Die MD2-Hashfunktion

Dorian Zedler, Finn Dröge und Thomas Florian

Gliederung

1. Einleitung - Die MD2-Hashfunktion
2. Kriterien einer Hashfunktion
3. Alternativen zur MD2-Hashfunktion
4. Berechnung des MD2-Hashes
    a. Padding
    b. Prüfsumme
    c. Finale Berechnung
5. Implementierungen
6. Unsere Ergebnisse (Performance)
    a. Vorteil der zweiten Implementierung
    b. Laufzeitanalyse
    c. Optimierungen
7. Analyse der Ergebnisse mit Ausblick

1. Einleitung - Die MD2-Hashfunktion

- Message-Digest Algorithm 2
- Erstellt von Ronald Rivest in 1989
- Hauptanwendungsgebiete:
  - Passwortsicherung
  - Nachrichtenauthentifizierungscodes
  - digitale Signaturen
- 2004 wurde die MD2-Hashfunktion als unsicher erklärt [2]

[1]

2. Kriterien einer Hashfunktion

Damit eine Funktion als Hashfunktion definiert werden kann, muss sie folgende Kriterien erfüllen:

- Preimage-Resistance:
  - Es sollte schwer sein, eine passende Eingabe zu einer gegebenen Ausgabe zu finden
- Second-Preimage-Resistance:
  - Es sollte schwer sein, für eine gegebene Ein- und Ausgabe einer Hashfunktion, eine weitere Eingabe zu finden, die dieselbe Ausgabe produziert
- Collision-Resistance:
  - Es gibt nur sehr wenige gleiche Hashes für unterschiedliche Eingaben

[3]

2. Kriterien für eine Hashfunktion

- Vor 2004 galt MD2 als sicher:
  - Der Brute-Force-Preimage-Angriff hat die Komplexität 2^{128}
- MD2 gilt als unsicher seit dem Preimage-Angriff von F. Muller in 2004 [2]
  - Aufgrund von Mustererkennungen kann die Komplexität des Preimage-Angriffs verringert werden
- Weitere Angriffe:
  - Optimierter Preimage-Angriff von Søren S. Thomsen [3]
    - Komplexität: 2^{73}
  - Kollisionsangriff von L. Knudsen et al. [4]
    - Komplexität: 2^{54} vs. Komplexität mit Geburtstagsparadoxon: 2^{64}

3. Alternativen zur MD2-Hashfunktion

- MD4 und MD5 (Ron Rivest)
  - Beide auch als unsicher erklärt [5] [6]
- Secure Hashing Algorithms (SHA)
  - Beispielsweise SHA-256 (stand August 2015 noch sicher) [7]

