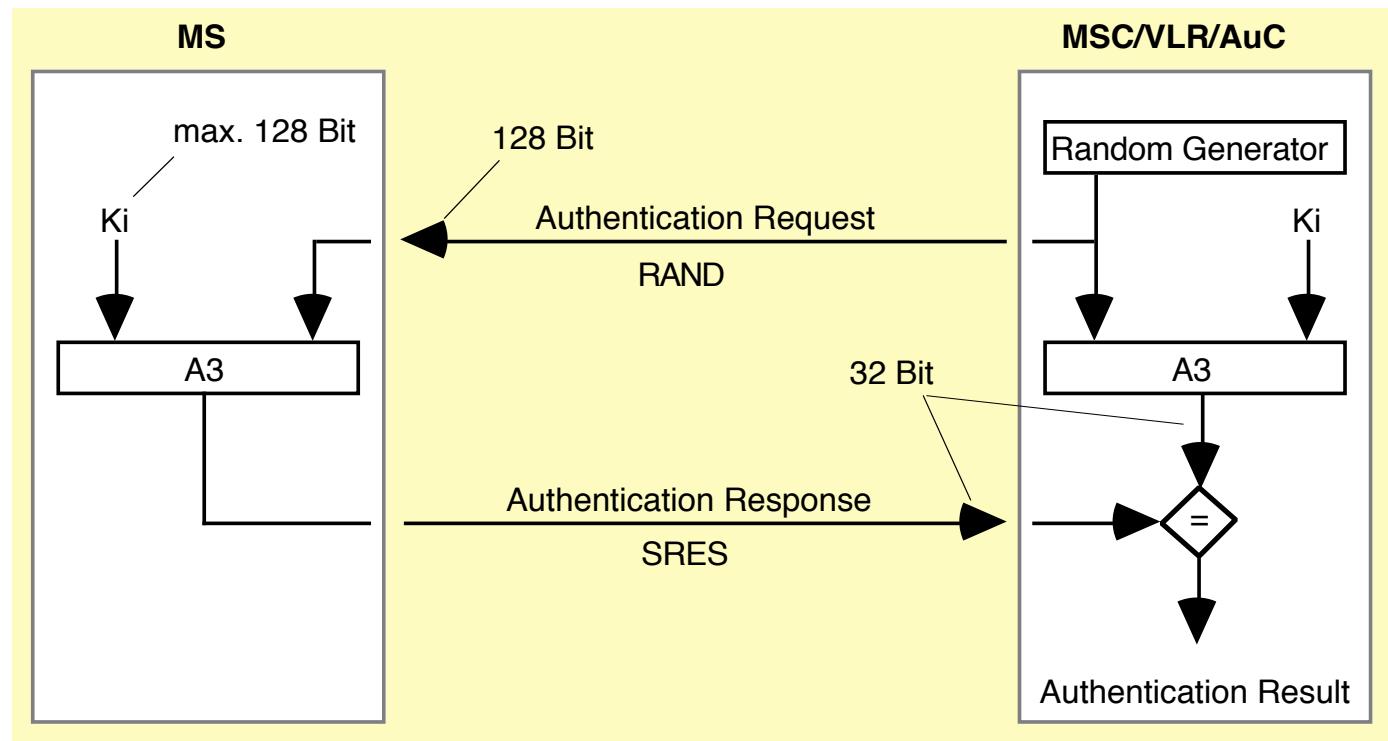


Subscriber Identity Module (SIM)

- Specialized smart card
 - Data stored on SIM:
 - IMSI (International Mobile Suscriber Identity)
 - individual symmetric key Ki (Shared Secret Key)
 - PIN (Personal Identification Number): admission control
 - TMSI (Temporary Mobile Subscriber Identity)
 - LAI (Location Area Identification)
 - Cryptographic algorithms:
 - A3: Challenge-Response-Authentication
 - A8: Session Key generation: Kc

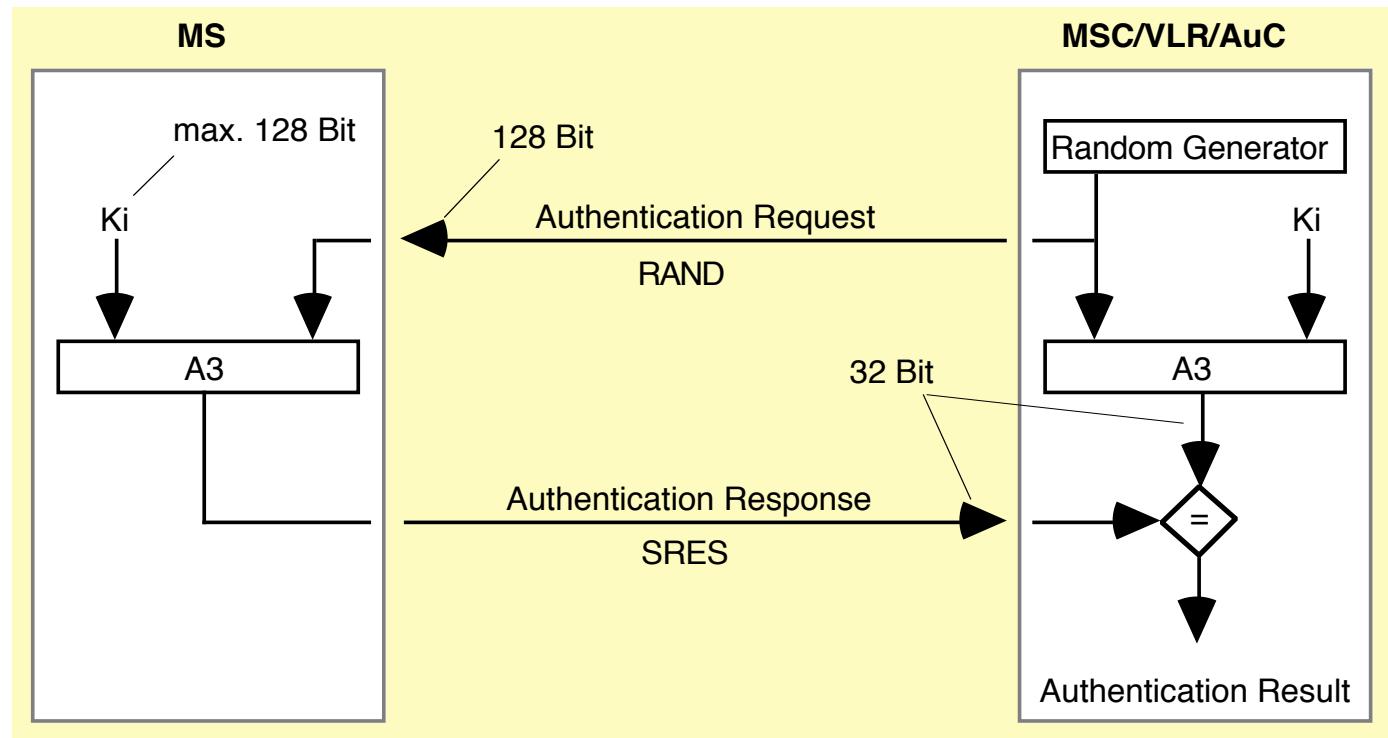
Challenge-Response-Authentication

- When initialized by the mobile network?
 - Location Registration
 - Location Update when changing the VLR
 - Call Setup (both directions)
 - Short Message Service



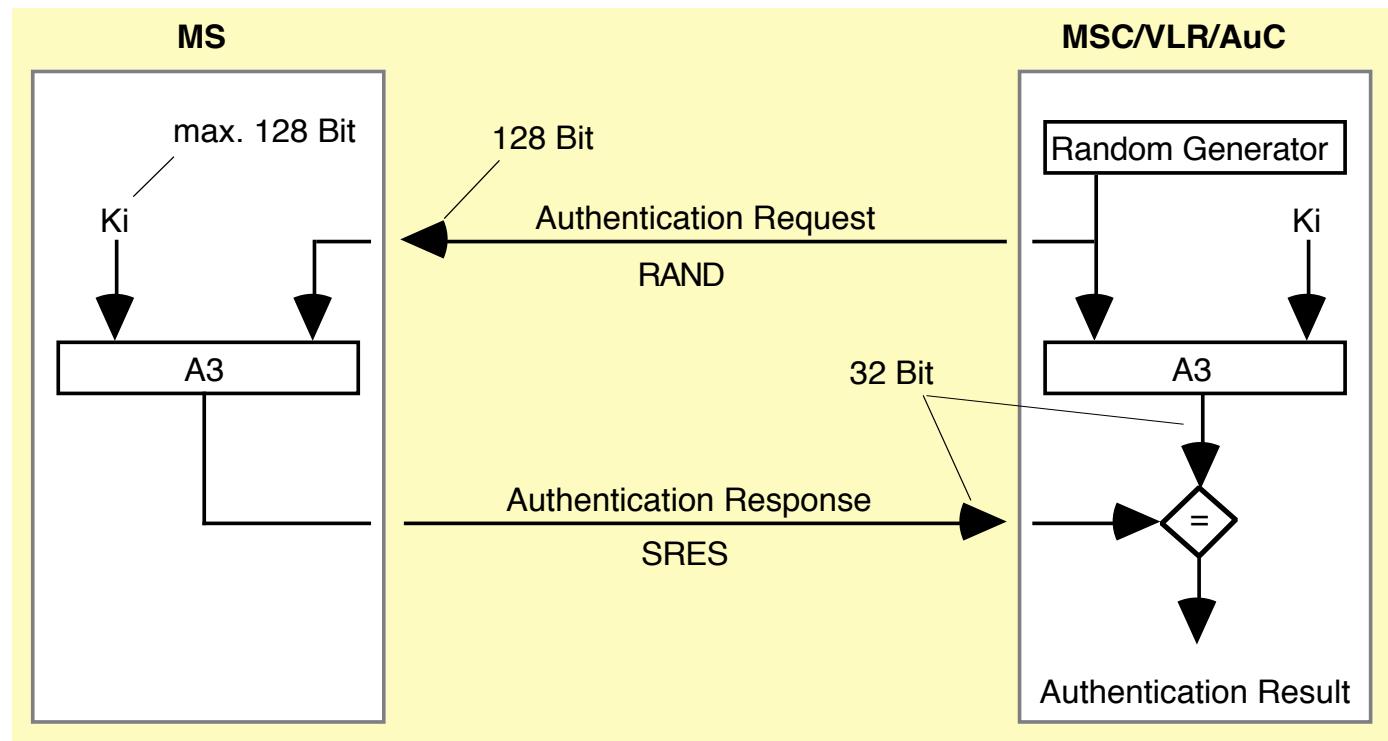
Challenge-Response-Authentication

- Algorithm A3
 - Implemented on SIM card and in Authentication Center (AuC)
 - Cryptographic one way function A3:
$$\text{SRES}' = \text{A3}(\text{Ki}, \text{RAND}) \quad (\text{Ki}: \text{individual user key})$$
 - Interfaces are standardized, cryptographic algorithm not



Challenge-Response-Authentication

- Specific algorithm can be selected by the network operator
 - Authentication data (RAND, SRES) are requested from AuC by the visited MSC
 - visited MSC: only compares SRES == SRES'
 - visited MSC has to trust home network operator

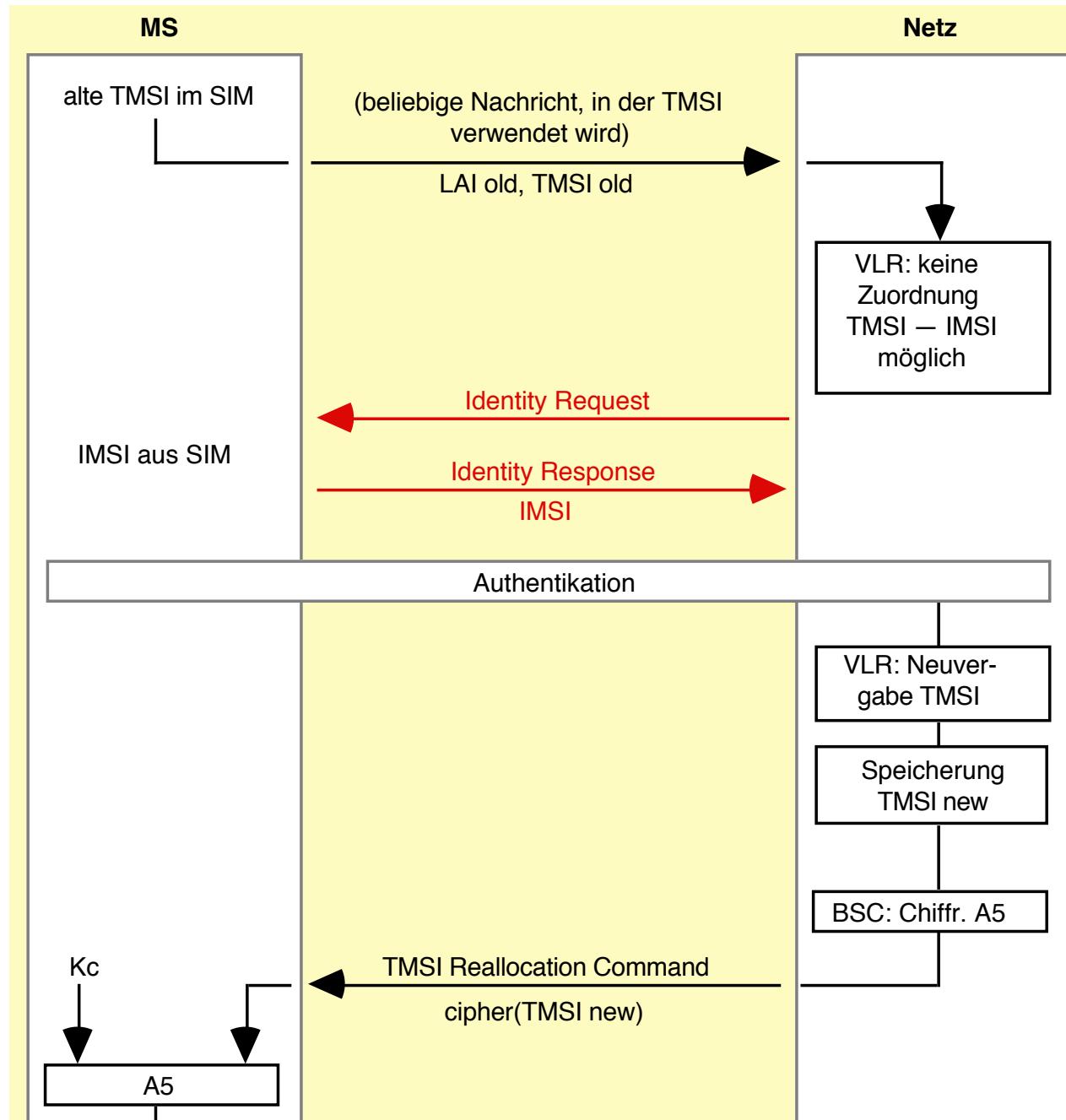


Pseudonymization on air interface

- **TMSI (Temporary Mobile Subscriber Identity)**
 - hides from traceability of mobile users by outsiders
 - on air interface: all (unencrypted) transactions from and to mobile user is addressed with TMSI
 - algorithm for TMSI generation is network individual (not standardized)
- **Identity Request**
 - first contact (home network)
 - after failure
 - IMSI is requested by serving network

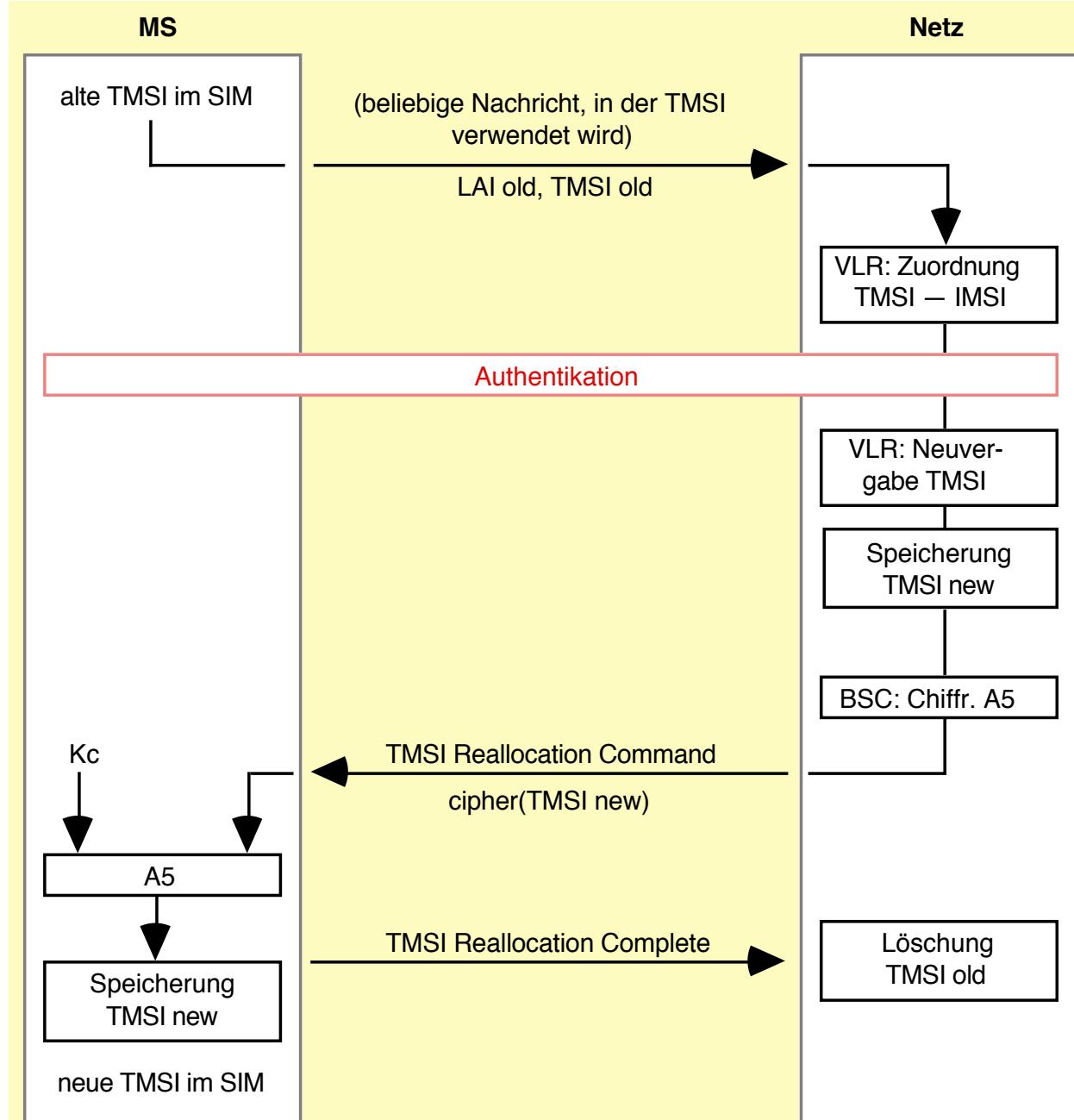
First contact
Failure

Identity Request



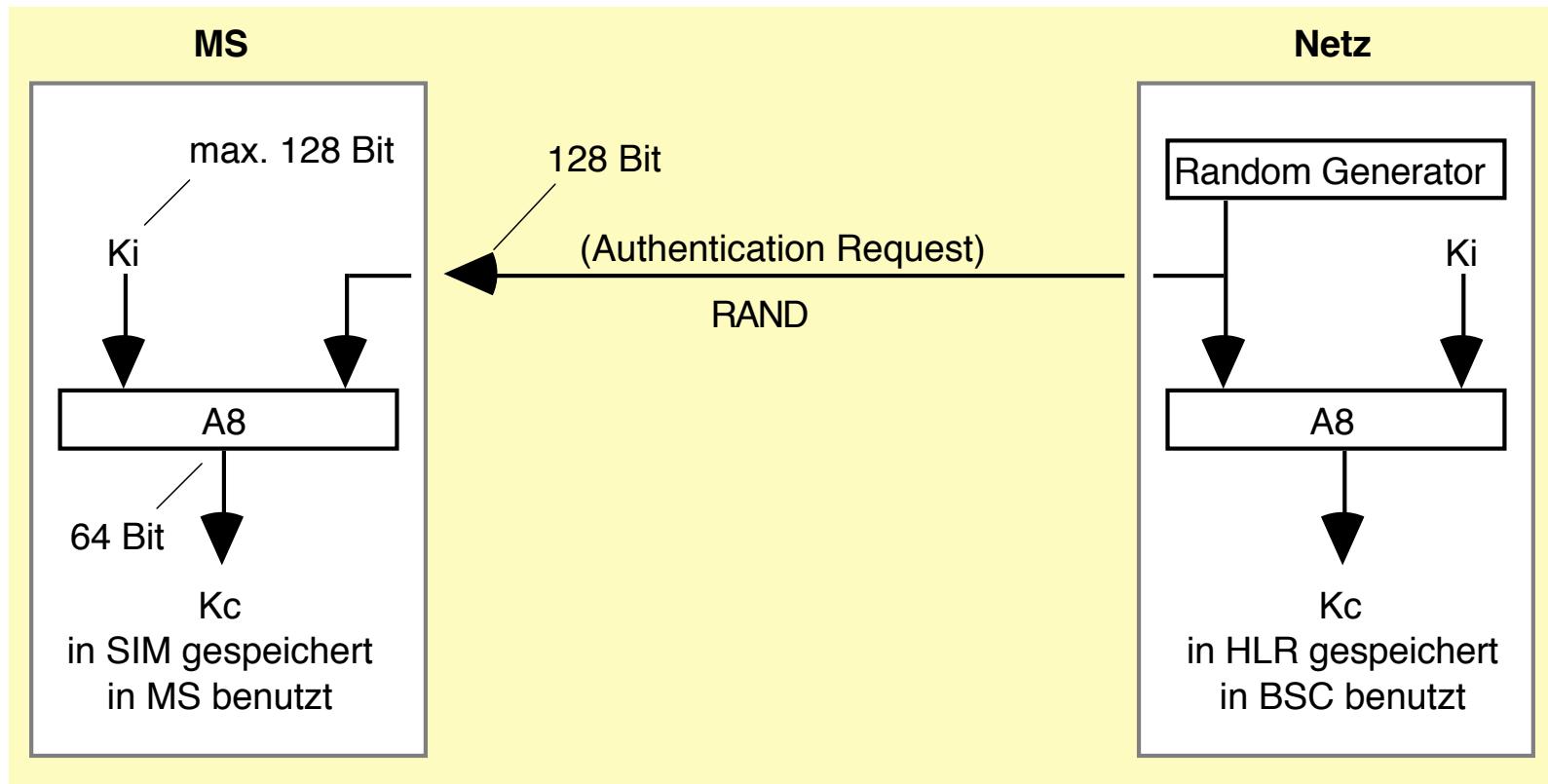
Normal case

TMSI used



Link encryption on air interface

- Session key generation: Algorithm A8

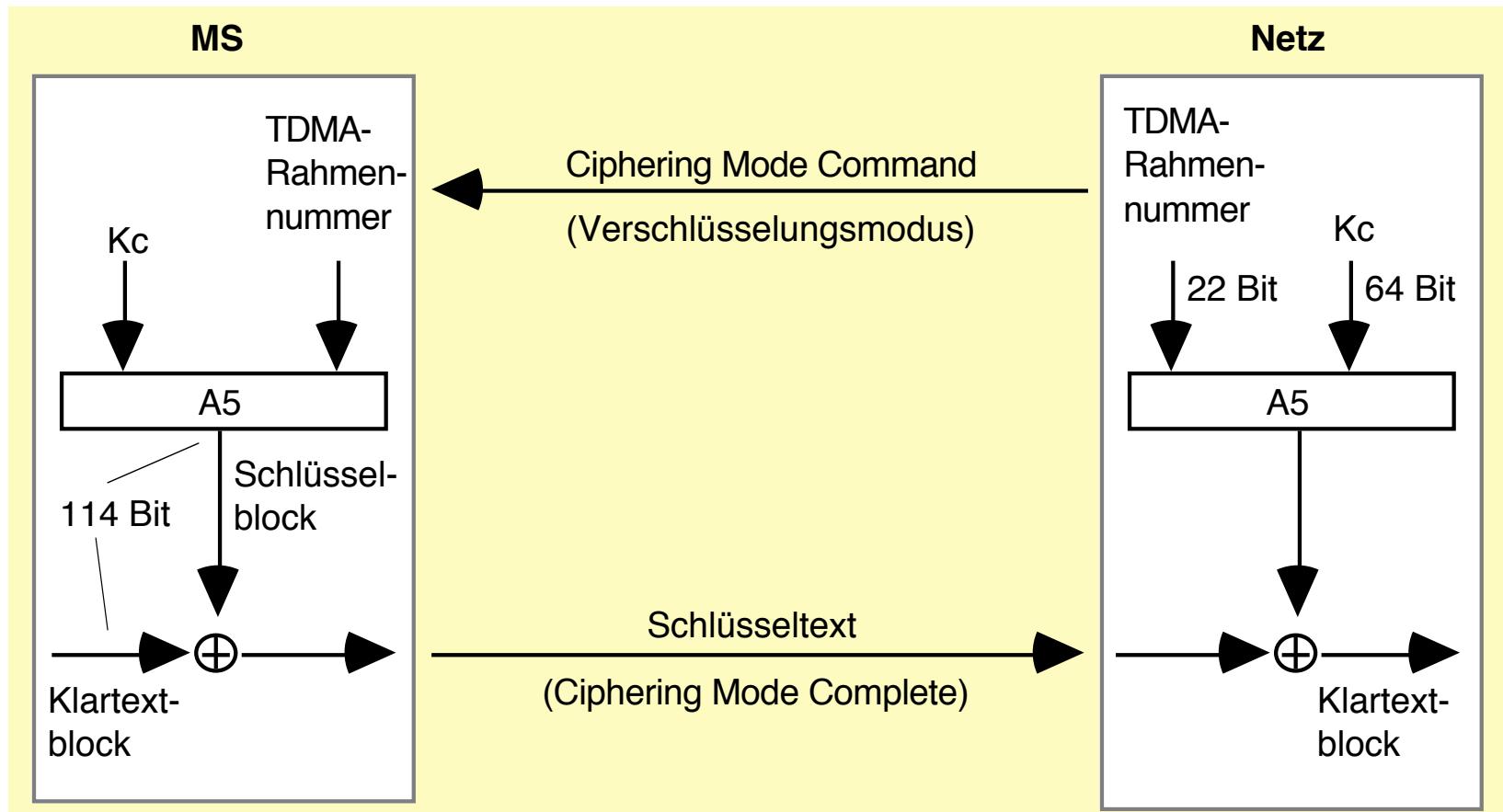


Link encryption on air interface

- Session key generation: Algorithm A8
 - implemented on SIM and in Authentication Centre (AuC)
 - cryptographic one-way function
 - interfaces are standardized
 - COMP128: well-known implementation of A3/A8

Link encryption on air interface

- Link encryption: Algorithm A5



Link encryption on air interface

- Link encryption: Algorithm A5
 - implemented in mobile station (not SIM!)
 - standardized algorithms:
 - A5 or A5/1
 - A5* or A5/2 »weak variant« of A5 — (deprecated)
 - [A5/3 based on KASUMI (UMTS) with $\text{length}(K_c)=64$ bit]
 - [A5/4 same as A5/3 with $\text{length}(K_c)=128$ bit]
- Security of A5/1 and A5/2
 - Cipher is based on non-linear shift registers
 - Algorithms considered insecure today
 - A5/1 broken by Nohl 2010
 - Attack uses ≈ 2 TByte of pre-calculated rainbow tables

Link encryption on air interface

- Ciphering Mode Command (GSM 04.08)

8	7	6	5	4	3	2	1	
TI flag	TI value		Protocol discriminator					octet 1
0	N(SD)	Message type						octet 2
Ciphering Mode Command								octet 3

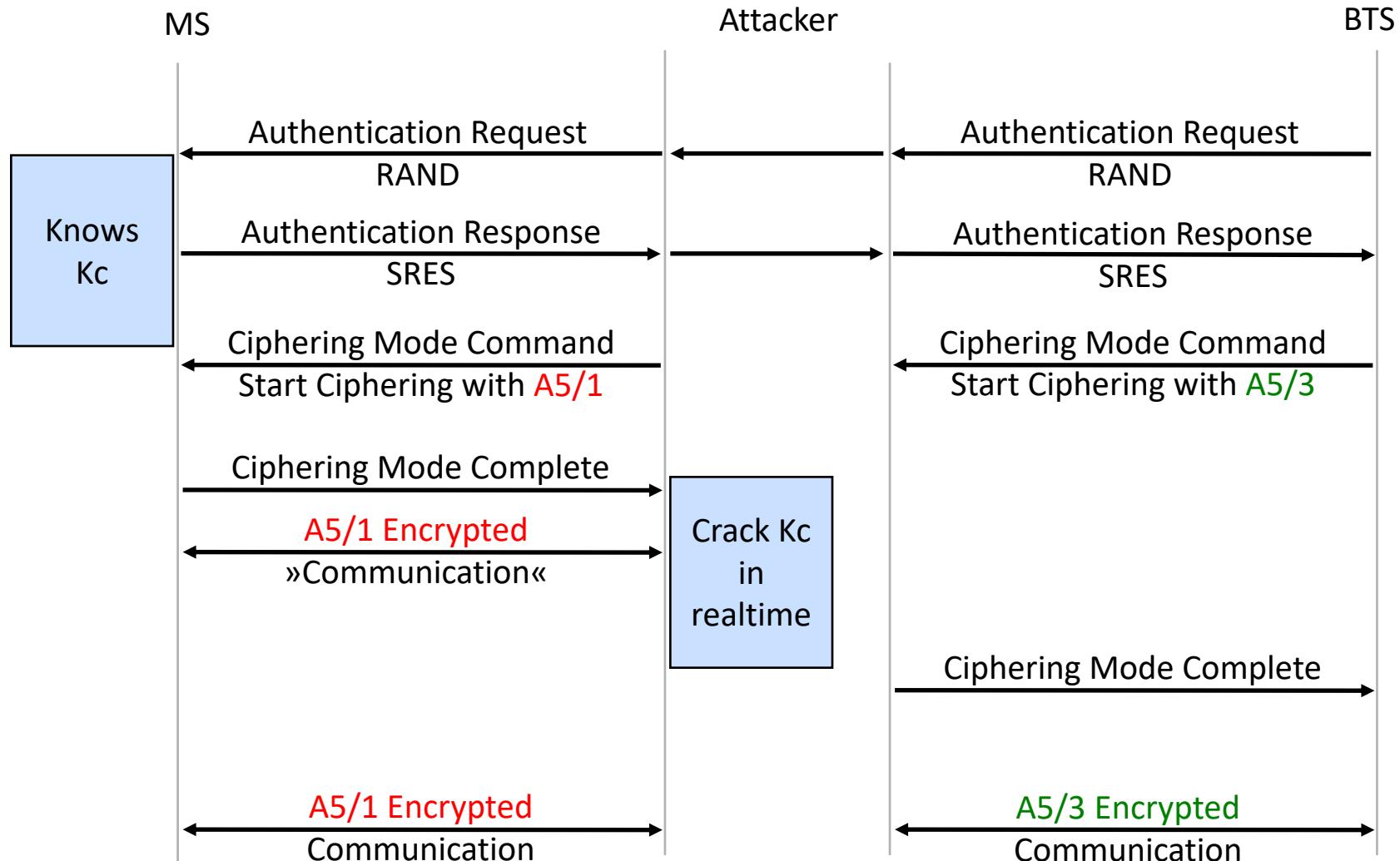
- Cipher mode setting information element

