Analyse und Implementierung des MD2-Algorithmus

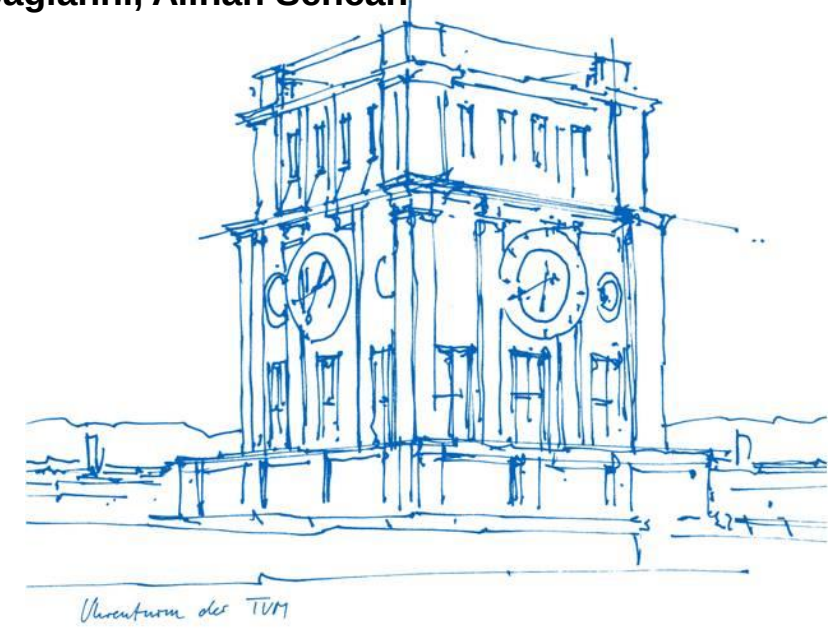
Maha Marhag, Magdalena Papagianni, Alihan Sencan

Technische Universität München

Lehrstuhl für Rechnerarchitektur und parallele Systeme

Grundlagepraktikum Rechnerarchitektur

München, 29 August 2022



Agenda

- Problemstellung
- Hashfunktionen
- Message-Digest 2
- Padding
- Checksum und Hashfunktion
- Korrektheit
- Performanzanalyse
- Optimierungen
- Lösungsalternativen

Problemstellung

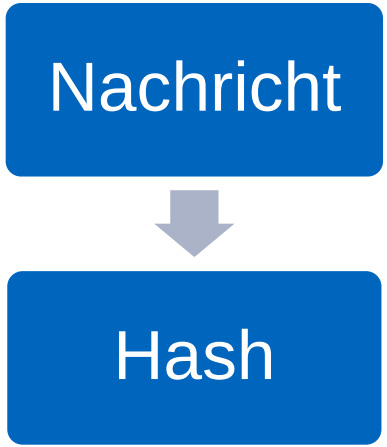
- MD2 Hashfunktion analysieren und implementieren
- Lösungsalternativen
- Optimierungen
- Korrektheit
- Performanzanalyze

Hashfunktionen - Grundaufbau

Hashfunktionen - Sicherheitseigenschaften

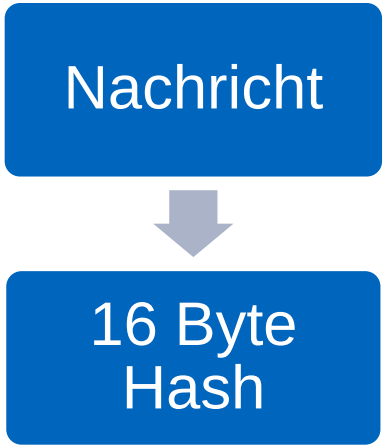
Kryptographische Hashfunktionen zeichnen sich aus durch:

