SS 2023 · Grundlagenpraktikum: Rechnerarchitektur

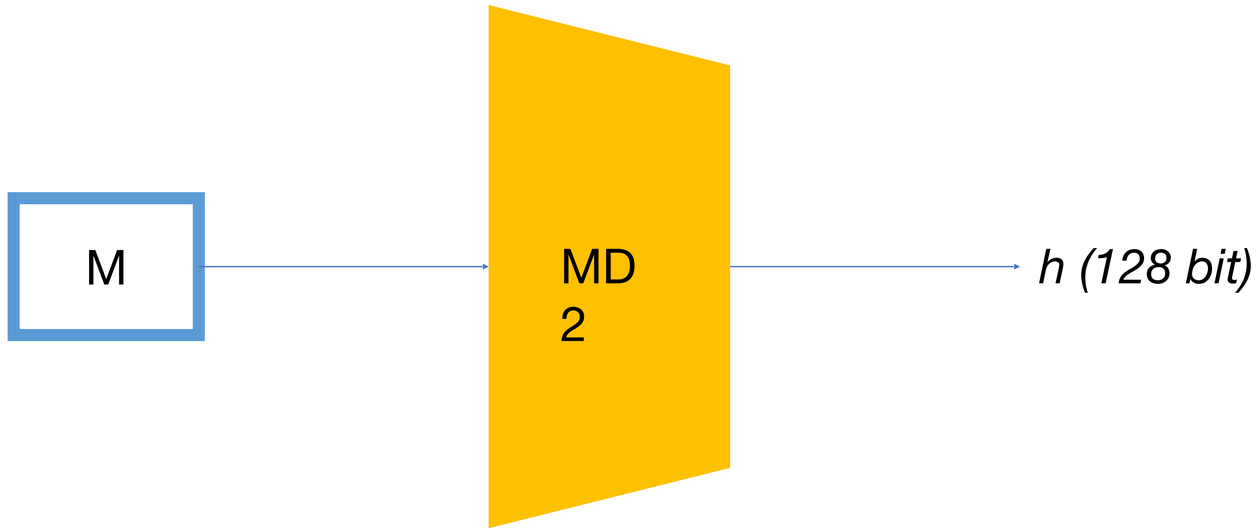
MD2 Hashing

Yiyang Xie, Awar Satar, Jingjing Yu

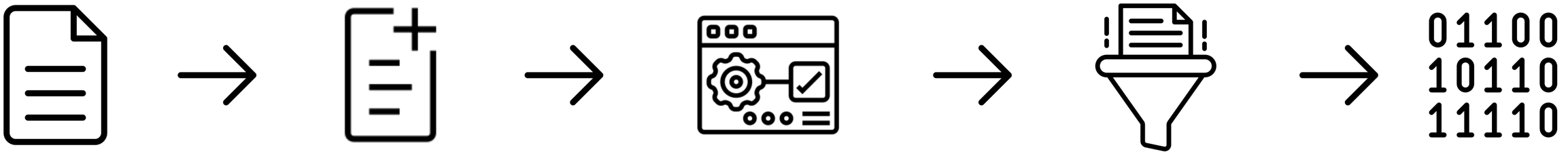
Inhaltsverzeichnis

Einleitung

- MD2 (Message Digest Algorithm 2) ist eine kryptografische Hash-Funktion, die 1989 von Ronald Rivest entwickelt wurde.
- Erzeugt für einen Eingabetext beliebiger Länge eine Hash-Ausgabe der Länge 128 bit
- Der berechnete „Hashwert“ soll nicht umkehrbar sein
- MD2 ist für 8-Bit Maschinen ausgelegt, die zum Zeitpunkt seiner Entwicklung die gängigste Architektur waren



