09.03.2021

Dr. Felix Heuer
fheuer.lba@fh-salzburg.ac.at

Kryptographie und Mediensicherheit

Sommersemester 2021

Übungszettel 2

Abgabe: 06.04.2021 vor der Vorlesung

1 Aufgabe (10 Punkte)

Betrachten Sie das folgende LFSR bestehend aus 60 Bit die wir mit $r_0, \ldots, r_{59}$ bezeichnen. In einem Takt passiert jeweils:

1. $r_{-1} = 1 \oplus r_2 \oplus r_6 \oplus r_{12} \oplus r_{17}$
2. $out = r_{59}$
3. $r_i = r_{i-1}$ für $i = 59, \ldots, 0$
4. Gib *out* aus

Die Ausgaben der ersten 512 Takte werden verworfen. Ab dem 513. Takt wird der eigentliche Strom ausgegeben.

a) Implementieren Sie das LFSR in C#. Sie finden einen Testvektor online.

b) Online finden Sie ebenfalls einen Chiffretext. Hier wurde ein deutscher Text in (8-Bit) ASCII umgewandelt und dann mit einem Strom des obigen LFSR bei Ihnen *unbekannter initialer Belegung* gexort.

Schreiben Sie ein C# Programm, das den Text entschlüsselt.

Hinweis: Ihr Programm wird einige Minuten laufen, aber sollten Sie eine Schleife mit 2^{60} Iterationen verwenden, sind Sie eventuell nicht bis zur Abgabe fertig.

2 Aufgabe (10 Punkte)

Sei Π ein symmetrisches Verschlüsselungsverfahren. Zeigen Sie, dass Π nicht CPA-sicher ist, falls Π deterministisch ist.

3 Aufgabe (10 Punkte)

Eine simple Methode, um Pseudozufallsfolgen zu erzeugen, sind *Lineare Kongruenzgeneratoren (LCG)*. Diese verwenden iterativ eine Vorschrift der Art:

$$X_{i+1} = (a \cdot X_i + c) \bmod m$$

um basierend auf einem Startwert X_0 eine Pseudozufallsfolge $(X_0, X_1, \ldots)$ zu bestimmen. Wir werden in dieser Aufgabe sehen, dass diese Generatoren schlecht sind und nicht verwendet werden sollten. Ein "truly horrible" (Donald Knuth) LCG war zum Beispiel IBM's RANDU.[1]

a) In welchem Wertebereich kann X_i maximal liegen?

b) Programmieren Sie einen LCG mit den folgenden Parametern und lassen Sie sich 1000 Werte erzeugen:

$$a = 24,\ c = 42,\ m = 529,\ X_0 = 42$$

c) Ändern Sie m auf 524. Was fällt Ihnen auf? Wie lässt sich dieses Verhalten erklären? Was schließen Sie daraus für die optimale Wahl von m?

d) Wählen Sie nun wieder die ursprünglichen Parameter und plotten Sie aufeinanderfolgende Folgeglieder paarweise als Punkte:

$$(X_0, X_1), (X_1, X_2), \ldots$$

Was fällt Ihnen auf? Angenommen Sie kennen nur die Ausgabefolge $(X_0, X_1, \ldots)$, wie können Sie daraus Parameter des LCG bestimmen?

4 Aufgabe (10 Punkte)

In der Praxis bezeichnen wir PRGs, denen ein sicheres mathematisches Design zu Grunde liegt, und deren Ausgaben hohen statistischen Anforderungen für Zufälligkeit genügen als *kryptographisch sichere PRGs*. Einen dieser kryptographisch sicheren Generatoren bietet die RNGCryptoService-Provider Klasse.

a) Erzeugen Sie für diesen sicheren Generator eine Folge von Pseudozufallszahlen.

b) Microsoft bietet einen standardisierten Testkatalog für statistische Eigenschaften von (Zufalls)folgen an.[2]

Machen Sie sich mit dem Frequency, Block und Runs Test vertraut und führen Sie diese Tests für

- die Ausgabefolge des LCGs aus Aufgabe 3c (1000 Ausgaben) und
- die Ausgabefolge des LCGs aus Aufgabe 3b (1000 Ausgaben) und
- die Ausgabefolge des LCGs aus Aufgabe 3b, aber mit 10.000 Ausgaben und
- eine Ausgabefolge (1000 Ausgaben) des PRGs der RNGCryptoServiceProvider Klasse durch.

Was beobachten Sie und warum?

[1] `https://en.wikipedia.org/wiki/RANDU`

[2] `https://docs.microsoft.com/en-us/archive/msdn-magazine/2013/government-special-issue/test-run-implementing-the-national-institute-of-standards-and-technology-tests-of-randomness-using-csharp`

5 Aufgabe (0 Punkte, aber unbezahlbar ;-))

Lernen Sie folgenden Satz auswendig:
Verwenden Sie in kryptographischen Anwendungen IMMER einen kryptographisch sicheren Zufallszahlengenerator!

Wie wir sehen werden kann auch das beste Verschlüsselungsverfahren komplett unsicher sein, wenn ein schwacher Zufallszahlengenerator verwendet wird.