Nachweis der Güte von Kombinationen des CRC

Tina Mattes[1], Frank Schiller[1], Jörg Pfahler[2], Annemarie Mörwald[3], Thomas Honold[4]

[1] Institut für Informationstechnik im Maschinenwesen
Technische Universität München
Boltzmannstr. 15
D-85748 Garching
{mattes; schiller}@itm.tum.de

[2] Lehrstuhl für Automatisierungstechnik und Prozessinformatik
Ruhr-Universität Bochum
Universitätsstr. 150
D-44780 Bochum
pfahler@atp.rub.de

[3] sd&m AG, software design & management
Carl-Wery-Str. 42
D-81739 München
annemarie.moerwald@sdm.de

[4] Institute of Information and Communication Engineering
Zhejiang University
Zheda Road
310027 Hangzhou, P.R. China
honold@zju.edu.cn

Abstract: Der Cyclic Redundancy Check (CRC) ist ein weit verbreitetes Codierungsverfahren zur Erkennung von Fehlern bei Datenübertragungen. Da der CRC selbst sehr effizient ist, d.h. er garantiert eine geringe Wahrscheinlichkeit unerkennbarer Fehler, und einfach zu implementieren ist, werden in diesem Beitrag drei Kombinationen des CRC vorgestellt, die den Grad der Fehlererkennung mit geringem Zusatzaufwand verbessern können. Zudem wird der Nachweis der Güte einer vierten Kombination vorgestellt, die durch die typische schichtenorientierte Kommunikation gegeben ist. Diese Kombination konnte bis jetzt noch nicht explizit zur Berechnung der Restfehlerwahrscheinlichkeit und damit in den Sicherheitsnachweis einbezogen werden, da eine korrekte Nachweismöglichkeit fehlte. Daher mussten worst case Annahmen getroffen werden, die letztendlich zu unnötigem Aufwand führten.

1 Einleitung

Die Übertragung digitaler Daten (z.B. von einer Steuerungslogik zu Aktoren) ist eine wichtige Funktion automatisierungstechnischer Anlagen. Da man nicht davon ausgehen kann, dass Daten stets korrekt übertragen werden, ist es vor allem in sicherheitskritischen Anwendungen von großer Bedeutung, Übertragungsfehler zu erkennen, damit der Übergang der Anlage in einen sicheren Zustand veranlasst werden kann. Ein Verfahren zur Fehlererkennung ist der Cyclic Redundancy Check (CRC), der wegen seiner leichten Implementierbarkeit und guten Fehleraufdeckung weit verbreitet ist. Die Erkennung erfolgt anhand einer Prüfsumme, die mittels einer Polynomdivision gewonnen und den Nettodaten (den ursprünglichen Daten) hinzugefügt wird. Die genaue Funktion wird in Abschnitt 2 erläutert. Da der CRC selbst sehr effizient ist (d.h. er garantiert gute Fehleraufdeckung mit relativ wenigen Prüfbits), ist es nahe liegend, Kombinationen des CRC zu untersuchen mit dem Ziel, die Fehleraufdeckung mit möglichst wenig Zusatzaufwand zu verbessern. In Abschnitt 3.1 werden drei Kombinationen sowie die Ermittlung ihrer Fehleraufdeckung vorgestellt und verglichen.

Im Kapitel 3.2 wird eine weitere Kombination vorgestellt. Diese Kombination ist durch eine schichtenorientierte Kommunikation gegeben, in der jede Schicht ihren eigenen CRC durchführt und den Nettodaten zusätzlich zu den Prüfdaten weitere Zusatzdaten hinzufügt (vgl. [ISO96]). Dieser äußere CRC der überlagerten Schicht(en) wurde bis jetzt noch nicht in den Sicherheitsnachweis (s. [IEC05]) einbezogen, da keine Berechnungsmethoden für die Restfehlerwahrscheinlichkeit zur Verfügung standen. Somit wurde eventuell zusätzlicher Aufwand betrieben, um eine durch Gesetze vorgegebene Sicherheit zu erreichen, die aber schon gegeben war, nur noch nicht nachgewiesen werden konnte. Mit der vorgestellten Berechnungsmethode kann sich dieser zusätzliche Aufwand erübrigen.

2 Grundlagen des CRC

In diesem Abschnitt werden die Funktionsweise des CRC vorgestellt, unerkennbare Fehler charakterisiert sowie ein Kriterium zur Bestimmung der Güte des CRC eingeführt.

