

## Cyclic Redundancy Check (CRC)

Die zyklische Redundanzprüfung (engl. *cyclic redundancy check*, daher meist CRC) ist ein Verfahren (bzw. eine bestimmte Klasse von Verfahren) aus der Informationstechnik zur Bestimmung eines Prüfwerts für Daten (z. B. Datenübertragung in Rechnernetzen oder eine Datei), um Fehler bei der Übertragung oder Duplizierung von Daten erkennen zu können.

Vor Beginn der Übertragung bzw. Kopie eines Blocks der Daten wird ein CRC-Wert berechnet. Nach Abschluss der Transaktion wird der CRC-Wert erneut berechnet. Anschließend werden diese beiden Prüfwerte verglichen. CRC ist so ausgelegt, dass Fehler bei der Übertragung der Daten, wie sie beispielsweise durch Rauschen auf der Leitung verursacht werden könnten, fast immer entdeckt werden. Zum Beispiel werden bei den meisten Festplatten Übertragungen und Schreib-/Leseoperationen mit CRC-Verfahren geprüft.

CRC-Werte können jedoch nicht die Integrität der Daten bestätigen. Das heißt, es ist verhältnismäßig leicht, durch beabsichtigte Modifikation einen Datenstrom zu erzeugen, der den gleichen CRC-Wert wie eine gegebene Nachricht hat. Wenn eine solche Sicherheit gefordert ist, müssen kryptografische Hash-Funktionen wie z. B. MD5 zum Einsatz kommen.

Die CRC-Prüfsumme (=Cyclic Redundancy Code) beruht auf Polynomdivision. Die Folge der zu übertragenden Bits wird als Polynom mit den Koeffizienten 0 und 1 betrachtet. Bei  $k$  Bits hat man also  $k$  Terme, von  $x^{(k-1)}$  bis  $x^0$ .

Beispiel:

$$110001 \quad \Rightarrow \quad x^5 + x^4 + x^0$$

Für die Berechnung einer CRC-Prüfsumme müssen Sender und Empfänger ein Generator-Polynom definieren. Dieses Generatorpolynom  $G(x)$  hat  $d$  Bits bzw. einen Grad von  $r$  ( $r=d-1$ ). An den zu übertragenden Frame  $M(x)$  mit  $m$  Bits werden nun zunächst  $r$  Bits am Low-Order-Ende angehängen. Der erweiterte Frame hat jetzt also  $m+r$  Bits, entsprechend dem Polynom  $x^r * M(x)$ .

Die Bitfolge der Daten mit den angehängten 0-Bits wird durch dieses vorher festgelegtes Generatorpolynom  $G(x)$  bzw. das CRC-Polynom mit modulo-2-Subtraktion dividiert, wobei ein Rest bleibt. Dieser Rest ist die CRC-Prüfsumme und hat zwangsläufig  $r$  Stellen. Bei der Übertragung des Datenblocks hängt man die CRC-Prüfsumme an den originalen Frame  $M(x)$  an und überträgt ihn mit. Die übertragene Bitfolge wird als  $T(x)$  bezeichnet.

Die Idee der CRC-Prüfsumme ist dabei, einen gegebenen Rahmen von  $m$  Datenbits durch  $r$  Bits so zu ergänzen, dass das Polynom aus Datenbits und Prüfsumme durch das Generatorpolynom teilbar ist.

