

Rechnerarchitekturpraktikum: RC5

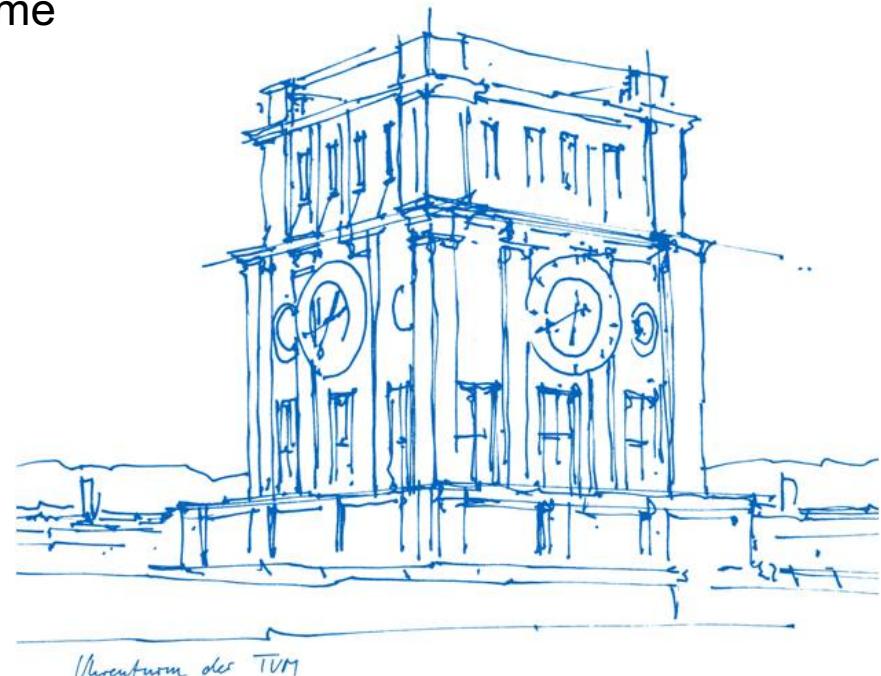
Mete Polat, Jonas Hübotter, Simon Bohnen

Technische Universität München

Fakultät für Informatik

Lehrstuhl für Rechnerarchitektur & Parallele Systeme

München, 20. August 2019



Inhalt

- Einleitung
- PKCS#7-Padding
- Betriebsmodi
- Aufbau von RC5
- Dateiaufbau
- Optimierung
- Performance
- Sicherheit

RC5

- Symmetrisch
- Blockchiffre
- Variable Blockgröße: Geeignet für unterschiedliche Registergrößen
- Iterativ, mit variabler Rundenanzahl
- Variable Schlüsselgröße
- Unsere Konfiguration: RC5-16/16/b

PKCS#7-Padding

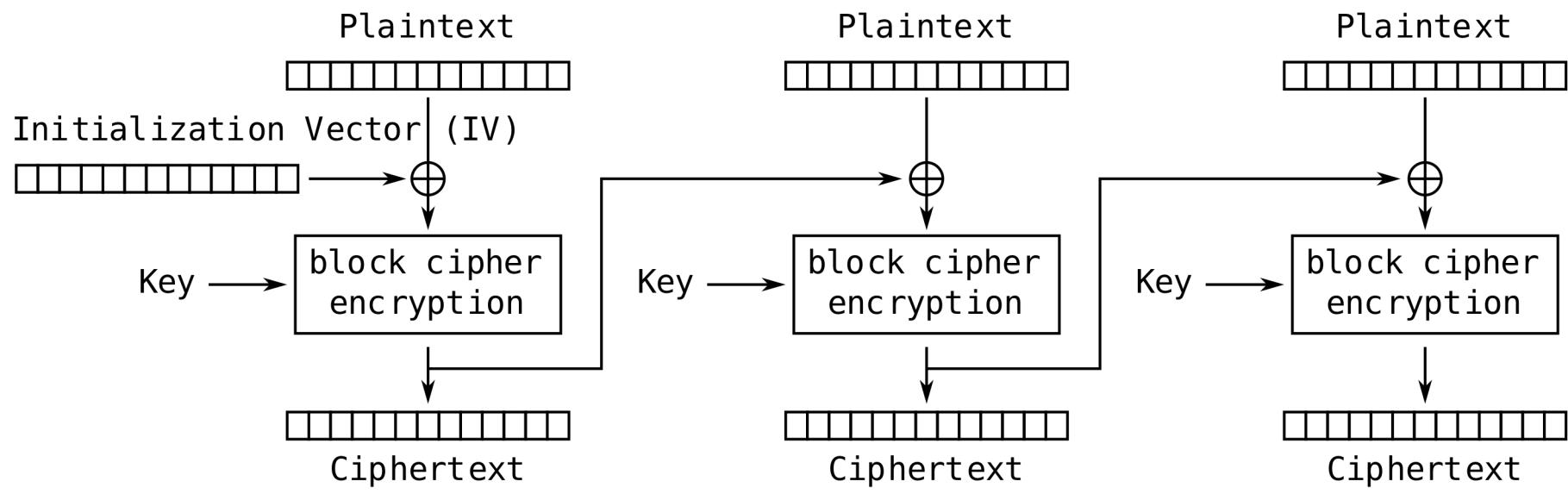
- Erweiterung des Plaintextes auf ein Vielfaches der Blockgröße
- Alle Padding-Bytes enthalten Länge des Paddings
- Padding kann immer eindeutig entfernt werden