1. Urbildresistenz
2. Zweite Urbildresistenz
3. Kollisionsresistenz

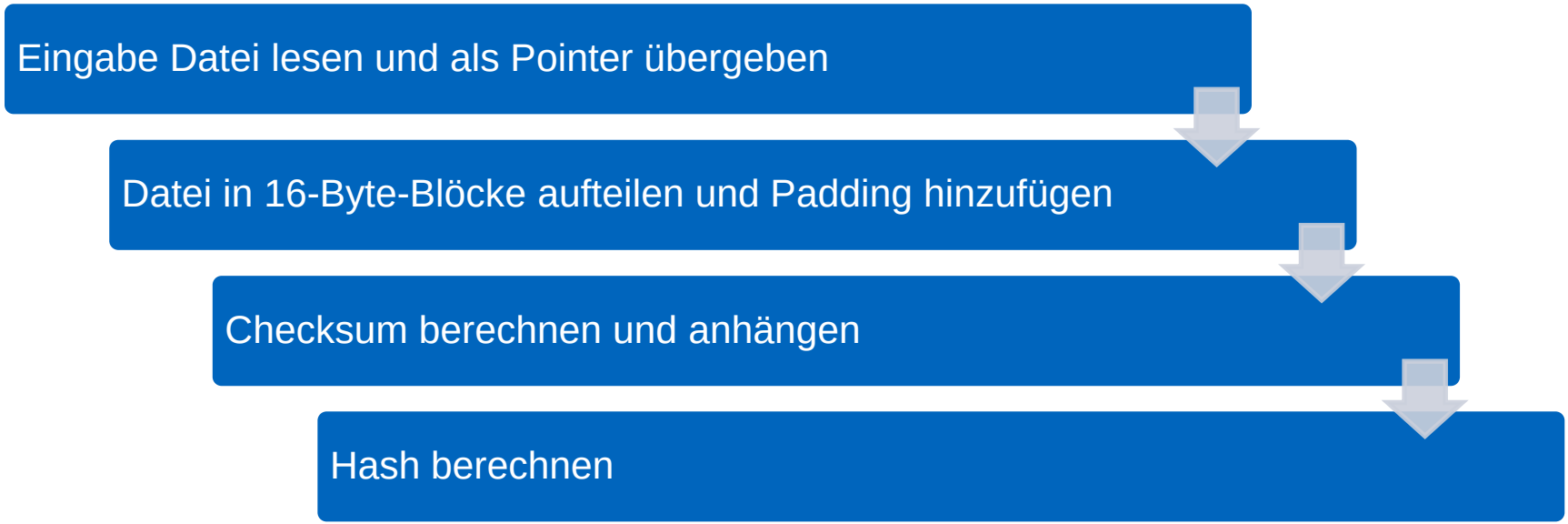


Message-Digest 2

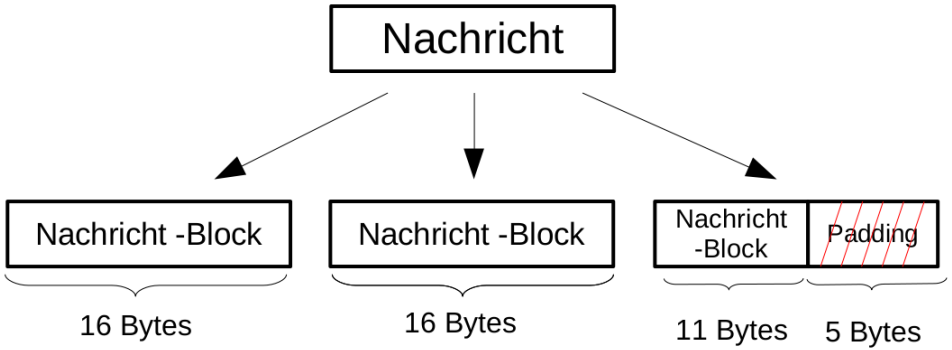
- 1989 von Ronald Rivest entwickelte Hashfunktion
- für digitale Signaturen konzipiert
- Gilt nicht mehr als sicher
- 2011 durch die IETF als „historisch“ eingestuft

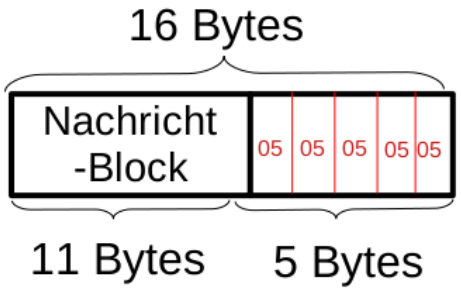


MD2 - Grundfunktionen

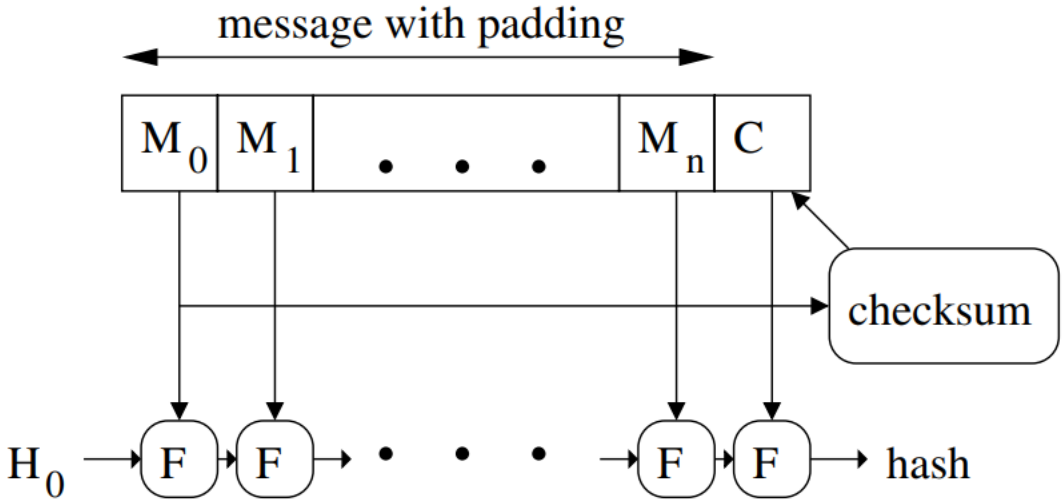


Padding


Nachricht
Nachricht -Block
Nachricht -Block
Nachricht -Block
Padding
16 Bytes
16 Bytes
11 Bytes
5 Bytes