Dies soll hier an einem Beispiel verdeutlicht werden. Dabei wird von folgenden Bedingungen:

$$\begin{aligned} \text{Datenpaket } M(x) &= 1101011011 \\ \text{Generatorpolynom } G(x) &= x^4 + x + 1 = 10011 \end{aligned}$$

Als erster Schritt wird  $M(x)$  um vier 0-Bits ergänzt. Die Anzahl ergibt sich aus dem Grad  $r$  des Generatorpolynoms. Wir erhalten:

$$11010110110000$$

Diese neue Bitfolge wird anschließend durch  $G(x)$  dividiert:

$$\begin{array}{r} 11010110110000 / 10011 = 1100001010 \\ \underline{10011} \\ 10011 \\ \underline{10011} \\ 000010110 \\ \quad \underline{10011} \\ \quad 010100 \\ \quad \quad \underline{10011} \\ \quad \quad 01110 \\ \quad \quad \quad \underline{00000} \\ \quad \quad \quad 1110 = \text{Rest} \end{array}$$

Der Rest der Division bildet die Prüfsumme. Aus dem Datenpaket  $M(x)$  und der angehängten Prüfsumme ergibt sich die zu übertragende Bitfolge  $T(x)$ :

$$T(x) = 11010110111110$$

Um zu verifizieren, dass die Daten keinen Fehler beinhalten, wird der empfangene Datenblock mit angehängter CRC-Prüfsumme erneut durch das CRC-Polynom geteilt und der Rest ermittelt. Wenn kein Rest bleibt, ist entweder kein Fehler aufgetreten oder es ist ein Fehler aufgetreten, der in Polynomdarstellung das CRC-Polynom als Faktor hat. Der Empfänger kann also überprüfen, ob der Frame korrekt übertragen wurde, in dem er  $T(x)$  durch  $G(x)$  dividiert, das Ergebnis muss 0 sein.

Wenn ein Fehler bei der Übertragung auftritt, kommt beim Empfänger nicht  $T(x)$ , sondern  $T(x) + E(x)$  an. Die zu  $E(x)$  gehörende Bitfolge hat an jeder Bitposition, die bei der zu übertragenden Bitfolge invertiert bzw. verfälscht wurde, eine 1. Wenn der Empfänger die um den CRC-Wert erweiterte Bitfolge erhält, berechnet er  $(T(x) + E(x)) / G(x)$ . Da  $T(x) / G(x) = 0$  (per Definition von  $T(x)$ ), ist das Ergebnis  $E(x) / G(x)$ .

Ist das CRC-Polynom gut gewählt, können mit dem oben beschriebenen Verfahren alle Einbitfehler, jede ungerade Anzahl von verfälschten Bits, sowie alle Burst-Fehler der Länge  $< r$  erkannt werden, wobei  $r$  der Grad des CRC-Polynoms ist. Zusätzlich werden alle Fehler (also auch unabhängige Vierbit-, Sechsbite-, Achtbitfehler, etc.) erkannt, deren Polynomdarstellung einen kleineren Grad als das CRC-Polynom hat.

Die Gesamtwahrscheinlichkeit, dass ein gestörter Frame durchkommt, ist  $0,5^r$ . Dies gilt unter der Voraussetzung, dass alle Bitmuster gleichmäßig verteilt sind.

Als Generatorpolynome werden folgende Standard-Polynome verwendet:

- CRC-12      -  $x^{12} + x^{11} + x^3 + x^2 + x + 1$
- CRC-16      -  $x^{16} + x^{15} + x^2 + 1$
- CRC-CCITT   -  $x^{16} + x^{12} + x^5 + 1$

Das CRC-Verfahren lässt sich sowohl in einfachen Hardware-Bausteinen als auch in Software realisieren. Verwendet wird ein Schieberegister mit  $n$  Bits und ein Bit-Datenstrom (String) beliebiger Länge. Durch Verwendung einer Tabelle, die etwa bei einer CRC-8 für jedes der 256 möglichen Bytes die zugehörige CRC-Prüfsumme enthält, lässt sich obiger Algorithmus auf das achtfache beschleunigen. Durch die Operationen Linksschieben und Exklusiv-Oder wird das Berechnen der CRC-Prüfsumme häufig durch Logikschaltungen gelöst. Die CRC-Prüfsumme eines Datenstroms kann bitweise berechnet und vom Sender an die Daten angehängt werden.