Betriebsmodi

- Erlaubt Verschlüsselung von Nachrichten mit variabler Länge

Wie kombiniert man die Ciphertextblöcke?

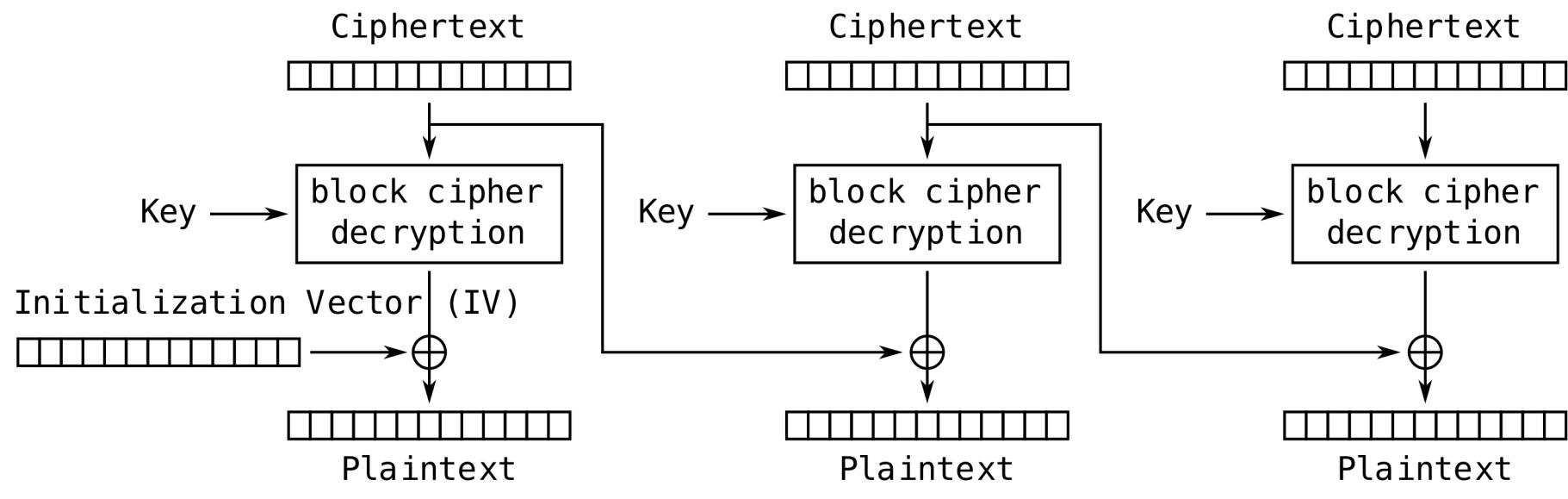
Cipher Block Chaining



Cipher Block Chaining (CBC) mode encryption

Bild 1)

Cipher Block Chaining



Cipher Block Chaining (CBC) mode decryption

Bild 2)