4. Berechnung des MD2-Hashes

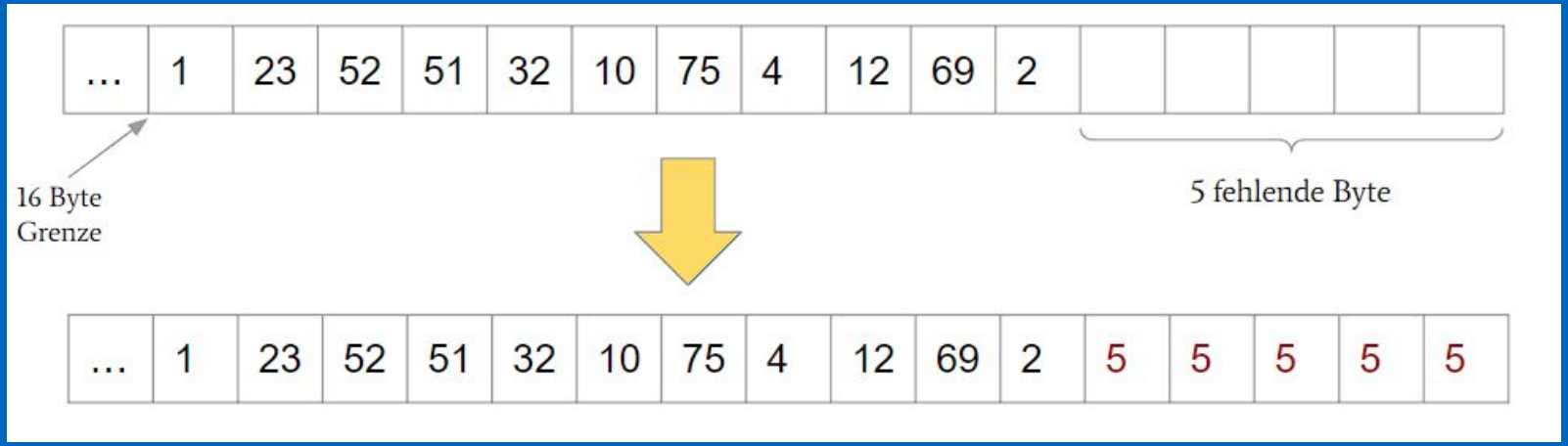
a. Padding

b. Prüfsumme

c. Finale Berechnung

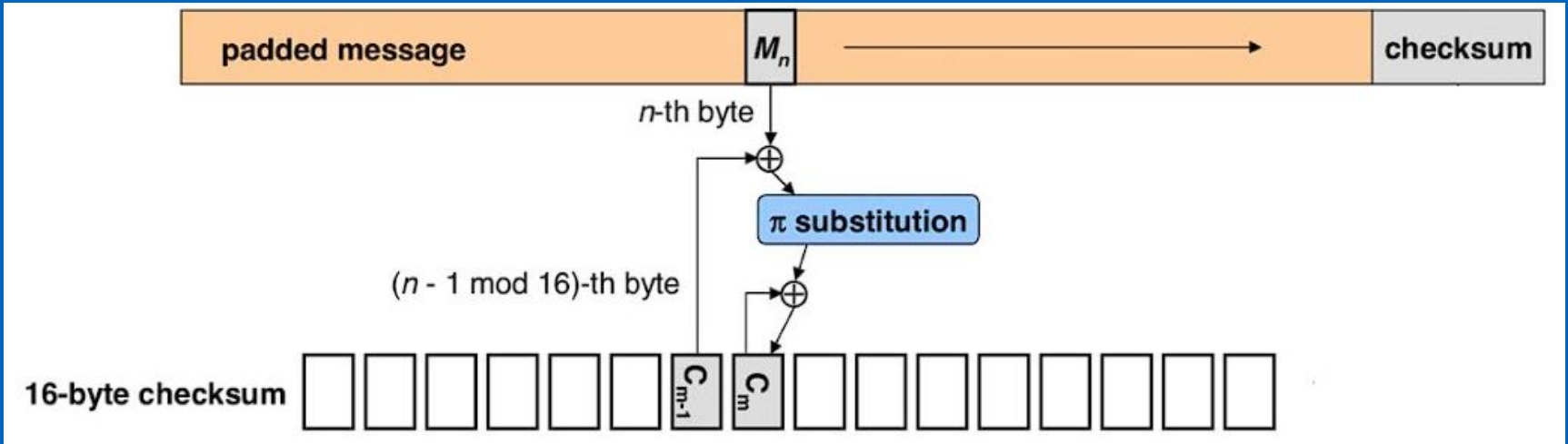
4.a. Padding

- Die Nachricht wird auf eine 16-Byte Grenze ausgerichtet
- Beispiel:



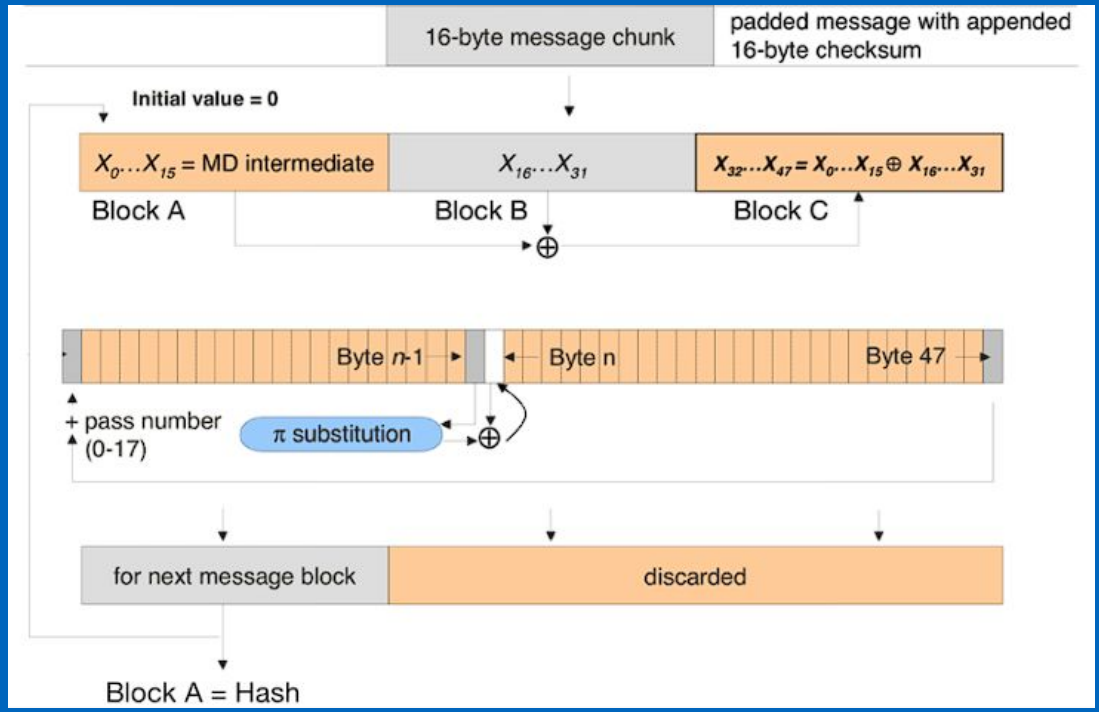
4.b. Prüfsumme

- Weitere 16 Byte werden angehangen
- Die Werte der Bytes werden mithilfe der Nachricht gebildet:



4.c. Finale Berechnung

- Ein Buffer mit 48 Byte wird initialisiert
- Die Werte des Buffers werden mithilfe der Nachricht mit Padding und Prüfsumme berechnet
- Das Ergebnis steht zum Schluss in den ersten 16 Byte des Buffers