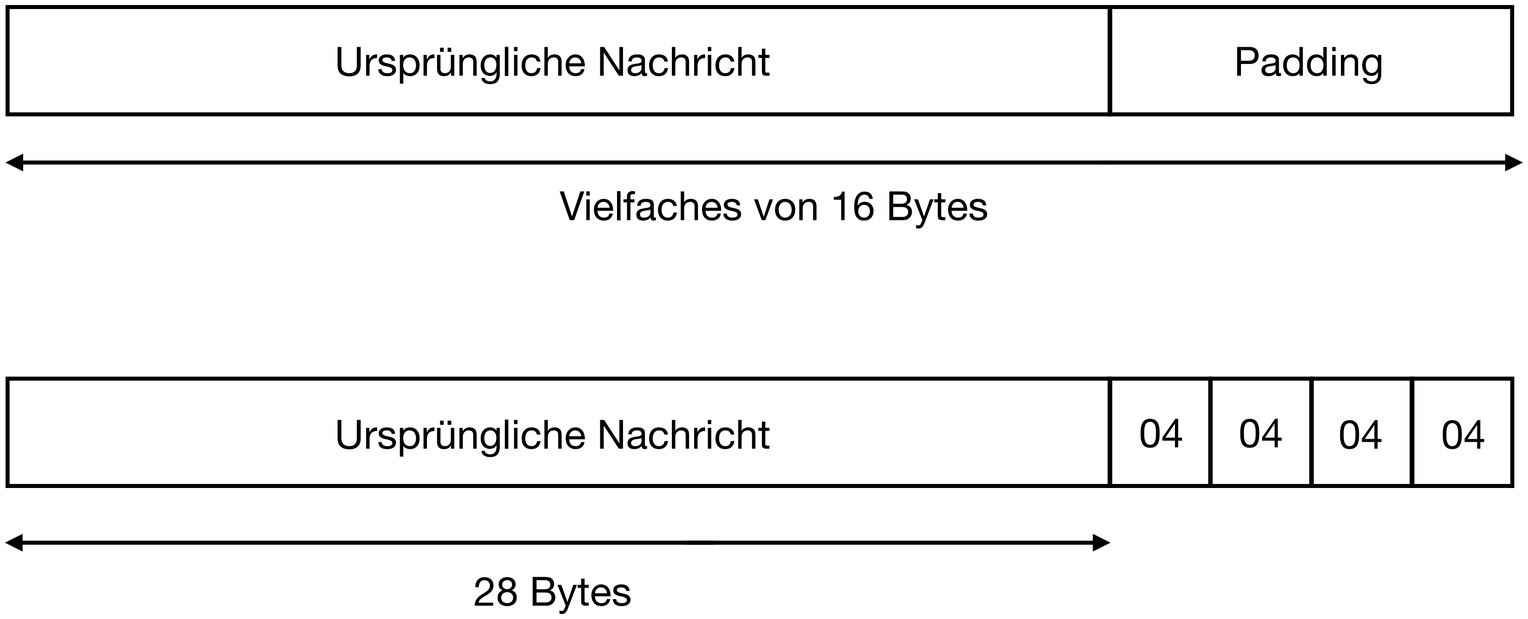
Lösungsansatz



Datei → Padding → Prüfsumme → Hash → Hashwert

Lösungsansatz - Padding

PKCS#7: Padding Wert = Padding Größe

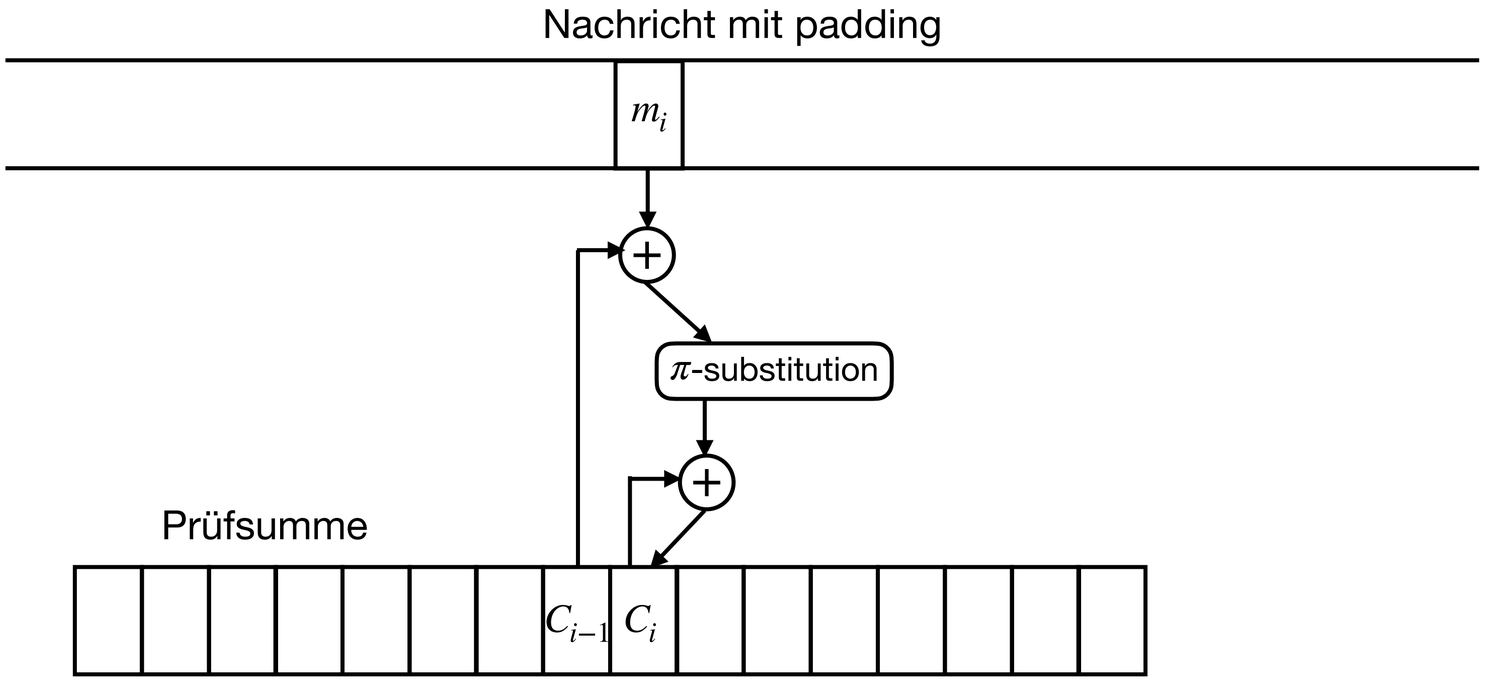


Lösungsansatz - Prüfsumme

Initialisiere einen 16-Byte-Block $C = C_0 C_1 \ldots C_{15}$ auf 0

Verarbeite M in 16-Byte-Blöcken.

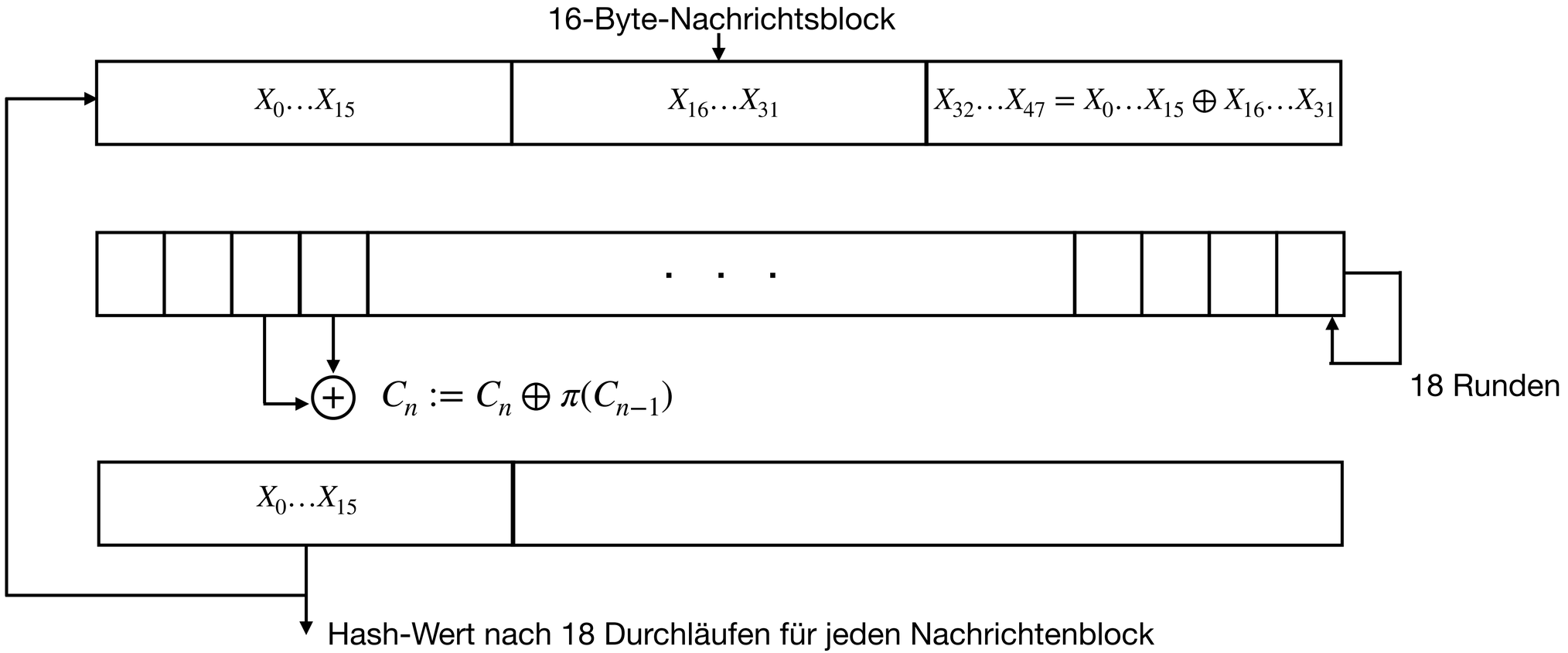
Byte C_i der Prüfsumme: $C_i = C_i \oplus S[m_i \oplus C_{i-1}]$



Lösungsansatz - Hash

Initialisiere einen 48-Byte-Block $X = X_0 X_1 \ldots X_{47}$ auf 0

Verarbeite M in 16-Byte-Blöcken



Lösungsansatz

V0:

SIMD-Anweisungen werden verwendet, um Daten zwischen Puffern auszutauschen.

V1:

Memcpy() and memset() werden verwendet, um Daten zwischen Puffern auszutauschen.

Sicherheitsanalyse - Sicherheitsanforderungen

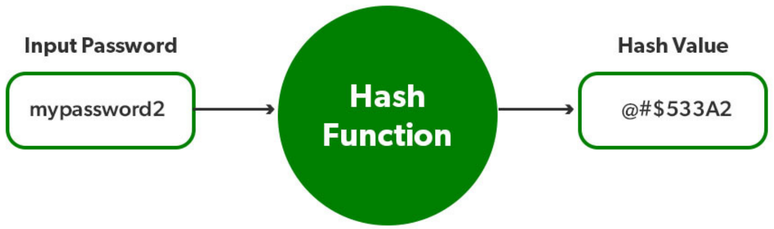
Die folgenden Anforderungen werden an eine kryptografische Hashfunktion H gestellt, damit sie als „sicher“ klassifiziert werden kann:

- Einwegeigeschaft (auch gen. pre-Image resistance): Sei $h = H(m)$, so ist es nicht möglich ein m mit $m = H^{-1}(h)$ in effizienter Zeit zu berechen
- Schwache Kollisionsresistenz (auch gen. second pre-image resistance): Es ist nicht in effizienter Zeit möglich ein $m \neq m'$ zu finden mit $H(m) = H(m')$
- Starke Kollisionsresistenz: Sei $h = H(m)$, so ist es nicht möglich ein m_k mit $H(m_k) = H(m)$ in effizienter Zeit zu berechen

Sicherheitsanalyse - Anwendungsbereiche

MD2 ist mittlerweile aus Sicherheitsgründen obsolet. Trotzdem gab es verschiedene Anwendungsbereiche für den Algorithmus

Nutzerauthentifizierung:



Dateiintegrität:

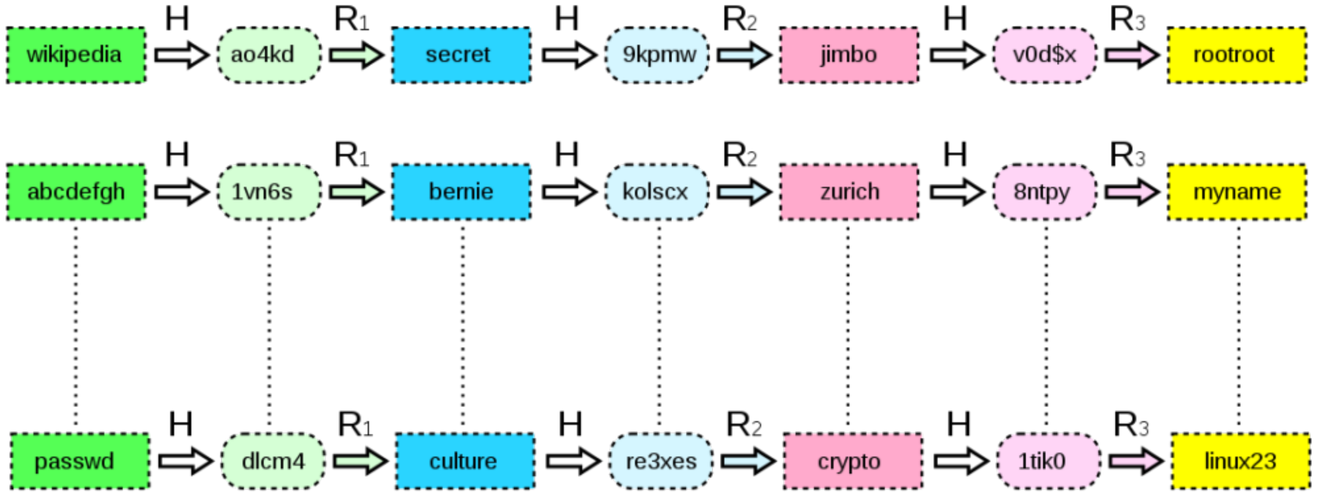
Sicherheitsanalyse - MD2 Sicherheitsprofil

Von der Verwendung von MD2 im Internet wird seit 2011 offiziell abgeraten. Seit 2004 wird der Algorithmus schon von moderneren Alternativen (z.B. MD5) abgelöst. Gründe:

- Angriffsvektor ermöglicht durch Initalvektor 0.
- Kollision in nur 2^{64} Rechenoperationen mit hoher Wahrscheinlichkeit auffindbar
- Korrektes Pre-Image seit 2004 in nur2^{104} Rechenoperationen auffindbar
- NIST schreibt eine Bit Sicherheit von 128 als Mindestmaß vor

Sicherheitsanalyse - Angriffsszenario

Vorberechnete Tabelle an gehashten Passwörter und Klartext. Dies erlaubt einen Angreifer sich die Rechenzeit zu sparen, solange er Zugriff auf diese Tabelle hat



Die Tabelle besteht aus Ketten von Hashwerten und Klartext, die mit der Hashfunktion und der sog. Reduktionsfunktion verbunden sind.

Sicherheitsanalyse - Maßnahmen

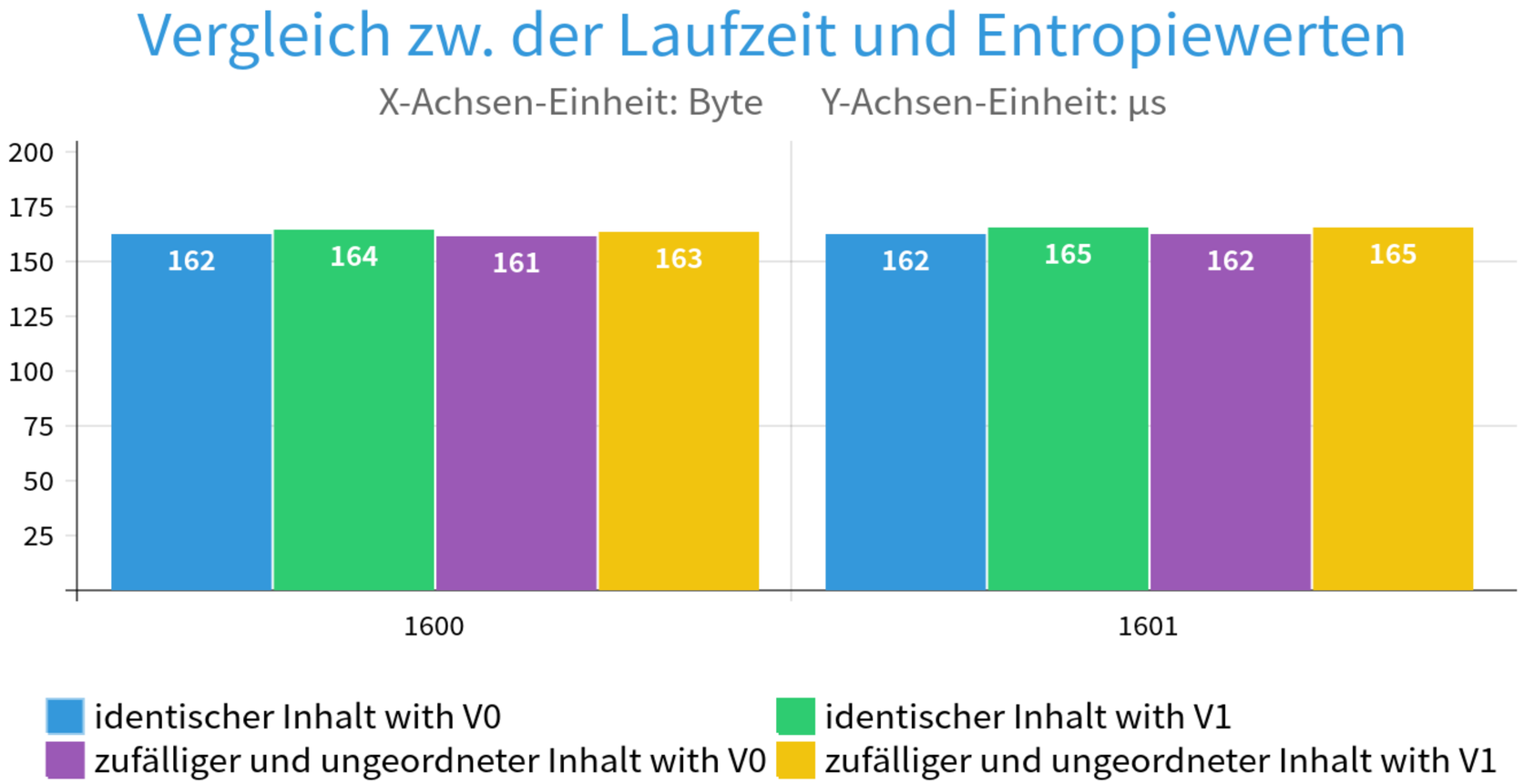
Einige Angiffe kann man durch einfache Maßnahmen deutlich erschweren

- Salting: Durch Hinzufügen eines unbekannten, zufälligen salts an den cleartext vor dem Hashing wird die Verwendung von Rainbow Tables redundant
- Rehashing: Durch Verwendung einer zweiten Hashfunktion werden die Größe der Hashmap drastisch erhöht und Kollisionen unwahrscheinlicher
- Bessere Hashfunktionen verwenden: NIST rät von der Verwendung von MD2, MD4 und MD5 ab und empfiehlt die Nutzung von SHA-512 sowie SHA-3Einige Angiffe kann man durch einfache Maßnahmen deutlich erschweren

Korrektheit

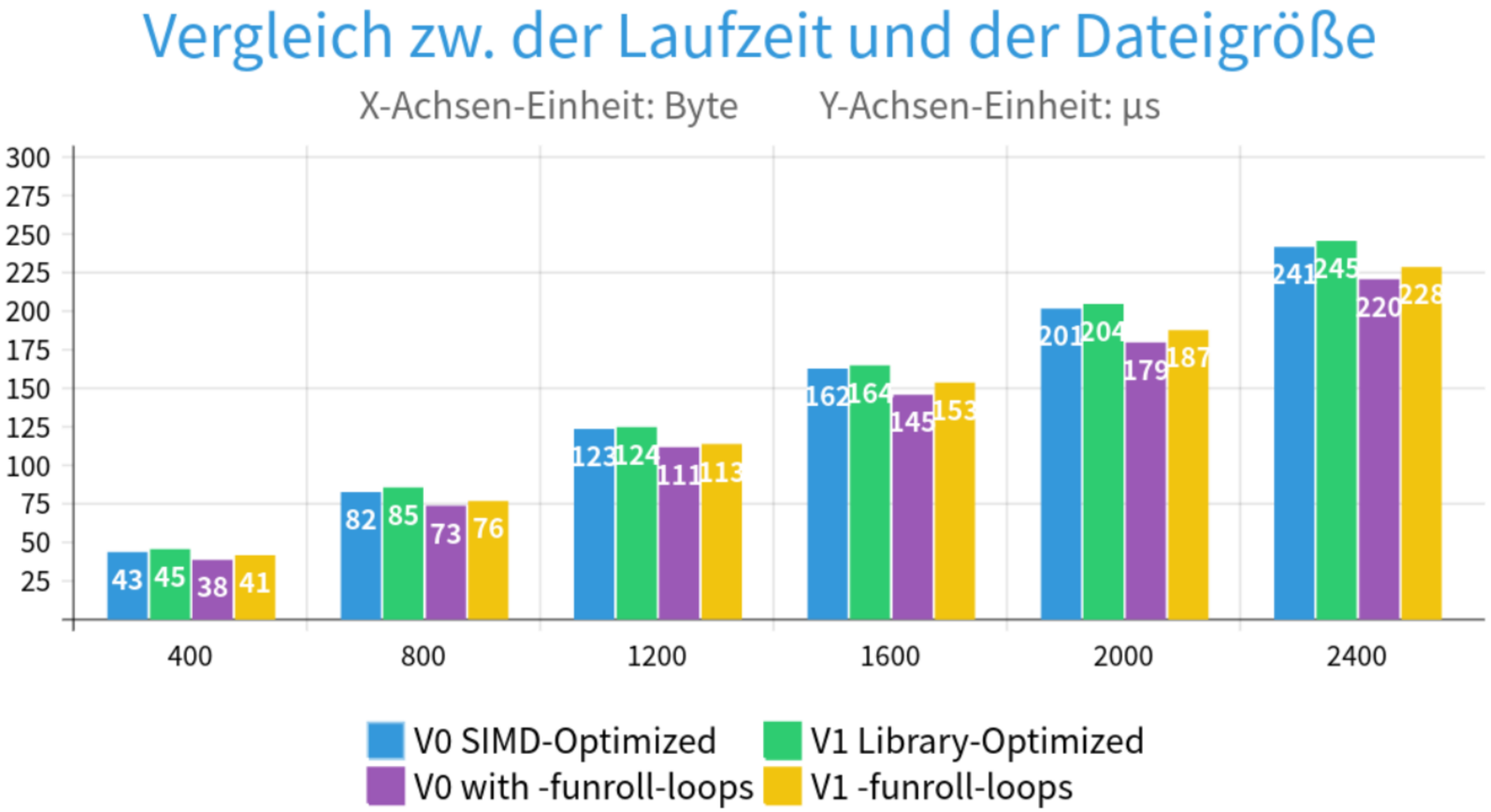
- **Testcases erstellen** – Testcases mit Texteditor generieren
- **Überprüfungen** – Online-MD2-Rechnern
- **Behandlung von Padding –** size_t paddingSize = 16 - (len % 16);
- **Behandlung von Dateigröße –** fstat(fileno(file), &statbuf)
- **Behandlung von Nicht-Text-Dateien –** uint8_t *buf