Counter Mode

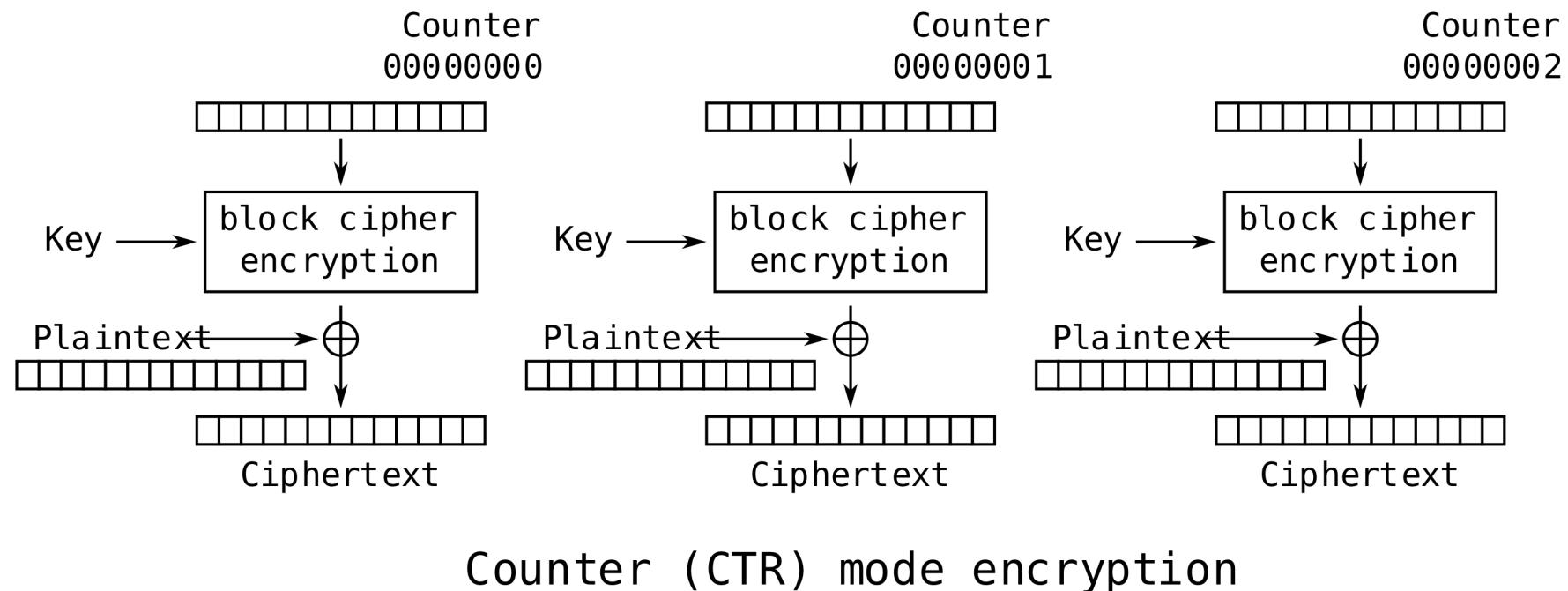


Bild 3)