5. Implementierungen

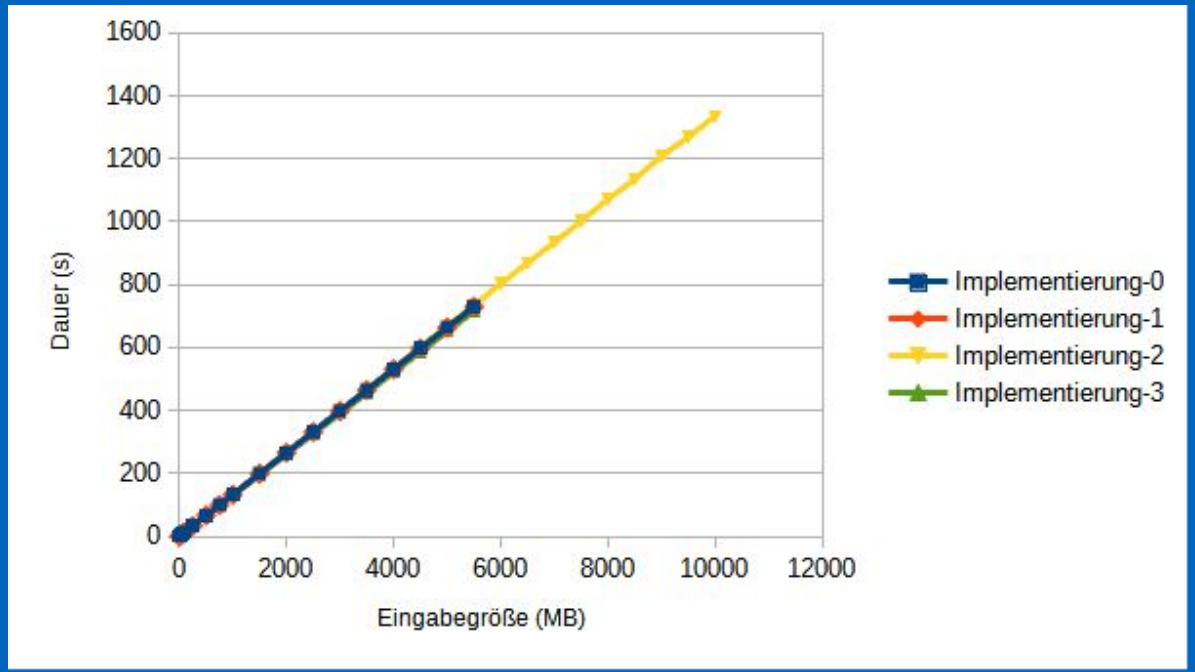
- Basis-Implementierung
- Optimierte Implementierung mit SIMD
- Implementierung mit partiellem Einlesen der Datei
- Optimierte Implementierung mit Threading
- Referenzimplementierung

6. Unsere Ergebnisse (Performance)

a. Vorteil der zweiten Implementierung
b. Laufzeitanalyse
c. Optimierungen

6.a. Vorteil der zweiten Implementierung

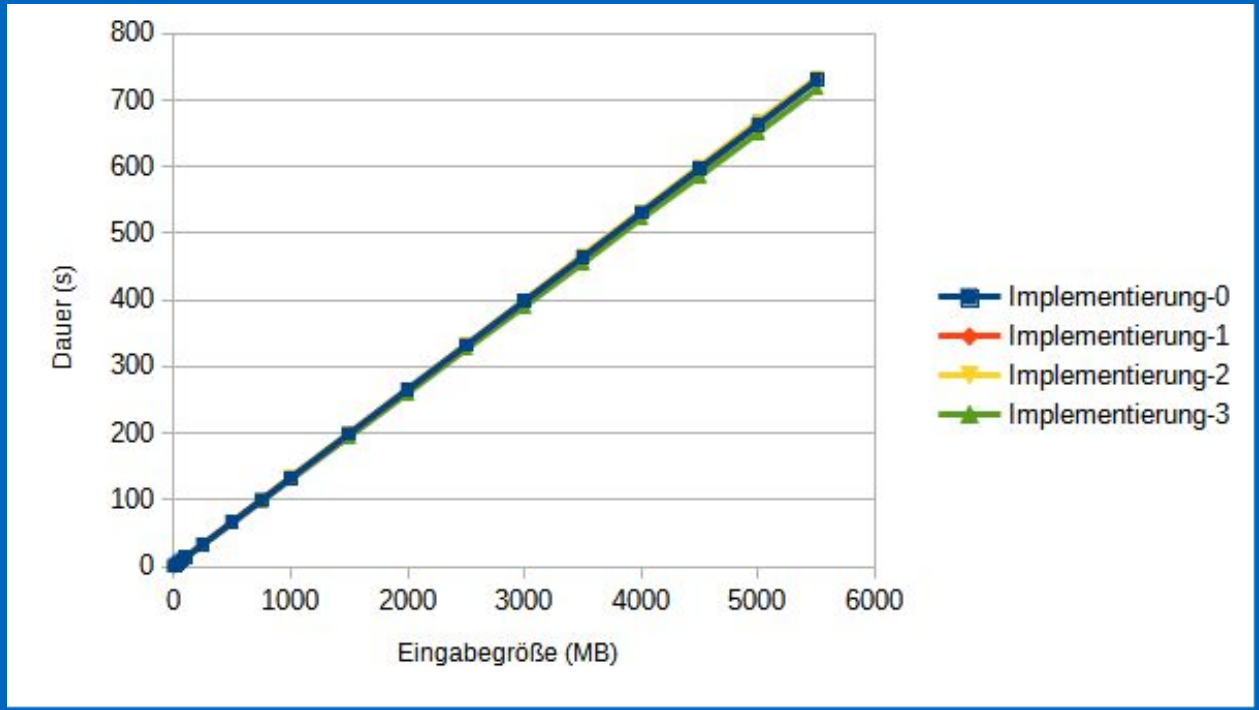
Beliebig große Eingaben können gewählt werden