8	7	6	5	4	3	2	1	
1	0	0	1	0	0	0	SC=0	

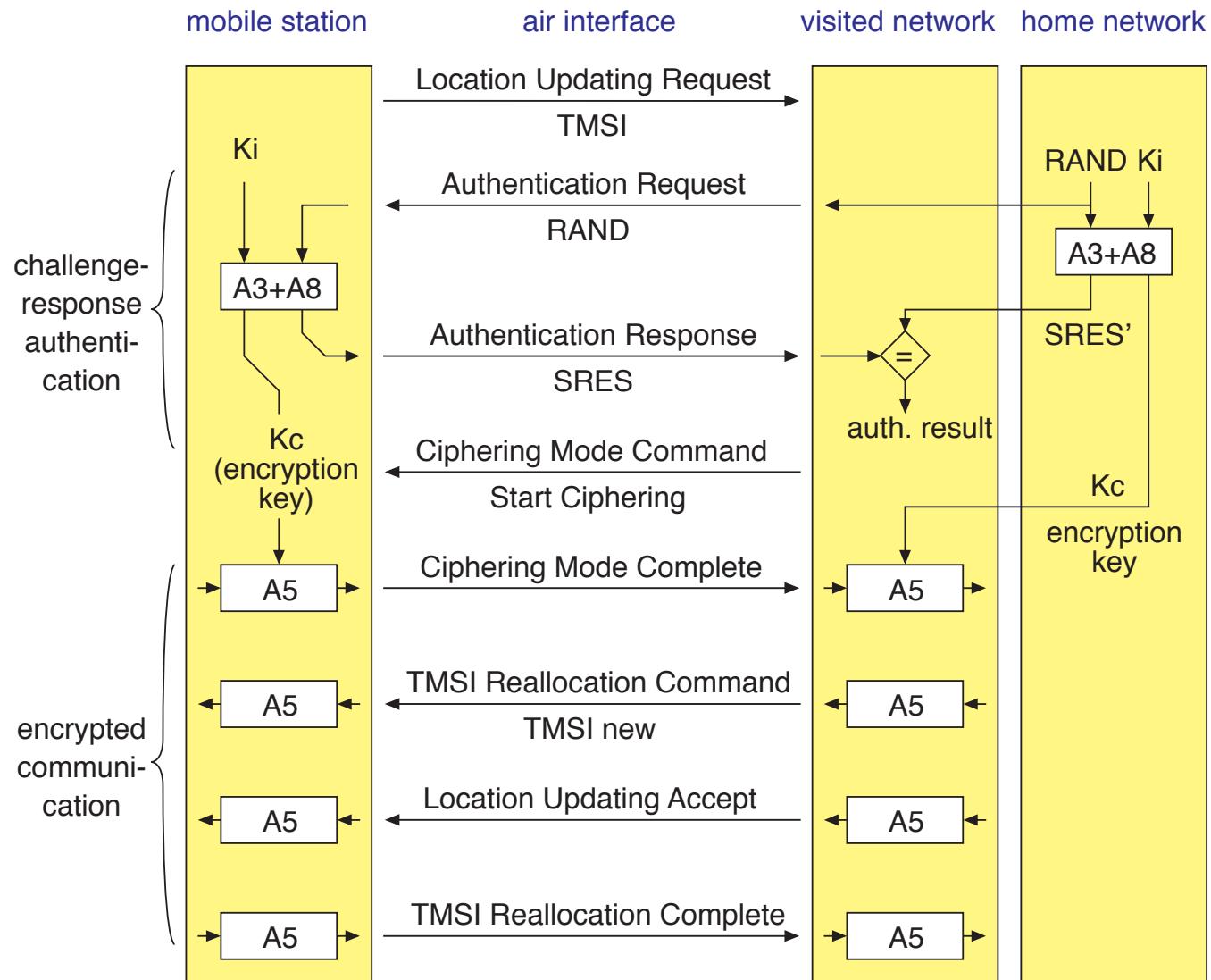
SC=0: No ciphering

SC=1: Start ciphering

Active Man-in-the-Middle Attack on A5/3



GSM security functions overview



Attacks – Telephone at the expense of others

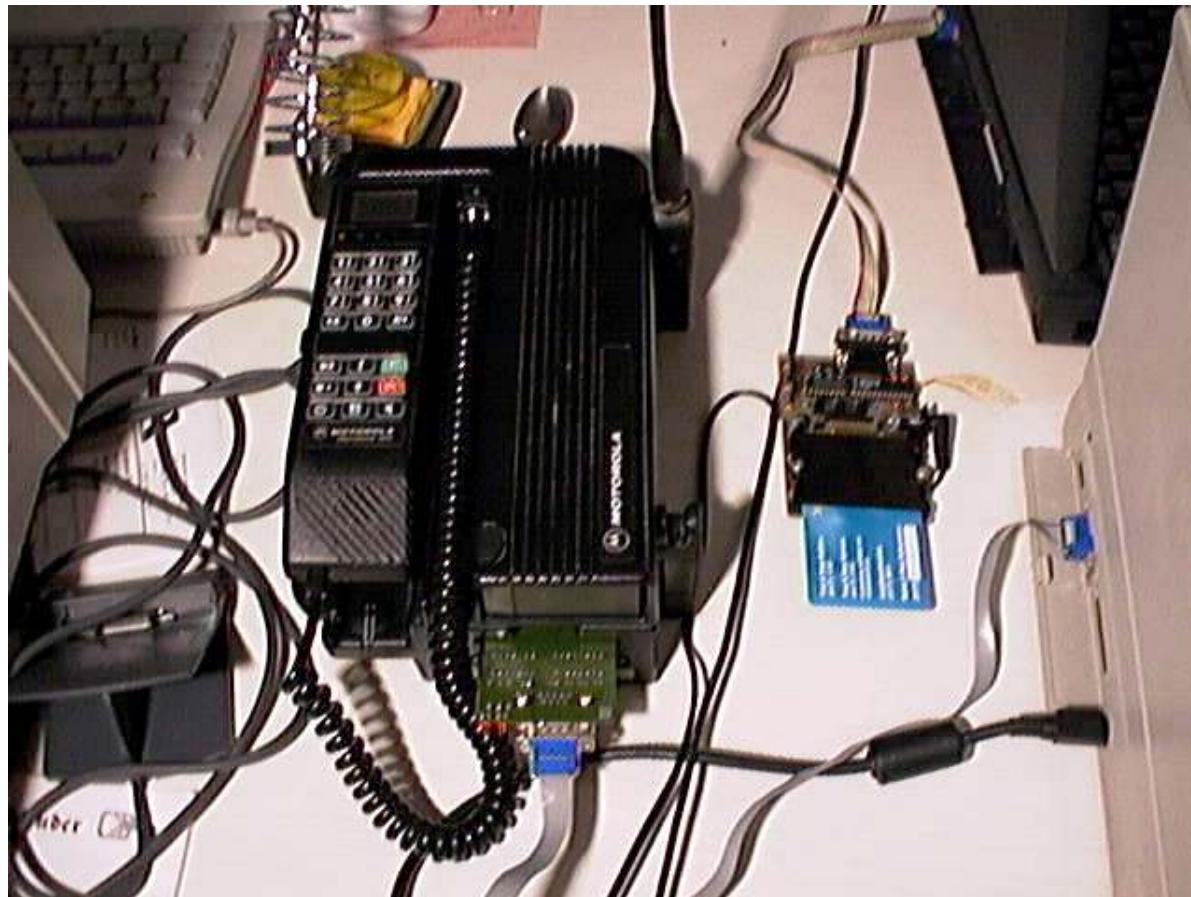
- SIM cloning
 - Weakness of authentication algorithm
- Interception of authentication data
 - Eavesdropping of internal communication links
- IMSI catcher
 - Man-in-the-middle attack on the air interface

SIM cloning

- Scope
 - Telephone at the expense of others
 - Determine Ki in SIM card
- Attack 1
 - Marc Briceno (Smart Card Developers Association), Ian Goldberg and Dave Wagner (both University of California in Berkeley)
 - <http://www.isaac.cs.berkeley.edu/isaac/gsm.html>
 - Attack uses a weakness of algorithm COMP128, which implements A3/A8
 - SIM card (incl. PIN) must be under control of the attacker for at least 8-12 hours
 - Needs 2^{17} RAND values (≈ 150.000 calculations) to determine Ki (max. 128 bit)
 - 6,25 calculations per second only, due to slow serial interface of SIM card

SIM cloning

- Scope
 - Telephone at the expense of others
 - Determine Ki in SIM card



SIM cloning

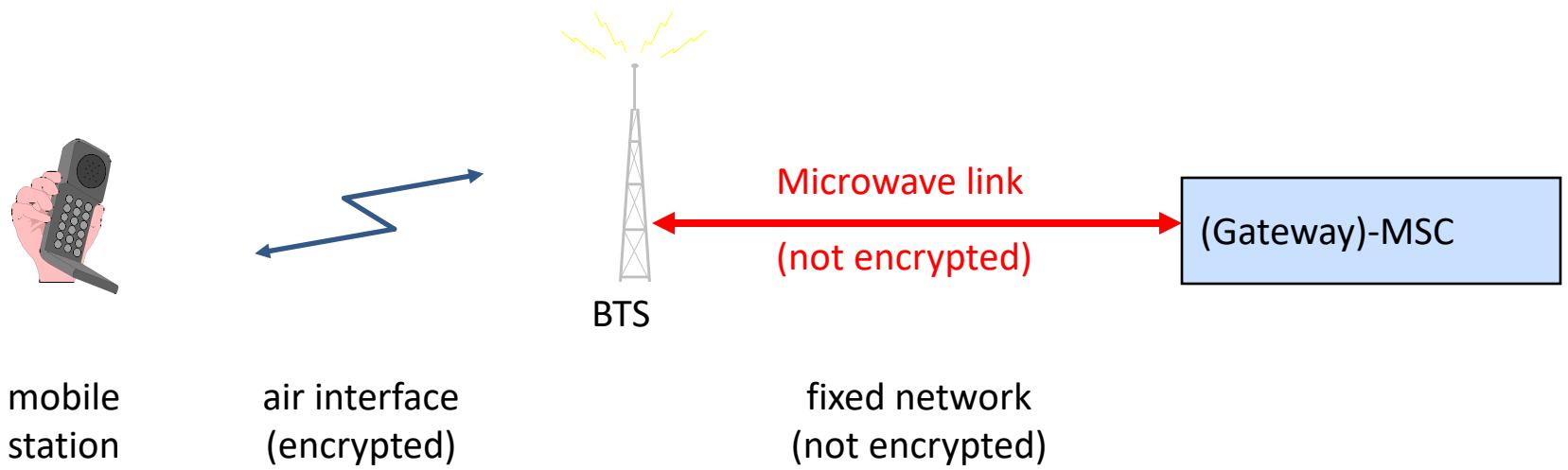
- Scope
 - Telephone at the expense of others
 - Determine Ki in SIM card
- Attack 2
 - Side Channel Attack on SIM card
 - Measurement of chip power consumption during authentication reveals Ki
 - Attack on the implementation of COMP 128, not the algorithm itself
 - Very fast: 500-1000 random inputs used for practical attack
 - More reading:
 - Rao, Rohatgi, Scherzer, Tinguely: Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards. Proc. 2002 IEEE Symposium on Security and Privacy, 2002

Interception of authentication data

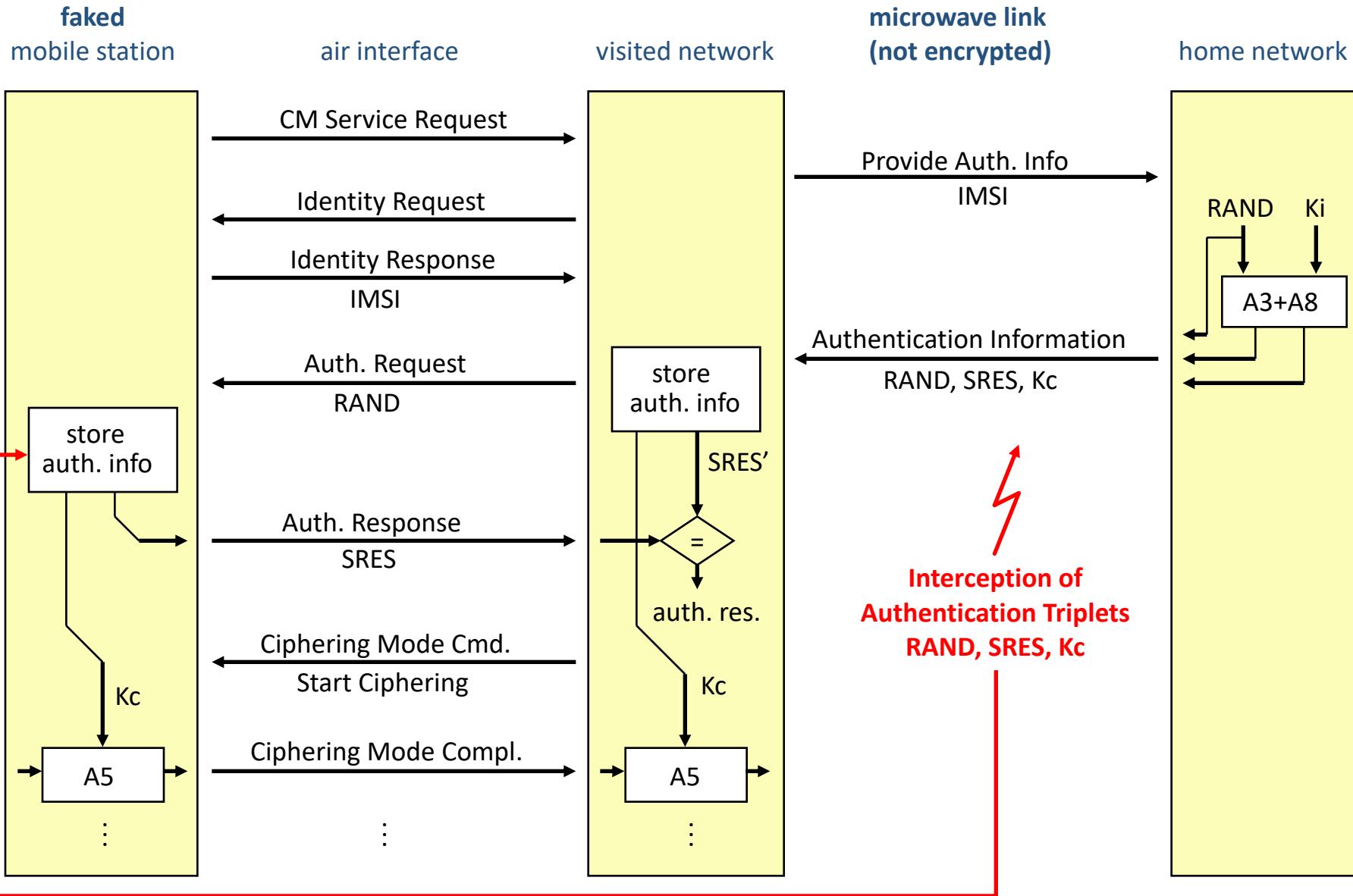
- Scope
 - Telephone at the expense of others
 - Described by Ross Anderson (University of Cambridge)
 - Eavesdropping of unencrypted internal transmission of authentication data (RAND, SRES, Kc) from AuC to visited MSC

- Weakness
 - GSM standard only describes interfaces between network components.
 - They forgot the demand for internal encryption.
 - Microwave links are widely used for internal linkage of network components.

No encryption of internal links



Interception of authentication data



IMSI-Catcher

- Scope
 - Identities of users of a certain radio cell
 - Eavesdropping of communications
 - (Telephone at the expense of others)
- Man-in-the-middle attack (Masquerade)
- Weakness
 - No protection against malicious or faked network components
- EP 1 051 053 B1
 - April 2000 by Rohde & Schwarz

IMSI-Catcher

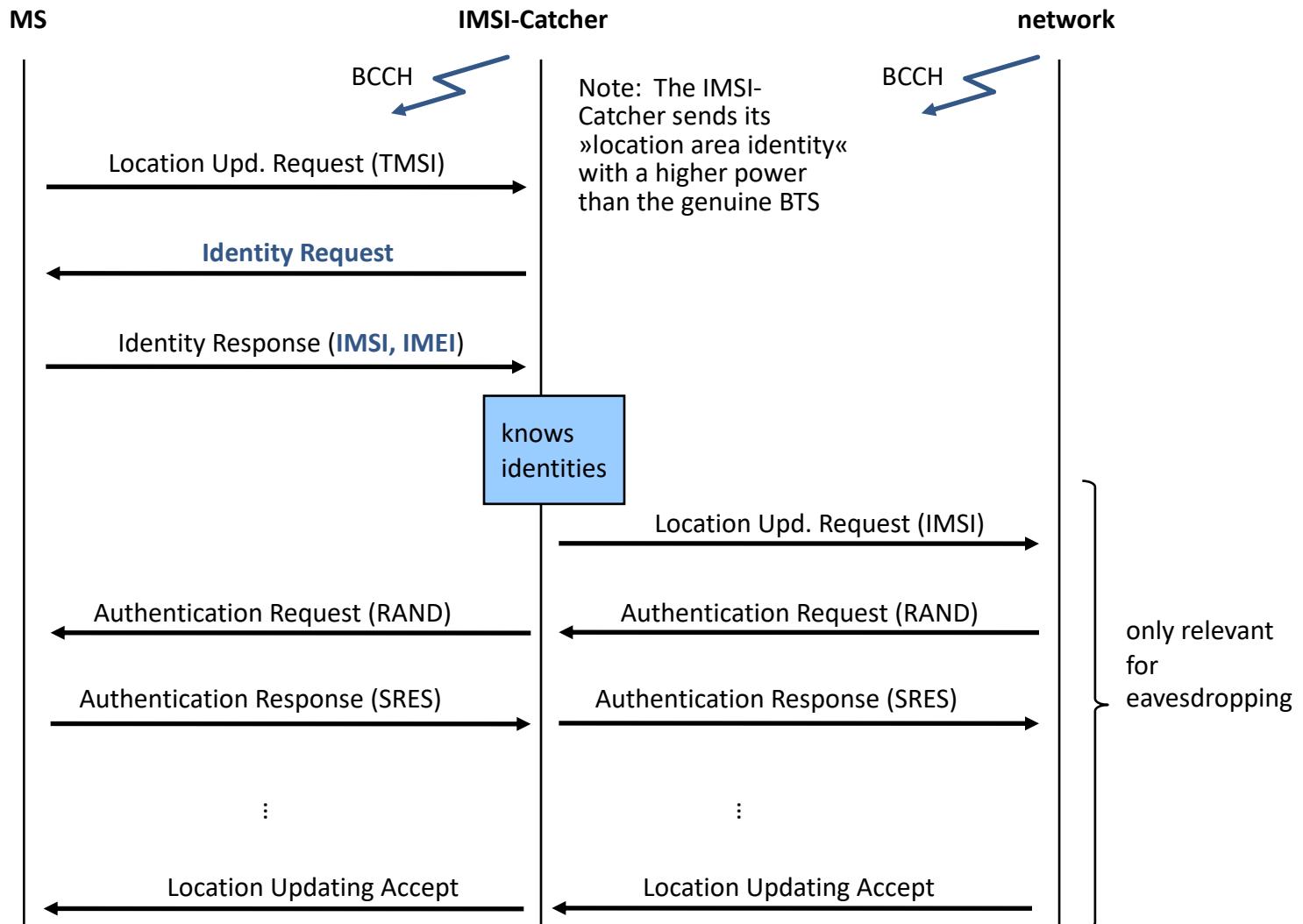


Pictures: Verfassungsschutz,

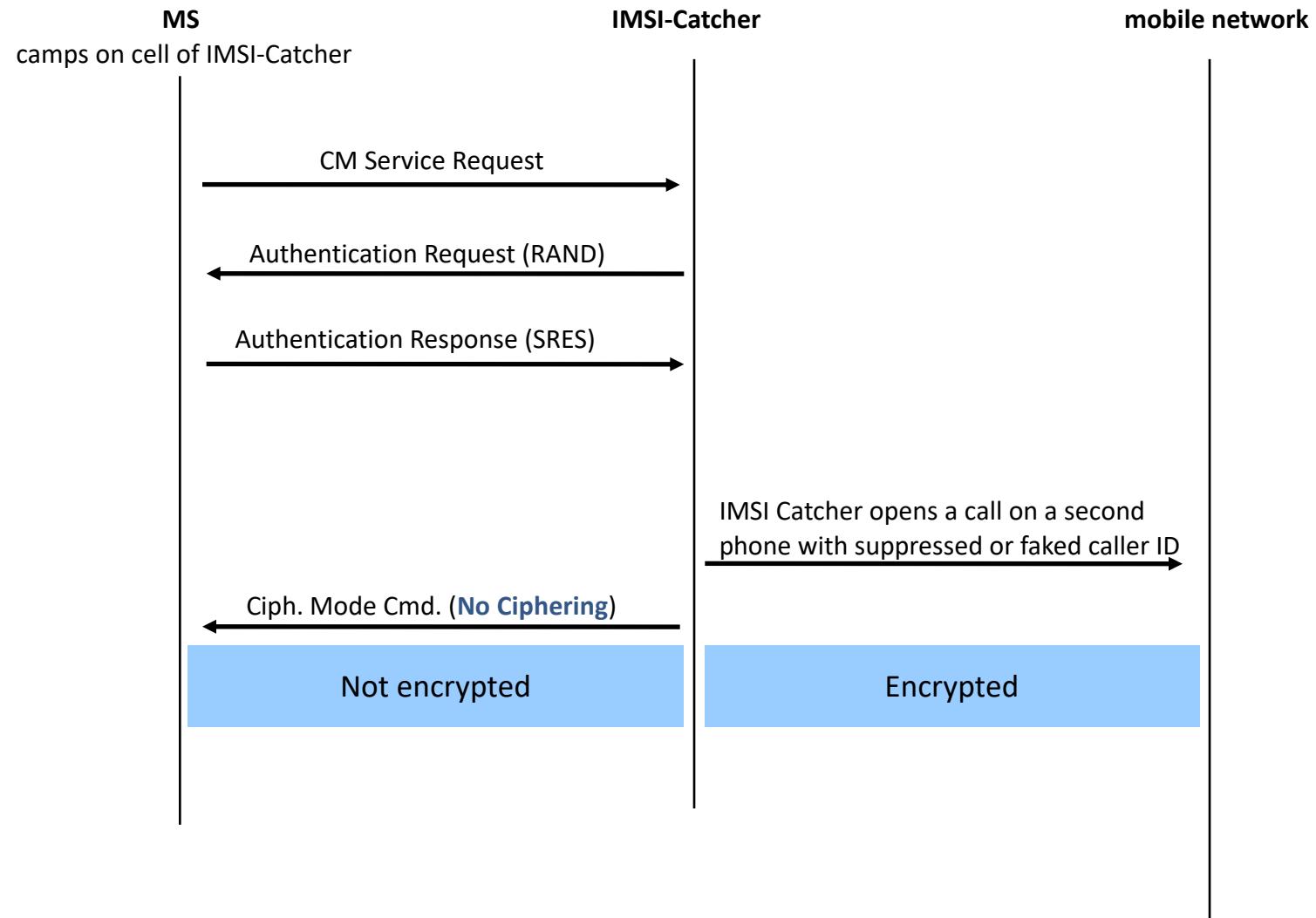
<http://www.datenschutz-und-datensicherheit.de/jhrg26/imsicatcher-fox-2002.pdf>

<http://www.heise.de/ct/artikel/Digitale-Selbstverteidigung-mit-dem-IMSI-Catcher-2303215.html>

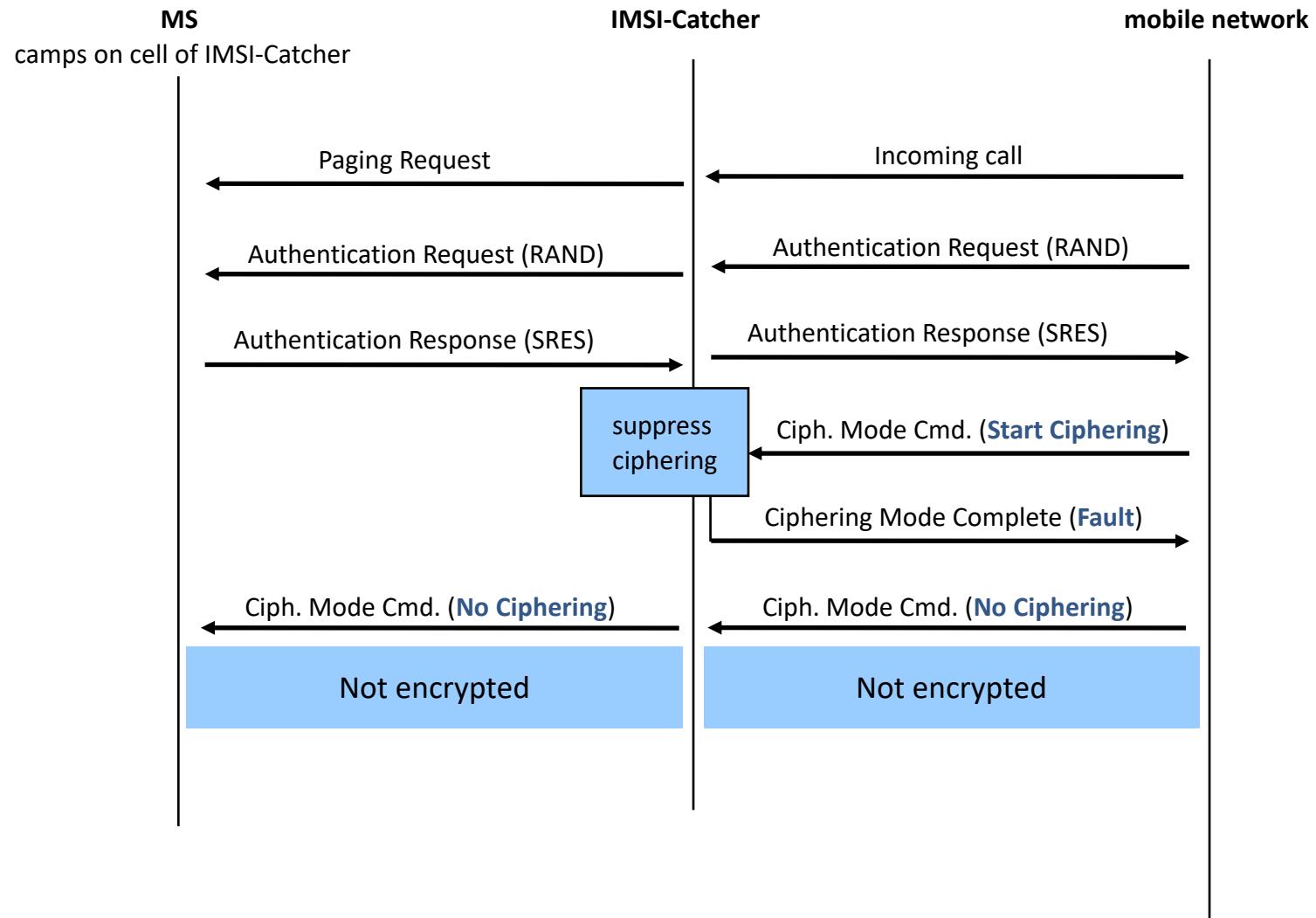
IMSI-Catcher: Getting IMSI and IMEI



IMSI-Catcher: Eavesdropping Mobile Originated Calls



IMSI-Catcher: Eavesdropping Mobile Terminated Calls



IMSI-Catcher (1)

- All BTS' send a list of frequencies of BCCHs of their neighboring cells and the own LAI
- Examples:
 - BTS 7: f4, f5, f8; LA 2
 - BTS 8: f7, f4, f5, f6, f9; LA 2

BTS 1: f1 / LA 1	BTS 4: f4 / LA 1	BTS 7: f7 / LA 2
BTS 2: f2 / LA 3	BTS 5: f5 / LA 1	BTS 8: f8 / LA 2
BTS 3: f3 / LA 3	BTS 6: f6 / LA 3	BTS 9: f9 / LA 2



IMSI-Catcher (2)

- IMSI-Catcher
 - receive from BCCH of current cell (5)
 - BTS 5: f1, f2, f3, f4, f6, f7, f8, f9; LA 1
 - select any frequency (e.g. f4) and receives from BCCH on f4
 - BTS 4: f1, f2, f5, f8, f7; LA 1
 - choose any LAI which differs from actual LAIs in neighborhood (e.g. LA 9)
 - send on f4 with high power
 - IMSI-C.: f1, f2, f5, f8, f7; LA 9

BTS 1: f1 / LA 1	BTS 4: f4 / LA 1	BTS 7: f7 / LA 2
BTS 2: f2 / LA 3	BTS 5: f5 / LA 1	BTS 8: f8 / LA 2
BTS 3: f3 / LA 3	BTS 6: f6 / LA 3	BTS 9: f9 / LA 2



IMSI-Catcher (3)

- MS (camps on cell 5)
 - monitors BCCHs of cells 1-9
 - finds best signal on f4 (transmitted by IMSI-Catcher) and learns that cell belongs to a new LA
 - send a LUP request to IMSI-Catcher
- IMSI-Catcher
 - responds with a Identity Request
- MS
 - answers with IMSI and IMEI

BTS 1: f1 / LA 1	BTS 4: f4 / LA 1	BTS 7: f7 / LA 2
BTS 2: f2 / LA 3	BTS 5: f5 / LA 1	BTS 8: f8 / LA 2
BTS 3: f3 / LA 3	BTS 6: f6 / LA 3	BTS 9: f9 / LA 2

● IMSI-Catcher

IMSI-Catcher (4)

- IMSI-Catcher
 - sends junk (non-decodable data) on Paging Channel (PCH) and
 - sends a frequency list of BTS which do not send the frequency of IMSI-Catcher (f4) in their frequency lists
 - IMSI-C.: f3, f6, f9; LA 9

BTS 1: f1 / LA 1	BTS 4: f4 / LA 1	BTS 7: f7 / LA 2
BTS 2: f2 / LA 3	BTS 5: f5 / LA 1	BTS 8: f8 / LA 2
BTS 3: f3 / LA 3	BTS 6: f6 / LA 3	BTS 9: f9 / LA 2

● IMSI-Catcher

IMSI-Catcher (5)

- MS

- receives junk on PCH and (according to GSM05.05) does a cell reselection:
- MS monitors signal strengths of f3, f6, f9
- changes to the best cell (LUP)

BTS 1: f1 / LA 1	BTS 4: f4 / LA 1	BTS 7: f7 / LA 2
BTS 2: f2 / LA 3	BTS 5: f5 / LA 1	BTS 8: f8 / LA 2
BTS 3: f3 / LA 3	BTS 6: f6 / LA 3	BTS 9: f9 / LA 2



IMSI-Catcher (5)

- Result
 - MS is back in the network again
 - because BTS 3, 6 and 9 do not send f4 in their frequency lists, the MS does not recognize the powerful IMSI-Catcher signal again (and subsequently does not change back to it)

BTS 1: f1 / LA 1	BTS 4: f4 / LA 1	BTS 7: f7 / LA 2
BTS 2: f2 / LA 3	BTS 5: f5 / LA 1	BTS 8: f8 / LA 2
BTS 3: f3 / LA 3	BTS 6: f6 / LA 3	BTS 9: f9 / LA 2



IMSI-Catcher detectors

- AIMSICD
 - <https://github.com/CellularPrivacy/Android-IMSI-Catcher-Detector>
- SnoopSnitch
 - from SRLabs (Karsten Nohl)
- Darshak
 - TU Berlin
- GSMK CryptoPhone
 - special Smartphone
- IMSI-Catcher-Catcher (ICC)
 - SBA Research (Adrian Dabrowski)

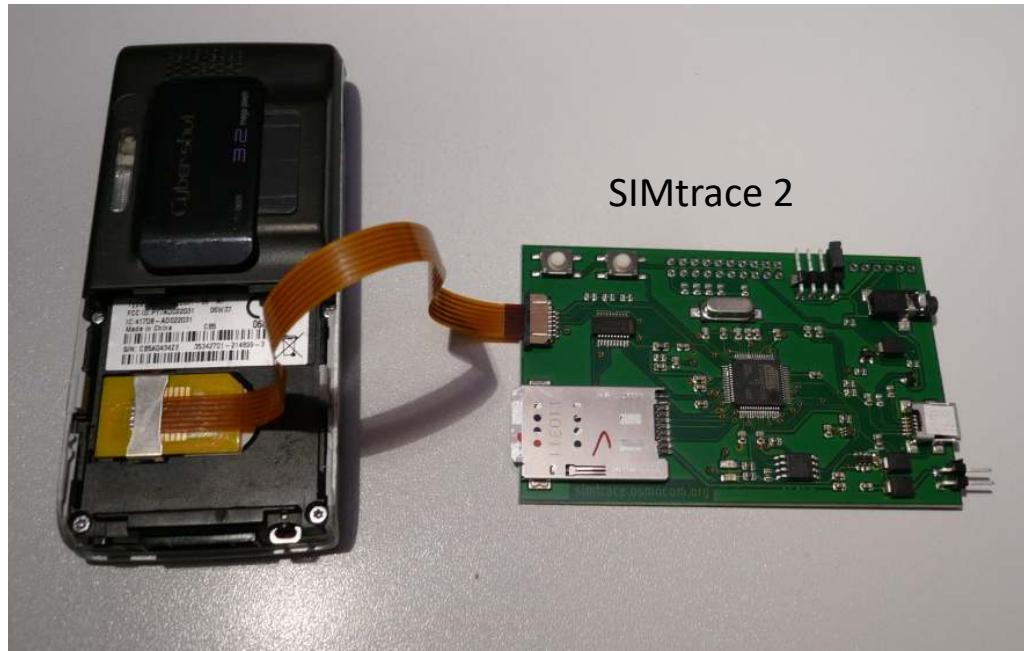


Picture (ICC): heise.de

Sources: https://www.privacy-handbuch.de/handbuch_75.htm
<http://www.heise.de/ct/artikel/Digitale-Selbstverteidigung-mit-dem-IMSI-Catcher-Catcher-2303215.html>

Mobile Communication Security Analysis (Tools)

- Osmocom SIMtrace 2
 - combination of software, firmware and hardware system
 - main purpose: sniff the communication between a phone and a SIM card
 - <https://osmocom.org/projects/simtrace2/wiki>



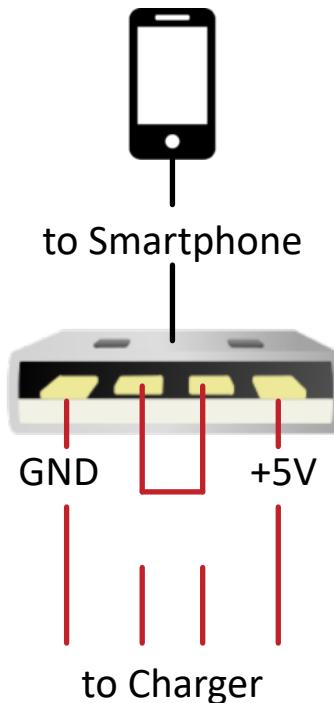
Turbo SIM: Earlier solution for sniffing communication between SIM and MS (introduced 2004, updated 2007)



<https://arstechnica.com/gadgets/2007/08/turbo-sim-add-on-allows-full-iphone-unlocking/>

USB charging condom

- USB-A has 4 wires
- cut 2 inner data wires and short-circuit
- connect power wires only



The screenshot shows a web browser displaying the INT3.CC website. The URL in the address bar is int3.cc/products/usbcondoms. The page features the INT3.CC logo and navigation links for HOME, CATALOG, BLOG, and ABOUT US. The main content area is titled "The Original USB Condom" and includes a product image of a black USB cable with a purple condom套住 its midsection. Below the title, a red banner reads "*** The USB Condom now sold on SyncStop.com ***". A text block explains that SyncStop.com offers the product, and a larger paragraph discusses the purpose of the USB condom to prevent data theft via "juice jacking".

INT3.CC

HOME CATALOG BLOG ABOUT US

The Original USB Condom

*** The USB Condom now sold on SyncStop.com ***

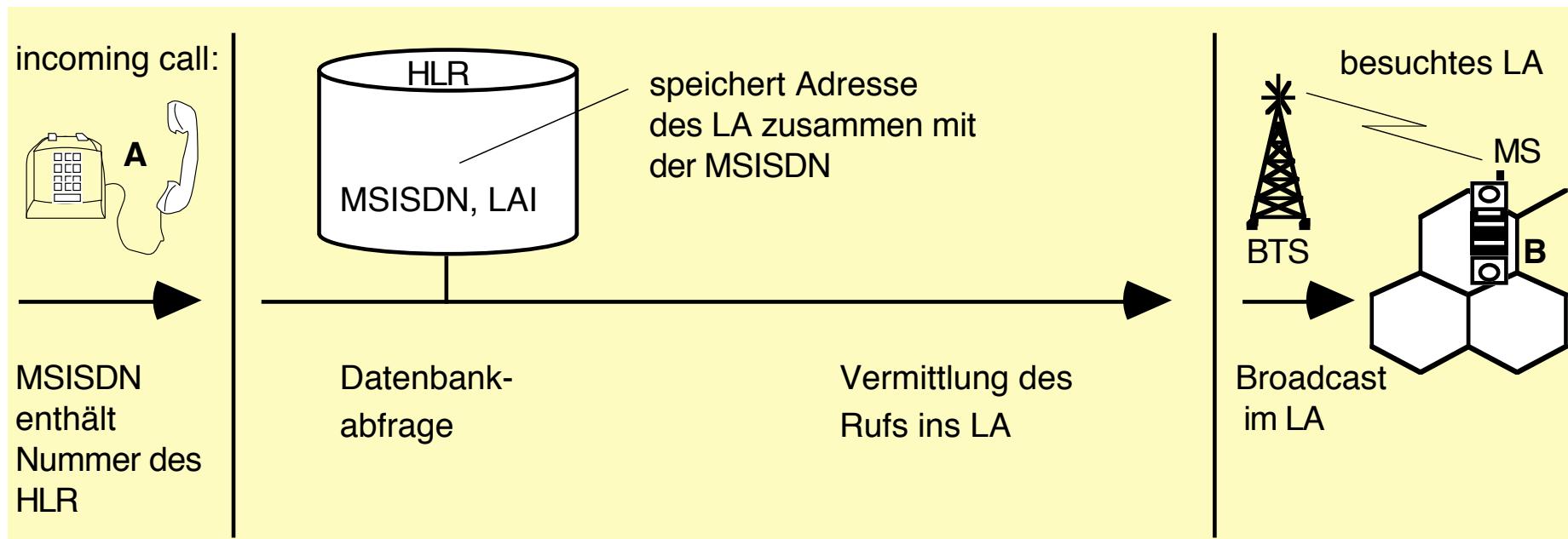
At SyncStop.com you can purchase the Original USB Condom and the new "cased" version called SyncStop :-)

Have you ever plugged your phone into a strange USB port because you **really** needed a charge and thought: "*Gee who could be stealing my data?*". We all have needs and sometimes you just **need** to charge your phone. "Any port in a storm." as the saying goes. Well now you can be a bit safer. "[USB Condoms](#)" prevent accidental data exchange when your device is plugged in to another device with a USB cable. USB Condoms achieve this by cutting off the data pins in the USB cable and allowing only the power pins to connect through. Thus, these "[USB Condoms](#)" prevent attacks like "[juice jacking](#)".

Location Management

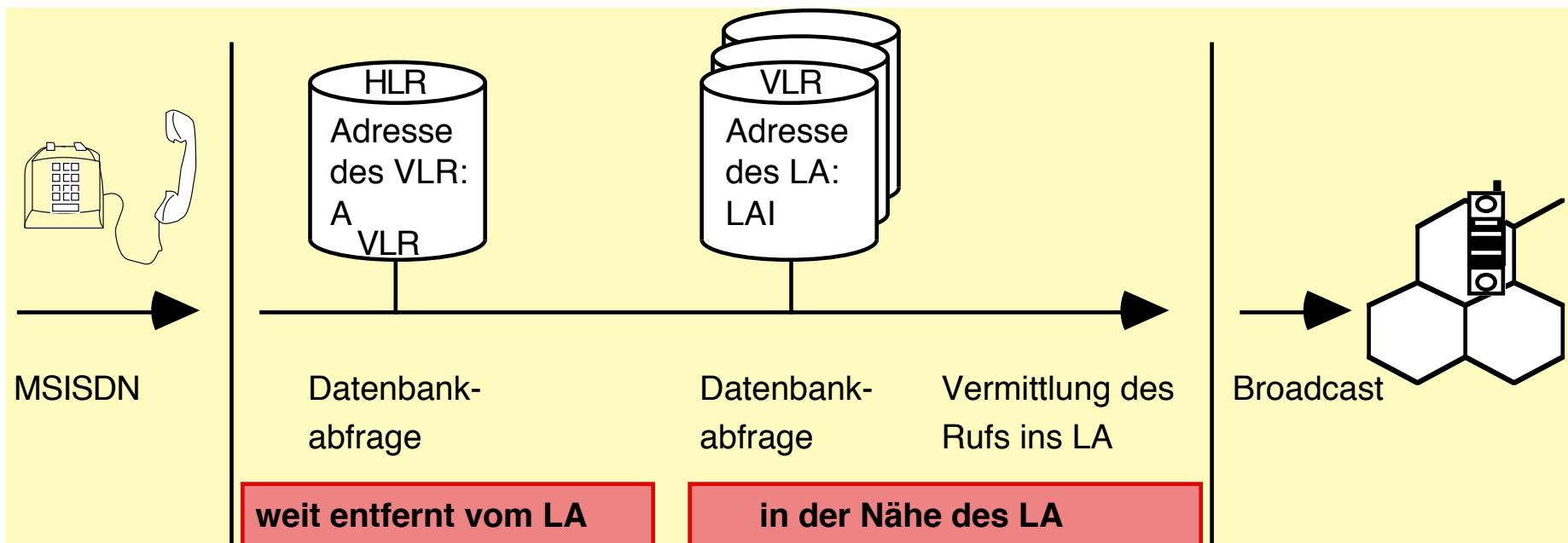
- Centralized approach
 - Change of Location Area (LA), i.e. Location Updating, needs communication with HLR (far away from LA)
 - Efficiency: Good at low Location Updating rates

- Used in Mobile IP
 - HLR = Home Agent



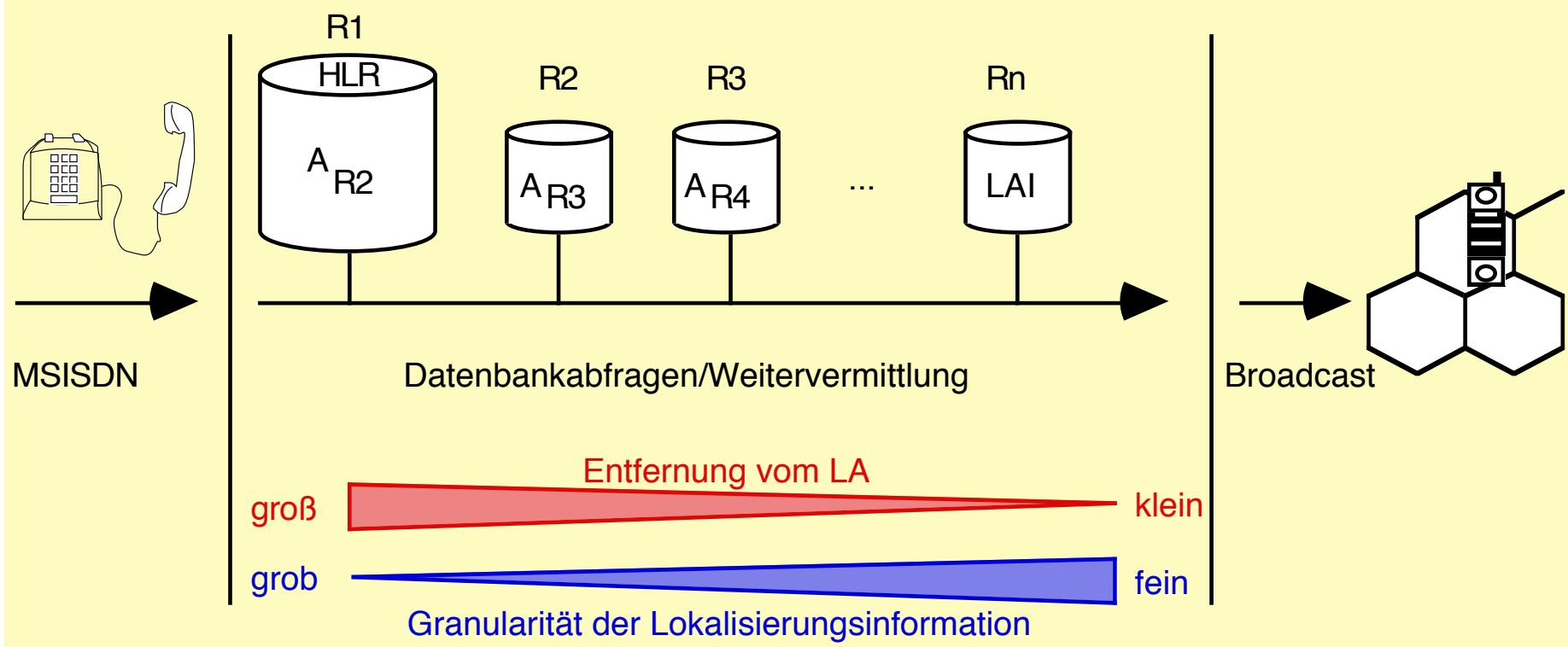
Location Management

- 2-staged approach
 - Change of Location Area (LA) changes VLR entry
 - VLR serves geographically limited area (VLR-Area)
 - Rare changes of VLR-Area changes HLR entry
 - Reduced signaling costs in wide area network
 - Tradeoff: Delayed call setup (mobile terminated)

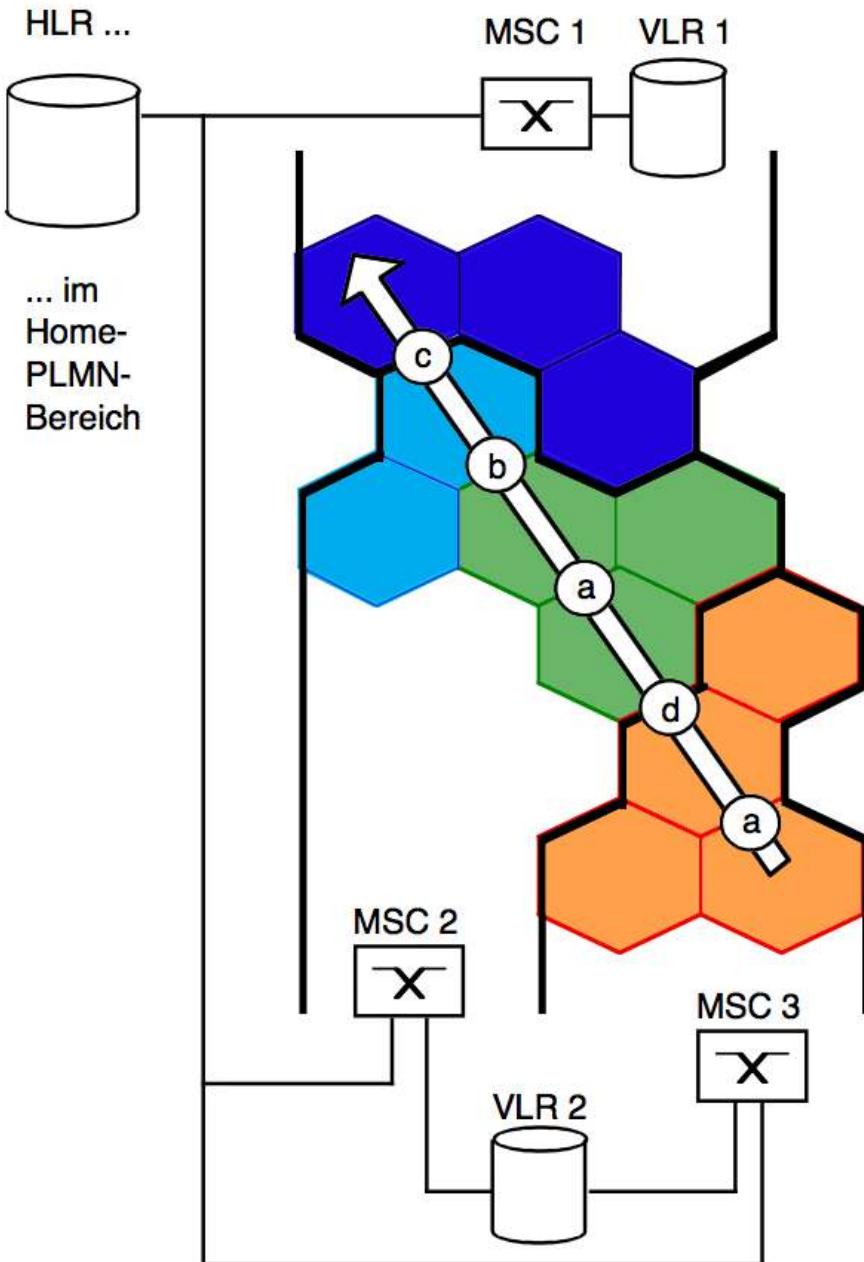


Location Management

- Multi-staged storage
 - Many proposals for 3rd Generation Systems (UMTS), never realized in the field
 - Variations: Hierarchical storage, Forwarding strategies



Location Updating Situations



Legend:

- a) Change of radio cell
- b) Change of LA
- c) Change of VLR/MSC area
- d) Change of MSC area

LA 1 (belongs to MSC 1 and VLR 1)

LA 2 (belongs to MSC 2 and VLR 2)

LA 3 (belongs to MSC 2 and VLR 2)

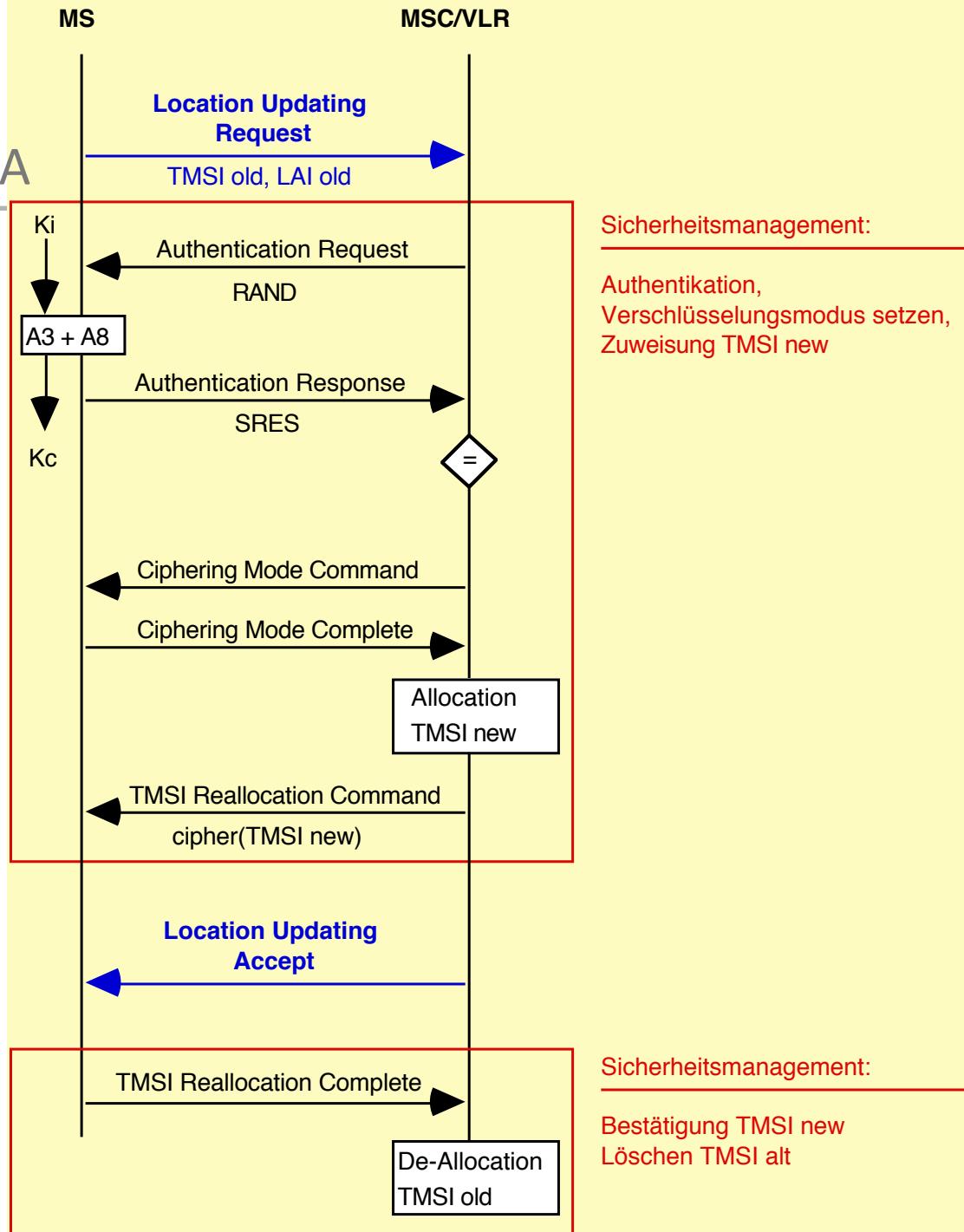
LA 4 (belongs to MSC 3 and VLR 2)

Movement of MS

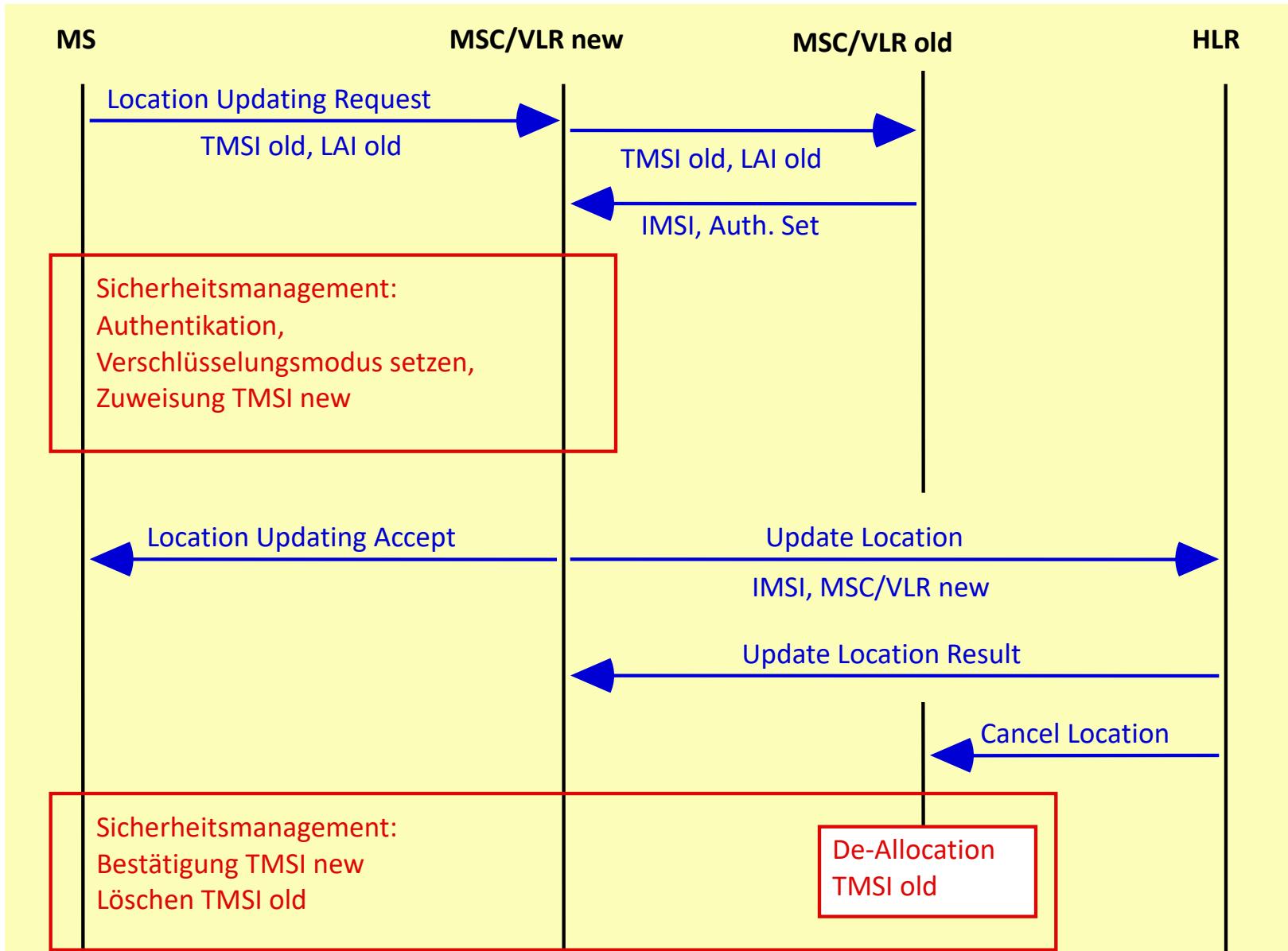
Radio cell

Location Updating: New LA

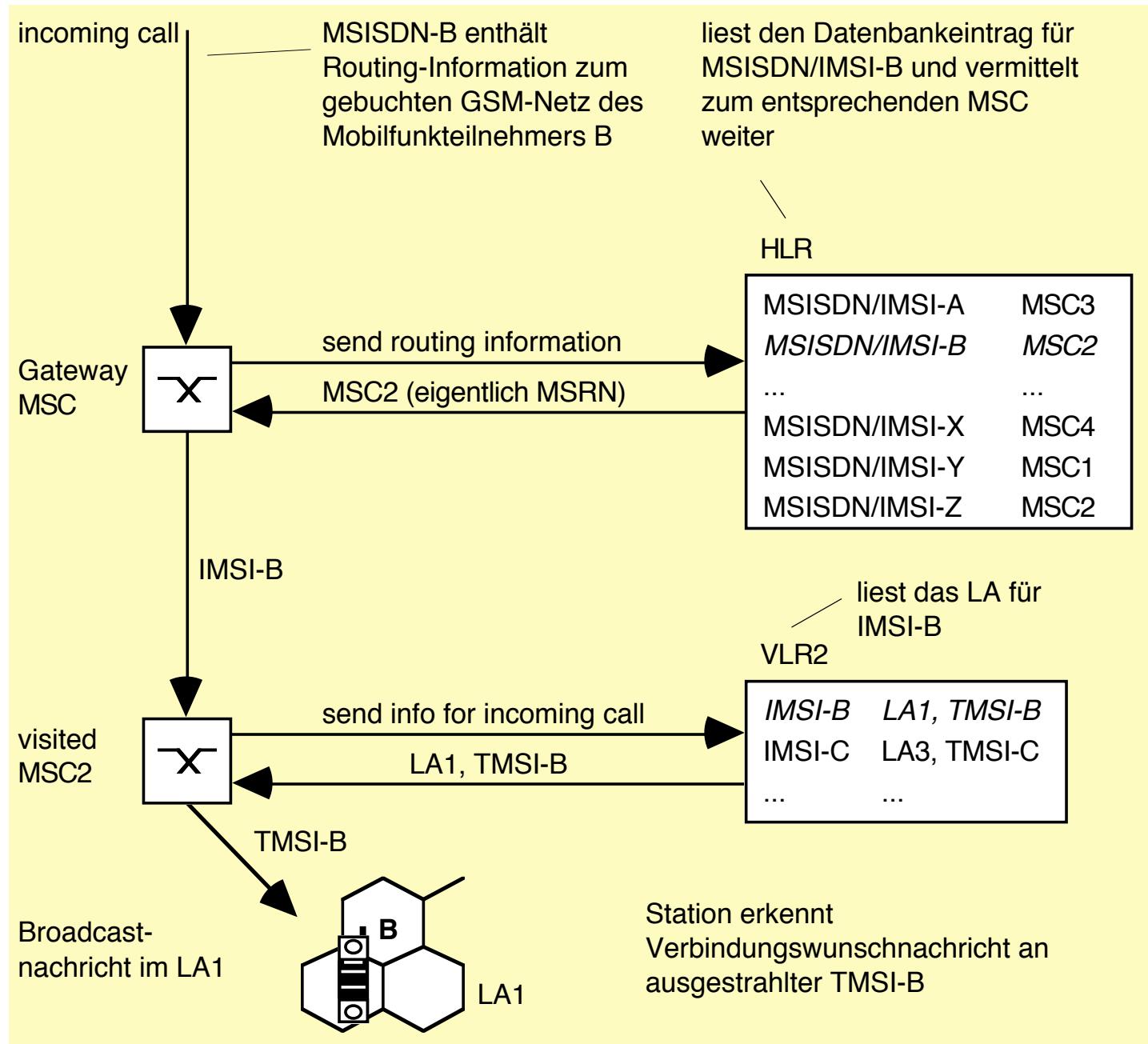
- New LA, old VLR (TMSI found)
 - Location Updating Request (TMSI, LAI)old
 - Security management
 - Authentication
 - Ciphering Mode
 - TMSI Reallocation
 - Location Updating Accept

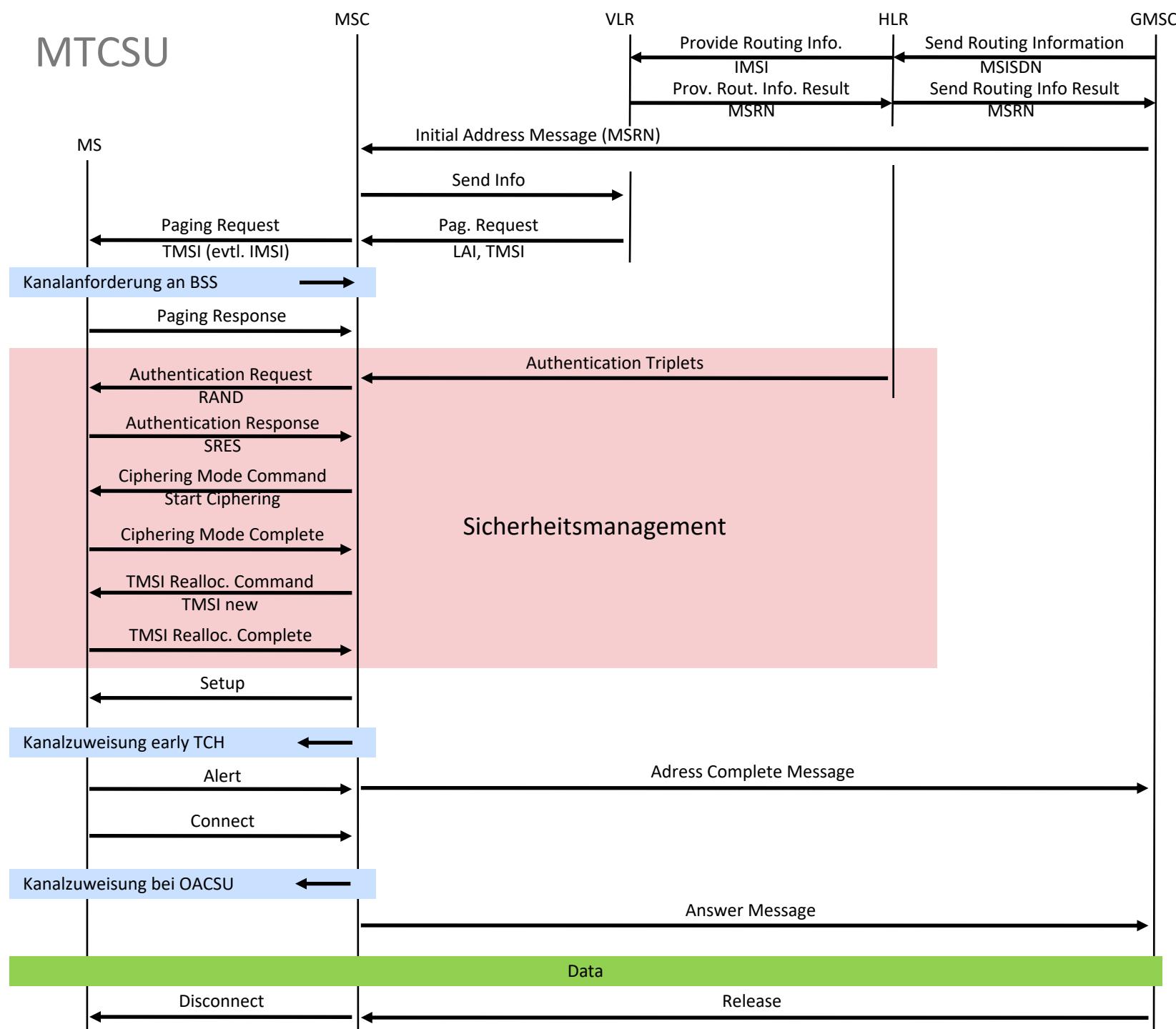


Location Updating: New VLR area

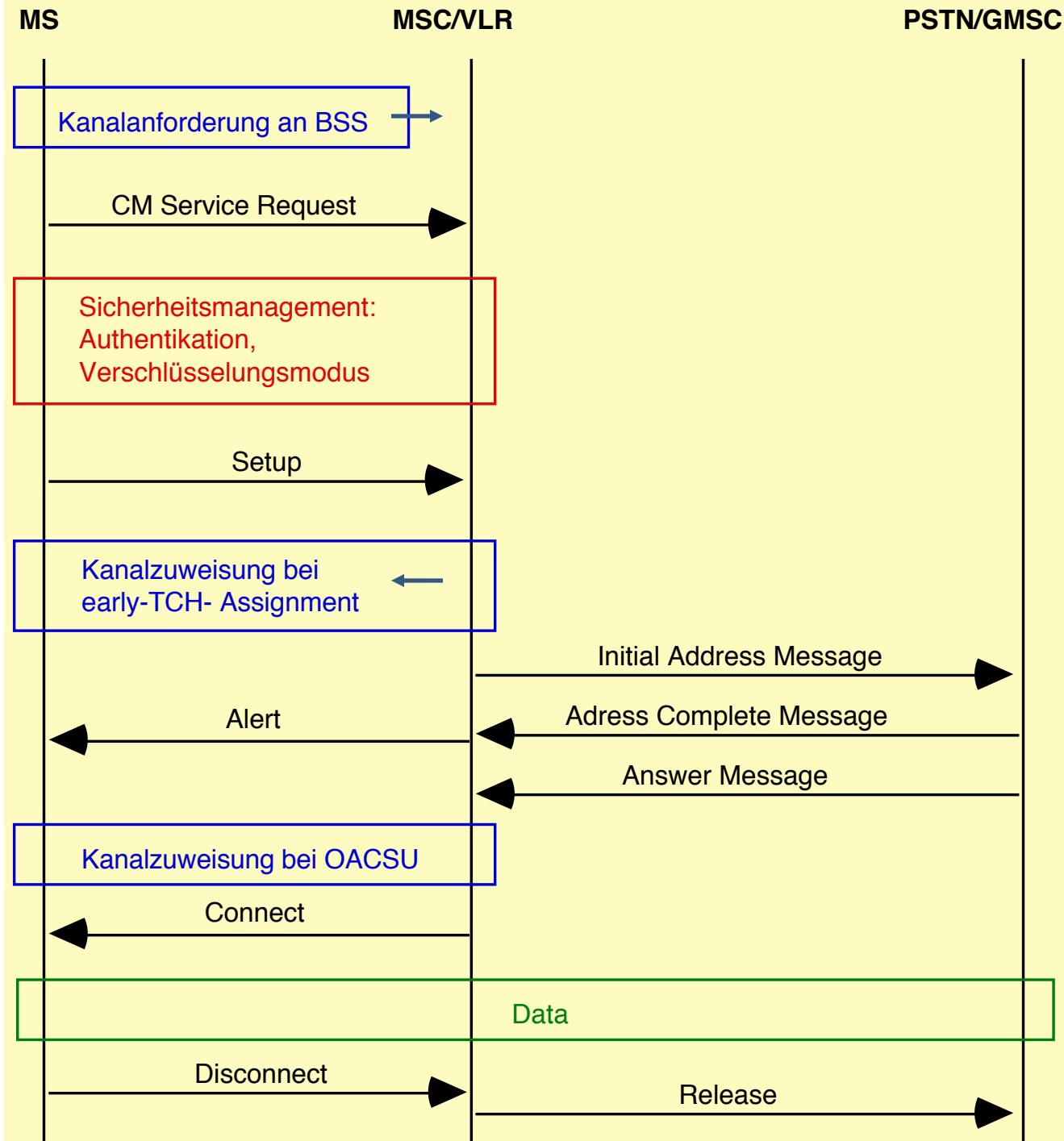


Mobile Terminated Call Setup (MTCSU)





Mobile Originated Call Setup



Message format GSM 04.08

■ Protocol discriminator

4 3 2 1 bit number

0 0 1 1 call control, packet-mode, connection control and call related SS msgs

0 1 0 1 mobility management messages

0 1 1 0 radio resources management messages

1 0 0 1 short message service messages

1 0 1 1 non call related SS messages

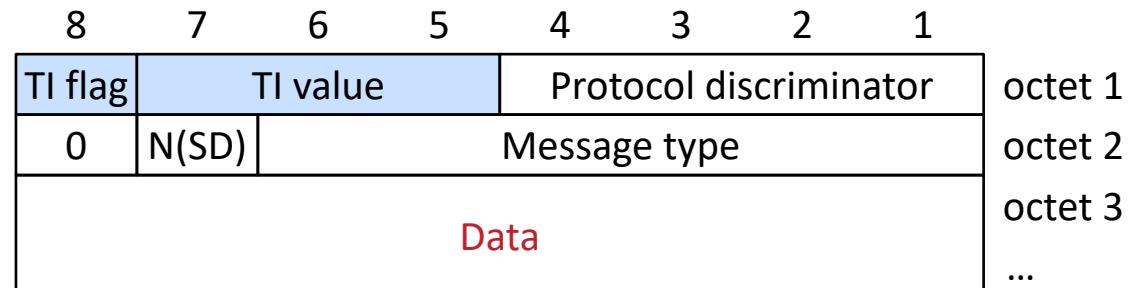
1 1 1 1 reserved for tests procedures

All other values are reserved

8	7	6	5	4	3	2	1				
TI flag			TI value		Protocol discriminator			octet 1			
0		N(SD)	Message type					octet 2			
Data								octet 3 ...			

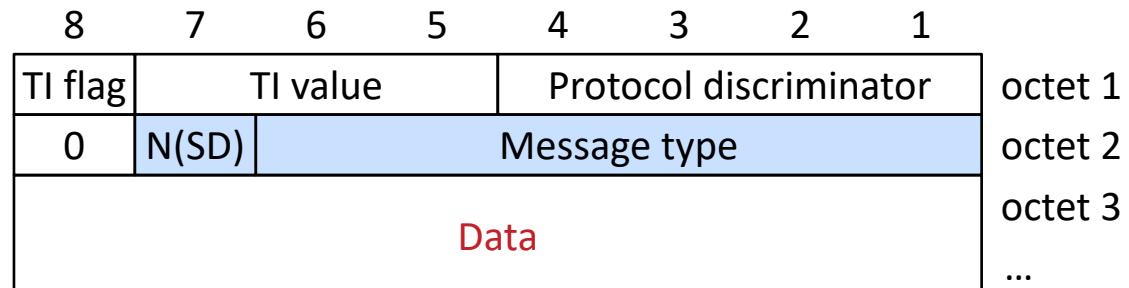
Message format GSM 04.08

- Transaction identifier (TI)
 - Used for distinction of parallel activities of MS
 - TI flag:
 - 0: message sent from the originated TI side
 - 1: message sent to the originated TI side
- TI value
 - Number 000...110 (bin: 0...6)
 - 111 reserved



Message format GSM 04.08

- 3 Classes:
 - Radio resources management
 - Mobility management
 - Call control
- N(SD)
 - Sequence number or Extension Bit



Message type (1)

- Radio resources management (1)

8	7	6	5	4	3	2	1	bit number
<hr/>								
0 0 1 1 1 - - - Channel establishment messages								
0	1	1	1	1	1	1	1	ADDITIONAL ASSIGNMENT
1	1	1	1	1	1	1	1	IMMEDIATE ASSIGNMENT
0	0	1	1	1	1	1	1	IMMEDIATE ASSIGNMENT EXTENDED
0	1	0	1	1	1	1	1	IMMEDIATE ASSIGNMENT REJECT
0 0 1 1 0 - - - Ciphering messages								
1	0	1	1	0	1	0	1	CIPHERING MODE ASSIGNEMT
0	1	0	1	0	1	0	1	CIPHERING MODE COMPLETE
0 0 1 0 1 - - - Handover messages								
1	1	0	1	0	1	0	1	ASSIGNEMT COMMAND
0	0	0	1	0	1	0	1	ASSIGNEMT COMPLETE
1	1	1	0	1	1	0	1	ASSIGNMENT FAILURE
0	1	1	0	1	1	0	1	HANOVER COMMAND
1	0	0	1	0	1	0	1	HANOVER COMPLETE
0	0	0	1	0	1	0	1	HANOVER FAILURE
1	0	1	0	1	0	1	0	PHYSICAL INFORMATION
0 0 0 0 1 - - - Channel release messages								
1	0	1	0	1	0	1	0	CHANNEL RELEASE
0	1	0	0	1	0	0	1	PARTIAL RELEASE
1	1	1	0	0	1	0	0	PARTIAL RELEASE COMPLETE

...

Message type (1)

- Radio resources management (2)

8	7	6	5	4	3	2	1	bit number

...								
0 0 1 0 0 - - - Paging messages								
0 0 1 PAGING REQUEST TYPE 1								
0 1 0 PAGING REQUEST TYPE 2								
1 0 0 PAGING REQUEST TYPE 3								
1 1 1 PAGING RESPONSE								
0 0 0 1 1 - - - System information messages								
0 0 1 SYSTEM INFORMATION TYPE 1								
0 1 0 SYSTEM INFORMATION TYPE 2								
0 1 1 SYSTEM INFORMATION TYPE 3								
1 0 0 SYSTEM INFORMATION TYPE 4								
1 0 1 SYSTEM INFORMATION TYPE 5								
1 1 0 SYSTEM INFORMATION TYPE 6								
0 0 0 1 0 - - - Miscellaneous messages								
0 0 0 CHANNEL MODE MODIFY								
0 1 0 RR-STATUS								
1 1 1 CHANNEL MODE MODIFY ACKNOWLEDGE								
1 0 0 FREQUENCY REDEFINITION								
1 0 1 MEASUREMENT REPORT								
1 1 0 CLASSMARK CHANGE								

Message type (2)

- Mobility management

- Bits 7 and 8 (value: 00) reserved as extension bits
- Bit 7: mobile originated only: 1, if sequence number is sent

8	7	6	5	4	3	2	1	bit number
<hr/>								
0 x 0 0 — — — Registration messages								
0	0	0	1					IMSI DETACH INDICATION
0	0	1	0					LOCATION UPDATING ACCEPT
0	1	0	0					LOCATION UPDATING REJECT
1	0	0	0					LOCATION UPDATING REQUEST
0 x 0 1 — — — Security messages								
0	0	0	1					AUTHENTICATION REJECT
0	0	1	0					AUTHENTICATION REQUEST
0	1	0	0					AUTHENTICATION RESPONSE
1	0	0	0					IDENTITY REQUEST
1	0	0	1					IDENTITY RESPONSE
1	0	1	0					TMSI REALLOCATION COMMAND
1	0	1	1					TMSI REALLOCATION COMPLETE
0 x 1 0 — — — Connection management messages								
0	0	0	1					CM SERVICE ACCEPT
0	0	1	0					CM SERVICE REJECT
0	1	0	0					CM SERVICE REQUEST
1	0	0	0					CM REESTABLISHMENT REQUEST
0 x 1 1 — — — Connection management messages								
0	0	0	1					MM STATUS

Message type (3)

- Call control (1)
 - Bits 7 and 8 (value: 00) reserved as extension bits
 - Bit 7: mobile originated only: 1, if sequence number is sent
 - Nationally specific messages: next octets contain message

8 7 6 5 4 3 2 1	bit number
<hr/>	
0 x 0 0 0 0 0 0	Escape to nationally specific message types
0 x 0 0 - - -	Call establishment messages
0 0 0 1	ALERTING
1 0 0 0	CALL CONFIRMED
0 0 1 0	CALL PROCEEDING
0 1 1 1	CONNECT
1 1 1 1	CONNECT ACKNOWLEDGE
1 1 1 0	EMERGENCY SETUP
0 0 1 1	PROGRESS
0 1 0 1	SETUP
0 x 0 1 - - -	Call information phase messages
0 1 1 1	MODIFY
1 1 1 1	MODIFY COMPLETE
0 0 1 1	MODIFY REJECTED
0 0 0 0	USER INFORMATION

...

Message type (3)

- Call control (2)
 - Bits 7 and 8 (value: 00) reserved as extension bits
 - Bit 7: mobile originated only: 1, if sequence number is sent

8	7	6	5	4	3	2	1	bit number

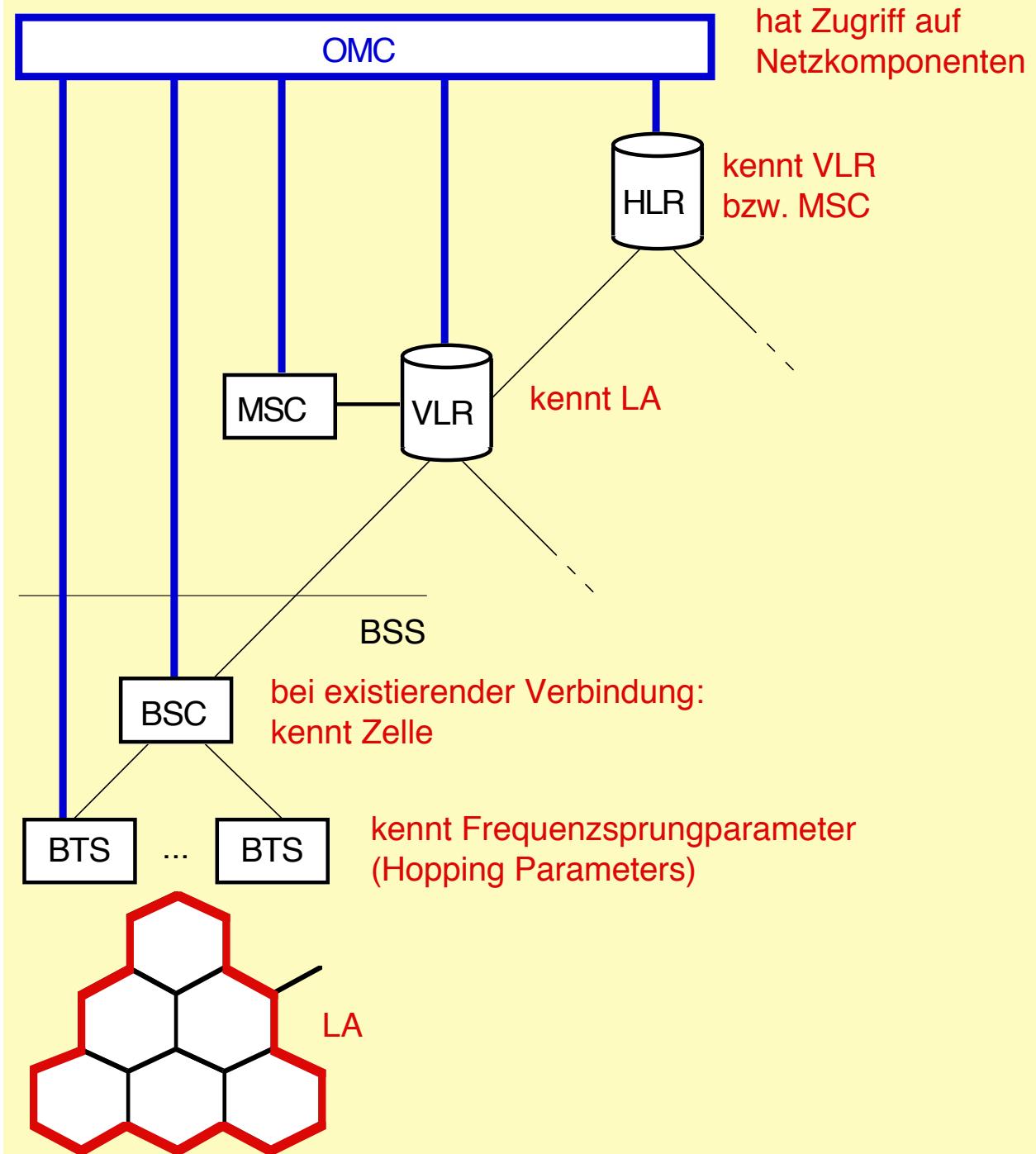
...								
0								
x	1	0	—	—	—	—	—	Call clearing messages
0	1	0	1					DISCONNECT
1	1	0	1					RELEASE
1	0	1	0					RELEASE COMPLETE
0	x	1	1	—	—	—	—	Miscellaneous messages
1	0	0	1					CONGESTION CONTROL
1	1	1	0					NOTIFY
1	1	0	1					STATUS
0	1	0	0					STATUS ENQUIRY
0	1	0	1					START DTMF
0	0	0	1					STOP DTMF
0	0	1	0					STOP DTMF ACKNOWLEDGE
0	1	1	0					START DTMF ACKNOWLEDGE
0	1	1	1					START DTMF REJECT

Movement profiling in GSM

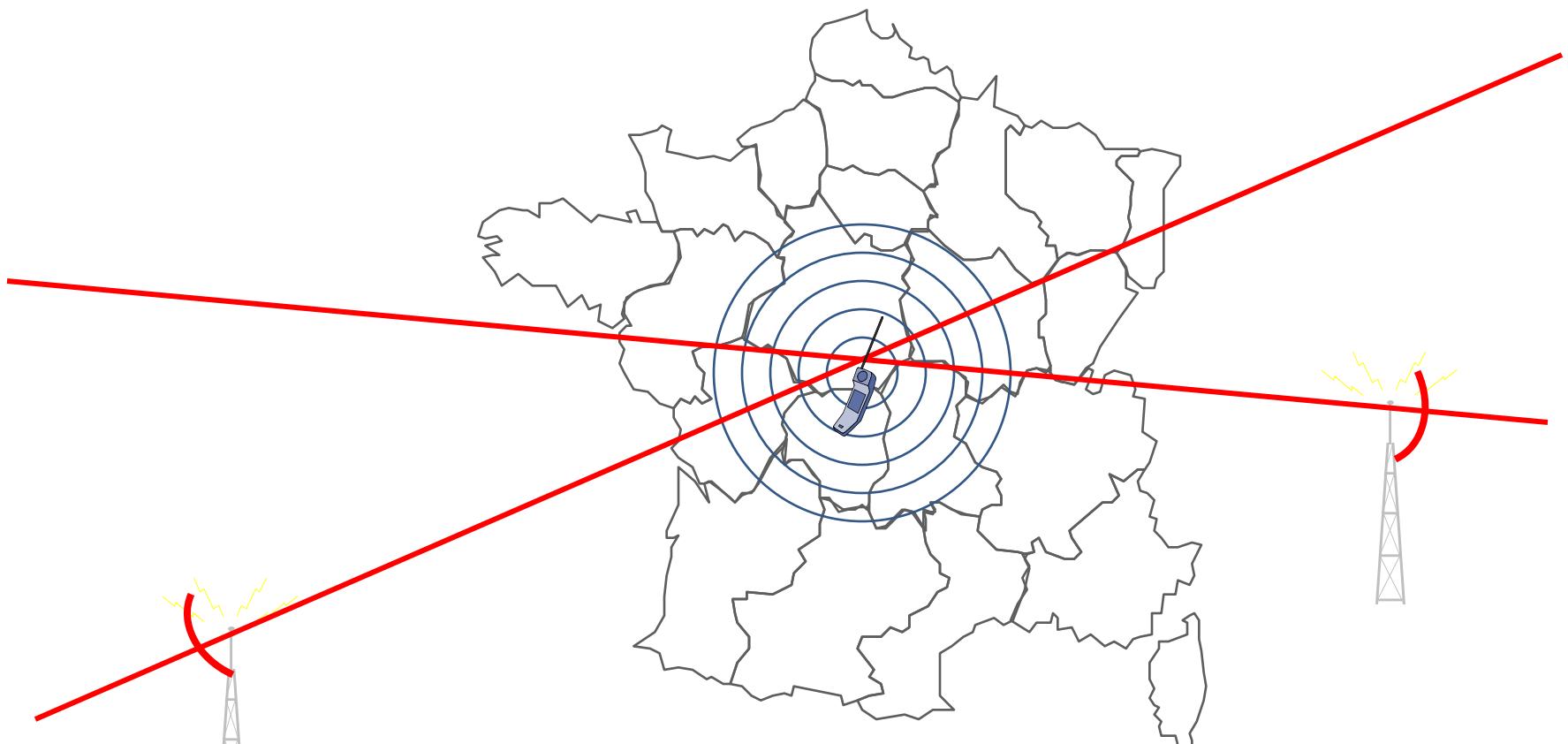
- **Variants:**
 - Access HLR and VLR data (insiders only)
 - Direction finding (*German: »Peilung«*)

- **Protection:**
 - Privacy protection of database entries
 - Direct Sequence Spread Spectrum

Access HLR and VLR data

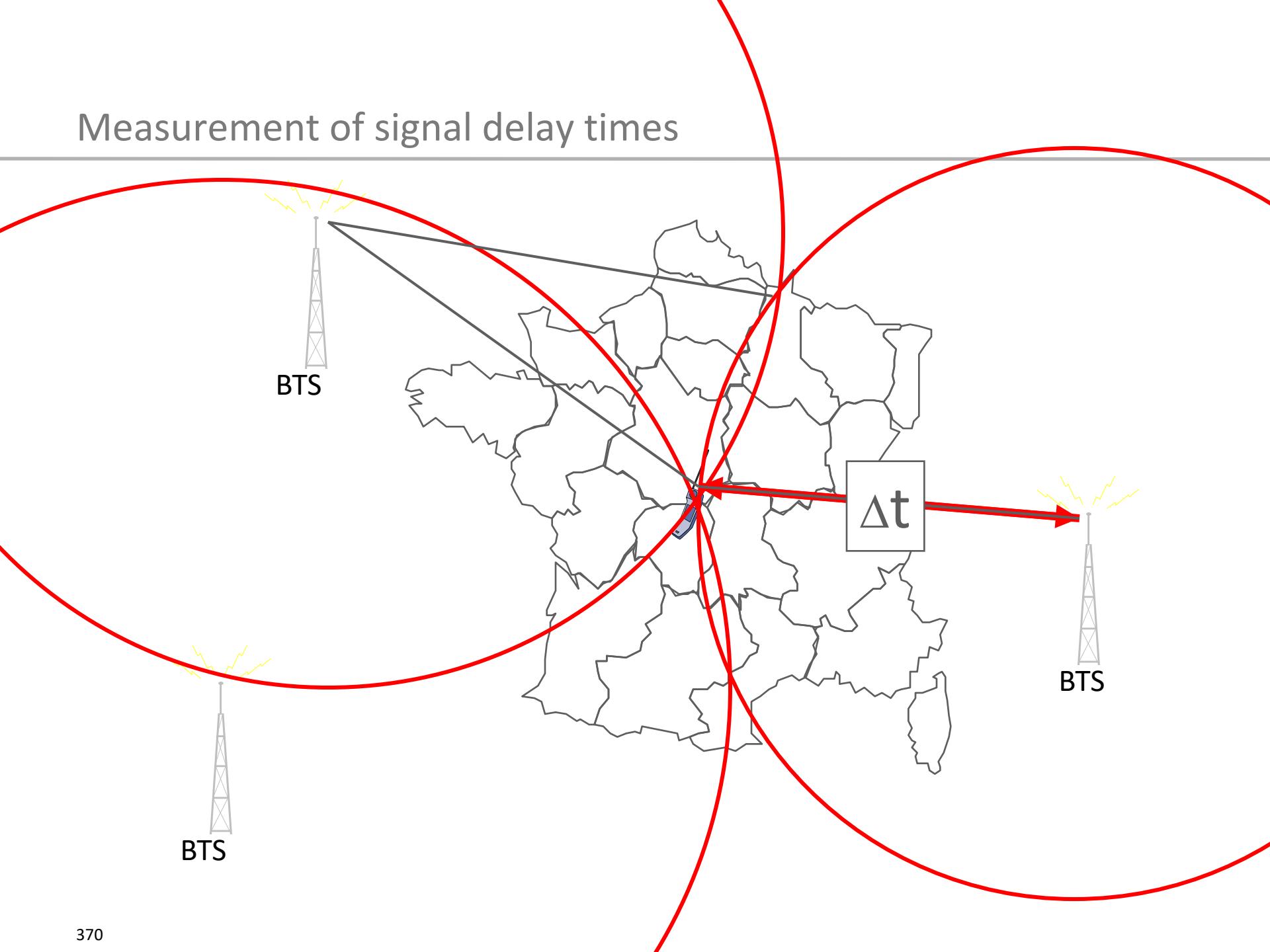


Direction finding with directional antennas



Richtantennen
notwendig

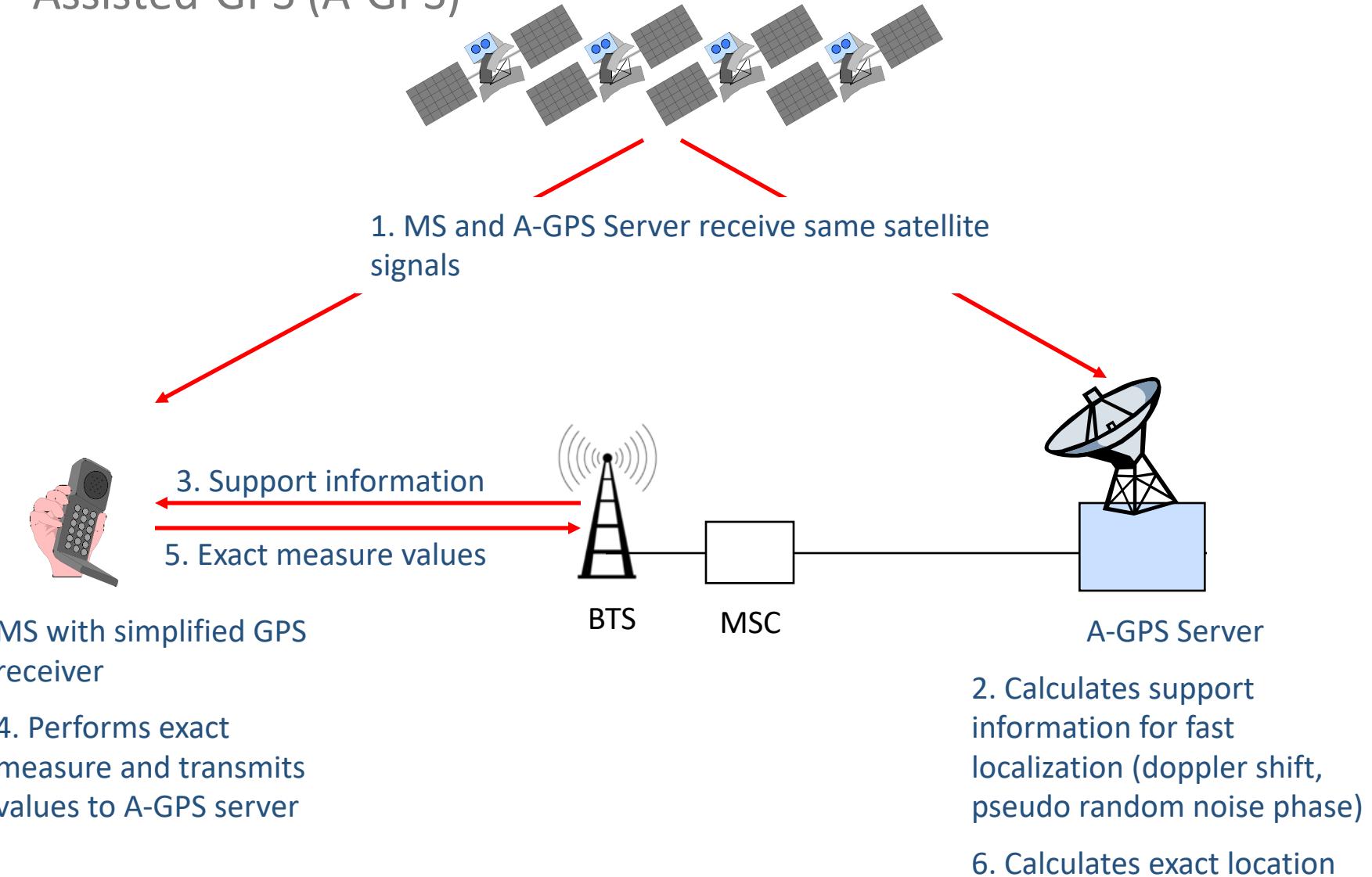
Measurement of signal delay times



Location Based Services

- Terminal-based locating
 - Global Positioning System (GPS)
 - Accuracy: 10...100 m
 - Location time: up to 30 sec
 - Assisted-GPS (A-GPS)
 - GPS signals re-broadcasted by BTS
 - Increased location speed (and accuracy)
 - Observed Time Difference (OTD)
 - BTS1 ... BTS3 send a location signal
 - Received after Δt_1 , Δt_2 and Δt_3 by MS
 - If $\Delta t_i == \Delta t_j$ then OTD=0

Assisted-GPS (A-GPS)