Performanzanalyse - Laufzeit



Kein Zusammenhang zwischen Entropie (Inhalt Wiederholung) und Laufzeit

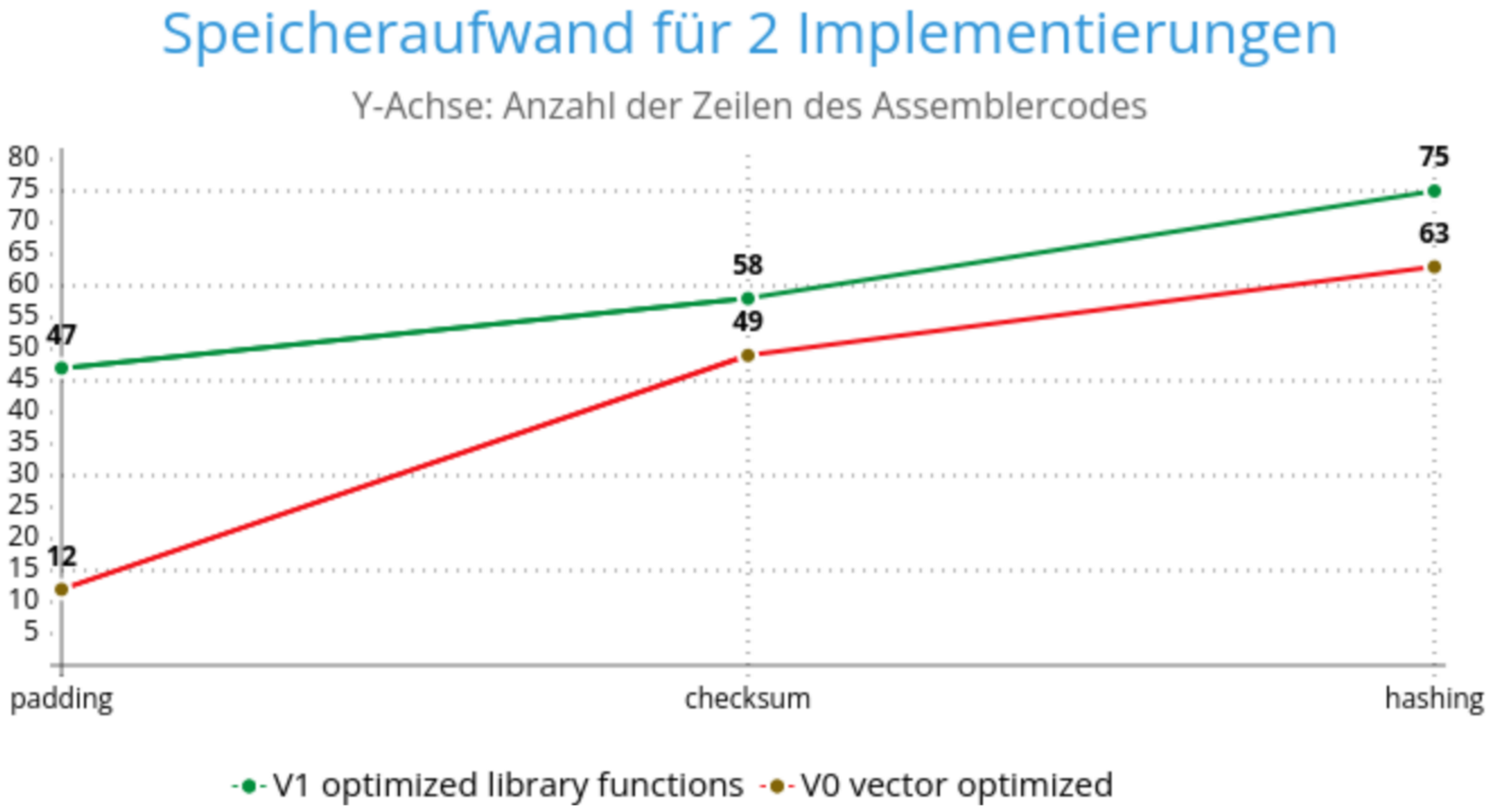
Performanzanalyse - Laufzeit



Die Laufzeit ist proportional zur Dateigröße

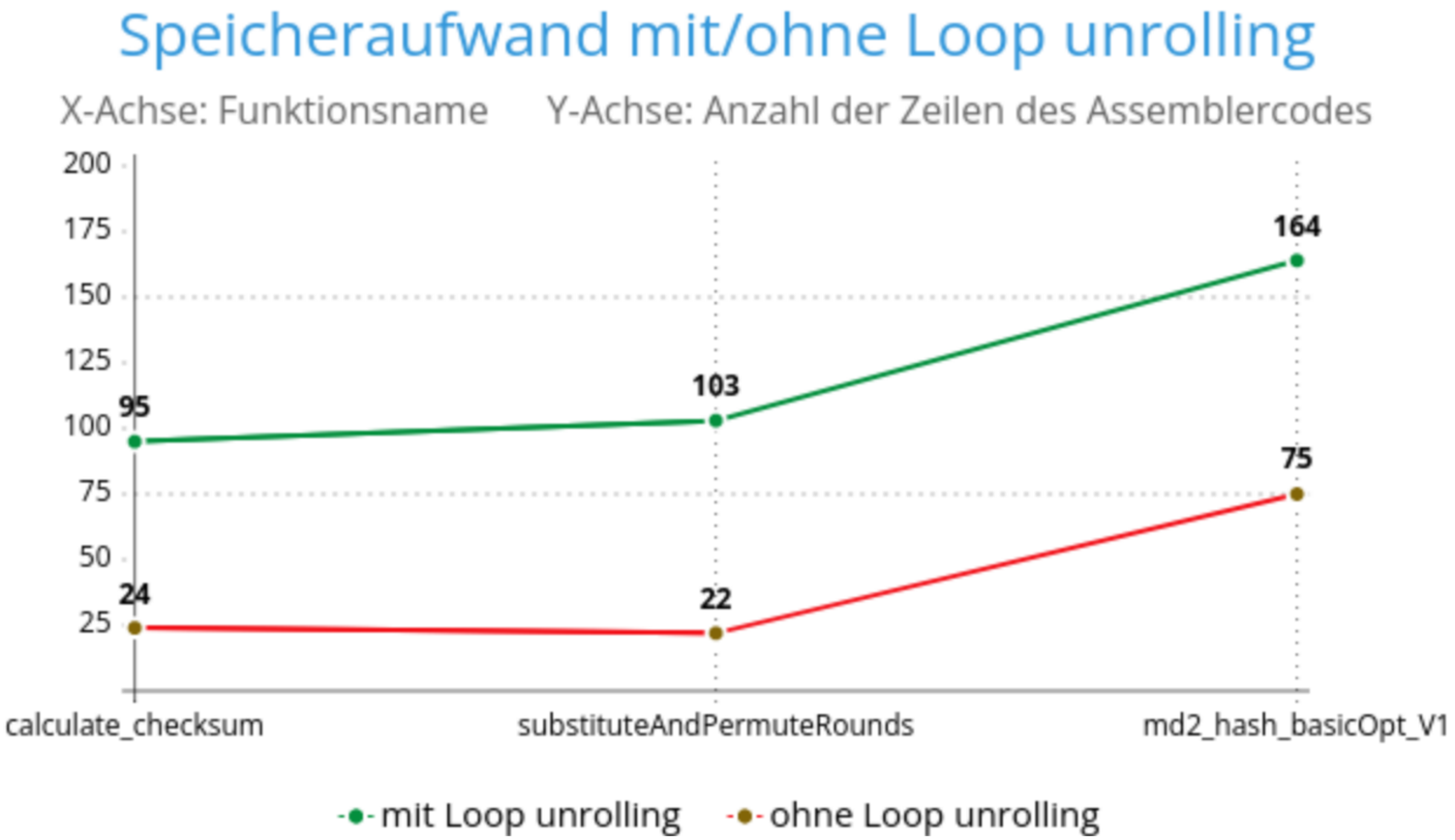
Mittels Loop-Unrolling kann die Laufzeit reduziert werden

Performanzanalyse - Programmgröße



Intrinsics können die Anzahl der Anweisungen reduzieren

Performanzanalyse - Programmgröße



Durch Loop-Unrolling wird die Anzahl der Anweisungen erheblich erhöht

Zusammenfassung und Ausblick

- MD2 hat die Grundlage für viele moderne Ansätze an Hashing gesetzt
- Der Algorithmus ist aufgrund der 8-bit Zielarchitektur kaum optimierbar
- Obwohl von der Verwendung von MD2 inzwischen abgeraten wird, ist das Verständnis der Funktion für die Entwicklung fortschrittlicherer Methoden essentiell
- SHA-3 und SHA-512 könnten für ein zukünftiges Projekt näher betrachtet werden
- Der Einfluss von Quantencomputern auf Hashing ist ein laufendes Forschungsprojekt und wird spannend mitzuverfolgen