6.b. Laufzeitanalyse

Laufzeitklasse:

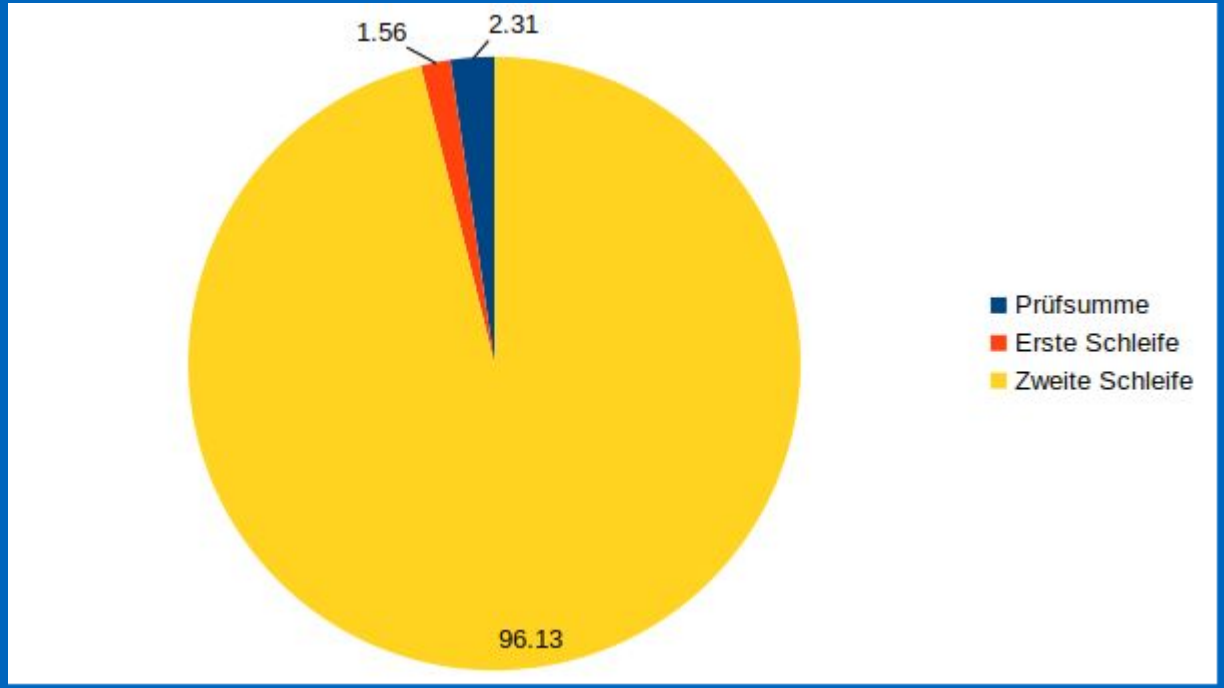
O(n)