Counter Mode

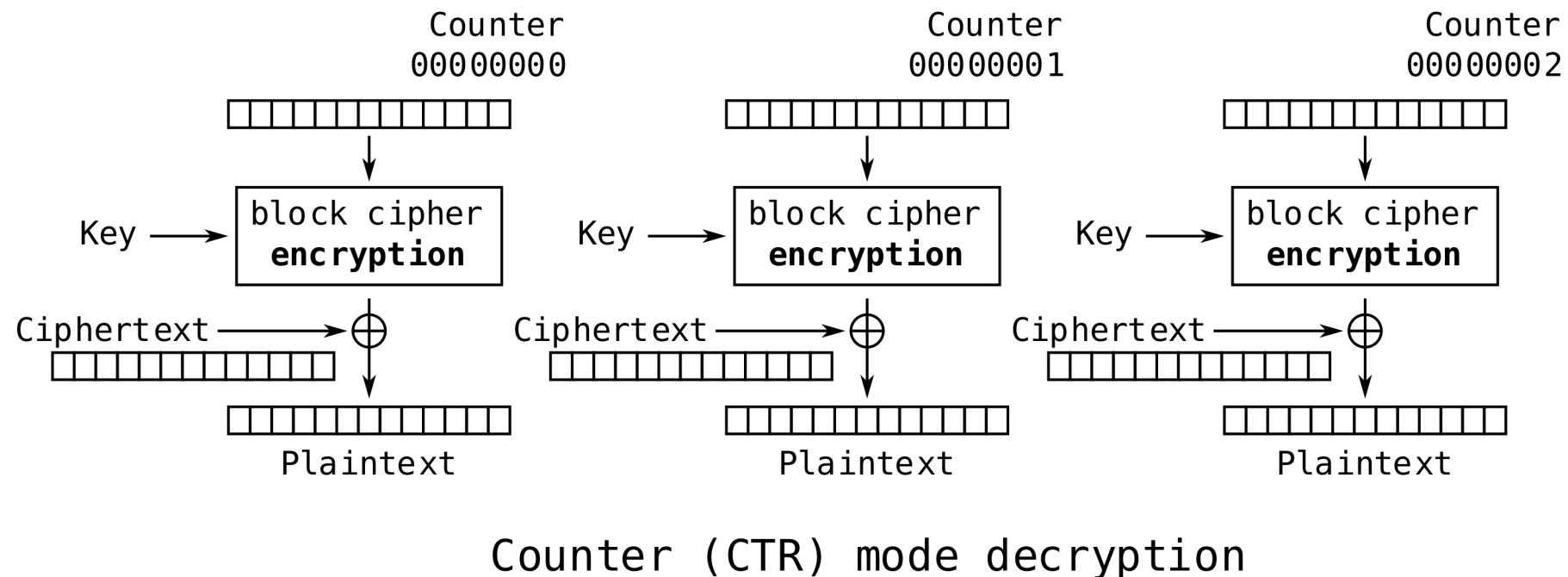
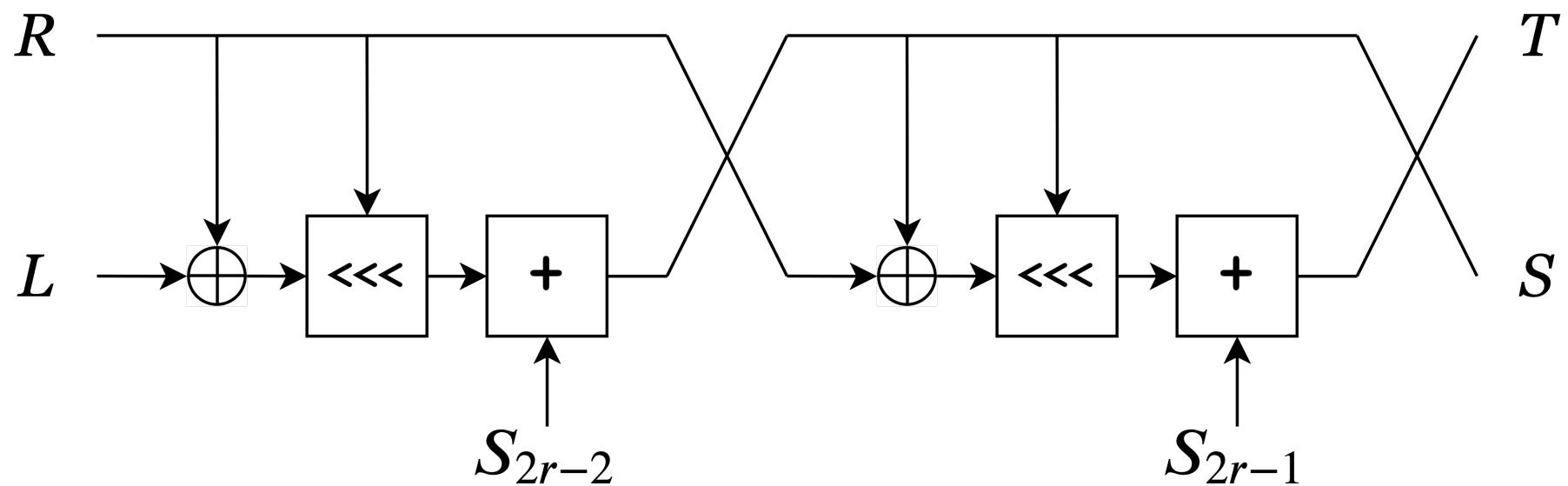


