Grundkurs Codierung

Lösungsvorschläge zu den Fragen in den Unterkapiteln „Was blieb?“
Stand 22.04.2007

Unterkapitel 4.4 Seite 261

Zu Frage 1:

Nein, damit bleibt das one time pad-Verfahren nicht perfekt. Man kann sich das klar machen, wenn man den “anderen” Extremfall nur eines Schlüssels betrachtet, mit dem ja gemäß den Bildern 4.14 und 4.15 die Verteilung der Geheimtextzeichen überhaupt nicht verändert wird.

Verwendet man nun zwei zufällig verteilte Schlüssel, so stehen hinter jedem Geheimtextzeichen zwei mögliche Klartextzeichen, nämlich die, deren Summen mit den beiden Schlüsseln immer denselben Wert ergeben. Die Verteilung der Geheimtextzeichen wird also im Mittel “Doppelsäulen” aufweisen und man muss durch Probieren nur noch ermitteln, welche Klartextzeichen mit welchen beiden Schlüsseln zum Wert dieser Doppelsäulen führen. Diese Verschlüsselung ist also noch weit davon entfernt, perfekt zu sein.

Bei drei zufällig verteilten Schlüsseln verhält es sich ähnlich, in der Verteilung der Geheimtextzeichen erhält man nun “Dreifachsäulen”, was eine weitere Abflachung des Profils bewirkt. Damit wird allerdings auch das Probieren zum Bestimmen der dann drei verschiedenen Klartextzeichen aufwändiger usw.

Mit jedem neu hinzu genommenen Schlüssel flacht die Häufigkeitsverteilung weiter ab, bewegt sich immer mehr auf einen perfekten Zustand zu, siehe Bild 4.16, und erreicht diesen im Extremfall gleich vieler verschiedener Schlüssel wie Klartextzeichen.

Zu Frage 2:

Nein, es gibt dann keine Möglichkeit.

Zu Frage 3:

Wenn ein Angreifer einige zusammenhängende Teilstücke von Geheim- und Klartext in die Hände bekommt, kann er die zugehörigen Schlüssel berechnen. Dann hat er eine Chance, mit Hilfe des in Unterkapitel 4.3, Seiten 258 ff, beschriebenen Verfahrens Länge und Rückkopplungspolynom des Schieberegisters zu ermitteln. Damit wäre ihm auch die gesamte Pseudo-Zufallsfolge bekannt.

Zu Frage 4:

Nein, maximale Längen von Schieberegisterfolgen (= Maximallängenfolgen) werden nur mit denjenigen irreduziblen Rückkopplungspolynomen erzeugt, deren zugehörige Wurzeln die maximale Ordnung aufweisen, oder, etwas genauer: Deren Wurzeln primitive Elemente sind. Ein über die Koeffizienten eines Polynoms m-ten Grades g(x) mit Koeffizienten im Z_2 rückgekoppelten binären Schieberegisters mit m Speichern gemäß Bild 4.1 in Unterkapitel 4.1, Seite 235, wirkt wie ein MOD g(x)-Dividierer.

Ein Beispiel für $g(x) = x^3 + x + 1$ ist in Bild 3.12 im Unterkapitel 3.7.2, Seite 109, dargestellt. Für die obere Tabelle in Bild 3.12 wurden die 3 Speicher zum Startzeitpunkt von links nach rechts mit 0 0 1 belegt. Am Eingang rechts steht ein Folge von Nullen. Beim Ablauf stellen sich $2^3 - 1 = 7$ verschiedene Speicherzustände ein, der achte ist die Wiederholung des ersten. Bei der Division 0010000 MOD g(x) wurde also eine Zustandsfolge maximaler Länge erzeugt – oder – als gleichwertige Beobachtung – eine Binärfolge des Ausgangsignals, die sich erst nach 7 Stellen wiederholt.

Dieses Ergebnis erhält man auch bei fortlaufender Berechnung und Linksverschiebung des

Syndrompolynoms s(x)

$s(x) = s_1(x) = 0001 \text{ MOD } g(x) \quad = 001,$

Linksverschiebung von $s_1(x)$ über $s_1(x)$ x und erneute Division ergibt nacheinander

$s_2(x) = s_1(x)\,x = 0010 \text{ MOD } g(x) \quad = 010$
$s_3(x) = s_2(x)\,x = 0100 \text{ MOD } g(x) \quad = 100$
$s_4(x) = s_3(x)\,x = 1000 \text{ MOD } g(x) \quad = 011$
$s_5(x) = s_4(x)\,x = 0110 \text{ MOD } g(x) \quad = 110$
$s_6(x) = s_5(x)\,x = 1100 \text{ MOD } g(x) \quad = 111$
$s_7(x) = s_6(x)\,x = 1000 \text{ MOD } g(x) \quad = 101$
$s_8(x) = s_8(x)\,x = 1000 \text{ MOD } g(x) \quad = 001 \quad = s_1(x)$

Mit den Elementen α des Galoiskörpers GF(2^3), welcher über die Nullstelle $\alpha^3 = \alpha + 1$ des irreduziblen Polynoms g(x) definiert wird, kann dieser Ablauf mit Hilfe von s(x), $v(x) = x^0 = 1$ und $x = \alpha$ auch folgendermaßen dargestellt werden:

$s(\alpha) = s_1(\alpha) \quad = v(\alpha) \quad = \alpha^0 \quad = 1$
$s_2(x) = s_1(x)\,x \quad = s_1(\alpha)\,\alpha \quad = \alpha$
$s_3(x) = s_2(x)\,x \quad = s_2(\alpha)\,\alpha \quad = \alpha^2$
$s_4(x) = s_3(x)\,x \quad = s_3(\alpha)\,\alpha \quad = \alpha^3$
$s_5(x) = s_4(x)\,x \quad = s_4(\alpha)\,\alpha \quad = \alpha^4$
$s_6(x) = s_5(x)\,x \quad = s_5(\alpha)\,\alpha \quad = \alpha^5$
$s_7(x) = s_6(x)\,x \quad = s_6(\alpha)\,\alpha \quad = \alpha^6$
$s_8(x) = s_7(x)\,x \quad = s_7(\alpha)\,\alpha \quad = \alpha^7 \quad = s_1(\alpha) \quad = 1.$

Hier ist direkt zu sehen, dass die Maximallängenfolgen mit der Ordnung der Wurzel des definierenden irreduziblen Polynoms g(x) zusammenhängen. Nach Unterkapitel 3.7.5 eignet sich hierzu im GF(2^m) aber nicht jedes irreduzible Polynom vom Grad m.

Zu Frage 5:

Die Frage ist also, ob und wie sich die mittlere Entropie eines nach dem one time pad-Verfahren erzeugten Geheimtextes G gegenüber der Entropie des Klartextes verändert. Wenn zur Darstellung eines Zeichens n Bits verwendet werden, dann ist die Entropie H bei Gleichverteilung $p(x_1) = p(x_2) = \ldots p(x_s) = 1/s$ aller $s=2^n$ Zeichen am größten, also

$$H = -\sum_{i=1}^{s} p(x_i)\cdot\log_2 p(x_i) \;\rightarrow\; \max_{p(x_i)} \quad \text{mit der Nebenbedingung} \quad \sum_{i=1}^{s} p(x_i) = 1, \quad \text{wenn} \quad p(x_1) = p(x_2) = \ldots = p(x_s)$$

Das one time pad-Verfahren bewirkt aber gerade eine Gleichverteilung bei der Abbildung des Klartext-Zeichensatzes in die Geheimtextzeichen. Also erhöht sich die Entropie der Geheimtextzeichen, wenn die Klartextzeichen nicht gleich verteilt sind. Sie ändert sich nicht, wenn sie gleich verteilt sind (Zusatzfrage: bringt das one time pad-Verfahren überhaupt etwas? Siehe unten *), jedoch wird sie in keinem Fall größer.

Dass sich die maximale Entropie bei Gleichverteilung ergibt, sieht man, wenn man z. B. die Wahrscheinlichkeit $p(x_s)$ gemäß der Nebenbedingung durch die Summe der übrigen Wahrscheinlichkeiten ausdrückt

$$p(x_s) = 1 - \sum_{i=1}^{s-1} p(x_i)$$

und in das Entropie-Funktional H einsetzt:

$$H=-\sum_{i=1}^{s-1} p(x_i)\cdot\log_2 p(x_i)\ -p(x_s)\log_2 p(x_s)\ =\ -\sum_{i=1}^{s-1} p(x_i)\cdot\log_2 p(x_i)\ -(1-\sum_{i=1}^{s-1} p(x_i))\cdot\log_2(1-\sum_{i=1}^{s-1} p(x_i))$$

Die Ableitungen dieses Funktionals nach den s-1 verbleibenden unabhängigen Wahrscheinlichkeiten ergeben s-1 Gleichungen, die zur Bestimmung des Extremwertes von H alle Null sein müssen:

$$\frac{\partial H}{\partial p(x_k)}=-\sum_{i=1}^{s-1} p(x_i)\cdot\log_2 p(x_i)\ -(1-\sum_{i=1}^{s-1} p(x_i))\cdot\log_2(1-\sum_{i=1}^{s-1} p(x_i))=0 \quad \text{für } k=1,\ 2,\ \ldots,\ s-1$$

$$\frac{\partial H}{\partial p(x_k)}=-\log_2 p(x_k)-\frac{-p(x_k)}{p(x_k)\cdot\ln 2}+\log_2(1-\sum_{i=1}^{s-1} p(x_i))-\frac{(1-\sum_{i=1}^{s-1} p(x_i))\cdot(-1)}{(1-\sum_{i=1}^{s-1} p(x_i))\cdot\ln 2}=0$$

nach Elimination des zweiten und vierten Terms sowie Exponenzieren der beiden verbleibenden Summanden erhält man das Gleichungssystem

$$-p(x_k)+1-\sum_{i=1}^{s-1} p(x_i)=0 \quad \text{für} \quad k=1,\ 2,\ \ldots,\ s-1 \quad \text{oder}$$

$$a_1\cdot p(x_1)+a_2\cdot p(x_2)+\ \ldots\ +a_{s-1}\cdot p(x_{s-1})=1, \quad a_i=2 \text{ für } i=k,\ \ a_i=1 \text{ für } i\neq k,\ \ k=1,\ 2,\ \ldots,\ s-1$$

Es ist nur für

$p(x_k)=\frac{1}{s}$, $i=1,\ 2,\ \ldots\ ,\ s-1$ erfüllt, wie man sich durch Einsetzen überzeugen kann. Die

Wahrscheinlichkeit $p(x_s)$ ergibt sich aus der oben genannten Nebenbedingung ebenfalls als

$$p(x_s)=\frac{1}{s}$$

Beispiel: Die Koeffizientenmatrix A und ihre Inverse A^{-1} des Gleichungssystems bei s=5:

$$A=\begin{bmatrix}2&1&1&1\\1&2&1&1\\1&1&2&1\\1&1&1&2\end{bmatrix}, \qquad A^{-1}=\begin{bmatrix}\frac{4}{5}&-\frac{1}{5}&-\frac{1}{5}&-\frac{1}{5}\\-\frac{1}{5}&\frac{4}{5}&-\frac{1}{5}&-\frac{1}{5}\\-\frac{1}{5}&-\frac{1}{5}&\frac{4}{5}&-\frac{1}{5}\\-\frac{1}{5}&-\frac{1}{5}&-\frac{1}{5}&\frac{4}{5}\end{bmatrix}$$

*) Da gleiche Geheimtextzeichen hier immer zu gleichen Klartextzeichen gehören, ist die Anzahl der “brute force”-Entschlüsselungsversuche bedeutend geringer als bei einer one time pad-Verschlüsselung, bei der jedes Geheimtextzeichen nacheinander **allen** Klartextzeichen zugeordnet werden muss. Die perfekte Verschlüsselung ist also - unabhängig von der Verteilung der Klartextzeichen - nur mit dem one tim pad-Verfahren möglich. Trotzdem eine weitere Frage: Gibt es überhaupt natürliche Alphabete mit gleich verteilten Zeichen?

Zu Frage 6:

... mit dem Verfahren der Kreuzkorrelation zwischen dem Sende- und dem Empfangssignal, siehe Unterkapitel 4.2.

Zu Frage 7:

Das zum irreduziblen Polynom $f(x) = x^m + ...+ 1$ vom Grad m reziproke Polynom $f^*(x)$ hat dieselben Nullstellen wie f(x). Die Koeffizienten erscheinen wegen

$$f^*(x) = x^m f(1/x)$$

in umgekehrter Reihenfolge bezogen auf die Potenzen von x.

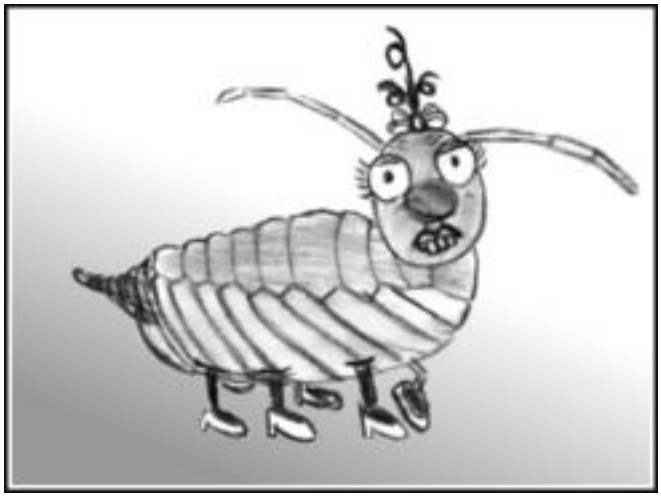
Mit $f(x) = x^m + a_{m-1} x^{m-1} + ... + 1$ ist $f^*(x) = 1 + a_{m-1} x + a_{m-2} x^2 + ... + x^m$

Beispiel: $f(x) = x^3 + x + 1 \rightarrow f^*(x) = x^3 (x^{-3} + x^{-1} + 1) = x^3 + x^2 + 1$

Anmerkung: Irreduzible Polynome haben bei x^0 immer einen von Null verschiedenen Koeffizienten (warum?).

Zu Frage 8:

Der Entdecker der Steinlaus (Gattung *Petrophaga* aus der Ordnung der Fabelnager - *Rodentia inexista*) war der Frankfurter Zoologe Dr. Bernhard Grzimek, veröffentlicht wurde diese Entdeckung allerdings erst 1976 von Loriot in einem Wissenschaftsbeitrag der ARD. Die genaue Klassifikation findet man z. B. in *Wikipedia* oder im medizinischen Wörterbuch *Pschyrembel*. Hier das Foto eines geschlechtsreifen Steinlaus-Weibchens:



Zu Frage 9:

Gemäß Unterkapitel 4.3, Seite 258 ff, kann ein Angreifer versuchen, aus zwei zusammengehörigen Stücken Klar- und Geheimtext das Rückkopplungspolynom des die Schlüsselfolge erzeugenden Schieberegisters zu ermitteln. Bei Annahme zu niedriger oder zu hoher Polynomgrade ergeben sich linear abhängige Zeilen im bestimmenden Gleichungssystem, so dass sich der passende Grad durch systematisches Probieren ermitteln lässt. (**Anmerkung:** Heute - 2007 - ist dies bei der Absicherung im WLAN mit Hilfe des WEP-

Verfahrens akut geworden. Zuletzt hat eine Forschergruppe der TU Darmstadt gezeigt, dass der 128-Bit-WEP-Schlüssel in etwa einer Minute “geknackt” werden kann. WEP ist als Sicherungsverfahren damit unbrauchbar).

Zu Frage 10:

Die Autokorrelationsfunktion des Empfangssignals enthält keine Information über die zeitliche Verschiebung (= Laufzeit) zum Sendesignal, die aber für diesen Zweck gerade bestimmt werden soll.