2.1 Funktionsweise des CRC

Damit im Empfänger Verfälschungen erkannt werden können, werden die zu versendenden Daten (Nettodaten, ND) zunächst im Sender aufbereitet. Dazu wird aus ND eine Prüfsumme, die Frame Check Sequence (FCS), mittels einer Polynom-Modulo-Division mit einem gewählten Polynom, dem so genannten Generatorpolynom $g(x)$, gemäß Gleichung (1) berechnet:

$$(nd(x) \cdot x^r) \bmod g(x) = fcs(x) \tag{1}$$

Dabei bezeichnen $nd(x)$ und $fcs(x)$ die als binäre Polynome interpretierten Bitmuster von ND bzw. FCS, und r bezeichnet den Grad von $g(x)$, der mit der Anzahl der Bits der FCS übereinstimmt. Die FCS wird mit den Nettodaten als Telegramm T = [ND, FCS] an den Sender verschickt. Der Empfänger überprüft mit Gleichung (2) die Korrektheit des empfangenen Telegramms T':

$$t'(x) \bmod g(x) = 0 ? \tag{2}$$

Ist die Gleichheit (2) erfüllt, geht man davon aus, dass T korrekt übertragen wurde (d.h. T=T'). Im Falle der Ungleichheit ist auf jeden Fall ein Übertragungsfehler aufgetreten, denn für $t(x)$ gilt: $t(x)$ mod $g(x)$ = 0, da $t(x)$ mod g(x) = ($nd(x)$*x^r+$fcs(x)$) mod $g(x)$ = ($nd(x)$*x^r) mod $g(x)$ + $fcs(x)$ mod $g(x)$ = $fcs(x)$ mod $g(x)$ + $fcs(x)$ mod $g(x)$ = 0.[1]

Beispiel: Das Generatorpolynom sei $g(x) = x^3 + x + 1$ und ND = 1110011. Die Nettodaten korrespondieren zum binären Polynom $nd(x) = 1*x^6+1*x^5+1*x^4+0*x^3+0*x^2+1*x^1+1*x^0 = x^6+x^5+x^4+x+1$. Da r = grad ($g(x)$) = 3, berechnet sich das zur FCS korrespondierende Polynom nach (1) durch $fcs(x) = ((x^6+x^5+x^4+x+1)* x^3) \bmod (x^3+x+1) = x$. Die an die Nettodaten anzufügende FCS ist somit FCS = [010], und das Telegramm ergibt sich demnach zu T = [1110011010]. Erhält der Empfänger nun T' = [1110011001] (die letzten beiden Bits wurden verfälscht), so ist $t'(x)$ mod $g(x) = x+1$. Die Gleichheit (2) gilt nicht, der Fehler wurde also erkannt.

Die Berechnung der FCS im Sender und der Test im Receiver können auch mittels einer Matrix-Vektor-Multiplikation modelliert werden. Sei nd ein Vektor, dessen Koeffizienten die m Bits von ND darstellen, (d.h. nd hat die Form $(d_{m-1}\ d_{m-2}\ \ldots\ d_0)$), t ein Vektor, der aus den Bits von T besteht (d.h. $t = (d_{m-1}\ d_{m-2}\ \ldots\ d_0\ c_{r-1}\ \ldots\ c_0)$, dabei bezeichnen $c_{r-1}\ \ldots\ c_0$ die Bits der FCS), I_m die Einheitsmatrix der Dimension $m \times m$, dann kann t mittels einer vom Generatorpolynom $g(x)$ abhängigen $m \times r$-Matrix A berechnet werden durch:

$$t = nd \cdot (I_m | A) \tag{3}$$

Der Test im Empfänger sieht mit einer Matrix-Vektor-Multiplikation aus wie folgt:

$$(A^T | I_r) \cdot t' = 0 ? \tag{4}$$

Wenn die Gleichheit (2) nicht gilt, ist wiederum ein erkennbarer Fehler aufgetreten. Diese Art der Berechnung sowie das Aufstellen der Matrix A sind ausführlich beschrieben in [Ma04], [Pf06], [PW96]. Die Matrix-Vektor-Multiplikation wird genutzt, um in den Abschnitten 3 und 4 die Berechnung der Güte der Kombinationen des CRC-Verfahrens herzuleiten.

[1] Im Raum der binären Polynome und binären Wörter entspricht ‚+' der Exklusiv-Oder-Operation.

2.2 Unerkennbare Fehler und Restfehlerwahrscheinlichkeit

Mit dem CRC-Verfahren kann nicht jeder Übertragungsfehler erkannt werden. Wird z.B. im Beispiel aus Abschnitt 2.1 das Telegramm T' = [1100101010] empfangen, so ist die Gleichheit (2) erfüllt, und T' wird somit als korrekt übertragen angesehen, obwohl drei Bits verfälscht wurden. Die Verfälschung von Bits kann durch überlagerte Fehlermuster F modelliert werden, die dieselbe Länge wie das Telegramm haben. Ein Bit von F ist 0, wenn das entsprechende Bit im Telegramm korrekt übertragen wurde, und 1, wenn das entsprechende Bit verfälscht wurde. Somit gilt: T' = T+F. Ein Übertragungsfehler ist unerkennbar, wenn das zu F korrespondierende Polynom $f(x)$ durch das Generatorpolynom teilbar ist, denn es gilt: $t'(x) \bmod g(x) = (t(x)+f(x)) \bmod g(x) = t(x) \bmod g(x) + f(x) \bmod g(x) = f(x) \bmod g(x)$, und somit ist $t'(x) \bmod g(x) = 0$ genau dann, wenn $f(x) \bmod g(x) = 0$. Es sei noch erwähnt, dass die unerkennbaren Fehlermuster zuzüglich des Nullwortes[2] einen linearen Code bilden.