16 Bytes
Nachricht -Block
05 05 05 05 05
11 Bytes
5 Bytes

Checksum und Hashfunktion



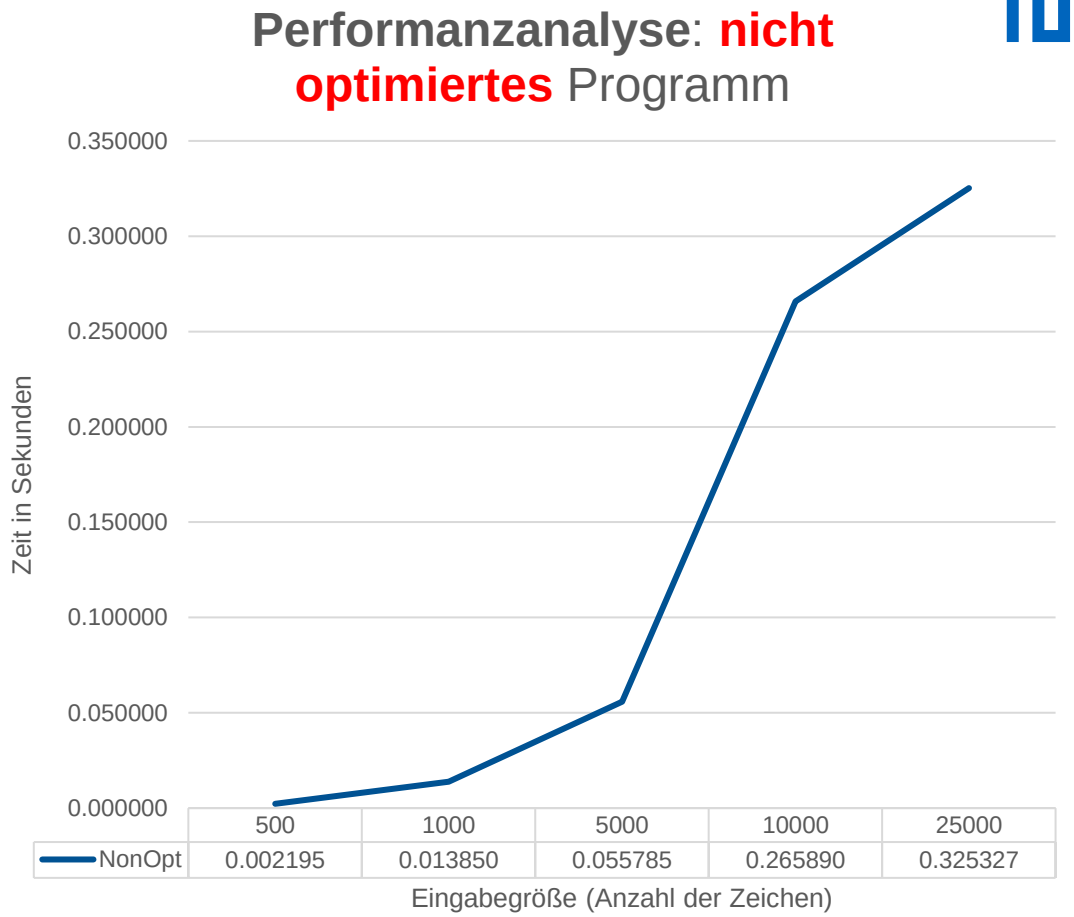
Korrektheit und Beispiele

- Eine Reihe von Tests mit verschiedenen Eingabegrößen:
  - 500, 1.000, 5.000, 10.000, 25.000 und 100.000 Zeichen
- Getestet wurde sowohl lokal als auch auf der Rechnerhalle
- Korrektheit mit Beispielen aus der RFC Dokumentation geprüft

Eingabe	Hash
Leere Datei	8350e5a3e24c153df2275c9f80692773
.... . .-.. .-.. --- / .-- --- .-. .-.. -.. / -.-.--	c6dc54d633087cbe5046d25129e1467c
abc	da853b0d3f88d99b30283a69e6ded6bb
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789	da33def2a42df13975352846c30338cd

Performanzanalyse

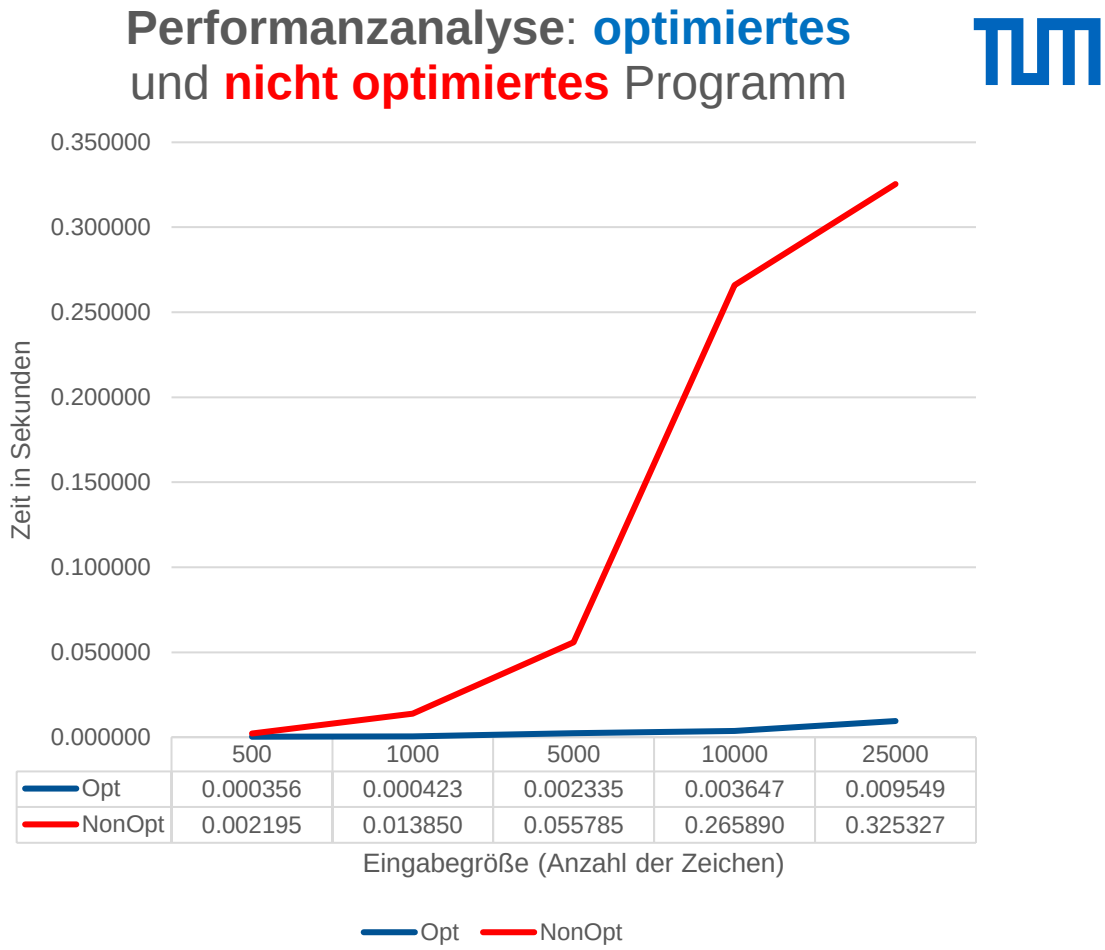
- Gemeßen mit `MONOTONIC` Clock
- Verschiedenen Eingabegrößen:
  - ➢ von 500 bis 25.000 Zeichen
- Mehrfach lokal gemessen und Ergebnisse gemittelt



Optimierungen

- Pad-Bytes werden nicht mehr iterativ kopiert, sondern mit `MEMSET` geschrieben
- Arrays werden auch mit `MEMSET` initialisiert
- Effizientere vektorisierte `SIMD-XOR`-Operationen werden eingesetzt

Performanzanalyze nach den Optimierungen



Lösungsalternative

Verwendung von Struct: benutzerdefinierter Datentyp "MD2_HASH„

Pros:

+ Kombination von Daten unterschiedlicher Typen
+ Vereinfacht den Aufbau des Quelltexts
+ Lesbarkeit und Wartbarkeit des Programms

Cons:

- Enspricht nicht der vorgegebenen Signatur

Zusammenfassung und Ausblick

- MD2 selbst zu implementieren war eine interessante und lehrreiche Erfahrung
- Tiefen Einblick in den Aufbau und die Einzelteile von Hashfunktionen erhalten
- Optimierungen wirken sich sehr auf die Laufzeit aus
- Ausblick: wir würden vermehrt `SIMD`-Operationen verwenden

Vielen Dank für Ihre Aufmerksamkeit!

Maha Marhag

Magdalena Papagianni

Alihan Sencan