Bild 4)

RC5

- Block wird in zwei 16-Bit-Halbblöcke aufgeteilt
- In jeder Runde werden diese Halbblöcke mit zwei Rundenschlüsseln kombiniert
- Vor der Verschlüsselung werden aus dem Schlüssel $2r + 2 = 34$ Rundenschlüssel generiert

RC5



r bezeichnet die aktuelle Runde

S bezeichnet die erweiterte Schlüsseltabelle

Dateiaufbau

```
$ echo 'Hello World!' > hello.txt
```

```
$ hexdump hello.txt
```

```
0000000 6548 6c6c 206f 6f57 6c72 2164 000a  
000000d
```

```
$ rc5 enc -m cbc myKey hello.txt hello.enc
```

```
$ hexdump hello.enc
```

```
0000000 5374 270c 2cd3 1935 62d9 2f51 a34c af2b  
0000010 2a67 f82d  
0000014
```

Dateiaufbau

```
$ echo 'Hello World!' > hello.txt
```

```
$ hexdump hello.txt
```

```
0000000 6548 6c6c 206f 6f57 6c72 2164 000a  
000000d
```

```
$ rc5 enc -m cbc myKey hello.txt hello.enc
```

```
$ hexdump hello.enc
```

```
0000000 5374 270c 2cd3 1935 62d9 2f51 a34c af2b  
0000010 2a67 f82d  
0000014
```

● Verschlüsseltes Padding

● Initialisierungsvektor

Optimierung

- Schlüsselexpansion liefert bei gegebener Blockgröße immer das gleiche Ergebnis
- Berechnung vor dem Kompilieren
- Speicherung des Ergebnisses in der .data-Sektion

Optimierung

Unsere Implementierung

```
movd r11d, xmm0  
pshufd xmm0, xmm0, 0b00111001  
xor r9w, r10w  
mov cl, r10b  
rol r9w, cl  
add r9w, r11w
```

```
shr r11d, 16  
xor r10w, r9w  
mov cl, r9b  
rol r10w, cl  
add r10w, r11w  
add rax, 2
```

RFC2040-Implementierung

```
mov ecx, eax  
xor r8d, eax  
add rdx, 0x4  
and ecx, 0xf  
rol r8w, cl  
add r8w, WORD PTR [rdx-0x4]
```

```
mov ecx, r8d  
xor eax, r8d  
and ecx, 0xf  
rol ax, cl  
add ax, WORD PTR [rdx-0x2]
```

Optimierung

Problem

- CBC und RC5 erreichen kryptografische Sicherheit durch Abhangigkeit der Blocke
- Erschwert parallele Verarbeitung, z.B. durch SIMD

Alternative

- Counter Mode erlaubt parallele Verschlusselung

Counter Mode

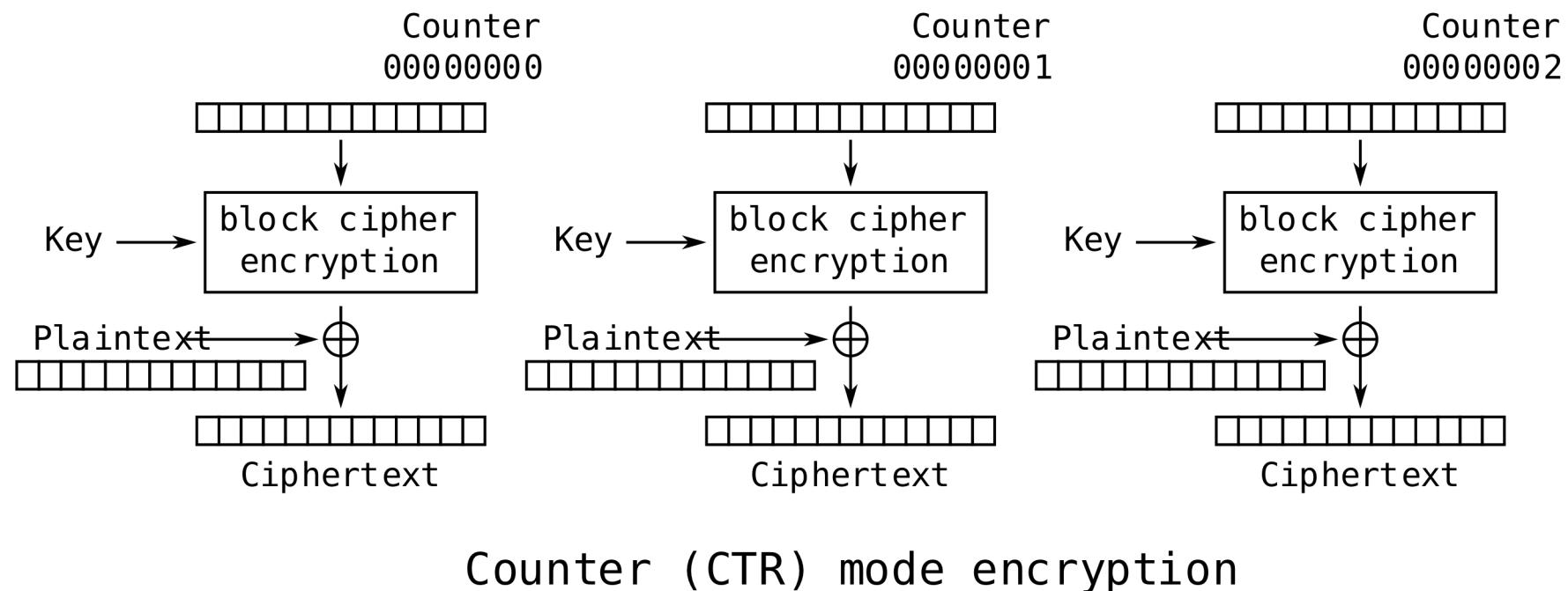
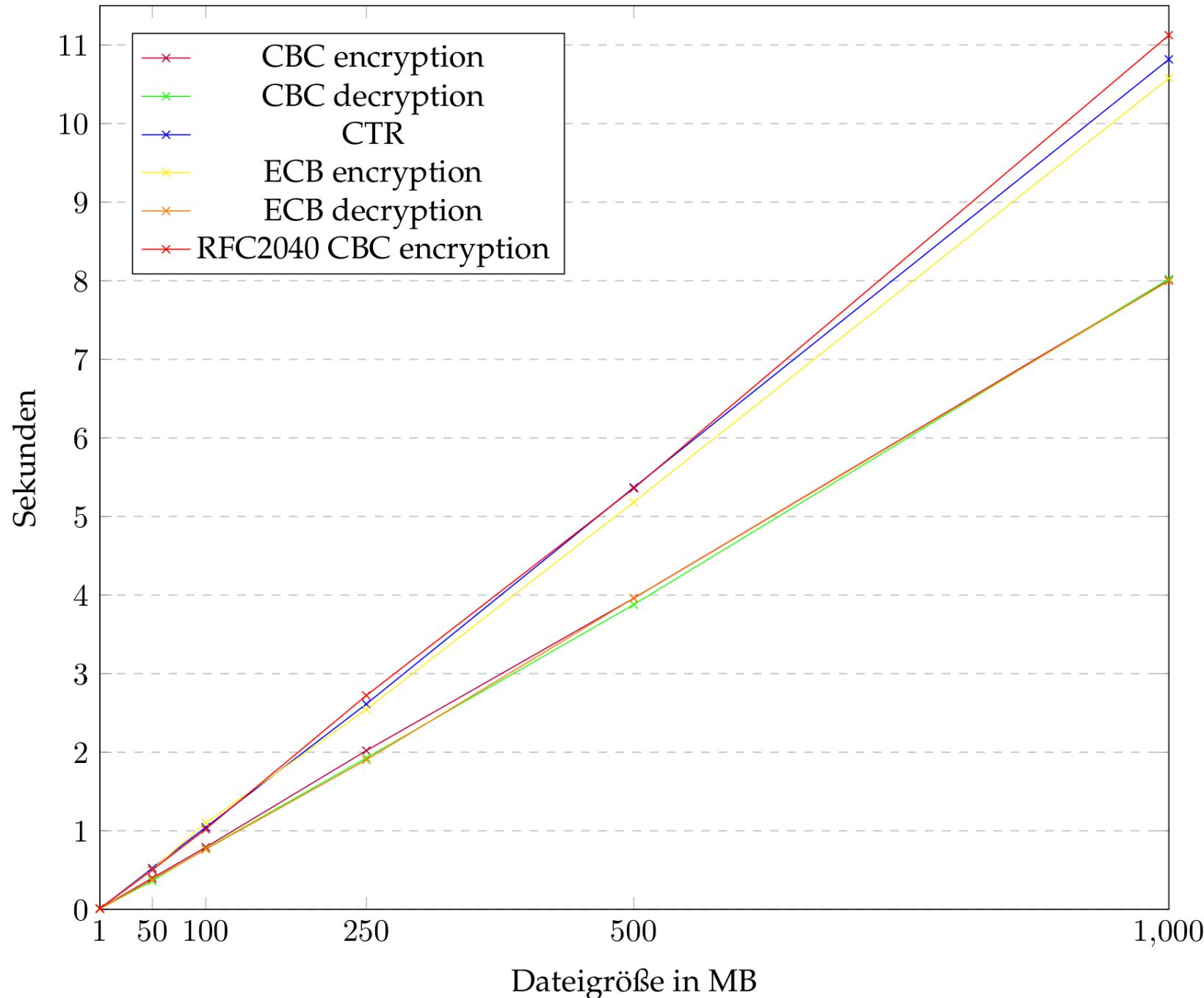