Da nicht alle Übertragungsfehler erkannt werden können, ist es wichtig, die Güte der Fehlererkennung zu messen. Die bekanntesten Gütekriterien sind die Hamming-Distanz und die Restfehlerwahrscheinlichkeit P_{re}. In diesem Beitrag wird nur P_{re} betrachtet, da sie das präzisere Kriterium ist. Die Restfehlerwahrscheinlichkeit ist definiert als die Wahrscheinlichkeit, dass ein Übertragungsfehler nicht erkannt wird. Deren Berechnung ist mitunter sehr komplex und auf verschiedenen Wegen möglich.

Bei der *direkten Codeanalyse* werden alle 2^m-1 unerkennbaren Fehlermuster generiert (wobei m wieder die Anzahl der ND-Bits bezeichnet). Die Anzahl A_i der unerkennbaren Fehlermuster, die i-Bits vom Wert 1 haben, wird ermittelt und benutzt, um letztendlich P_{re} nach (5) zu berechnen:

$$P_{re} = \sum_{i=1}^{n} A_i \cdot p^i \cdot (1-p)^{n-i} \tag{5}$$

Dabei bezeichnet $n = m+r$ die Anzahl der Bits der Telegramme und p die so genannte Bitfehlerwahrscheinlichkeit. Die Bitfehlerwahrscheinlichkeit ist die Wahrscheinlichkeit, mit der ein Bit während der Übertragung verfälscht wird.[3] Fügt man $A_0 = 1$ zu den A_i hinzu, so heißen die Zahlen $A_0, A_1, \ldots, A_n$ die Gewichtsverteilung des Codes. Da die Generierung von 2^m-1 Fehlermuster nur für kleines m in Frage kommt, scheidet die direkte Codeanalyse in den meisten Fällen aus.

---

[2] Das Nullwort ist hier ein Bitmuster der Länge $m+r$, dessen Bits alle Null sind. Es stellt in diesem Sinn keinen unerkennbaren Fehler dar, da ein Fehlermuster, dessen Bits alle Null sind, für die korrekte Übertragung steht.

[3] Wie üblich wird auch hier das Modell des Binären Symmetrischen Kanals angenommen, d.h. dass die Wahrscheinlichkeit der Verfälschung eines Bits vom Wert 0 zum Wert 1 die gleiche ist wie vom Wert 1 zum Wert 0 und dass Bits unabhängig voneinander verfälscht werden.

Bei der *dualen Codeanalyse* werden anstatt der 2^m-1 unerkennbaren Fehlermuster, dem originalen Code, die 2^r Elemente des dualen Codes generiert und dessen Gewichtsverteilung B_0, B_1, …, B_n hergeleitet. Mit der dualen Gewichtsverteilung kann P_{re} mit einer Formel berechnet werden. Diese liefert aufgrund numerischer Probleme häufig ungenaue Werte. Deshalb ist es ratsam, aus der Gewichtsverteilung des dualen Codes mit Hilfe der Mac-Williams-Identität (vgl. [MS91], [HQ95]) die des originalen Codes zu berechnen und dann mit Hilfe von (5) P_{re} zu bestimmen. Dieser Weg wurde bei der Berechnung der Restfehlerwahrscheinlichkeit der Kombinationen gewählt. Allerdings ist diese Art der Berechnung aufgrund numerischer Ungenauigkeiten auch nur bis zu einer Telgrammlänge von ca. 1000 Bits zu empfehlen (s. [Ma04]).

Eine dritte Alternative zu Berechnung von P_{re} ist die Methode mit stochastischen Automaten [SM06a], [SM06b]. Sie ist auch zur Berechnung der Güte der Kombinationen anwendbar, soll aber hier nicht näher vertieft werden.

3. Kombinationen des CRC

Im Folgenden werden die Kombinationen aus Abbildung 1 samt der Herleitung der Berechnung ihrer Restfehlerwahrscheinlichkeit vorgestellt. Abschnitt 3.1 beinhaltet die Kombinationen a)-c) sowie deren Vergleich. Diese Kombinationen werden hier kurz diskutiert, da sie dem Verständnis von Kombination d) dienen. Sie sind ausführlicher in [MP07] beschrieben. Der Abschnitt 3.2. beschreibt die Kombination d).