6.c. Optimierungen

Die zweite Schleife ist nicht optimierbar

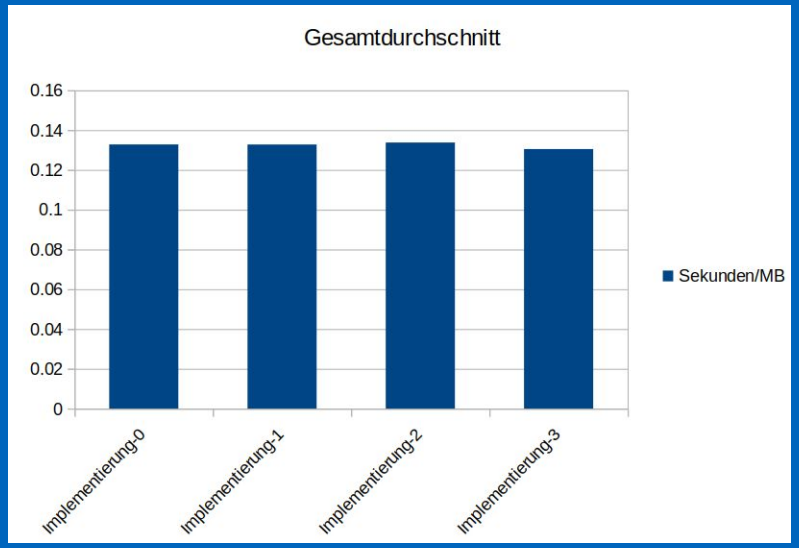
➔ Optimierungen an der ersten Schleife und Prüfsumme bewirken nur eine sehr geringe Verbesserung



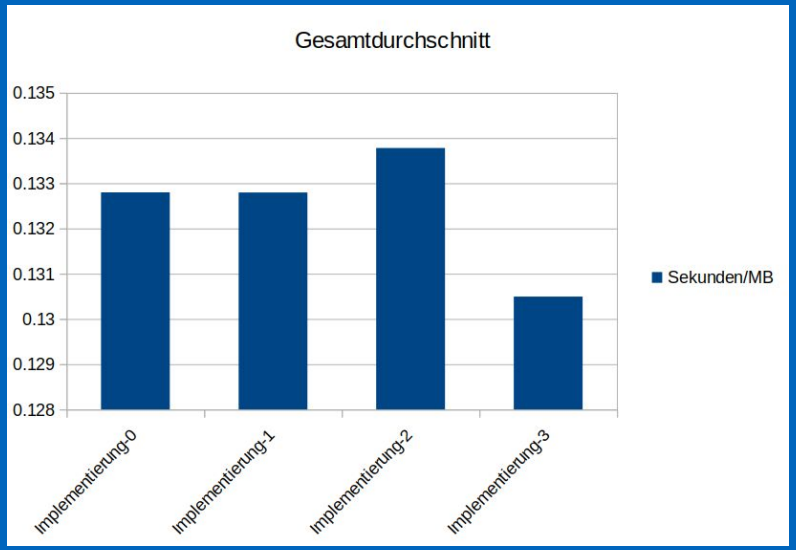
Gemessen wurde mit 100 Megabyte Eingabedaten und dem Durchschnitt über zehn Wiederholungen

6.c. Optimierungen

Daraus ergeben sich folgende Laufzeiten:



Laufzeiten (normiert)



Laufzeiten (nicht normiert)

7. Analyse der Ergebnisse (Ausblick)

- MD2 ist kaum optimierbar
- Optimierungen können sich möglicherweise aus dem Vorgehen der Preimage-Angriffe folgern lassen
- Durch Moore’s Law veralten nach gewisser Zeit alle Hashfunktionen
    - Man benötigt immer größere und komplexere Hashfunktionen

Quellen

1. https://en.wikipedia.org/wiki/Ron_Rivest
2. https://www.iacr.org/archive/asiacrypt2004/33290211/33290211.pdf
3. https://eprint.iacr.org/2008/089.pdf
4. https://link.springer.com/content/pdf/10.1007/s00145-009-9054-1.pdf
5. https://web.archive.org/web/20030523231212/http://dsns.csie.nctu.edu.tw/research/crypto/HTML/PDF/C91/194.PDF
6. http://merlot.usc.edu/csac-f06/papers/Wang05a.pdf
7. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf