Übung 3

Abgabe: 08.11.2018 **bis** 12:15 Uhr

1. One-Time-Pad Rechenaufgabe 15 Punkte

Das One-Time Pad (OTP) wurde in der Vorlesung zur Verschlüsselung des binären Alphabets ($\Sigma \in \{0, 1\}$) eingeführt. Hiermit können Daten von beliebiger Länge bitweise verschlüsselt werden.
Entschlüsseln Sie den folgenden Ciphertext, von dem Sie wissen, dass er einen ASCII-String enthält:

```
DC 48 13 3B 9C 4C 49 80 AC A7 B9 54 F2 7C 2B 9E D5 DF 0D 05 B3 1D 4E F8
```

Verwenden Sie den folgenden Schlüssel:

```
98 29 60 72 F2 38 2C F2 C2 C2 CD 1D 81 08 65 FB A0 B3 6C 6B D7 3C 6F D9
```

Entschlüsseln Sie mindestens drei Bytes exemplarisch Bit für Bit, danach können Sie die Entschlüsselung direkt auf den hexadezimalen Werten durchführen.
Geben Sie den Klartext als ASCII-Zeichen an.

2. Vollständige Schlüsselsuche beim One-Time-Pad 15 Punkte

Auf den ersten Blick scheint es möglich, das OTP durch vollständige Schlüsselsuche zu brechen, was einen Widerspruch zu der informationstheoretischen Sicherheit des OTP ist. Gegeben sei eine kurze Nachricht bestehend aus 5 ASCII-Zeichen, d. h. 40-Bit. Dieser Klartext wurde mit 40-Bit eines OTP verschlüsselt. Beschreiben Sie in max. **3 Sätzen**, warum eine vollständige Schlüsselsuche nicht zum Erfolg führt, obwohl genug Rechenleistung für die Suche zur Verfügung steht.
Bemerkung: Das Paradox muss aufgelöst werden, d. h. Antworten à la „Das OTP ist beweisbar sicher, daher funktioniert die vollständige Schlüsselsuche nicht.“ reichen nicht.

3. One-Time-Pad Angriff bei sich wiederholendem Schlüssel 35 Punkte

Sie haben bisher einige Verschlüsselungsübungen per Hand erledigt. Im Übungsordner befindet sich eine verschlüsselte PNG Datei `lage.hex`. Zudem werden Sie das Programm „`CrypTool`“ (aktuelle Release-Version: 1.4.40) benötigen, welches Sie sich kostenlos herunterladen können[1].

Zu Halloween ist eine Gruppe Sozialwissenschaftler durch die Universität gezogen und hat die erbeuteten Süssigkeiten der Professoren gestohlen. Unsere Fakultät hat auf die Diebe ein Kopfgeld von 35 Übungspunkten ausgesetzt, um der Kriminalität ein Ende zu bereiten. Ein Undercover-Agent

[1] `https://www.cryptool.org/de/cryptool1`

hat für Sie eine Nachricht abgefangen, die die Diebe auf ihrem Krypto-Drucker ausgedruckt haben. Die Nachricht enthält die Lage des Versteckes des Diebesgutes. Ihnen ist bekannt, dass der Krypto-Drucker empfangene Dateien automatisch mit einem sich wiederholendem Schlüsselstring XOR-verknüpft. Aus dem Handbuch des Druckers wissen Sie, dass dieser Schlüsselstring genau 8 Byte lang ist und sich dann wiederholt.

a) Öffnen Sie willkürlich einige PNG-Dateien mit dem CrypTool, und schauen sie sich jeweils den Anfang der Datei an. Beschreiben Sie, was alle Dateien gemeinsam haben. Für welche Art von Angriff können Sie dieses Wissen ausnutzen? (10 Pkt.)

b) Der Schlüssel zur Verschlüsselung der o. g. Datei hat eine Länge von acht Byte. Wie lautet er (hexadezimale Darstellung)? Zeigen Sie auch die Berechnung. (10 Pkt.)

c) Nutzen Sie dieses Wissen um mit Hilfe des CrypTools die Datei zu dechiffrieren (entschl..symmetrisch..klassisch..XOR).
Wo wurden die Süßigkeiten der Professoren versteckt? (15 Pkt.)

Historische Bemerkung: Es wird berichtet, dass das Ehepaar Rosenberg in den 50er Jahren eine solche Verschlüsselung verwendete, um der UDSSR Wissen über den Bau der Atombombe zukommen zu lassen. Da sie auf dem elektrischen Stuhl endeten, können wir die Mehrfachverwendung eines OTP-Schlüssels nicht uneingeschränkt empfehlen (siehe auch `https://de.wikipedia.org/wiki/Ethel_und_Julius_Rosenberg`).

4. Erweiterter linearer Kongruenzgenerator (LCG) 35 Punkte

Sie haben eine geheime Nachricht eines Kommilitonen abgefangen. Diese soll einen Link (https://) zu einem besonderen Konzentrationsmittel für Übungen und Klausurvorbereitungen enthalten. Sie wollen natürlich den Link entschlüsseln um selbst in Genuss des Mittels zu kommen.
Sie wissen das ein Erweiterter linearer Kongruenzgenerator (LCG) zu Erzeugung eines Schlüsselstroms genutzt wurde. Außerdem wissen Sie, dass der Schlüsselstrom S_i dazu verwendet wurde, den Link in ASCII-Kodierung zeichenweise zu verschlüsseln, d.h. das Chiffrat $Y_i = X_i \oplus S_i$ (XOR) wird für jedes ASCII-Zeichen X_i berechnet.
Eine erweiterte Variante des LCG berechnet jedes neue Schlüsselstromsymbol S_{i+1} aus den beiden vorherigen Elementen S_i und S_{i-1}. In diesem Fall benötigt der LCG aber auch zwei *seed* Werte S_0 und S_1 sowie insgesamt drei Schlüsselparameter A,B und C und den Modul m.