Bild 3)

Vergleich der Betriebsmodi inkl. der Referenzimplementierung



Sicherheit

Wird von folgenden Faktoren beeinflusst:

- Schlüssellänge
- Blockgröße
- Betriebsmodus
- Eigenschaften und Konfiguration der Chiffre

Sicherheit

Blockgröße

Die minimale Erfolgswahrscheinlichkeit eines Angriffs auf eine 32-Bit Blockchiffre mit CBC kann durch die Anzahl der verschlüsselten Blöcke q abgeschätzt werden:

$$\epsilon \geq \frac{q^2}{2^{32}}$$

Beispiel

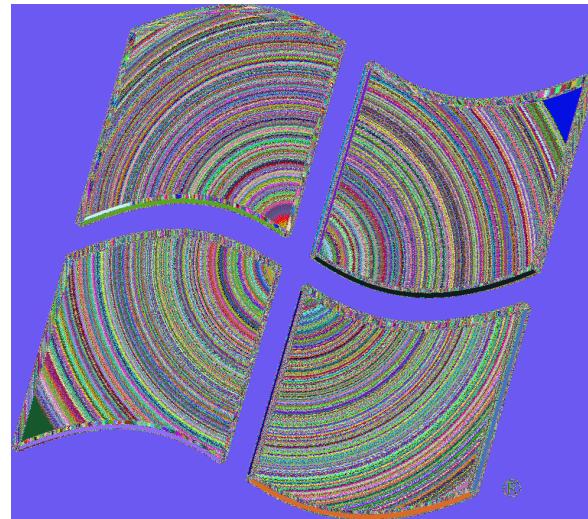
Für $\epsilon = 2^{-20}$ gilt $q \leq 2^6$

Sicherheit

Betriebsmodi



Original
Bild 5)



ECB



CTR und CBC

Sicherheit

Initialisierungsvektor

- Darf öffentlich gespeichert werden
- Muss nicht zufällig sein
- Darf nicht aus vorher bekannten Informationen generiert werden
- `arc4random()` als Pseudozufallsgenerator

Sicherheit

Verbleibende Daten

Speicherinhalt mit sensiblen Daten zurücksetzen

- Freigegebener Speicher könnte anderen Prozessen zugewiesen werden
- Swap-Partition ist gegebenenfalls nicht verschlüsselt

Bildquellen

- 1) https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#/media/File:cbc_encryption.svg
- 2) https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#/media/File:cbc_decryption.svg
- 3) https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#/media/File:ctr_encryption_2.svg
- 4) https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#/media/File:ctr_decryption_2.svg
- 5) https://en.wikipedia.org/wiki/Windows_7#/media/File:Windows_logo_-_2006.svg