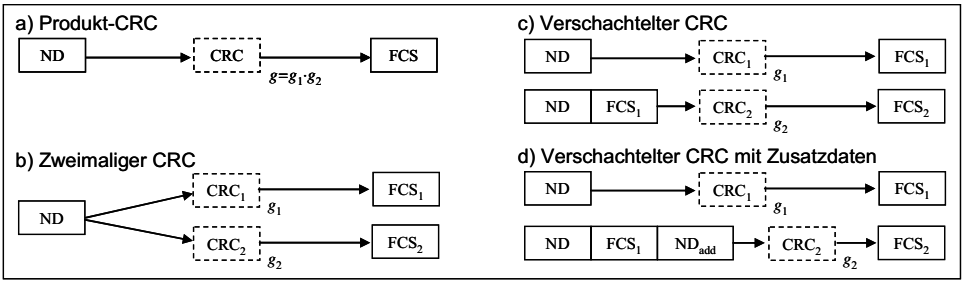

Abbildung 1: Überblick über die Kombinationen

3.1 Produkt-CRC, zweimaliger CRC und verschachtelter CRC

Zunächst werden nur die Kombinationen a)-c) vorgestellt und verglichen, da diese die gleiche Telegrammlänge haben. Die Berechnung der Restfehlerwahrscheinlichkeit erfolgt über die duale Codeanalyse, deswegen werden zudem die Berechnungen der unerkennbaren Fehlermuster des dualen Codes jeder Kombination erläutert. Aus dieser wird die duale Gewichtsverteilung abgeleitet, die in die originale umgewandelt wird, woraus schließlich mit Formel (5) die Restfehlerwahrscheinlichkeit berechnet wird. Ziel des Vergleichs ist herauszufinden, welche der drei Kombinationen die geringste Restfehlerwahrscheinlichkeit garantiert.

Produkt-CRC

Bei dieser Kombination ist das Generatorpolynom $g(x)$ ein Produkt aus zwei Polynomen $g_1(x)$, $g_2(x)$ mit Grad r_1 bzw. r_2 (s. Abbildung 1a)). Die FCS besteht somit aus $r = r_1 + r_2$ Bits und wird berechnet durch:

$$(nd(x) \cdot x^{r_1+r_2}) \bmod g(x) = fcs(x)$$

Das Telegramm setzt sich wiederum aus ND und FCS zusammen, und im Empfänger wird geprüft, ob Gleichung (2) erfüllt ist, um über die Gültigkeit des Telegramms zu entscheiden. Da das Produkt zweier Polynome wiederum ein Polynom ist, ist diese Kombination keine echte Kombination, sondern der herkömmliche CRC. Die Berechnung der Restfehlerwahrscheinlichkeit erfordert somit keine zusätzlichen Methoden. Der Produkt-CRC wurde zu Vergleichszwecken für die folgenden zwei Kombinationen in die Untersuchungen miteinbezogen.

Zweimaliger CRC

Beim zweimaligen CRC werden zwei Modulo-Berechnungen durchgeführt (vgl. Abbildung 1b)). Zum einen wird aus den Nettodaten mit Generatorpolynom $g_1(x)$ eine erste FCS, FCS_1, der Länge r_1 berechnet, und zum anderen wird aus ND in einem zweiten CRC mit Generatorpolynom $g_2(x)$ eine FCS_2 berechnet:

$$(nd(x) \cdot x^{r_1}) \bmod g_1(x) = fcs_1(x)$$
$$(nd(x) \cdot x^{r_2}) \bmod g_2(x) = fcs_2(x)$$

Das Telegramm hat die Form T = [ND, FCS_1, FCS_2]. Der Empfänger prüft in zwei Tests:

$$(nd'(x) \cdot x^{r_1} + fcs_1'(x)) \bmod g_1(x) = 0?$$
$$(nd'(x) \cdot x^{r_2} + fcs_2'(x)) \bmod g_2(x) = 0?$$

Nur wenn beide Tests den Wert Null liefern, wird das Telegramm als korrekt übertragen betrachtet. Die Matrix zur Generierung der Telegramme nach Formel (3) ist gegeben durch: $(I_m \mid A_1 \mid A_2)$, wobei A_1 die charakteristische Matrix für Polynom $g_1(x)$ ist und A_2 die für $g_2(x)$. Alle unerkennbaren Fehlervektoren f lassen sich durch Formel (6) berechnen, wobei $k = (k_{r-1}\ k_{r-2}\ \ldots\ k_0)$ alle möglichen Vektoren des $\{0;1\}^r$ mit $r = r_1 + r_2$ durchläuft:

$$f = k \cdot \left(\begin{array}{c|c} A_1^T & \\ A_2^T & I_r \end{array} \right) \tag{6}$$

Diese und die folgende Kombination wurden untersucht, um zu sehen, ob und wieweit die Restfehlerwahrscheinlichkeit durch einen zweiten CRC verringert werden kann.

Verschachtelter CRC

Bei dieser Kombination wird zunächst wie beim zweimaligen CRC aus ND in einem ersten CRC mit $g_1(x)$ die FCS_1 der Länge r_1 berechnet. Im Unterschied zum zweimaligen CRC erfolgt die Berechnung der zweiten FCS, FCS_2 der Länge r_2, mit Generatorpolynom $g_2(x)$ nicht nur aus den Nettodaten, sondern aus ND und FCS_1 (vgl. Abbildung 1c)):

$$(nd(x) \cdot x^{r_1}) \bmod g_1(x) = fcs_1(x)$$
$$((nd(x) \cdot x^{r_1} + fcs_1(x)) \cdot x^{r_2}) \bmod g_2(x) = fcs_2(x)$$

Das Telegramm besteht wiederum aus ND, FCS_1 und FCS_2. Der Empfänger prüft, ob:

$$((nd'(x) \cdot x^{r_1}) + fcs_1'(x)) \bmod g_1(x) = 0?$$
$$(((nd'(x) \cdot x^{r_1}) + fcs_1'(x)) \cdot x^{r_2} + fcs_2'(x)) \bmod g_2(x) = 0?$$

Auch hier wird das Telegramm wieder nur dann als korrekt angesehen, wenn beide Tests erfolgreich verlaufen. Die Matrix zur Erzeugung der Telegramme nach Formel (3) kann in folgender Weise hergeleitet werden. Sei $(I_m \mid A_1)$ die Matrix zur Erzeugung des temporären Telegramms $T_t = [ND, FCS_1]$, das für die Berechnung von FCS_2 dient, (d.h. $t_t = nd \cdot (I_m \mid A_1)$). Sei weiter $(I_{m+rl} \mid A_2)$ die Matrix zur Berechnung des Telegramms $T = [ND, FCS_1, FCS_2]$ aus T_t (d.h. $t = t_t \cdot (I_{m+rl} \mid A_2)$) im zweiten CRC, dann gilt:

$$t = t_t \cdot (I_{m+r_1} \mid A_2) = \left(nd \cdot (I_m \mid A_1)\right) \cdot (I_{m+r_1} \mid A_2) = nd \cdot (I_m \mid A_1 \mid B)$$

wobei $B = (I_m \mid A_1) \cdot A_2$. Also lassen sich alle unerkennbaren Fehlervektoren des verschachtelten CRC durch Formel (7) berechnen, wobei $k = (k_{r-1}\ k_{r-2}\ \ldots\ k_0)$ wiederum alle möglichen Vektoren des $\{0;1\}^r$ mit $r = r_1 + r_2$ durchläuft:

$$f = k \cdot \left(\begin{array}{c|c} A_1^T & \\ B^T & I_r \end{array} \right) \tag{7}$$

Der Vergleich der drei Kombinationen zeigt, dass keine der drei Kombinationen generell als die Beste bezeichnet werden kann. Die Güte der Fehleraufdeckung hängt von den gewählten Polynomen $g_1(x)$, $g_2(x)$, der Länge der Nettodaten m und der Bitfehlerwahrscheinlichkeit p ab.

In Abbildung 2 ist der Verlauf der Restfehlerwahrscheinlichkeiten für die drei Kombinationen für die Polynome 89h[4] und 185h für 64 ND-Bits über variable Bitfehlerwahrscheinlichkeit gegeben. Die gepunktete Linie dient zur Orientierung und gibt die Restfehlerwahrscheinlichkeit für $p = 0{,}5$ an. Die beste Alternative in diesem Beispiel ist der Produkt-CRC. Zweimaliger und verschachtelter CRC sind etwa gleich gut. Es sei erwähnt, dass das Produkt dieser Polynome das CAN-Bus-Polynom und 64 Bits die maximale Länge einer CAN-Bus Nachricht ist, sodass durch den Produkt-CRC die Restfehlerwahrscheinlichkeit des CAN-Busses gegeben ist.