$$S_{i+1} = A \cdot S_i + B \cdot S_{i-1} + C \bmod m \ , \ \ i = 0, 1, \ldots$$

Die abgefangene Nachricht ist wie folgt (in Hex):
`4E 7E 3D 88 8E 01 0D 84 B8 7E BF 1A 25 37 FA 4D 89 87 91 FA 50 51 FC 42 7A 9A 6A E4`

a) Berechnen Sie die Parameter A, B und C. (25 Pkt.)
Hinweis: Benutzen Sie die Tabelle 1 auf der nächsten Seite um die multiplikative Inverse modulo 257 zu bestimmen. Und benutzten Sie ihr Wissen, dass es sich um ein Link (https://) handelt.

b) Stellen Sie sich vor, dass ein LCG das Schlüsselstromsymbol S_{i+1} auf Basis der letzten n Elemente $S_i, S_{i-1}, S_{i-2}, \ldots, S_{i-n+1}$ berechnet. Wie viele Parameter $A, B, C, \ldots$ sowie initiale *seed*-Werte werden für diesen LCG benötigt? Wie viele Klartext-Chiffretext-Paare werden für einen Angriff benötigt? (10 Pkt.)

a	b	$a \cdot b \bmod 257$	a	b	$a \cdot b \bmod 257$	a	b	$a \cdot b \bmod 257$	a	b	$a \cdot b \bmod 257$
1	1	1	2	129	1	3	86	1	4	193	1
5	103	1	6	43	1	7	147	1	8	225	1
9	200	1	10	180	1	11	187	1	12	150	1
13	178	1	14	202	1	15	120	1	16	241	1
17	121	1	18	100	1	19	230	1	20	90	1
21	49	1	22	222	1	23	190	1	24	75	1
25	72	1	26	89	1	27	238	1	28	101	1
29	195	1	30	60	1	31	199	1	32	249	1
33	148	1	34	189	1	35	235	1	36	50	1
37	132	1	38	115	1	39	145	1	40	45	1
41	163	1	42	153	1	43	6	1	44	111	1
45	40	1	46	95	1	47	175	1	48	166	1
49	21	1	50	36	1	51	126	1	52	173	1
53	97	1	54	119	1	55	243	1	56	179	1
57	248	1	58	226	1	59	61	1	60	30	1
61	59	1	62	228	1	63	102	1	64	253	1
65	87	1	66	74	1	67	234	1	68	223	1
69	149	1	70	246	1	71	181	1	72	25	1
73	169	1	74	66	1	75	24	1	76	186	1
77	247	1	78	201	1	79	244	1	80	151	1
81	165	1	82	210	1	83	96	1	84	205	1
85	127	1	86	3	1	87	65	1	88	184	1
89	26	1	90	20	1	91	209	1	92	176	1
93	152	1	94	216	1	95	46	1	96	83	1
97	53	1	98	139	1	99	135	1	100	18	1
101	28	1	102	63	1	103	5	1	104	215	1
105	164	1	106	177	1	107	245	1	108	188	1
109	224	1	110	250	1	111	44	1	112	218	1
113	116	1	114	124	1	115	38	1	116	113	1
117	134	1	118	159	1	119	54	1	120	15	1
121	17	1	122	158	1	123	140	1	124	114	1
125	220	1	126	51	1	127	85	1	128	255	1
129	2	1	130	172	1	131	206	1	132	37	1
133	143	1	134	117	1	135	99	1	136	240	1
137	242	1	138	203	1	139	98	1	140	123	1
141	144	1	142	219	1	143	133	1	144	141	1
145	39	1	146	213	1	147	7	1	148	33	1
149	69	1	150	12	1	151	80	1	152	93	1
153	42	1	154	252	1	155	194	1	156	229	1
157	239	1	158	122	1	159	118	1	160	204	1
161	174	1	162	211	1	163	41	1	164	105	1
165	81	1	166	48	1	167	237	1	168	231	1
169	73	1	170	192	1	171	254	1	172	130	1
173	52	1	174	161	1	175	47	1	176	92	1
177	106	1	178	13	1	179	56	1	180	10	1
181	71	1	182	233	1	183	191	1	184	88	1
185	232	1	186	76	1	187	11	1	188	108	1
189	34	1	190	23	1	191	183	1	192	170	1
193	4	1	194	155	1	195	29	1	196	198	1
197	227	1	198	196	1	199	31	1	200	9	1
201	78	1	202	14	1	203	138	1	204	160	1
205	84	1	206	131	1	207	221	1	208	236	1
209	91	1	210	82	1	211	162	1	212	217	1
213	146	1	214	251	1	215	104	1	216	94	1
217	212	1	218	112	1	219	142	1	220	125	1
221	207	1	222	22	1	223	68	1	224	109	1
225	8	1	226	58	1	227	197	1	228	62	1
229	156	1	230	19	1	231	168	1	232	185	1
233	182	1	234	67	1	235	35	1	236	208	1
237	167	1	238	27	1	239	157	1	240	136	1
241	16	1	242	137	1	243	55	1	244	79	1
245	107	1	246	70	1	247	77	1	248	57	1
249	32	1	250	110	1	251	214	1	252	154	1
253	64	1	254	171	1	255	128	1	256	256	1

Tabelle 1: Multiplikative Inverse modulo $m = 257$.