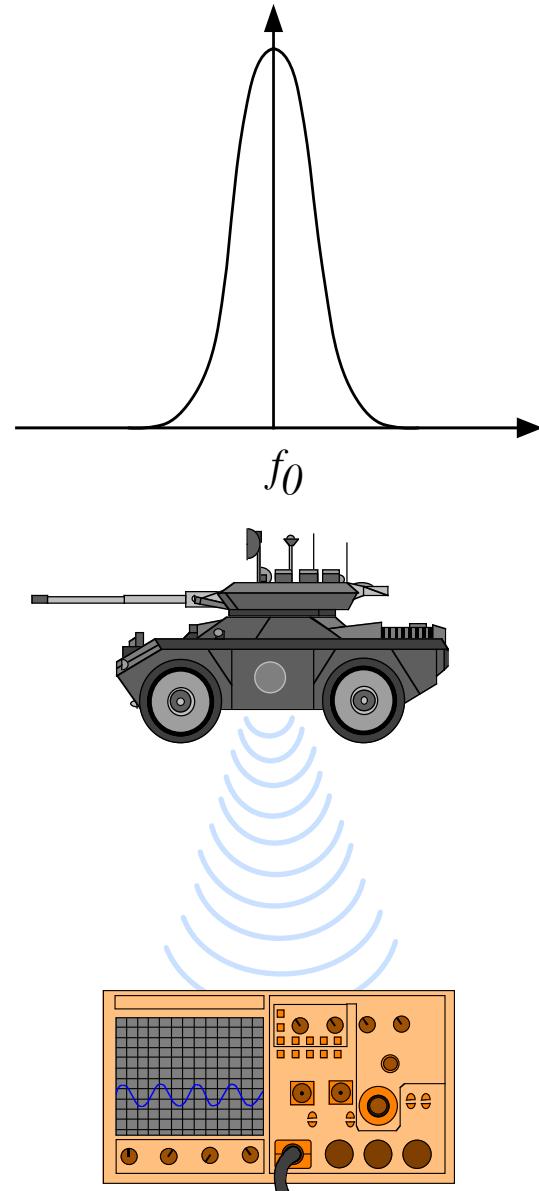


Location Based Services

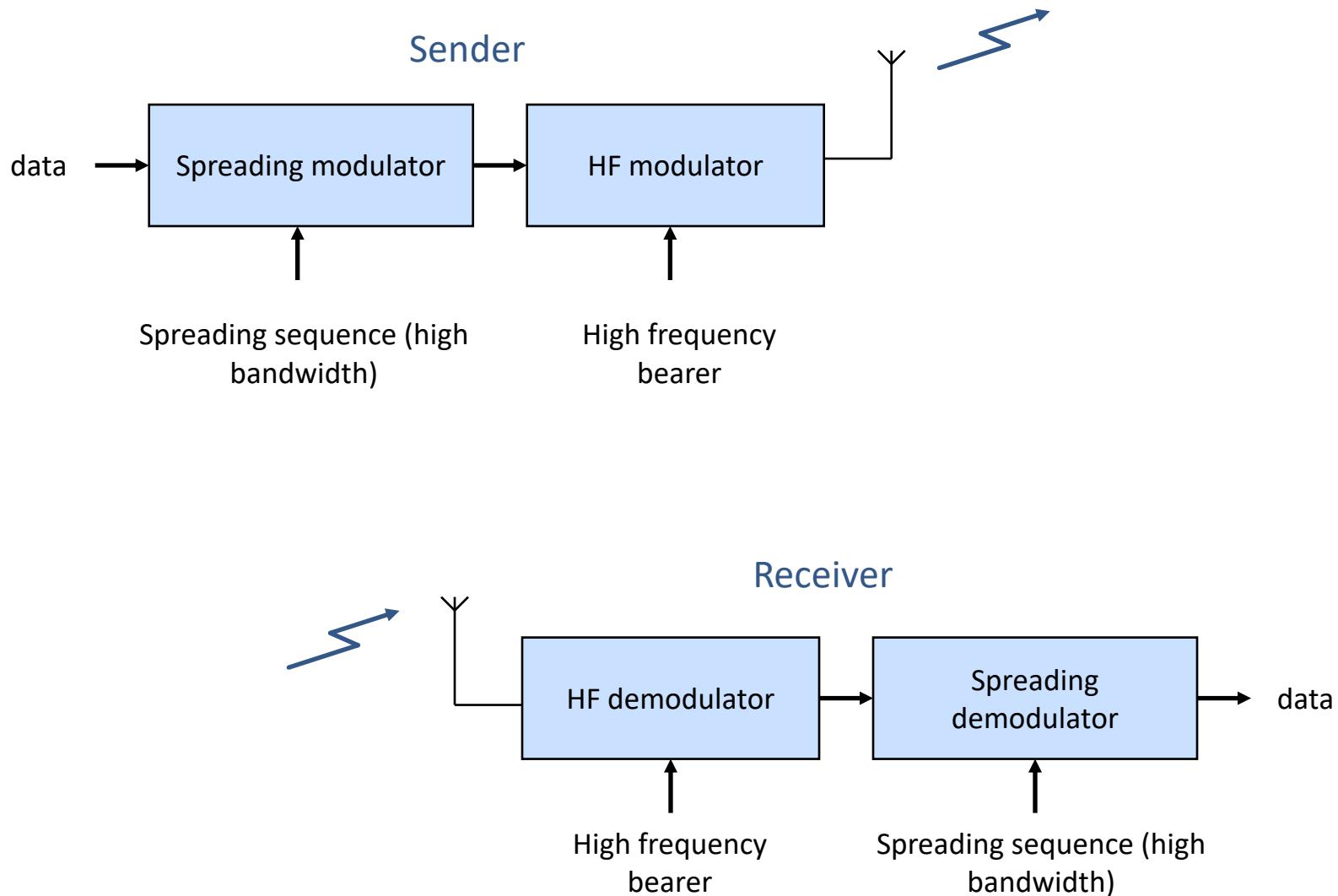
- Network-based locating
 - Time of Arrival (TOA)
 - Mobile station sends signal
 - BTS receive signal after Δt_i ($i=1,2,3$)
 - Cell of Origin (COO)
 - Cell-ID is associated with geographic location
 - Accuracy: 100 m ... 35 km

Spread Spectrum Systems

- Radio communication between military divisions
 - Sender sends on frequency f_0 with bandwidth B
- Problems:
 - Spectrum analyzer detects energy around f_0 and directional antennas locate source of signal
 - Jammer may interfere communication

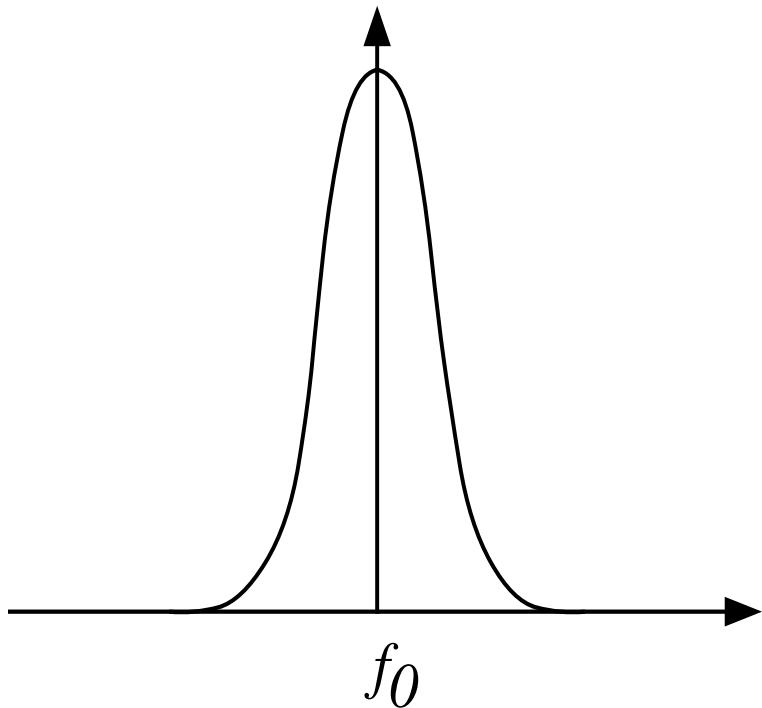


Transmision model Spread Spectrum Systems

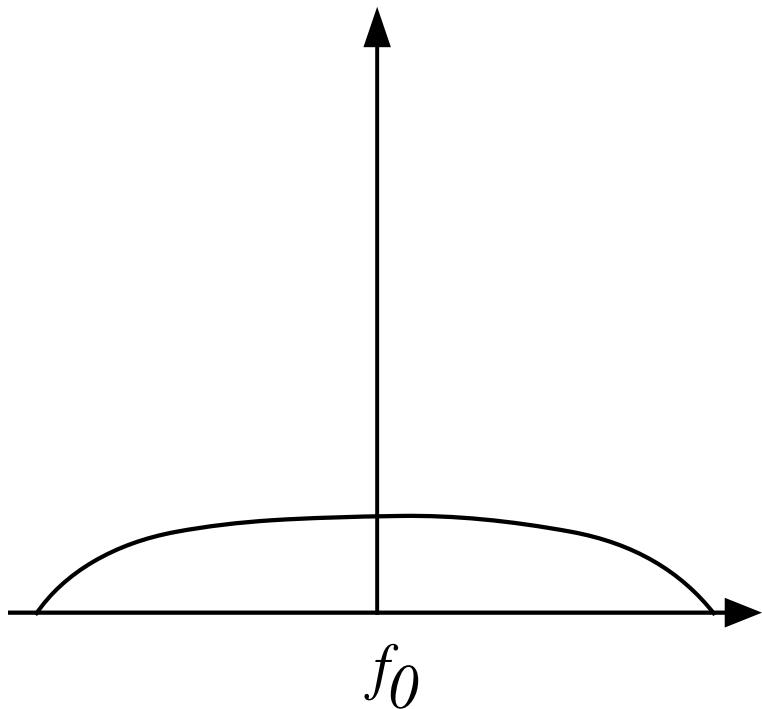


Spreading

- Data is modulated with high-bandwidth spreading sequence:
 - Walsh functions (orthogonal codes)
 - Pseudo-Noise-Sequence (PN-Code)



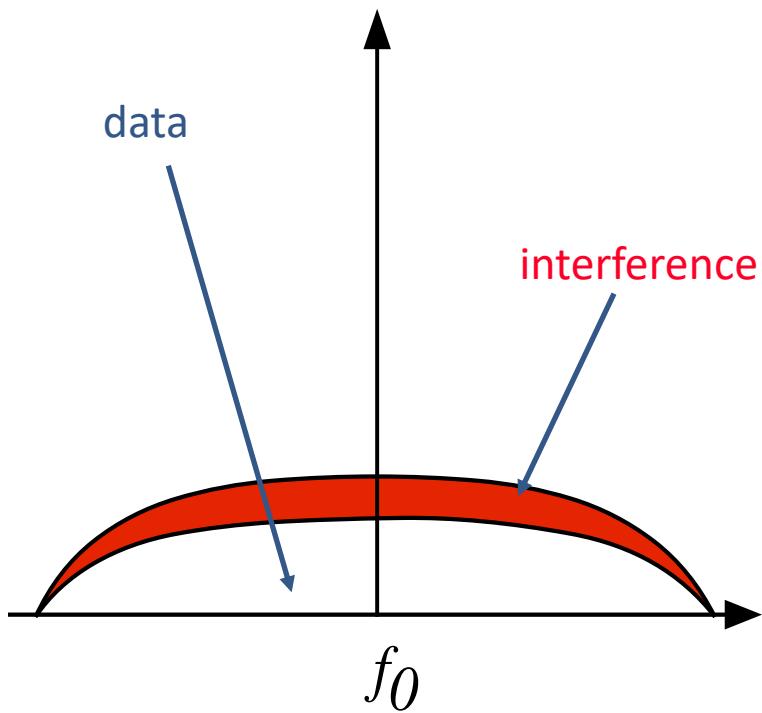
Spreading



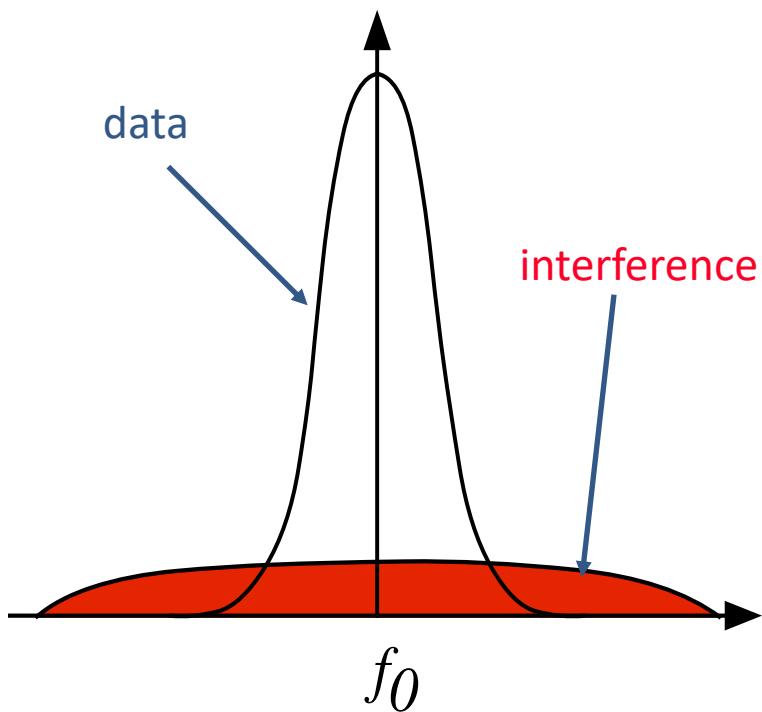
- Data is modulated with high-bandwidth spreading sequence:
 - Walsh functions (orthogonal codes)
 - Pseudo-Noise-Sequence (PN-Code)
- Spectral spreading of signal
- Dispersion of energy on a large frequency spectrum

De-Spreading

- Spread data interfered by (random) noise



De-Spreading

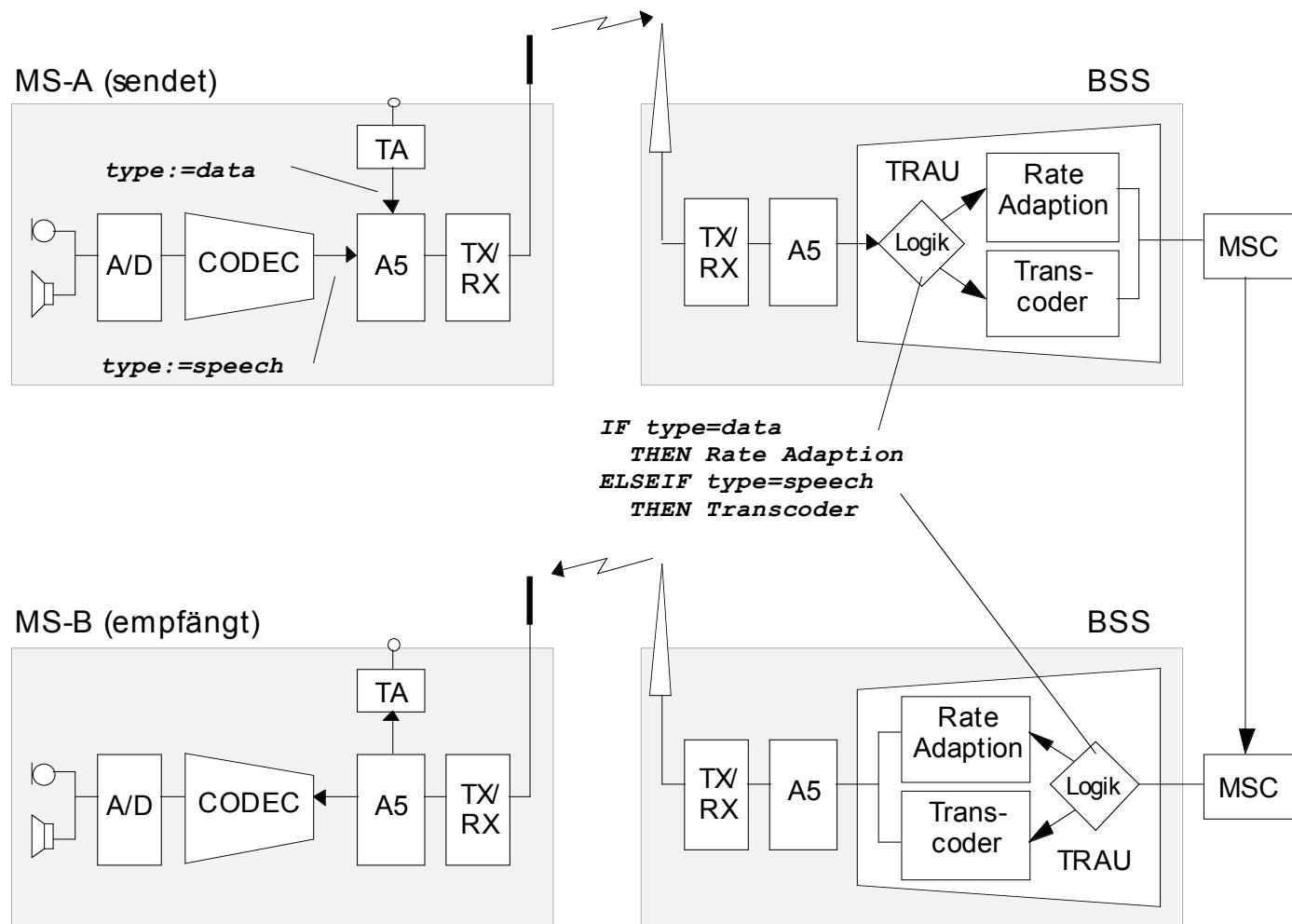


- Spread data interfered by (random) noise
- ↓
- Spectral spreading of noise
 - De-spreading of data

Missing end-to-end-Services in GSM

- Speech channels of GSM are not bit transparent channels
 - Lossy compression of speech channels
- Use data channel for additional end-to-end encryption
 - As an external add-on (e.g. GSM TopSec Med)
 - As integrated service (e.g. GSM TopSec GSM)
 - Both is not GSM standards conform add-on
 - Users need compatible devices or software on MS

Signaling of channel type (speech, data) in GSM



MS
BSS
A/D
CODEC
TA

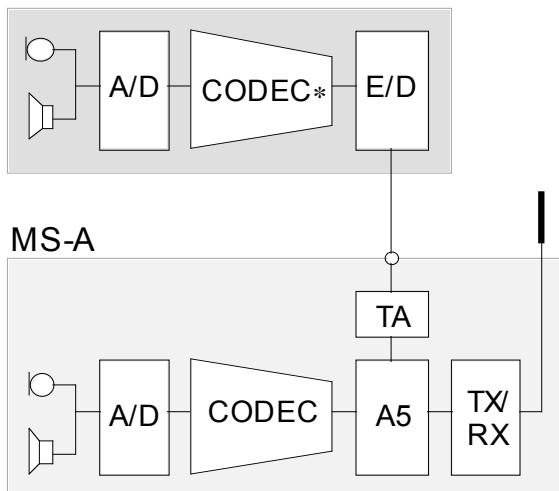
Mobile Station
Base Station Subsystem
Analog-Digital-Converter
Speech Coder/Decoder
Terminal Adaption

A5
TX/RX
TRAU
MSC

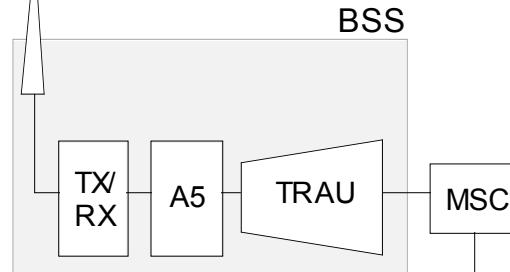
GSM Link Encryption
Transmitter/Receiver
Transcoder/Rate Adaption Unit
Mobile Switching Centre

Bit transparent data channel for end-to-end speech encryption

Zusatz zu MS-A



MS-A

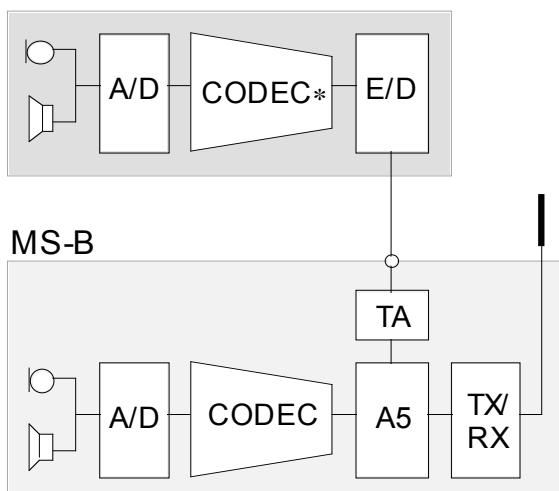


Example:

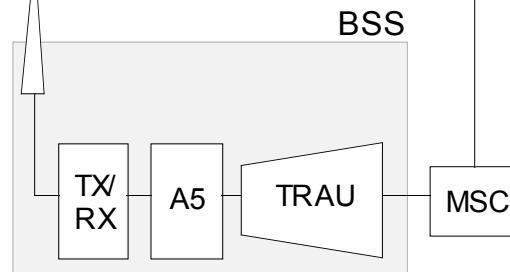
TopSec MED

(Rohde&Schwarz): external device bluetooth connected to mobile phone

Zusatz zu MS-B



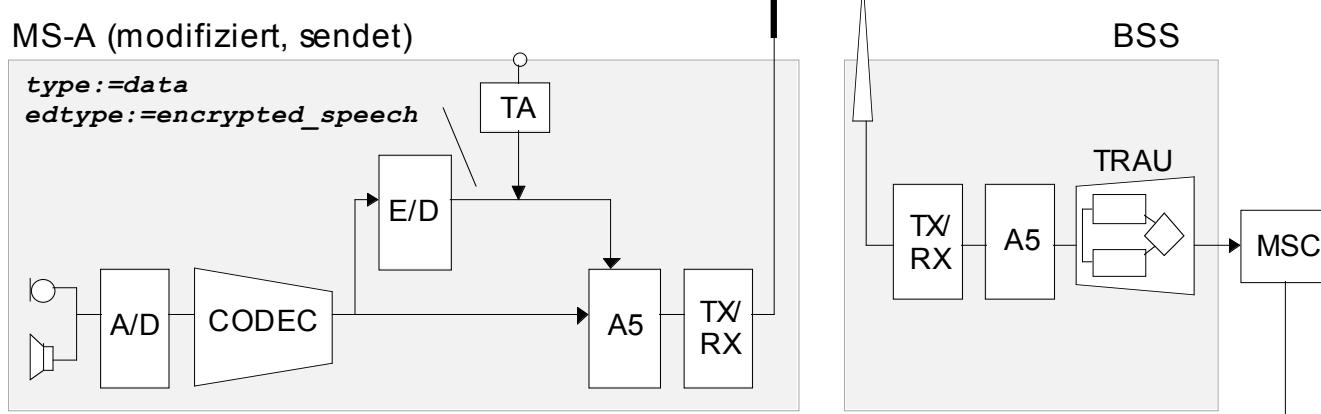
MS-B



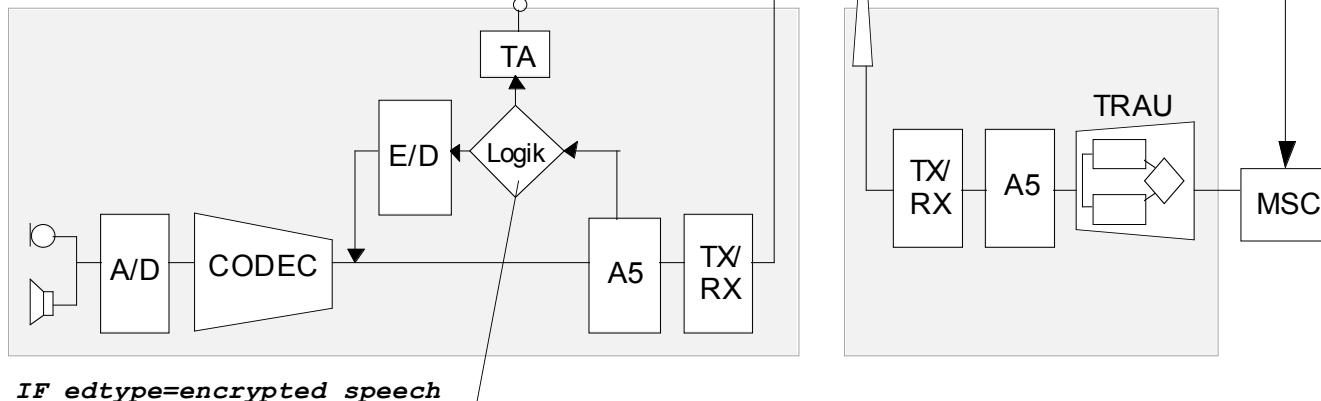
Bit transparent data channel – internal use for end-to-end enc.

MS-A (modifiziert, sendet)

*type := data
edtype := encrypted_speech*



MS-B (modifiziert, empfängt)



*IF edtype=encrypted_speech
THEN E/D
ELSE TA*

Example:

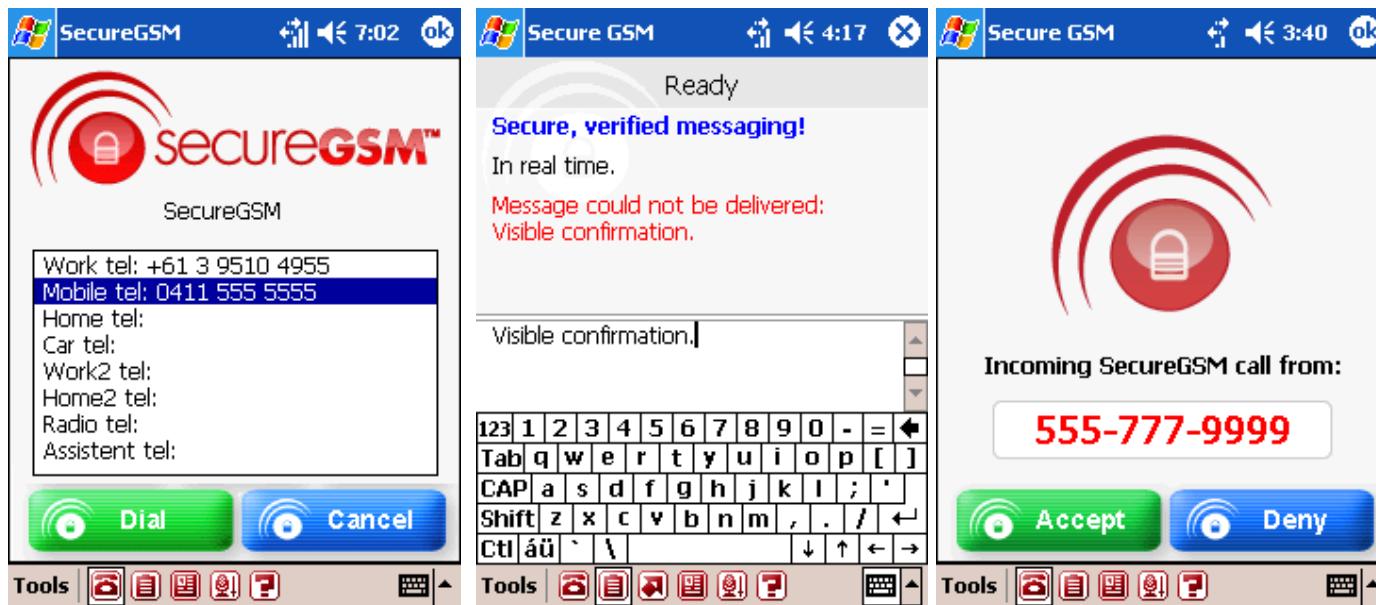
TopSec GSM

(Rohde&Schwarz): modified Siemens S35i with Crypto processor, 128 bit encryption



Software solutions for end-to-end encryption

- Example: **SecureGSM** · <http://www.securegsm.com>
 - For Windows Mobile Smartphones
 - Bit transparent data channel used
 - Asymmetric key agreement (»4Kbit«)
 - Triple encryption with AES, Serpent and Towfish with triple 256 bit session keys



Screenshots: <http://www.securegsm.com>

Summary of security problems in GSM

- Hard
 - Weak link encryption protects against outsiders only
 - No bit transparent speech channels → no end-to-end encryption
 - Location finding for insiders possible
 - Mutual authentication is missing

- Further
 - Symmetric encryption
 - No anonymous network usage possible
 - Trust into accounting is necessary

Security functions of further mobile Systems

UMTS and LTE

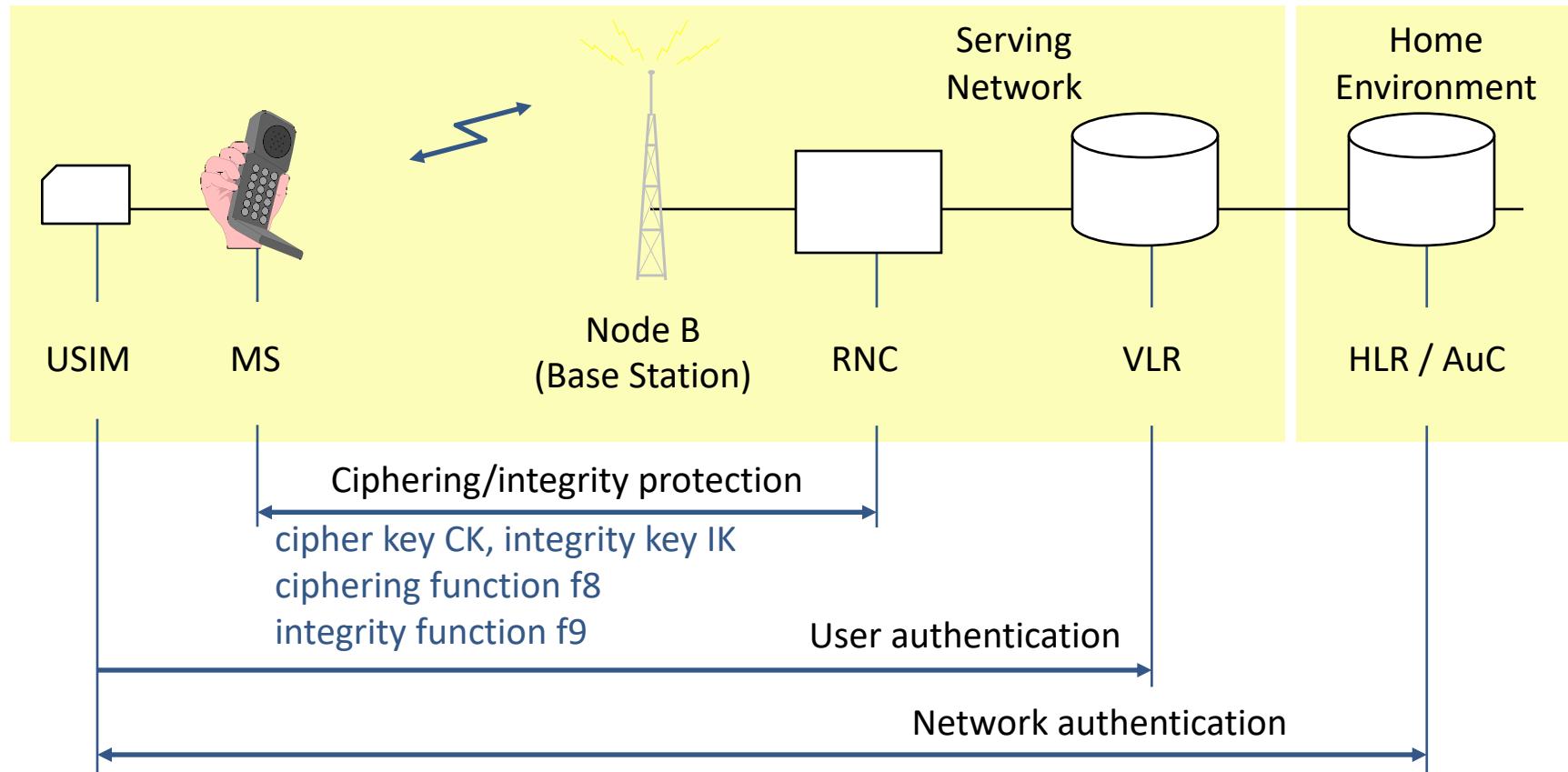
Bluetooth security

WiFi security

Universal mobile telecommunication system (UMTS)

- Security functions of UMTS → inspired by GSM security functions
- From GSM
 - Subscriber identity confidentiality (TMSI)
 - Subscriber authentication
 - Radio interface encryption
 - SIM card (now called USIM)
 - Authentication of subscriber towards SIM by means of a PIN
 - Delegation of authentication to visited network
 - No need to adopt standardized authentication algorithms
- Additional UMTS security features
 - Enhanced UMTS authentication and key agreement mechanism
 - Integrity protection of signaling information (prevents false-base-station attacks)
 - New ciphering / key agreement / integrity protection algorithms
 - ... and a few minor features

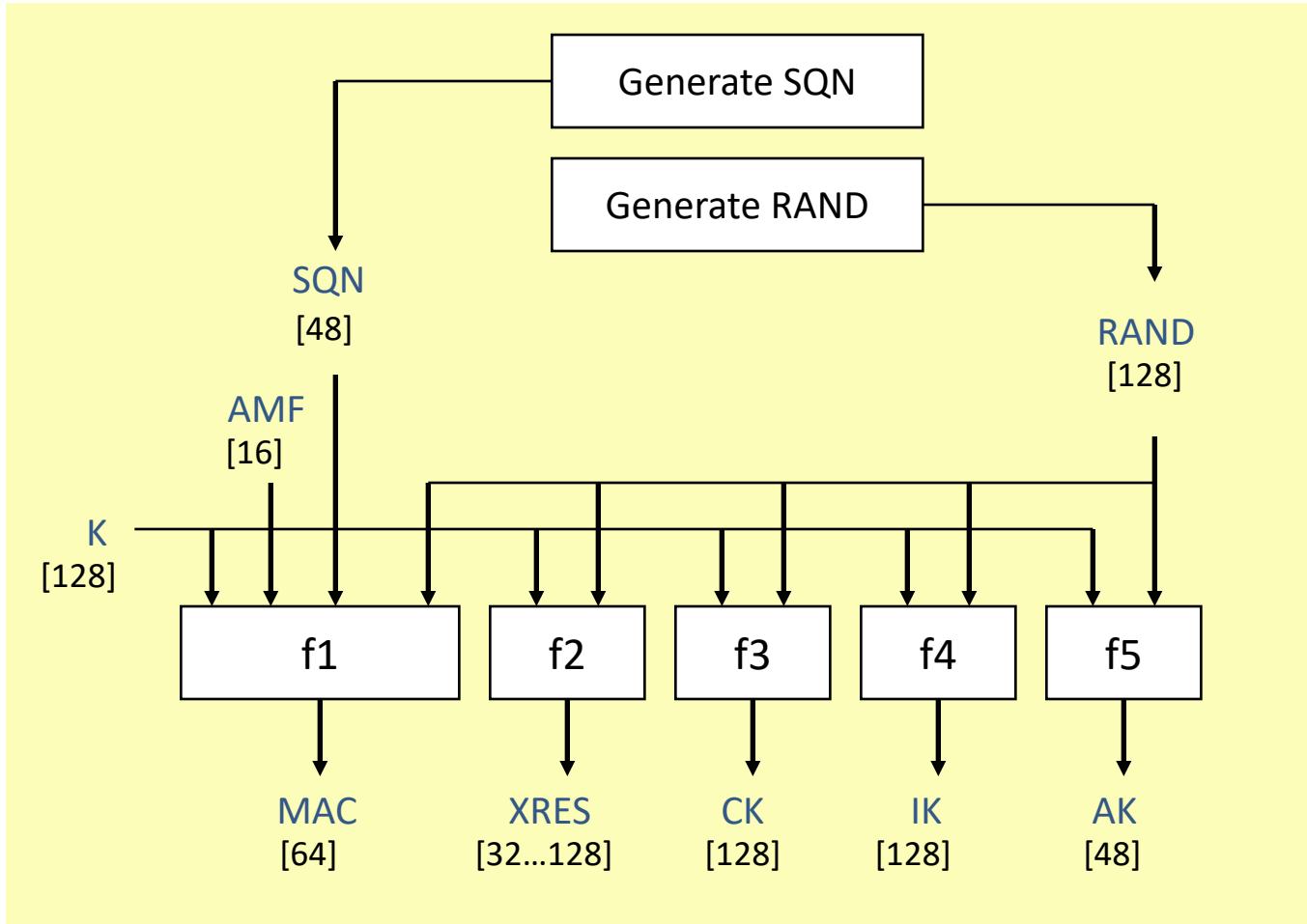
UMTS Security Architecture



USIM	UMTS Subscriber Identity Module
MS	Mobile Station
RNC	Radio Network Controller
VLR	Visitor Location Reg.
HLR	Home Location Register
AuC	Authentication Centre

authentication key K,
authentication function f1, f2
key generation function f3, f4, f5
sequence number management SQN

Generation of authentication vectors (network side)



$$\begin{aligned}
 \text{AUTN} &:= \text{SQN} \oplus \text{AK} \parallel \text{AMF} \parallel \text{MAC} \\
 \text{AV} &:= \text{RAND} \parallel \text{XRES} \parallel \text{CK} \parallel \text{IK} \parallel \text{AUTN}
 \end{aligned}$$

Abbreviations

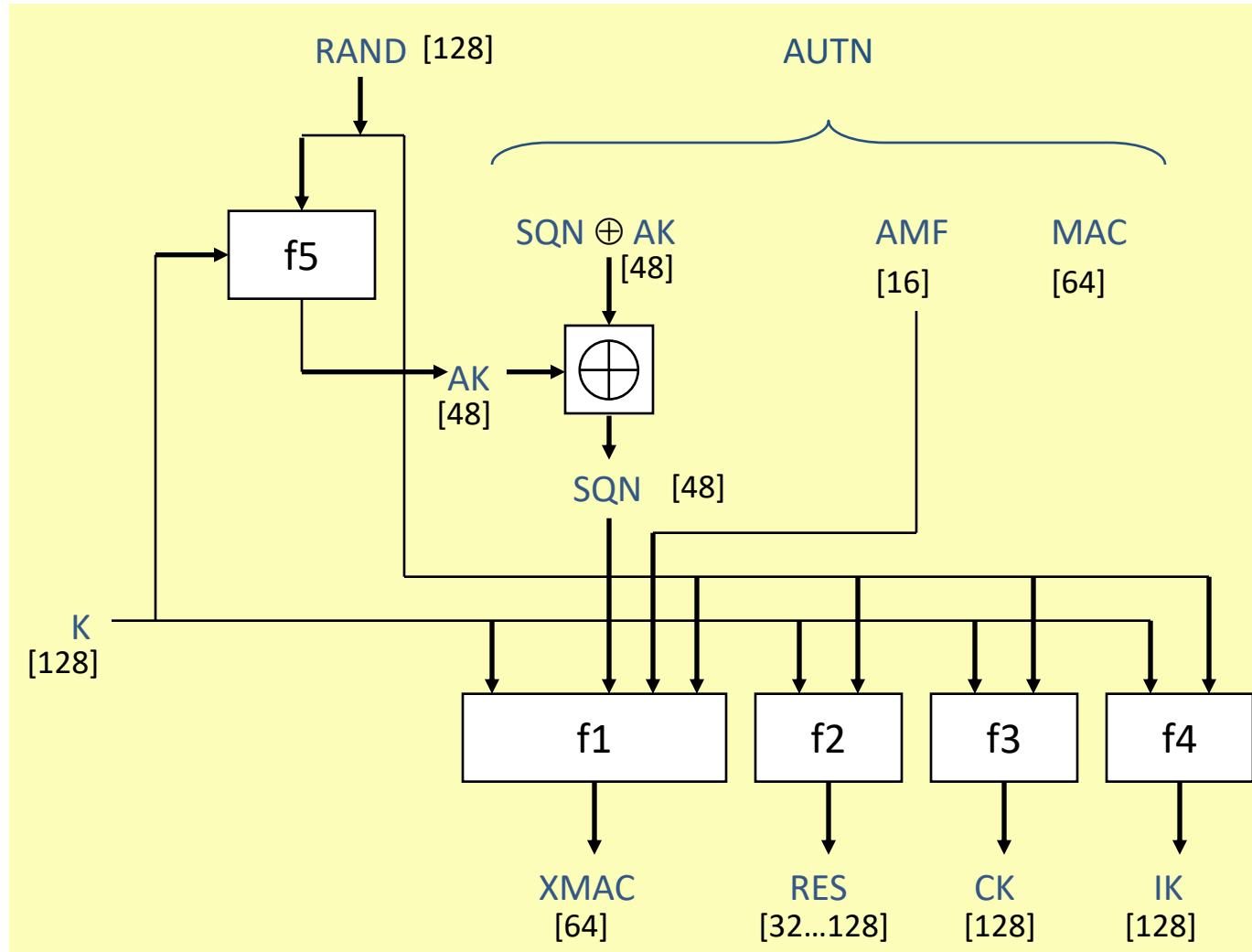
SQN Sequence number
RAND Random number
AMF Authenticated Management Field
K Secret Key

MAC Message authentication code
XRES Expected response
RES Response
CK Cipher key
IK Integrity key
AK Anonymity key

AUTN Authentication token
AV Authentication vector
[...] # of bits

False-base-station attacks possible if attacker can eavesdrop AV on network internal communication lines

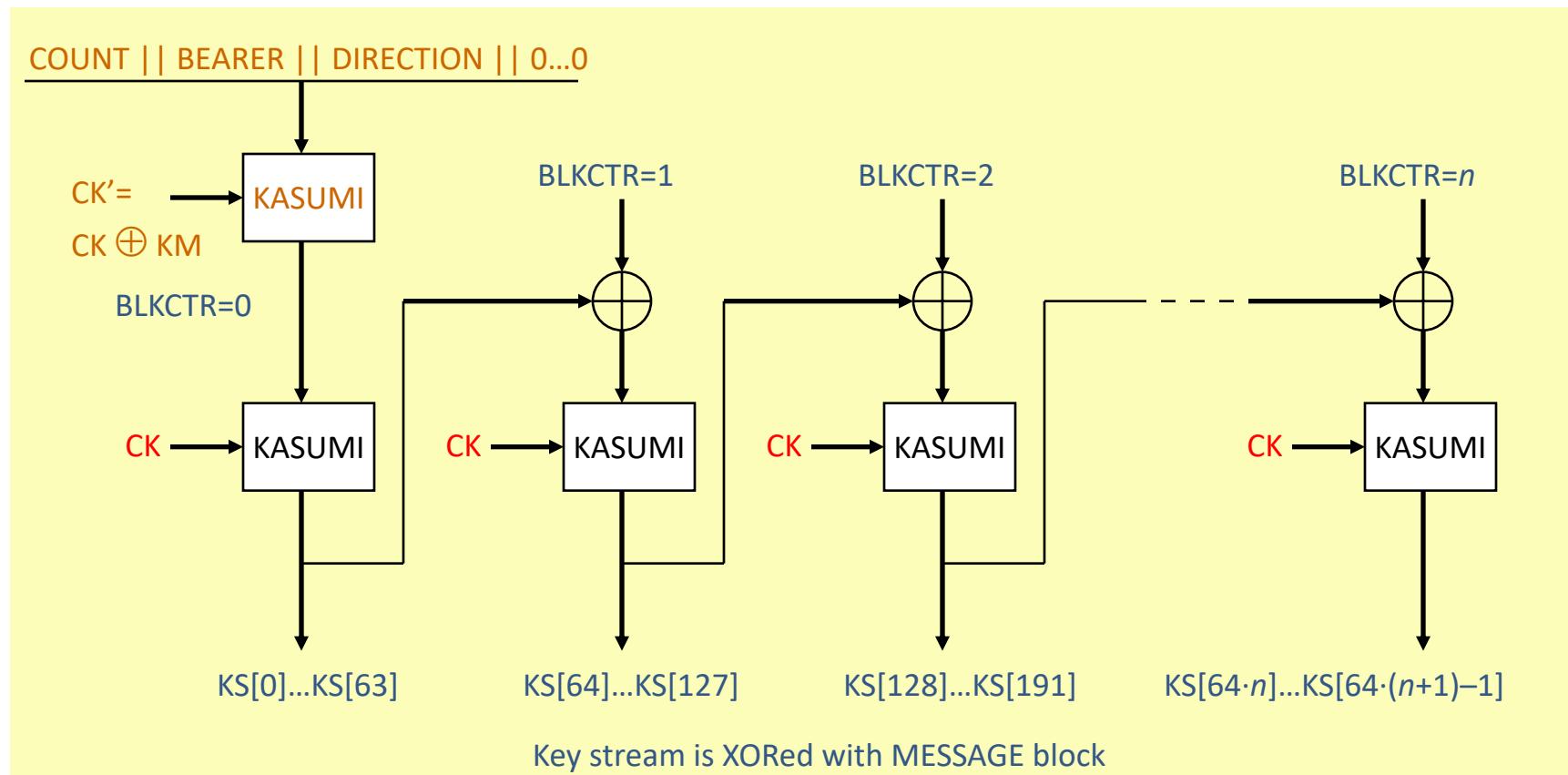
Authentication function in the USIM (user side)



Verify MAC == XMAC, than verify that SQN is in the correct range

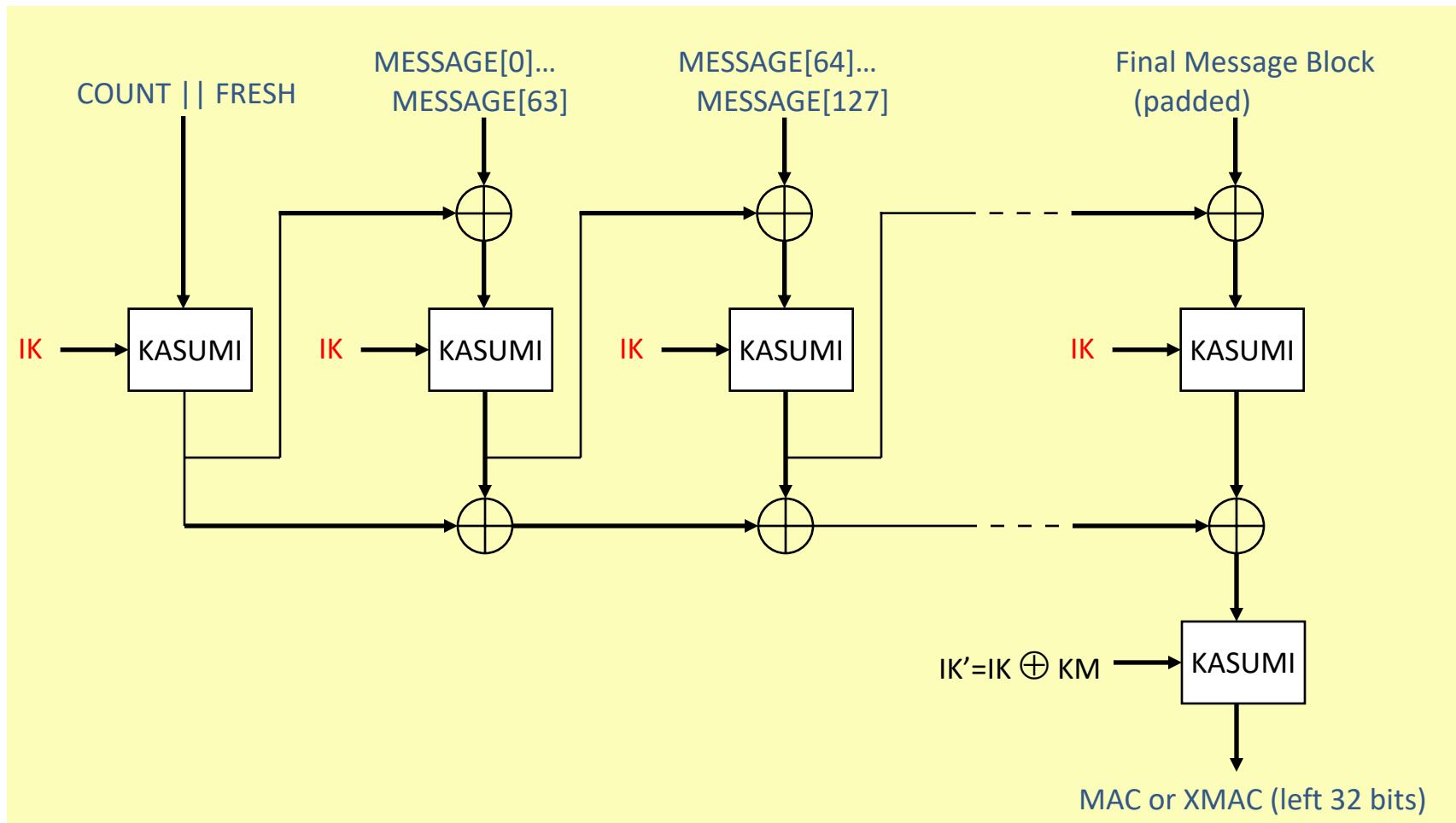
Cipher algorithm f8

- Combination of Output Feedback mode (OFB) and counter mode
- First encryption under CK' prevents chosen plaintext attacks (initialization vector is encrypted, KM: key modifier)



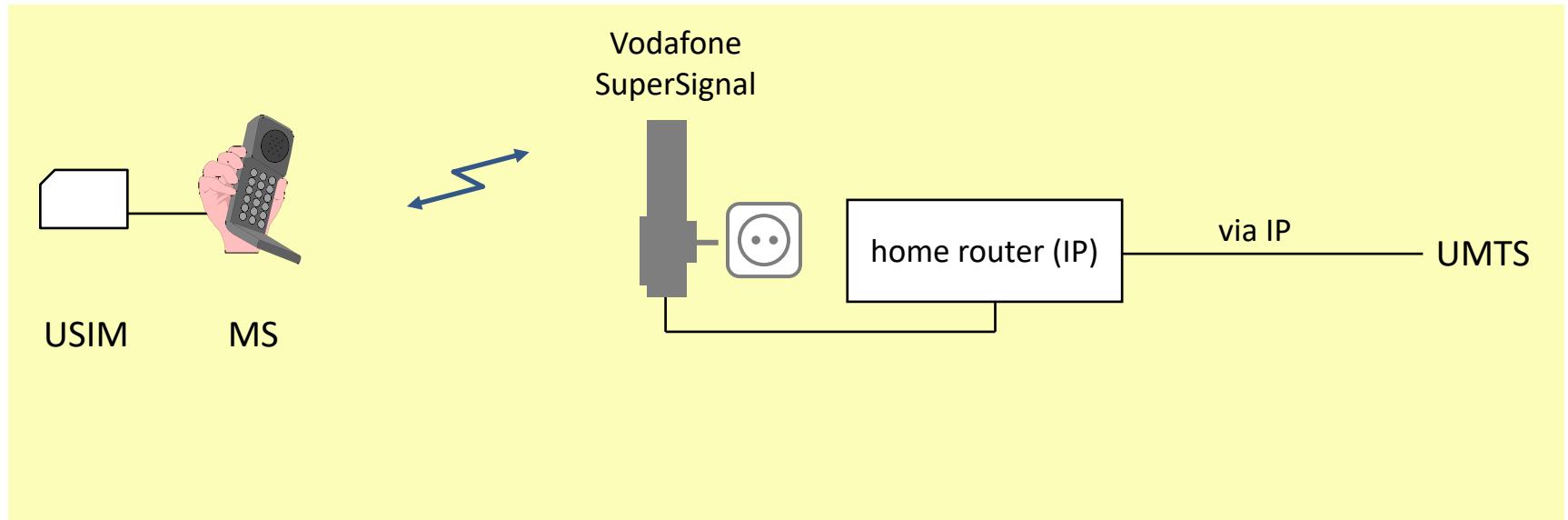
Integrity algorithm f9: ISO/IEC 9797-1 (MAC algorithm 2)

- Sender and receiver use f9
- Receiver verifies $\text{MAC} == \text{XMAC}$



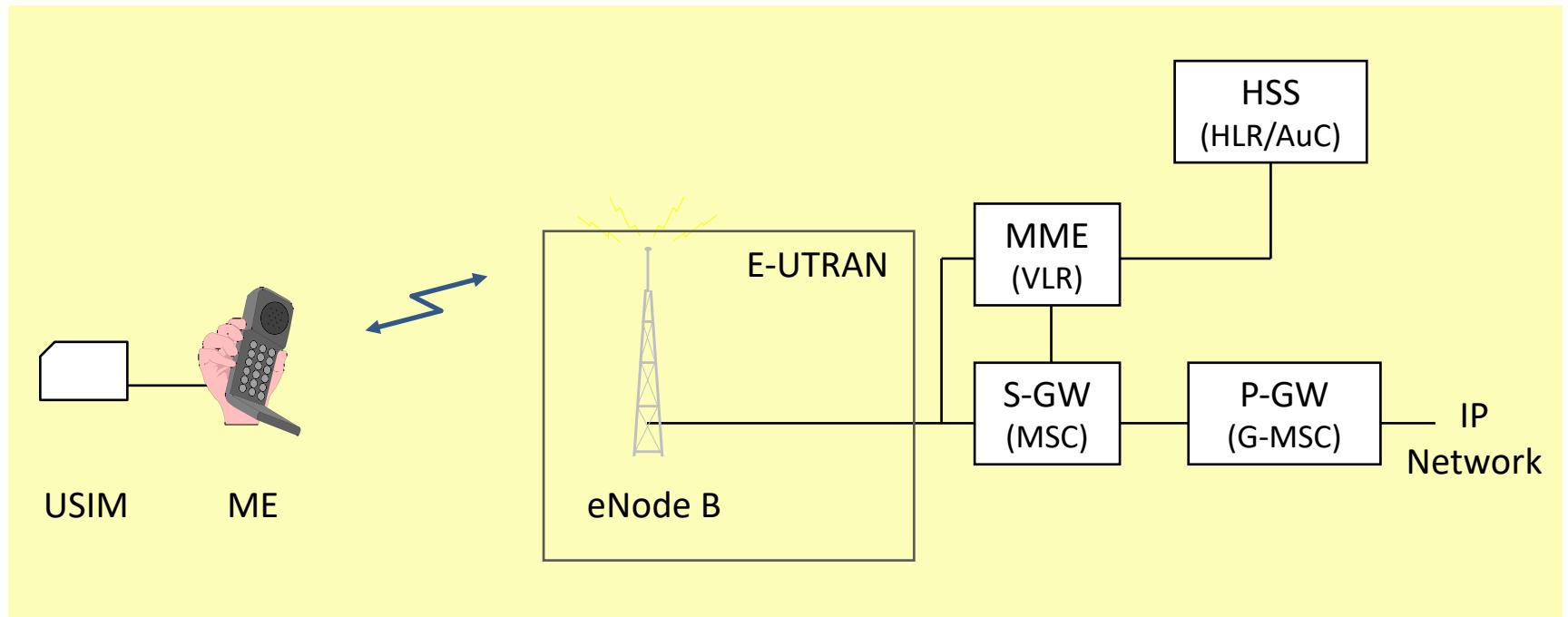
Own base station in UMTS

- Example: Vodafone SuperSignal
 - base station connected via IP with UMTS network
 - femto cell at home, not a repeater



Source: <http://www.vodafone.de/business/hilfe-support/umts-basisstation-vodafone-supersignal.html>

Long Term Evolution (LTE) Architecture



USIM	UMTS Subscriber Identity Module
ME	Mobile Equipment
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
MME	Mobility Management Entity
HSS	Home Subscriber Service
S-GW	Serving Gateway
P-GW	Packet Data Network Gateway
IP	Internet Protocol

Long Term Evolution (LTE)

- Characteristics
 - Traffic channels: Data services only, Speech is realized via Voice-over-IP
 - SMS is realized via signalling messages (similar to GSM)
- Security: inspired and closely related to UMTS
 - Individual symmetric key at USIM and HSS
 - Authentication vector
 - Calculated at USIM and HSS
 - Checked at MME
 - Pseudonymization on air interface:
 - Globally Unique Temporary Identity (GUTI)
 - Data encryption
 - Air interface: Advanced Encryption Standard (AES)
 - Network internal communication: IPSec
 - > False-base-station attacks: impossible

Bluetooth security



Bluetooth

- Development
 - Initiated by Ericsson
 - Bluetooth Special Interest Group (SIG)
 - Ericsson, Nokia, IBM, Toshiba, Intel and many other
- Standard
 - IEEE 802.15.1
- Benefits
 - Low energy consumption
 - Low interference sensibility (spread spectrum techniques)
- Disadvantages
 - Low Bandwidth
 - Limited signal coverage (radius)
 - Limited number of users



Technical Details

- Physical Layer
 - License free ISM-Band: 2,4GHz (ISM: Industrial, Scientific, Medical)
 - 2402 to 2480 MHz
 - 79 channels per 1 MHz bandwidth
 - Frequency-Hopping with 1600 chips (changes per second)
- Link Layer (DLL)
 - Modulation method:
 - Gaussian Frequency Shift Keying
 - Forward Error Correction (FEC)
 - Cyclic Redundancy Check (CRC)



Technische Details

- **Specifications**
 - 1.0: First spec, still immature, ca. 732 kbps data rate
 - 1.1: Broadly used
 - 1.2: Adaptive Frequency Hopping, improved error correction
 - 2.0 (Nov 2004): Data rates up to 2 Mbps
 - 3.0 (Apr 2009): Data rates up to 24 Mbps
 - 4.0 (Dec 2009): Bluetooth Low Energy
- **Classification**
 - Pico-Bluetooth
 - 2,5 mW / 1 mW transmission power (Class 2 and 3)
 - Radius up to 50 m / 10 m
 - Mega-Bluetooth
 - 100 mW transmitting power (Class 1)
 - Radius up to 100 m

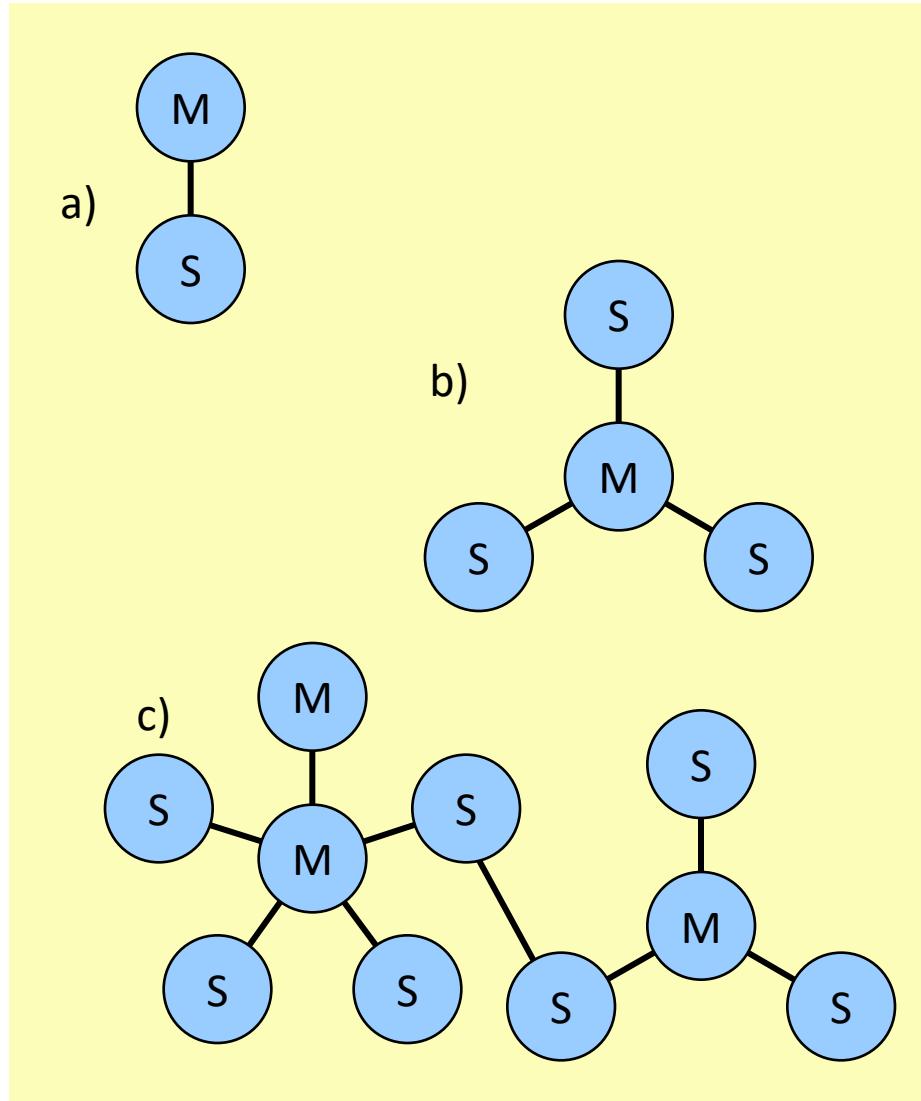


Development of networks

a) Point-to-Point

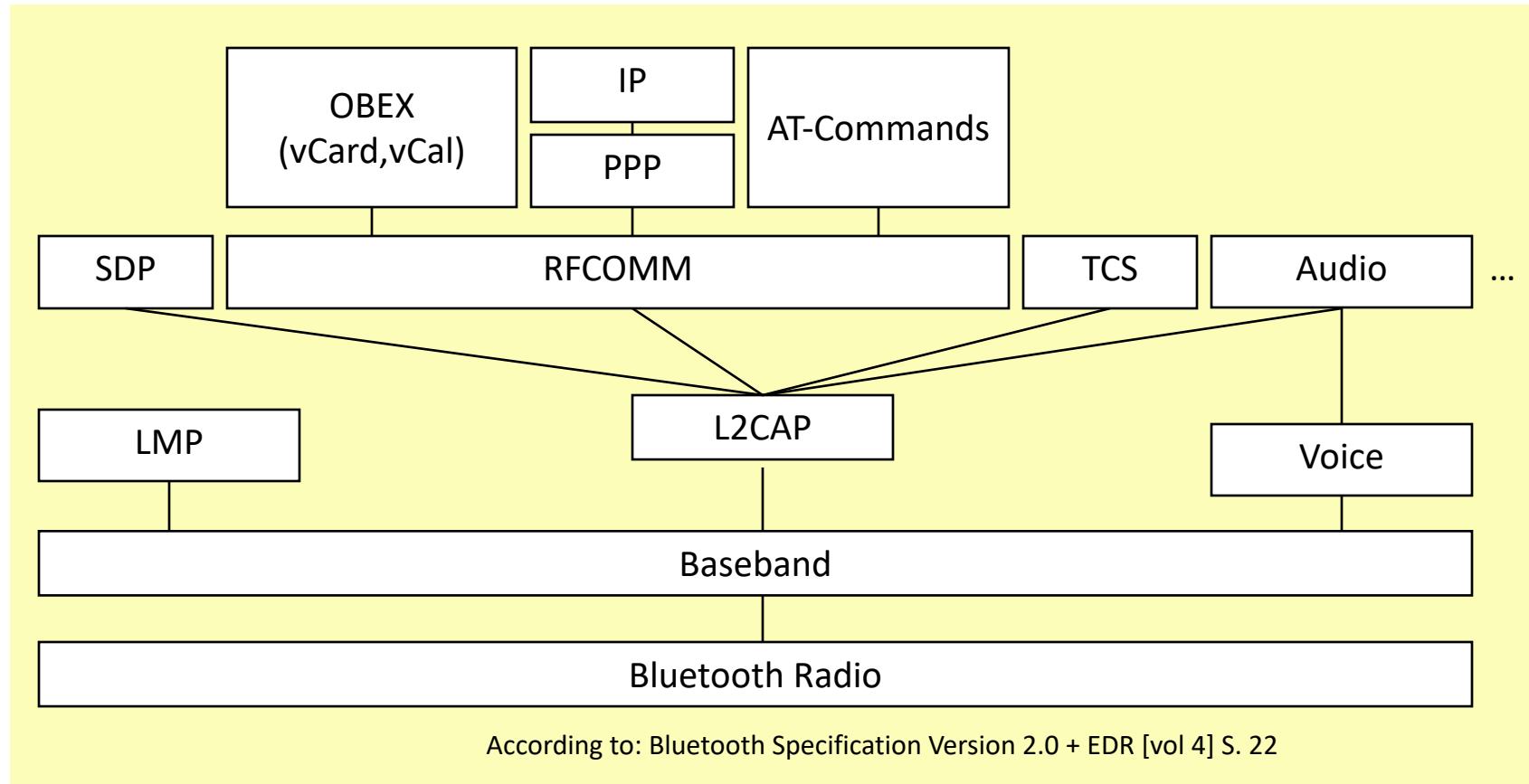
b) Pico-Network: 1 Master, up to 7 active slaves

c) Scatter-Network: various overlapping Pico-Networks





Protocols



OBEX OBject EXchange protocol
IP Internet Protocol
PPP Point-to-Point Protocol
SDP Service Discovery Protocol

RFCOMM Serial cable emulation protocol
TCS Telephony Control protocol Specification
LMP Link Manager Protocol
L2CAP Logical Link Control and Adaption Protocol



Protocols (2)

- Bluetooth Radio
 - Air Interface
- Baseband
 - Functions for Link connection, Frequency-Hopping, etc.
- Link Manager Protocol (LMP)
 - Security features, clock synchronisation
- Logical Link Control and Adaption Protocol (L2CAP)
 - Interface for higher protocol layers to access baseband
- Service Discovery Protocol (SDP)
 - Information about device types, services, etc.
- RFCOMM (Serial cable emulation protocol)
 - Based on ETSI TS 07.10; for universal use (Modem, IP, ...)
- Telephony Control protocol Specification (TCS)
 - For device control



Security

- Security functions
 - Secure device pairing
 - Symmetric authentication (one sided and mutual)
 - Symmetric encryption
- Basic algorithm for pairing and authentication
 - SAFER+
 - Publicly known
 - 1 of 15 candidates for AES (Advanced Encryption Standard)
 - Characteristics of SAFER+
 - Block cipher with 128 Bit block length
 - 8 rounds
 - Key length 128 Bit
 - Used in E21, E22, E1 und E3



Pairing

- Objectives
 - Identification of two devices A and B
 - Generates a symmetric key K_{AB}
- Pairing Procedure
 1. Exchange of device addresses BD_ADDR_A and BD_ADDR_B
 2. Generate Initialization key K_{init} (intermediate step)
 3. Generate K_{AB}



Pairing (1)

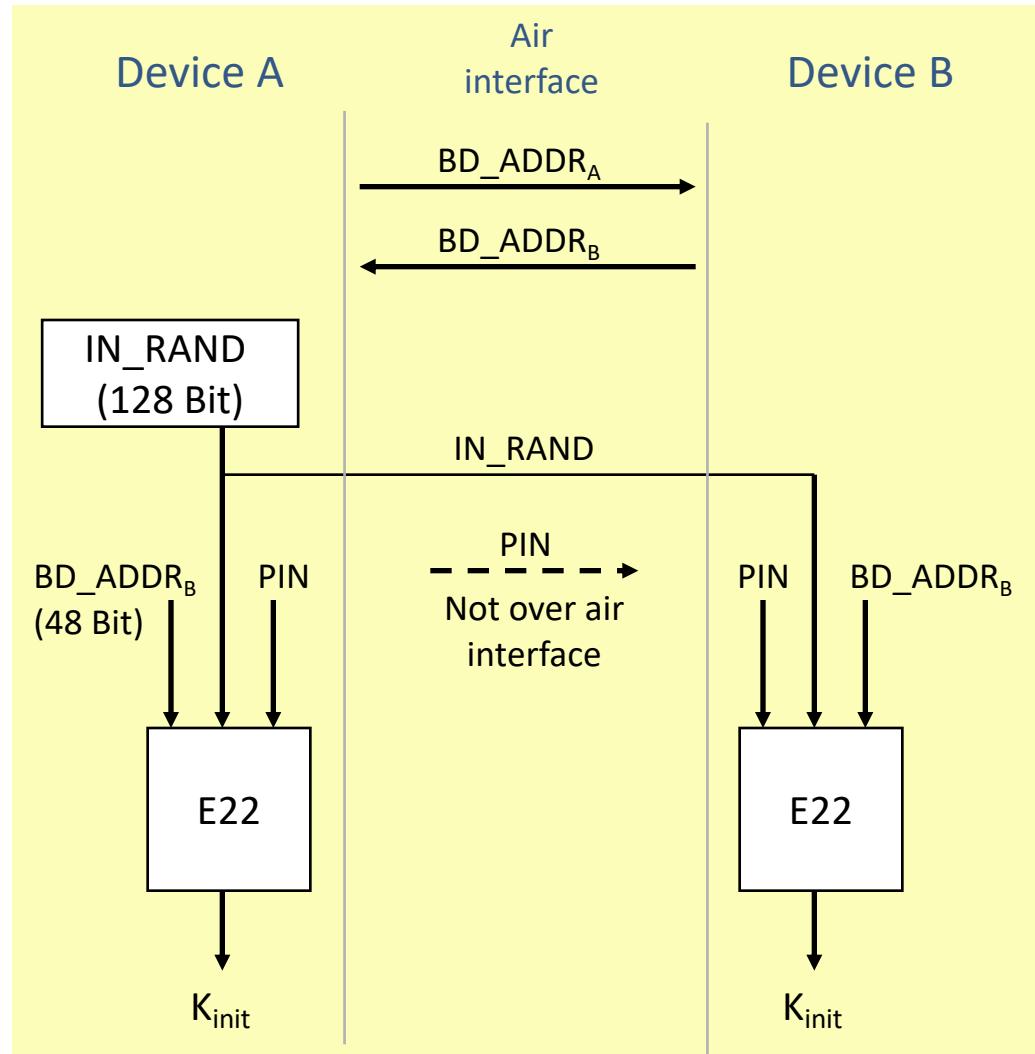
- Generate Initialization key K_{init} (Algorithm E22)

- Input:

- Device address
(BD_ADDR_B , 48 Bit)
 - PIN (8-128 Bit, typ.
at least 4 digits)
 - Random number
(IN_RAND ,
128 Bit)

- Output:

- K_{init} (128 Bit)





Pairing (2)

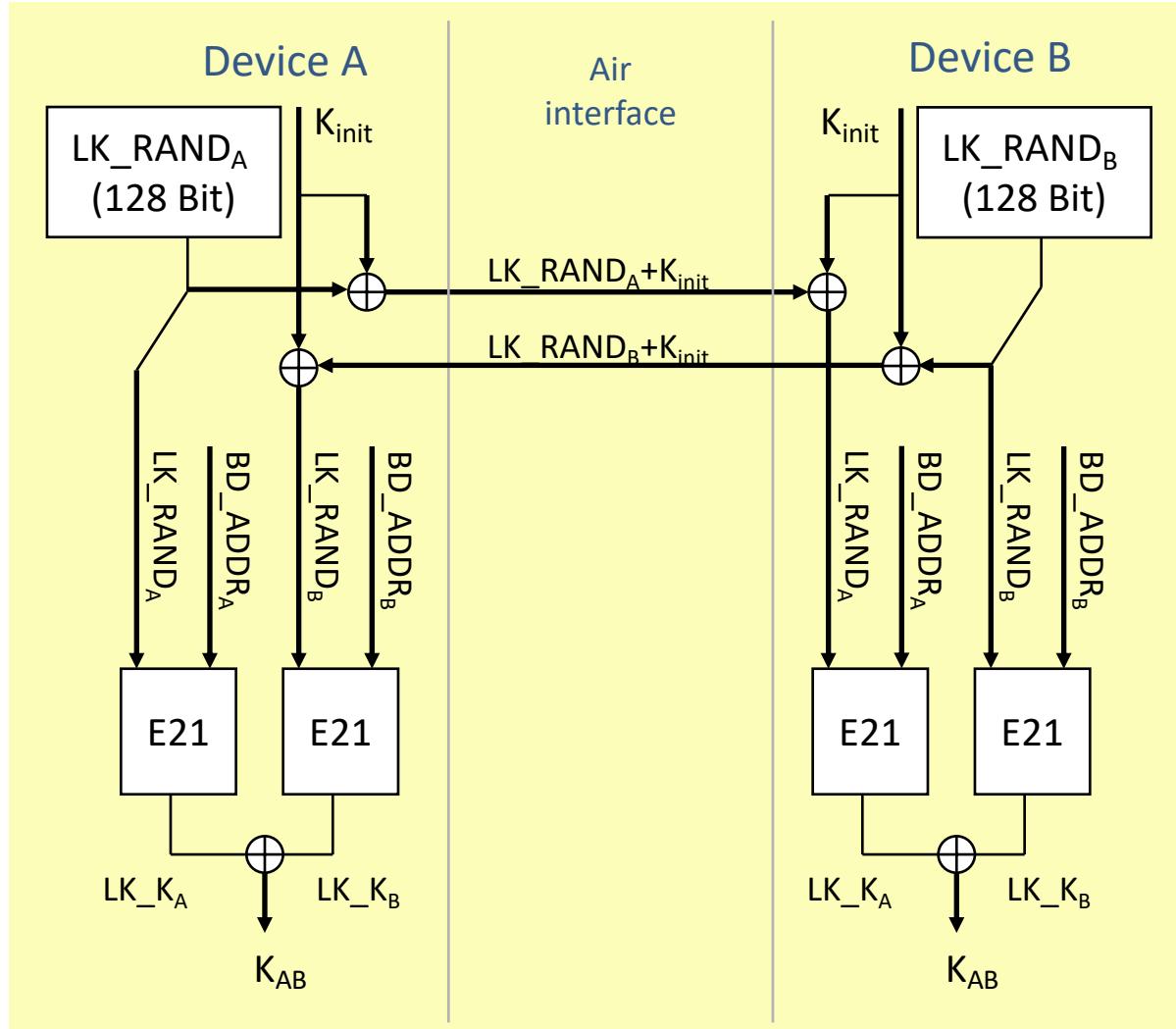
■ Generate K_{AB} (Algorithm E21)

■ Input:

- Random numbers ($LK_RAND_{A/B}$, 128 Bit)
- Device address ($BD_ADDR_{A/B}$, 48 Bit)
- Initialization key K_{init}

■ Output:

- K_{AB} (128 Bit)





Authentication (one sided or mutual)

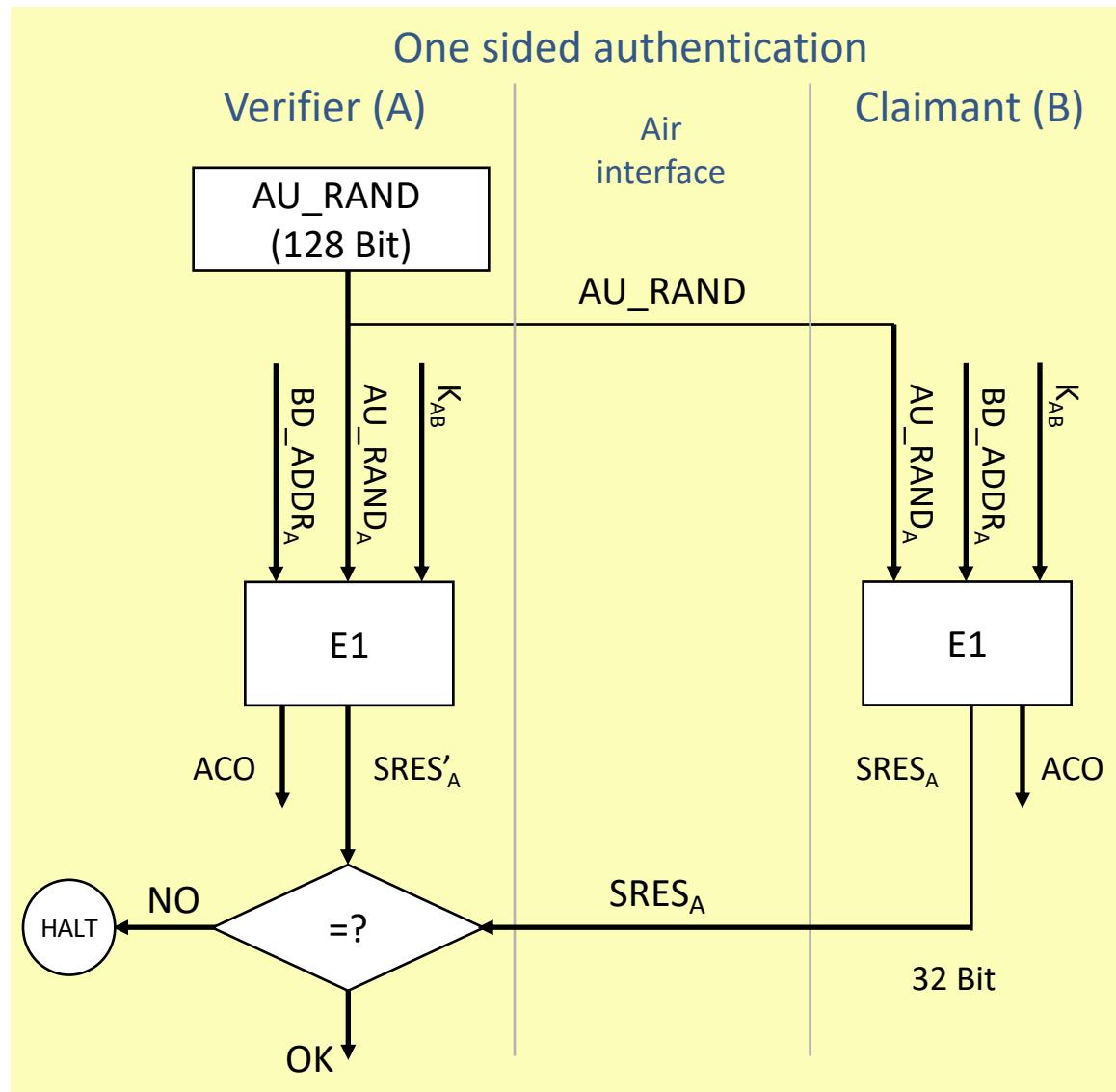
- Algorithm E1

- Input:

- Random number AU_RAND
- K_{AB}
- Device address A BD_ADDR_A

- Output:

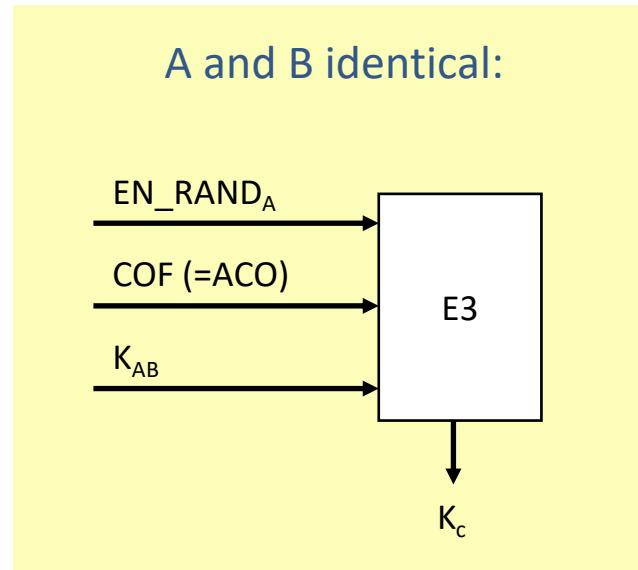
- true or false
- ACO (Authenticated Chiphering Offset, 96 Bit)





Encryption

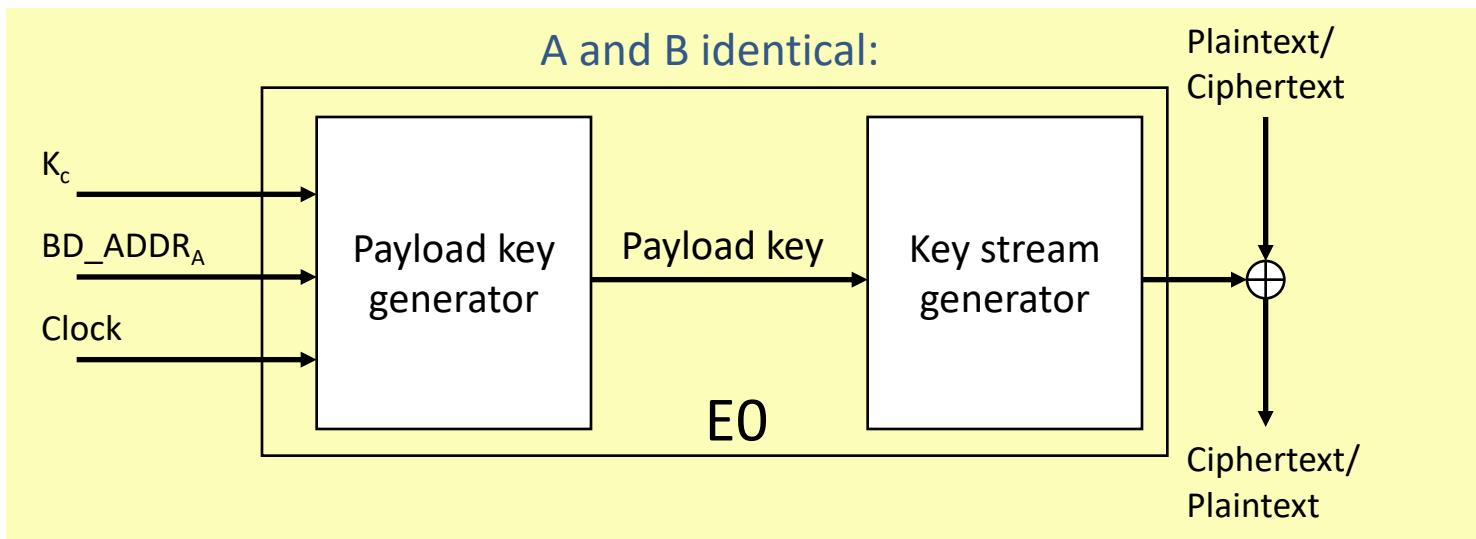
- 2 Steps
 - Generate key K_c with algorithm E3
 - Data encryption with stream cipher E0
- Algorithm E3
- Input:
 - Random number (EN_RAND_A , 128 Bit)
 - Ciphering Offset Number (COF, 96 Bit) = ACO (from Authentication)
 - K_{AB} (128 Bit)
- Output:
 - K_c (8-128 Bit, manufacturer specific)





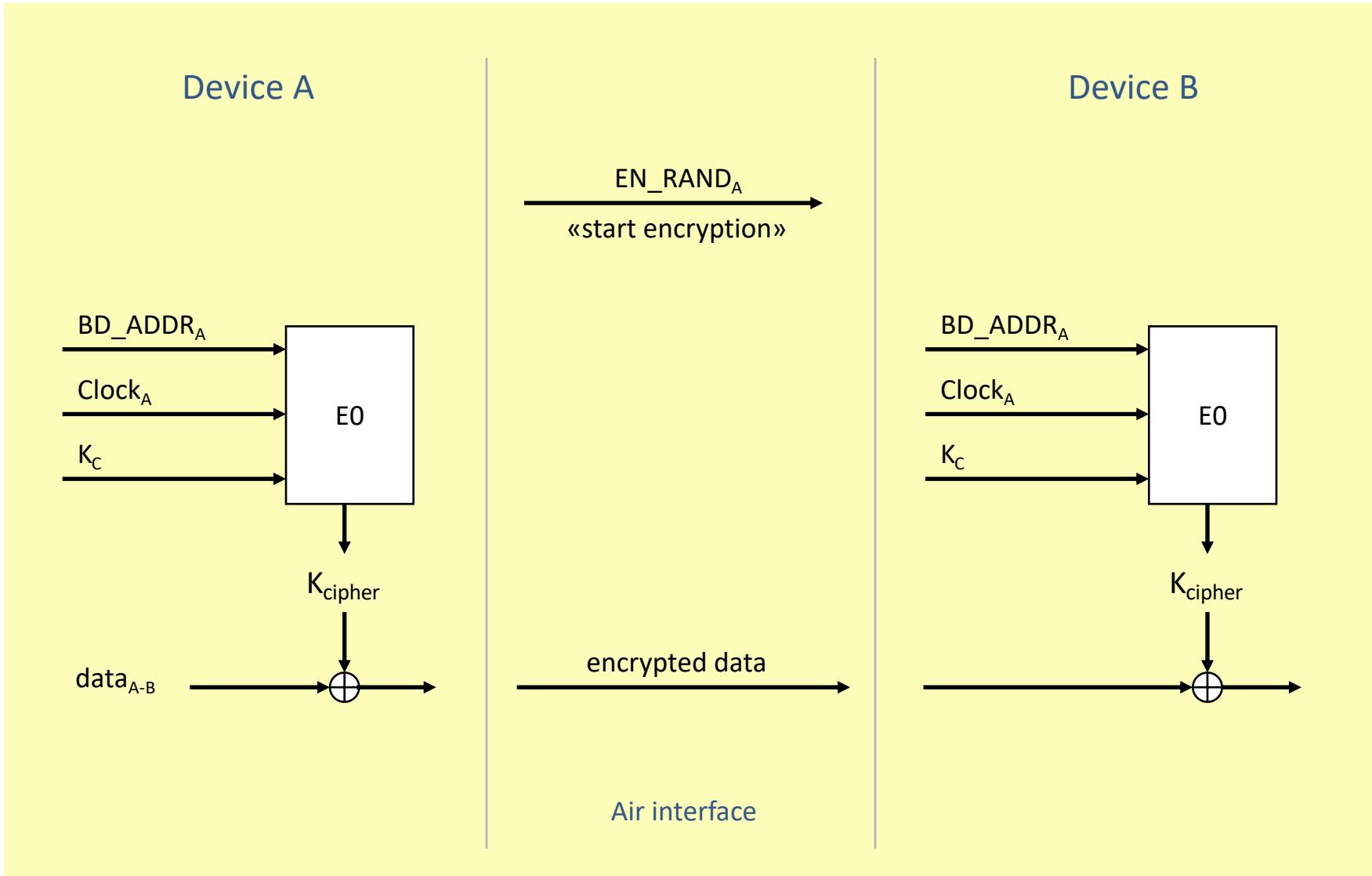
Encryption (2)

- Algorithm E0
 - Linear Feedback Shift Register
 - Stream cipher with variable block length up to 64 Bit
- Input:
 - K_c
 - Device address (BD_ADDR_A)
 - Clock (counter)
 - Plaintext or Ciphertext





Encryption (3)





Summary: Bluetooth security functions

- Initialization (Pairing)
 - Generate symmetric key K_{AB} between devices
 - K_{AB} saved
 - K_{init} no longer needed
- Authentication
 - Challenge-Response based on K_{AB}
- Encryption
 - Session key K_c generated from K_{AB}
 - Pseudo-One-Time-Pad
 - K_c can be changed automatically while being connected



Vulnerabilities

- Used PIN with Pairing
 - Often too short (4 digits)
 - Fixed in the device (1234 or 0000)
 - Often one for all devices used by user (convenience)
 - Some devices can only process max. 16-digit PINs
- Location finding is easy
 - BD_ADDR used to discover devices
 - Service Discovery Protocol (SDP)
 - Generating route profiles
- Device address can be faked
- High level of vulnerability to DoS-attacks
 - Repeated refused queries
 - Result: battery is discharged



Known attacks (selection) 1/2

- Range: with antenna up to 2 km
 - Salzburg research, August 2004
- BlueBug: Uses implementation errors
 - Marcel Holtmann, Sept 2003
 - BlueSnarf: change phone book, send SMS, ...
 - Chaos-Attack: initiate unnoticed calls, possibilities like BlueSnarf
 - No pairing necessary
- BlueSmack:
 - DoS-Attack (use echo-requests)



Known attacks (selection) 2/2

■ PIN Cracking

- Yaniv Shaked and Avishai Wool, Juni 2005
- Brute-force attack on K_{init} (and K_{AB})
- Passive attack
 - Pairing process is sniffed by attacker
- Active Attack
 - Attacker provokes Re-Pairing and hopes for weak PIN
- Not possible, if PIN>64 Bit \approx 19 digits

PIN lengths	Time in s
4	0,063
5	0,75
6	7,609
7	76,127

Results with Pentium IV 3GHz



Security

- In general
 - no use of Bluetooth, as far as possible
 - if not used, switch it off
 - disable visibility of device
- Pairing
 - no pairing in the public
 - pairing with other technology (e.g. NFC = Near Field Communication)
 - use (more than 18 digits) non-trivial PINs
 - multiple devices must have different PINs
- Hope for good implementation
 - firmware update if necessary

WiFi security

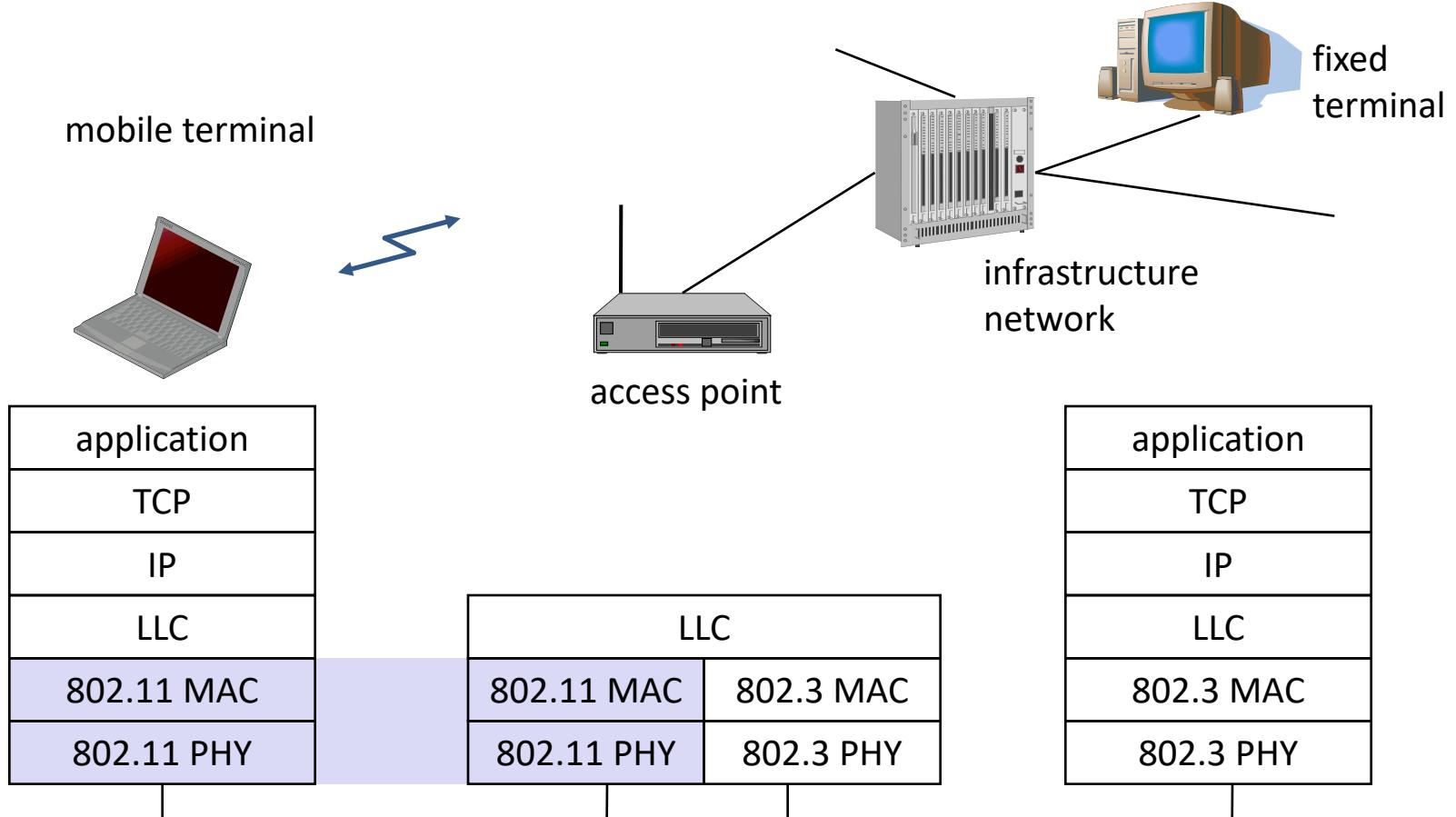


WLAN: Wireless Local Area Networks

- Wireless connection of systems
 - increased mobility
 - no physical (wired) connections
- Topologies
 - Ad-hoc mode: peer-to-peer connections (client-to-client)
 - Infrastructure mode: via Access Point (AP)
- IEEE 802.11 standard
 - IEEE: Institute of Electrical and Electronics Engineers
 - defines layer 1 and parts of layer 2 of OSI ref. model
 - has Logical Link Control (802.2) together with other 802 standards



IEEE 802.11 Standard





IEEE 802.11 Protocol family

- Well-known WLAN-standards:
 - IEEE 802.11:
 - Infrared (IR)
 - 1 or 2 Mbps via radio in 2,4-GHz ISM band
 - IEEE 802.11b: 11 Mbps in 2,4-GHz ISM band
 - IEEE 802.11a: 54 Mbps in 5-GHz ISM band
 - IEEE 802.11g: 54 Mbps in 2,4-GHz ISM band
 - IEEE 802.11n: 600 Mbps in 2,4-GHz and 5-GHz ISM band
 - IEEE 802.11p: 27 Mbps around 5-GHz Car-to-Car
- Security
 - IEEE 802.11i: Security (WPA2)
 - Outdated:
 - WEP (Wired Equivalent Privacy)
 - WPA (WiFi Protected Access) and others



WLAN

- Security demands
 - Confidentiality:
 - Protection against eavesdropping
 - Integrity:
 - Protection against modification of messages
 - Protection against unauthorized access
 - Availability
 - Protection against denial-of-service attacks



Protection against unauthorized access

- Weak protection: MAC addresses
 - Limit access to specific MAC addresses on the network
- Problem:
 - Management of valid MAC addresses
 - MAC addresses can be spoofed (MAC spoofing)

WIRELESS ACCESS FILTER

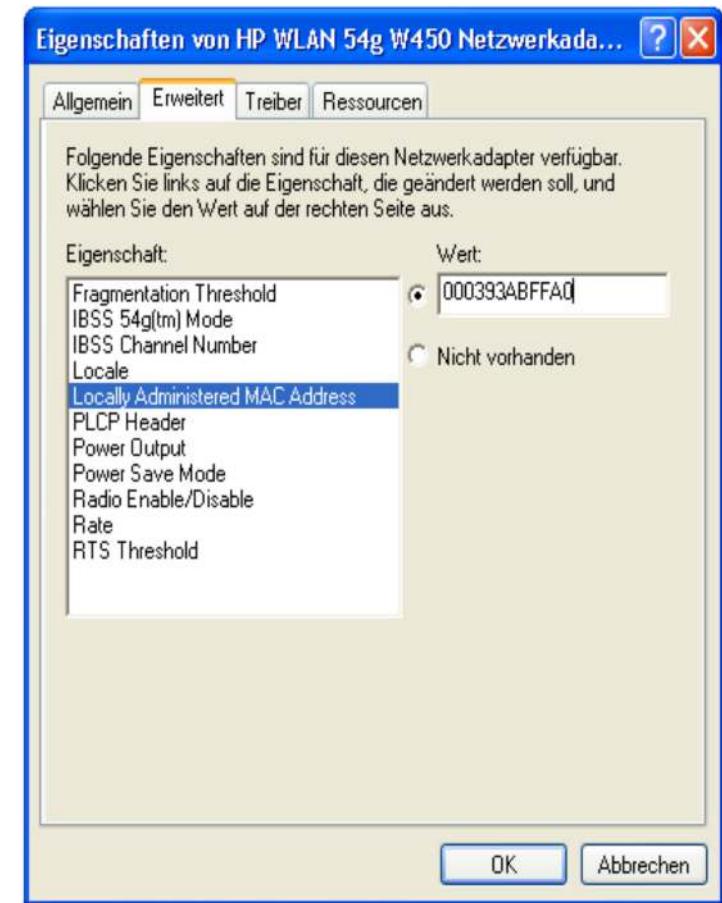
Setup Access Filter

Wireless Access Filter

Wireless Filter Activated Yes No

User	MAC Address	User	MAC Address
sec1	00:03:93:af:cc:09	Card 7	
sec2	00:30:bd:61:fb:b6	Card 8	
tom	00:0b:cde7:94:c2	Card 9	
Card 4		Card 10	
Card 5		Card 11	
Card 6		Card 12	

NOTE: When Activated, Only These Cards Will Be Able To Access The Router.