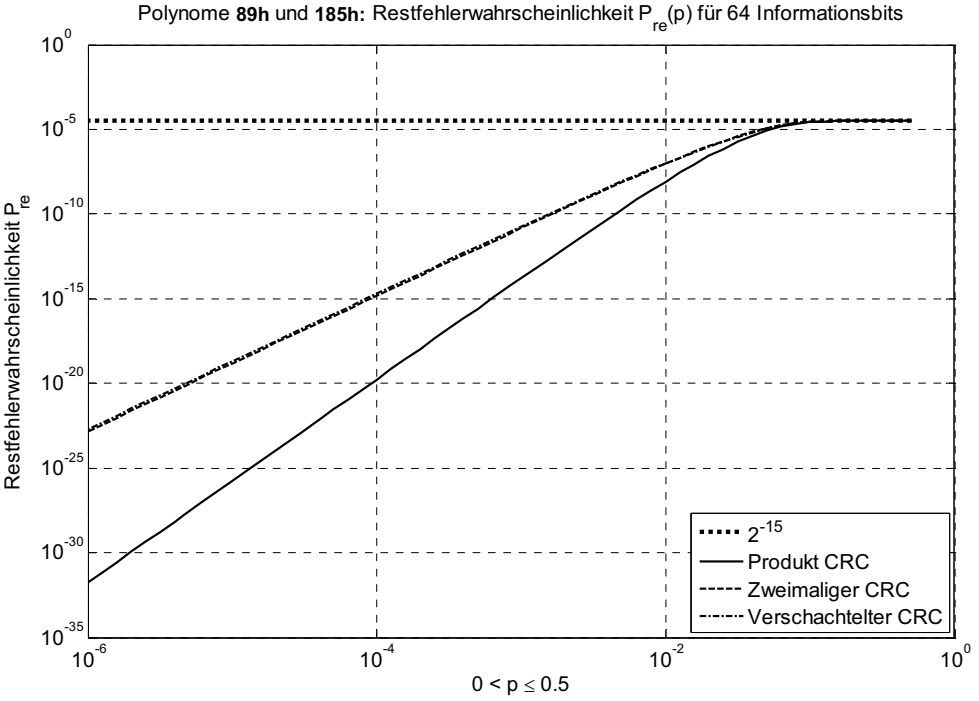

Abbildung 2: Restfehlerwahrscheinlichkeit der Kombinationen für Polynome 89h und 185h für 64 ND-Bits

In Abbildung 3 ist der Verlauf der Restfehlerwahrscheinlichkeit für die Polynome 89h und 185h für 128 ND-Bits dargestellt. In diesem Fall ist der Produkt-CRC die schlechteste Alternative und der zweimalige CRC die beste. Da die Polynome die gleichen sind wie in Abbildung 2, zeigt dieses Beispiel den Einfluss der Nettodatenlänge auf die Wahl der geeigneten Kombination. Erwähnenswert ist in Abbildung 3 auch die Überschreitung der gepunkteten Linie, was generell beim CRC auf eine schlechte Polynomwahl schließen lässt.

[4] Polynome werden in hexadezimaler Schreibweise angegeben. 89h ist als Dualzahl $1000\ 1001_2$ und entspricht somit dem binären Polynom x^7+x^3+1

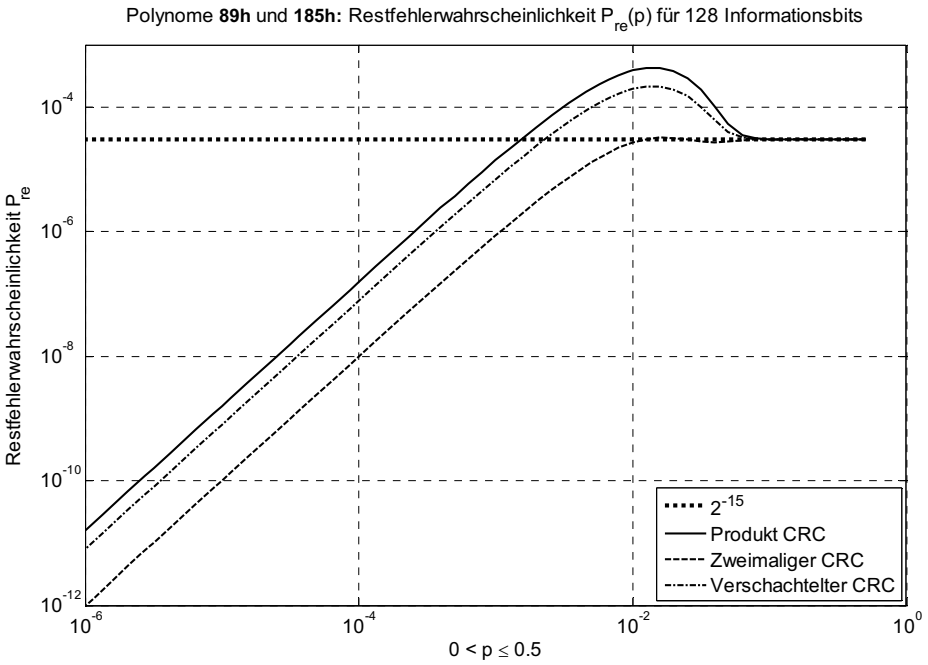

Abbildung 3: Restfehlerwahrscheinlichkeit der Kombinationen für Polynome 89h und 185h für 128 ND-Bits

3.2 Verschachtelter CRC mit Zusatzdaten

Bei dieser Kombination wird zunächst wie im verschachtelten CRC die FCS_1 aus den Nettodaten mit Generatorpolynom $g_1(x)$ berechnet. Die Berechnung der FCS_2 erfolgt im Unterschied zum verschachtelten CRC nicht nur aus ND und FCS_1, sondern auch aus zusätzlichen Nettodatenbits ND_{add} (vgl. Abbildung 1d)). Die Berechnungen werden mathematisch folgendermaßen notiert, wobei m_{add} die Anzahl der Zusatzdatenbits ist:

$$(nd(x) \cdot x^{r_1}) \bmod g_1(x) = fcs_1(x)$$

$$(((nd(x) \cdot x^{r_1} + fcs_1(x)) \cdot x^{m_{add}} + nd_{add}(x)) \cdot x^{r_2}) \bmod g_2(x) = fcs_2(x)$$

Das Telegramm hat folgende Form: T = [ND, FCS_1, ND_{add}, FCS_2]. Der Empfänger prüft, ob:

$$(nd'(x) \cdot x^{r_1} + fcs_1'(x)) \bmod g_1(x) = 0 ?$$

$$(((nd'(x) \cdot x^{r_1} + fcs_1'(x)) \cdot x^{m_{add}} + nd_{add}'(x)) \cdot x^{r_2} + fcs_2'(x)) \bmod g_2(x) = 0 ?$$

Diese Kombination ist üblicherweise bei einer schichtenorientierte Kommunikation z.B. in Feldbussen (vgl. [SB05]) gegeben. Der äußere CRC ist der CRC der unterlagerten Schichten. Er wurde bislang noch nicht in die Berechung der Restfehlerwahrscheinlichkeit miteinbezogen. Die Berechnung der Telegramme dieser Kombination in Anlehnung an Formel (3) ist gegeben durch:

$$t = (nd\ nd_{add}) \cdot \left(\begin{array}{c|c|c|c} I_m & A_1 & 0 & C \\ 0 & 0 & I_{m_{add}} & A_2 \end{array} \right)$$

Dabei bezeichne 0 Nullmatrizen passender Dimensionen. Die Matrix C ist wie folgt aufgebaut: Zeile i der Matrix C enthält die Koeffizienten des Polynoms, das aus der Modulo-Division des zur i-ten Zeile der Matrix $(I_m|A_1|0)$ korrespondieren Polynoms multipliziert mit x^{r_2} durch $g_2(x)$ resultiert. Die unerkennbaren Fehlermuster des dualen Codes sind somit berechenbar durch:

$$f = (k^1\ k^2) \cdot \left(\begin{array}{c|c|c|c} A_1^T & I_{r_1} & 0 & 0 \\ C^T & 0 & A_2^T & I_{r_2} \end{array} \right),$$

wobei k^1 alle möglichen Vektoren des $\{0;1\}^{r_1}$ annimmt und k^2 alle des $\{0;1\}^{r_2}$. Zur Berechnung der Gewichtsverteilung werden nur die Fehlervektoren betrachtet, die in den ersten m Koeffizienten mindestens einen Koeffizienten mit dem Wert 1 haben (d.h. nur die unerkennbaren Verfälschungen der ursprünglichen Nettodaten sind von Interesse).

Das wichtigste Ergebnis der Untersuchung dieser Kombination ist der Algorithmus zur Berechnung der Restfehlerwahrscheinlichkeit, die durch den zweiten CRC gegeben ist. Diese Restfehlerwahrscheinlichkeit der Verschachtelung kann somit in den geforderten Sicherheitsnachweis einbezogen werden. Wie in Abschnitt 2.2 erwähnt, steigt die Rechenzeit des Algorithmus exponentiell zum Grad der Polynome. Für zwei Polynome vom Grad 8, benötigt ein herkömmlicher Rechner[5] nur wenige Sekunden. Die Berechnungen für ein Polynom vom Grad 32 und eines vom Grad 3 dauern ca. zehn Tage. Abbildung 4 zeigt den Verlauf der Restfehlerwahrscheinlichkeit für die Polynome 6Dh und 103h für verschiedene Datenlängen.

[5] Pentium4 HT 3,2Ghz, 1GB RAM

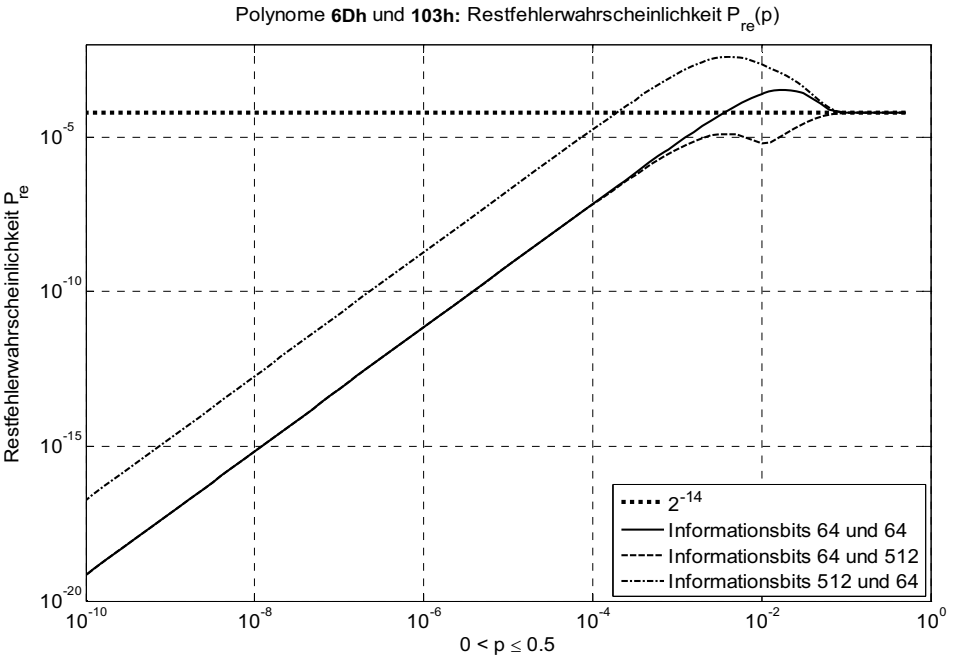

Abbildung 4: Restfehlerwahrscheinlichkeiten für den verschachtelten CRC für Polynome 6Dh und 103h für verschiedene Datenlängen

Des Weiteren wurde zu dieser Kombination eine Reihe von Untersuchungen von Datenlängen bzgl. Polynomeigenschaften gemacht. Damit ließen sich einige erste Kriterien ableiten, zu denen sich jedoch immer Ausnahmen finden lassen. Diese Kriterien unterstützen bei der Wahl von Polynomen. Tabelle 1 zeigt einige dieser Kriterien in Bezug auf die Länge der zusätzlichen Nettodatenlänge auf:

Fall	m_{add}
$g_1(x) = g_2(x)$	Relativ klein
$g_1(x)$, $g_2(x)$ primitiv oder irreduzibel, $g_1(x) \neq g_2(x)$	Relativ klein
$g_1(x)$ primitiv, $g_2(x)$ reduzibel, $g_2(x)$ nicht teilbar durch $g_1(x)$	Relativ klein
$g_1(x)$ primitiv, $g_2(x)$ reduzibel, $g_2(x)$ teilbar durch $g_1(x)$	Relativ groß
$g_1(x)$ reduzibel, $g_2(x)$ primitiv, $g_1(x)$ nicht teilbar durch $g_2(x)$	Relativ klein
$g_1(x)$ reduzibel, $g_2(x)$ primitiv, $g_1(x)$ teilbar durch $g_2(x)$	Relativ groß

Tabelle 1: Auszug bisher ermittelter Empfehlungen zur Wahl der Länge der zusätzlichen Daten m_{add} aufgrund der Eigenschaften der Generatorpolynome

4 Ausblick

Weiterführende Arbeiten haben zum Ziel, die Restfehlerwahrscheinlichkeit von Protokollen der existierenden Feldbusse für verschiedene Nettodatenlängen zu berechnen. Insbesondere die Einbettung von Telegrammen in Telegramme wird immer bedeutsamer. Durch die vorgestellten Verfahren ist es möglich, bei bekanntem Generatorpolynom des äußeren bzw. unterlagerten CRC ein passendes Generatorpolynom des inneren bzw. überlagerten CRC zu bestimmen.

Literaturverzeichnis

[HQ95] Heise, W.; Quattrocchi, P.: Informations- und Codierungstheorie. 3. Auflage, Springer Verlag, 1995.

[IEC05] International Electrotechnical Commission: Functional Safety of Electrical/Electronic/ Programmable Electronic Safety-related Systems (IEC 61508), 2005.

[ISO96] International Organization for Standardization, International Electrotechnical Commission (ISO/IEC), Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model, (ISO/IEC 7498-1), 1996.

[Ma04] Mattes, T.: Untersuchungen zur effizienten Bestimmung der Güte von Polynomen für CRC-Codes. Diplomarbeit, Universität Trier, Siemens AG Nürnberg, 2004.

[MS91] Mac Williams, F.J.; Sloane, N.J.A.: Theory of Error-Correcting Codes. North-Holland Mathematical Library, 1991.

[MP07] Mattes, T., Pfahler, J., Schiller, F., Honold, T.: Analysis of Combinations of CRC in Industrial Communication, 26th International Conference on Computer Safety, Reliability and Security, Nuremberg, Germany, in: Saglietti, F. and Oster, N. (Eds): SAFECOMP 2007, Lecture Notes in Computer Science, LNCS 4680, pp. 329-341, Springer, 2007

[Pf06] Pfahler, J.: Analyse von Kombinationen von Fehleraufdeckungsverfahren in der industriellen Kommunikation. Diplomarbeit, TU München, 2006.

[PW96] Peterson, W.; Weldon, E.J.: Error Correcting Codes. MIT Press, 1996.

[SB05] Stripf, W.; Barthel, H.: PROFIsafe - Safety Technology with PROFIBUS. In (Hrsg. Zurawski, R.): The Industrial Information Technology Handbook. CRC Press, 2005; S. 1-20

[SM06a] Schiller, F.; Mattes, T.: An efficient method to Evaluate CRC-Polynomials for Safety-Critical Industrial Communication, Journal of Applied Computer Science, Vol. 14, 1/2006, Technical University Press, Lodz, Poland, 2006; S. 57-80.

[SM06b] Schiller, F.; Mattes, T.: Analysis of CRC-Polynomials for Safety-Critical Communication by Deterministic and Stochastic Automata, 6th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS 2006, Beijing, China, 2006; S. 1003-1008.