WEP: Wired Equivalent Privacy

- General
 - Optional sub-protocol of IEEE 802.11
 - Encryption, integrity protection and authentication
 - Implemented in virtually all WLAN devices
- Encryption
 - Symmetric encryption with 40 or 104 bit keys, based on RC4
- Integrity protection
 - CRC (Cyclic Redundancy Check)
- Authentication
 - Method 1: »Open«: no authentication
 - Method 2: »Shared Key«: Challenge-Response-Authentication

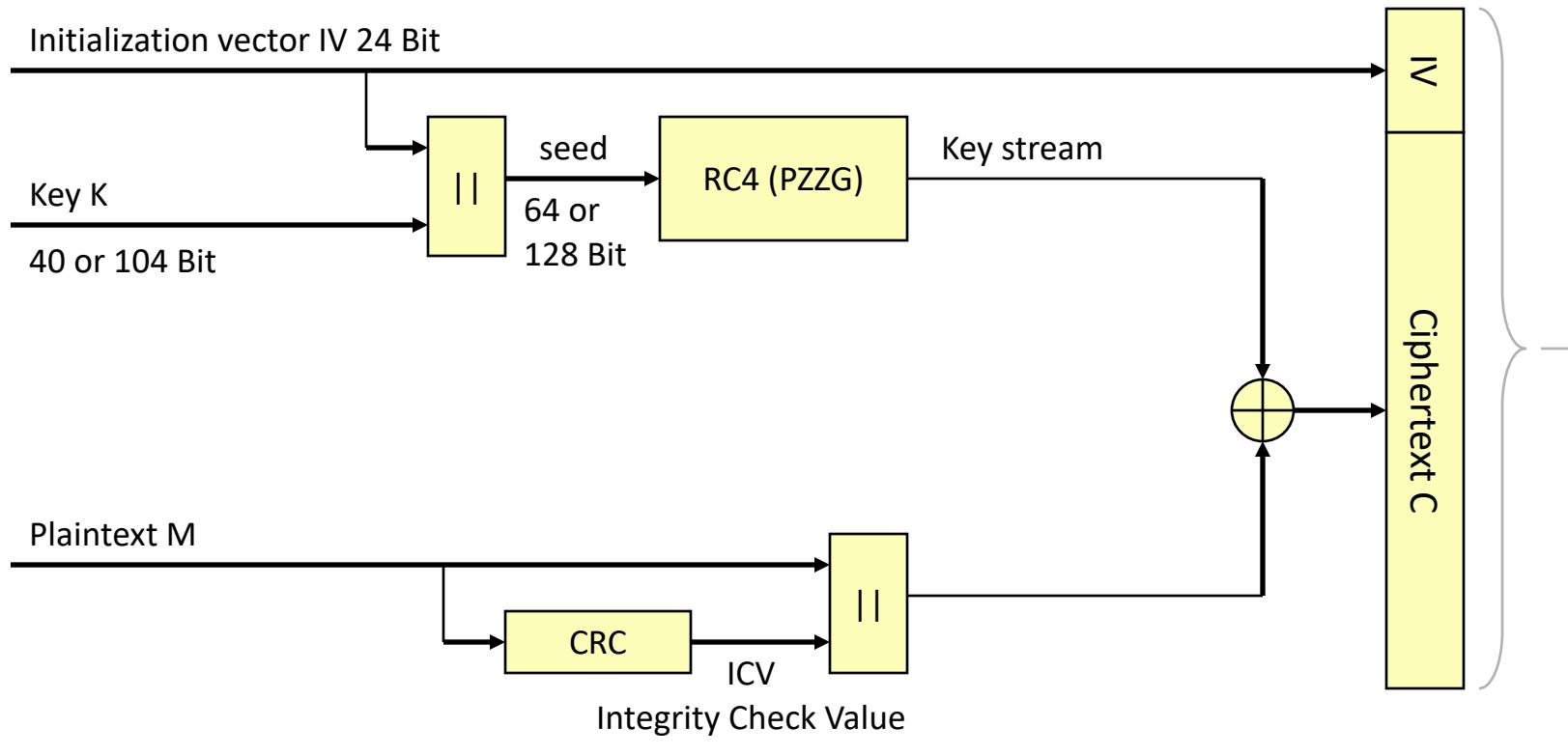


WEP: Encryption

- Symmetric stream cipher
 - Plaintext XORed with key stream
- Generation of key stream
 - Initialization vector (IV, 24 bit)
 - Key (K, 40 or 104 bit)
 - RC4 algorithm used as Pseudo Random Number Generator (PRNG)
- IV is send in clear
- Decryption
 - Receiver generates same key stream
 - Ciphertext XORed with key stream
 - Cipher text and key stream linked again with XOR



WEP: Encryption and Integrity protection



|| concatenation

⊕ XOR

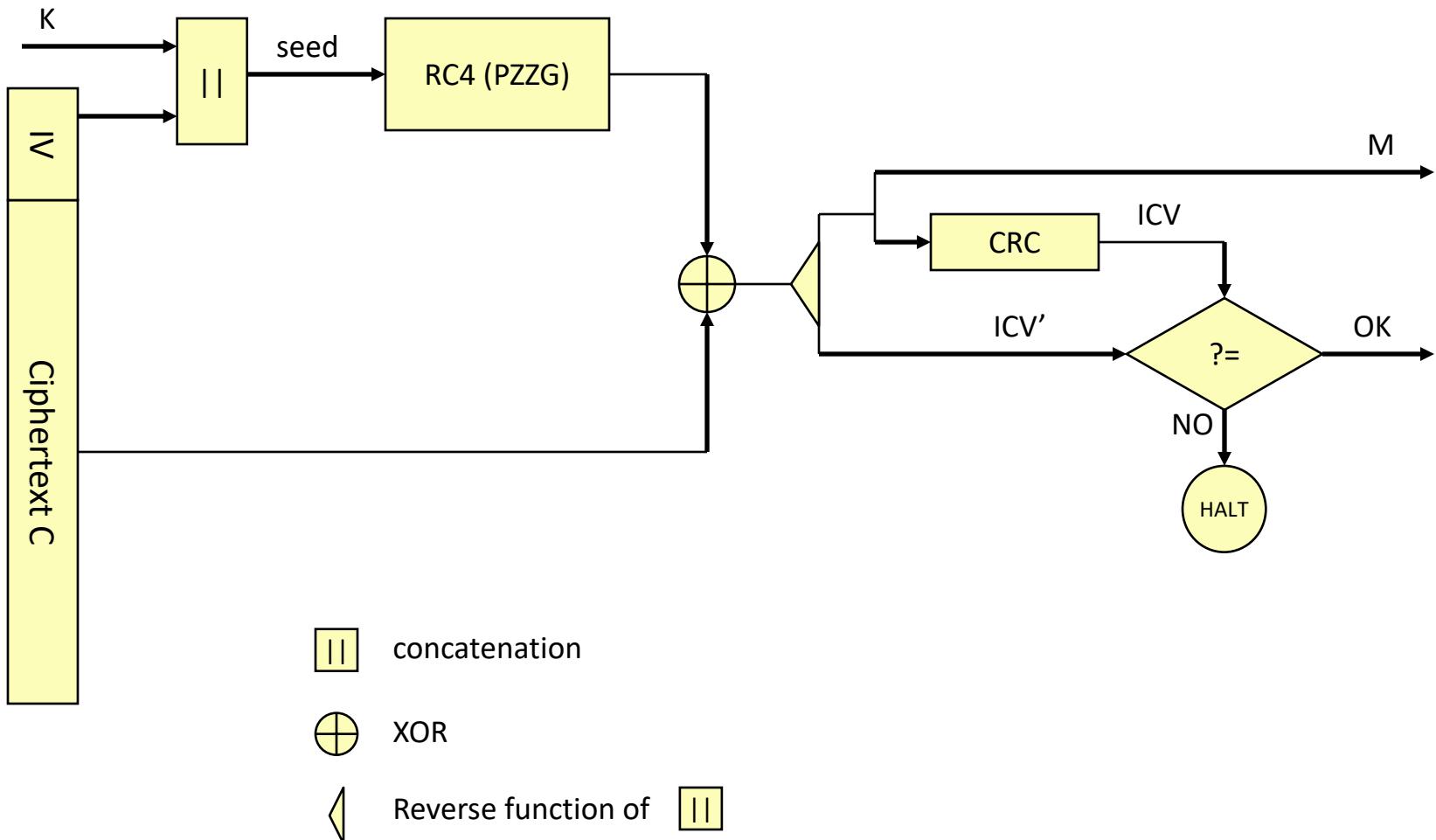
$IV \parallel (M \parallel CRC(M)) \oplus RC4(IV \parallel K)$

Or shorter:

$IV, (M, CRC(M)) \oplus RC4(IV, K)$



WEP: Decryption and integrity protection





WEP: Authentication

- Two options
 - Open and Shared Key
- Open (= no authentication)
 - disable authentication (only SSID, Server Set ID)
- Shared Key
 - Challenge-Response-Authentication
 - Access Point sends unencrypted challenge value
 - Client sends challenge value back as encrypted response
 - Access to network, if challenge is encrypted correctly



WEP: Vulnerabilities

1. Initialization vector

- IV too short, repeated usage of equal IVs
- Some products implement IV++ with start value IV=0
- Results in Known-Plaintext-Attack:
 - Attacker can store a table of (IV, Key stream):
 - Ciphertext $C = (M, \text{CRC}(M)) \oplus \text{RC4}(IV, K)$
 - Attacker knows ciphertext, IV and M:
Calculate Key stream = $\text{RC4}(IV, K)$
If IV again occurs, attacker can decrypt
 - Message-related break: Break individual messages, without finding the key K

1. Key K

- Too short key length with 40 Bit (Brute-Force-Attack)



WEP: Vulnerabilities

3. Weakness in RC4 and its usage

- Weak IVs can be used to calculate K with statistic attack:
 - Attacker knows IV, ciphertext and beginning of plaintext
 - Beginning of plaintext: Data packets start with M=0xAAAA03 (SNAP-Header, Sub Network Access Protocol)
 - Attacker knows first three bytes of key stream
 - Determine Key stream (output of RC4) from ciphertext and M
$$C = \text{Key stream} \text{ XOR } M$$
 - With knowledge of many IVs and many Key streams:
 - Possible exploitation of vulnerability from RC4: partial Linearity of RC4 allows determination of K

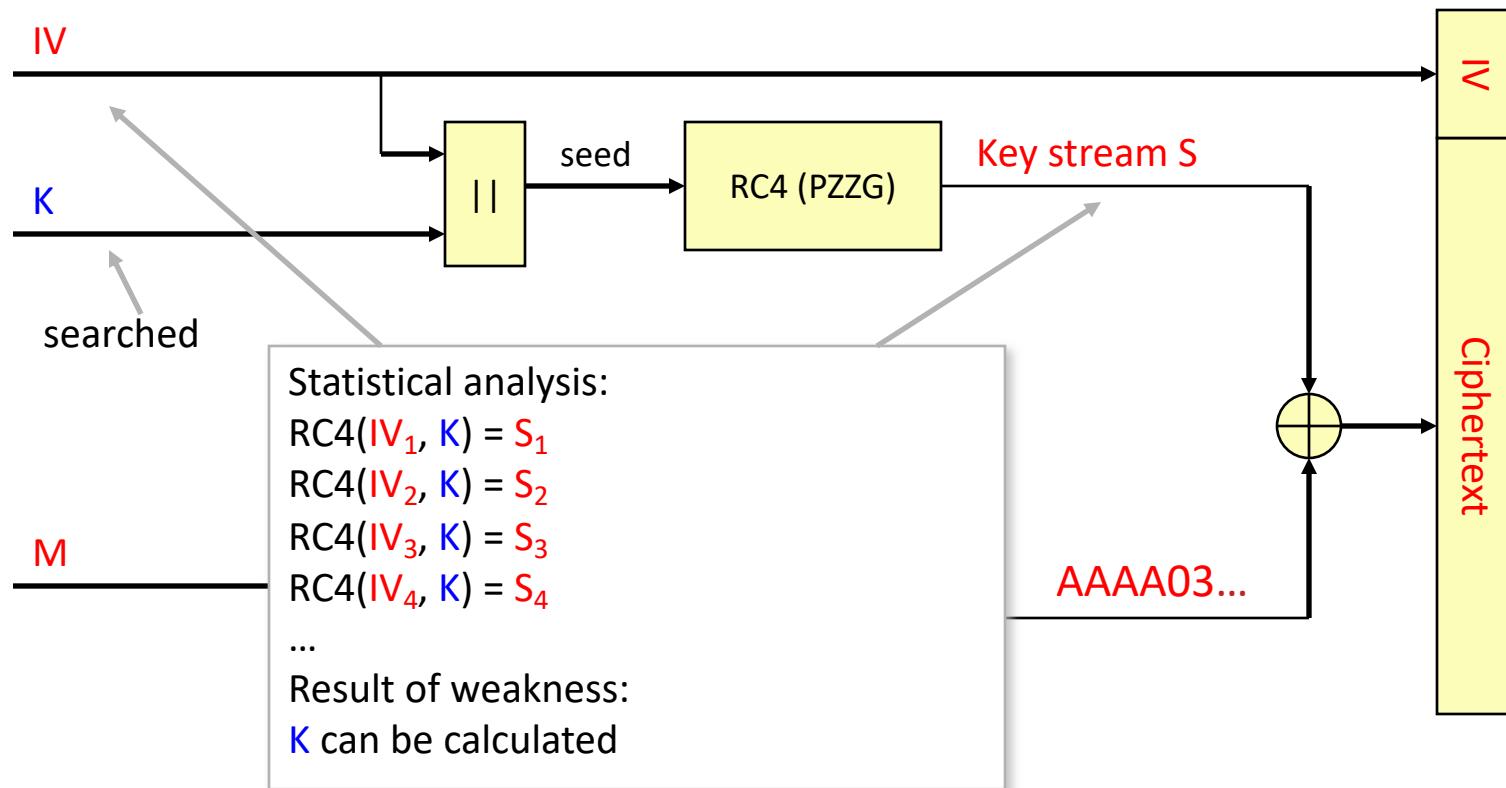
Key stream = RC4(**IV**, **K**)

red = known
blue = unknown



WEP: Vulnerabilities

3. Weakness in RC4 and its usage



II concatenation

XOR

red = known
blue = unknown



WEP: Vulnerabilities

3. Weakness in RC4 and its usage

- Practical attack
 - 4-6 million data packets required to gather weak IVs: $\approx 5\%$ IVs are weak (≈ 900.000 of 2^{24}).
 - needs 8-12 hours (avg. net load of 1 Mbps) and up to 12 GB HDD space
 - all data packets begins with SNAP pattern 0xAAAA03
 - partial linearity of RC4 on weak IVs
- Improvement 1:
 - Attacker can enforce usage of weak IVs to reduce network load by choosing the IV, and sending and receiving packets
- Improvement 2:
 - Tews et al (2007) found further weakness in RC4 to improve speed of WEP attack to ≈ 1 min and no need of weak IVs

Weak IVs: Attack only possible if certain bit combinations in IV



WEP: Vulnerabilities

3. Weakness in RC4 and its usage

- Literature:
 - Scott Fluhrer, Itsik Mantin, Adi Shamir: Weaknesses in the Key Scheduling Algorithm of RC4. 2001.
 - Adam Stubblefield, John Ioannidis, Aviel D. Rubin: Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. 2001.
 - Erik Tews, Ralf-Philipp Weinmann, Andrei Pyshkin: Breaking 104 bit WEP in less than 60 seconds. 2007



WEP: Vulnerabilities

4. Weakness of CRC

- CRC and encryption are linear:
 - $c(a \oplus b) = c(a) \oplus c(b)$
- Modification of data packets is easy:
 - XOR a random number to (encrypted) plaintext
 - XOR a CRC to (encrypted) checksum



WEP: Vulnerabilities

4. Weakness of CRC

– Let

$$(M, \text{CRC}(M)) \oplus \text{RC4}(\text{IV}, K) = C$$

– Attacker sends a $C \oplus X$: with $X = (M', \text{CRC}(M'))$

$$X \oplus (M, \text{CRC}(M)) \oplus \text{RC4}(\text{IV}, K) = C \oplus X$$

– Recipient decrypts:

$$X \oplus (M, \text{CRC}(M)) = (M', \text{CRC}(M')) \oplus (M, \text{CRC}(M))$$

– Because of the data format and the linearity of the encryption (or XOR) and CRC:

$$\text{CRC}(M \oplus M') = \text{CRC}(M) \oplus \text{CRC}(M')$$

– Result: Attacker has sent a valid message $M \oplus M'$

- CRC can be used to detect random errors, but not to detect modifications of data by an attacker



WEP: Vulnerabilities

4. Weakness of CRC

Ciphertext from sender:

$$C = (M, \text{CRC}(M)) \oplus \text{RC4}(\text{IV}, K)$$

X of attacker:

$$X = (M', \text{CRC}(M'))$$

Attacker sends $C \oplus X$:

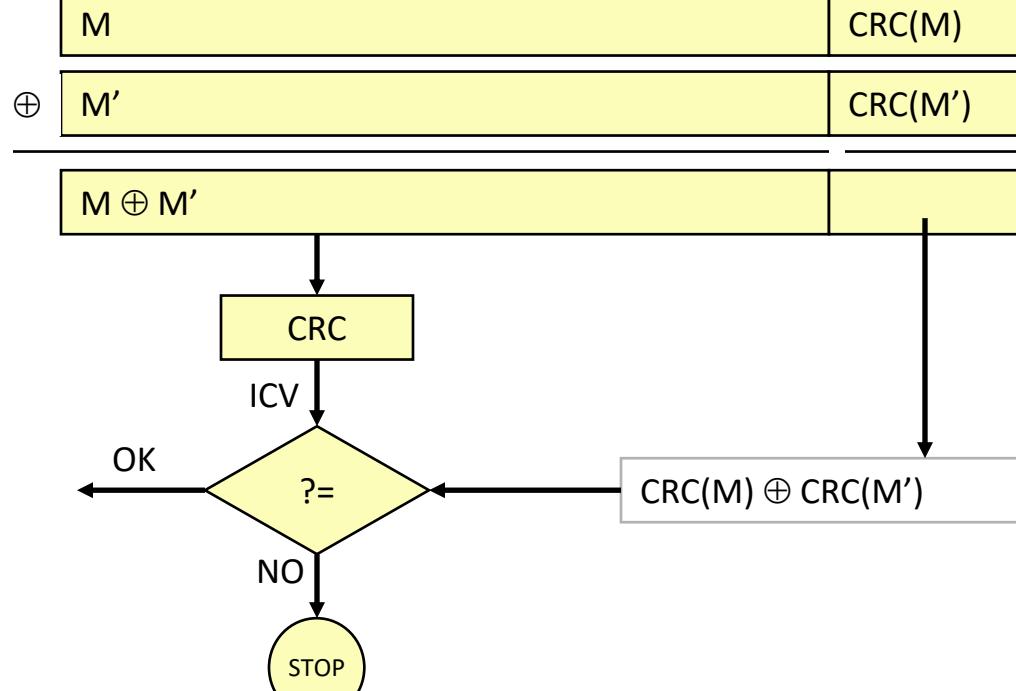
$$(M', \text{CRC}(M')) \oplus (M, \text{CRC}(M)) \oplus \text{RC4}(\text{IV}, K)$$

Receiver decrypts:

$$(M', \text{CRC}(M')) \oplus (M, \text{CRC}(M))$$

Receiver checks CRC
(always successful here),

$$\text{CRC}(a \oplus b) = \text{CRC}(a) \oplus \text{CRC}(b)$$





WEP: Vulnerabilities

5. No mutual authentication

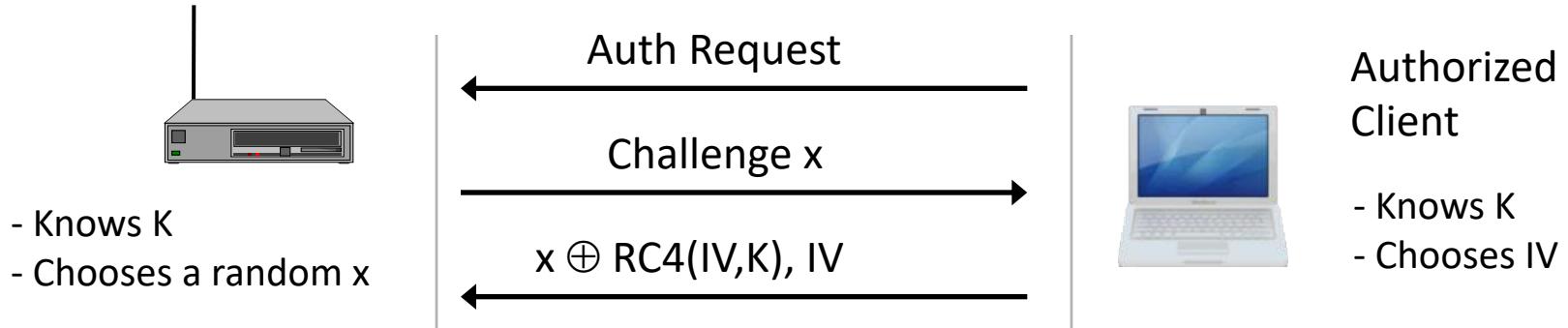
- No protection against false Access Points

6. Ineffective authentication

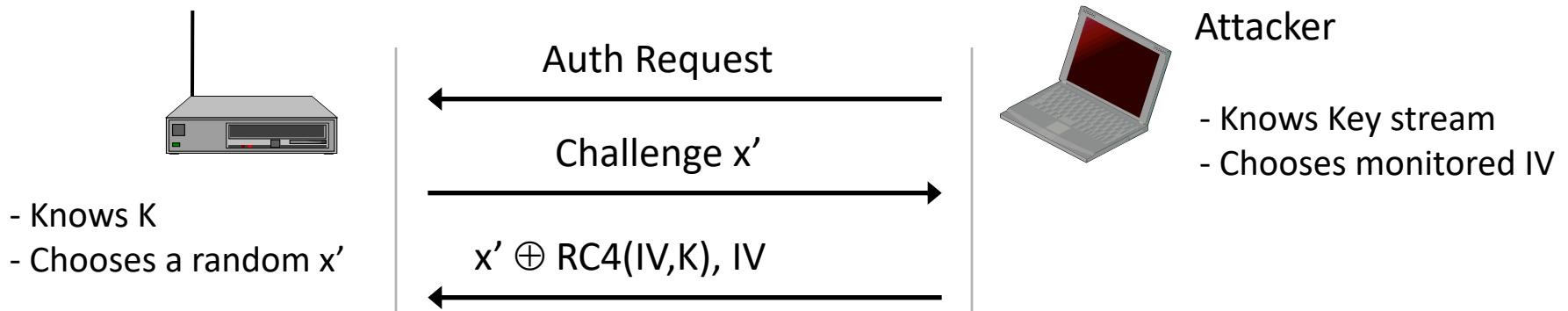
- Attacker eavesdrops Challenge-Response-Pairs (x/C)
 - Knows $x=M$ and C (and IV)
 - Calculates Key stream = $RC4(IV, K)$
- Attacker opens his own Session
 - Receives a Challenge x'
 - Calculates: $x' \oplus RC4(IV, K)$
 - Weakness: Attacker chooses same IV



WEP: Vulnerabilities: Ineffective authentication



- Attacker monitors IV, x and $x \oplus \text{RC4}(\text{IV}, K)$
- Calculates Key stream $\text{RC4}(\text{IV}, K)$ from x





Development of WiFi Security

- Evolution steps
 - WEP128
 - WEPplus
 - Fast Packet Keying
 - WEP2
 - EAP (Extensible Authentication Protocol)
 - WPA (WiFi Protected Access)
- IEEE 802.11i
 - »WPA2«
 - covers some of the evolutional extensions by one standard



Comparison of WEP, WPA, WPA2

	WEP	WPA	WPA2
Encryption	RC4	RC4	AES
Key length	40 Bit	128 Bit	128 Bit
IV	24 Bit	48 Bit	48 Bit
Data integrity	CRC-32	Michael	CCM
Header integrity	–	Michael	CCM
Replay attacks	–	IV sequence	IV sequence
Key management	–	Based on EAP	Based on EAP



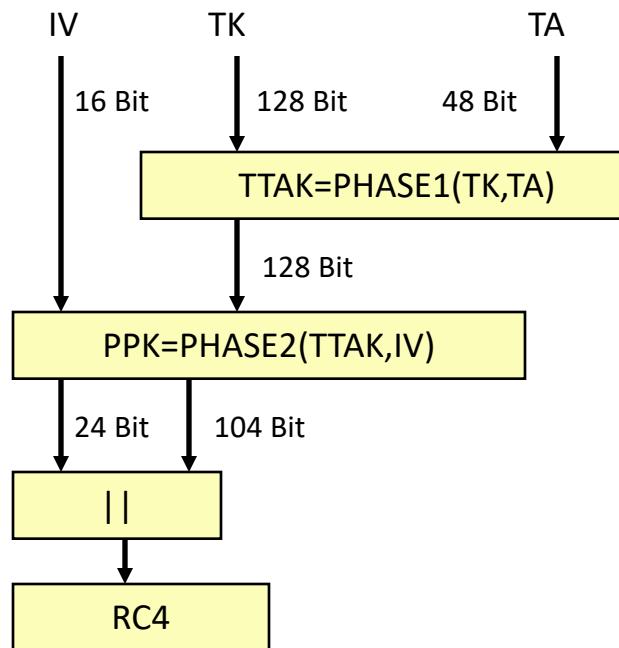
Evolutionary solutions

- WEP128
 - Proprietary extension of WEP standard
 - WEP with 128 bit encryption (24 Bit IV plus 104 Bit key)
- WEPplus
 - Another proprietary extension of WEP standard
 - Defined by Agere Systems (ORiNOCO-Chipset producer)
 - Prevent occurrence of weak IVs
- Unsolved:
 - No useful authentication
 - No cryptographic integrity
 - Replay/repetition of IVs still very likely



Fast Packet Keying

- Extension for WEP by RSA Security Inc. (Developer of RC4)
 - prevent weak IVs
 - prevent repeated combinations of IV and Key
 - Key stream = $\text{RC4}(\text{PHASE2}(\text{PHASE1}(\text{TK}, \text{TA}), \text{IV}))$



TK

Temporal Key

TA

Transmitter Address

PPK

Per Packet Key

TTAK

Key Mixing of TK and TA

Phase 1:

Key Mixing

Phase 2:

Generating a Per Packet Key



Fast Packet Keying

- **Functionality**
 - Symmetric key TK (Temporal Key), 128 Bit
 - Key Mixing: new key is generated from TK and device address TA (Transmitter Address), 48 Bit
 - Packet Key Generation: 24 Bit IV and WEP Key is generated from a 16 Bit IV and mixed key
 - Input of RC4 is repeated after $4 \cdot 10^{21}$ years
- **Unsolved:**
 - No useful authentication
 - No cryptographic integrity



WEP2

- Task Group i (TGi) within IEEE:
 - Objective: Improvement of WEP
 - New standards: WEP2, WPA, WPA2
- WEP2
 - Extension of IV to 128 Bit
 - Optional authentication of Access Points and Clients via Kerberos
 - Introduction of Session Keys
- Problems:
 - Replay of IVs still possible
 - Weak IVs not excluded
 - Security vulnerability in Kerberos
 - Ineffective authentication



EAP

- Extensible Authentication Protocol
 - Introduced for Remote Access with Dial-In connections
 - Part of 802.1X standard
 - Authentication and key management
 - Low implementation costs in Access Points (AP)
 - No firmware-Upgrade necessary
- Functionality
 - Three systems involved: Client, AP, Authentication server
 - AP works as a proxy between client and Authentication server
 - AP grants access to network after successful authentication



WPA (WiFi Protected Access)

- WPA is part of IEEE 802.11i
- Functionality
 - Authentication via EAP
 - Encryption based on RC4 with 128 Bit keys
 - New cryptographic integrity protection by alg. »Michael«
 - Mechanism to negotiate key length and authentication procedure
 - either: Session Key Distribution over RADIUS servers (Remote Authentication Dial-In User Service)
 - or: without server via Broadcast/Multicast
 - IV is incremented with each packet (prevent replay of IV)



WPA (WiFi Protected Access)

- WPA is part of IEEE 802.11i
- Problems
 - Broadcast/Multicast key is known to all stations
 - »Michael« is relatively weak: $O(2^{20..30})$
 - 1-minute shut-down of AP while receiving more than one wrong authenticate packet (within a given time)
 - Denial-of-Service attacks easy
 - Possible improvements:
 - Reduction of deactivation/disconnection time (ca. 100ms)
 - After n authentication errors, renegotiate Session Keys



802.11i

- WPA2-Standard adopted in July 2004
 - Includes WPA
 - Requires hardware upgrade of AP and Client
- Functionality
 - Authentication via EAP
 - AES for encryption
 - New protocol for integrity protection