Grundlagen der IT-Sicherheit

Übung 2

Gruppenmitglieder: Alexander Baumann
Johannes Theiner

Semester: Wintersemester 2018/19
letzte Änderung: 07.12.2018

1 Berechnen Sie jeweils ohne Taschenrechner.

1.1

$$\begin{aligned} & 19 * 23 \mod 17 \\ = & \ 2 * 6 \mod 17 \\ = & \ 12 \mod 17 \end{aligned}$$

1.2

$$\begin{aligned} & -15 * 23 \mod 17 \\ = & \ 2 * 6 \mod 17 \\ = & \ 12 \mod 17 \end{aligned}$$

2 Vervollständigen Sie die Verknüpfungstafeln für $\mathbb{Z}/5\mathbb{Z}$

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$*$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

3 Modulare Arithmetik II

3.1 Welche Elemente in $\mathbb{Z}/8\mathbb{Z}, \mathbb{Z}/11\mathbb{Z}$ haben keine multiplikative Inverse?

$\mathbb{Z}/8\mathbb{Z}$: $2, 3, 6$
$\mathbb{Z}/11\mathbb{Z}$: keine

3.2 Wie unterscheidet sich $\mathbb{Z}/p\mathbb{Z}$ von $\mathbb{Z}/8\mathbb{Z}$, falls p eine Primzahl ist?

$\mathbb{Z}/p\mathbb{Z}$ und $\mathbb{Z}/8\mathbb{Z}$ unterscheiden sich darin, dass bei $\mathbb{Z}/p\mathbb{Z}$ alle Zahlen im Bereich $1 - p - 1$ multiplikative Inverse besitzen und in $\mathbb{Z}/8\mathbb{Z}$ nicht.

3.3 Gegeben sei die Primzahl 7. Berechnen Sie für $5 \in \mathbb{Z}/7\mathbb{Z}^*$ die Potenzen $5^i, i = 1, 2, \ldots$ bis zum ersten Mal $5^i \equiv 1 \mod 7$ auftritt.

$$\begin{aligned} 5^1 &\mod 7 = 5 \\ 5^2 &\mod 7 = 4 \\ 5^3 &\mod 7 = 6 \\ 5^4 &\mod 7 = 2 \\ 5^3 &\mod 7 = 3 \\ 5^6 &\mod 7 = 1 \end{aligned}$$

3.3.1 Ist die Existenz eines solchen i a priori klar?

Bei jeder Primzahl gibt es ein multiplikatives Inverses.

3.3.2 Was ändert sich, wenn man die Potenzen von $2 \in \mathbb{Z}/8\mathbb{Z}$ betrachtet. Warum existiert jetzt kein i mit $2^i \equiv 1 \mod 8$.

2 ist ein Teiler von 8 und somit kann es kein multiplikatives Inverses von 2 in $\mathbb{Z}/8\mathbb{Z}$ geben.

3.4 Gegeben sei $n = 1065023 = 1031 * 1033$. Bestimmen Sie für $7, 9$ und 11 ein $i \in N > 0$ mit $7^i \equiv 9^i \equiv 11^i \equiv 1 \mod n$.

$$\begin{aligned} A^{\Phi(n)} &= 1 \mod n \\ n &= p * q \\ \Phi(n) &= (p-1) * (q-1) \end{aligned}$$

$$\begin{aligned} \Phi(1065023) &= (1031-1) * (1033-1) = 1062960 \\ 7^{1062960} &= 9^{1062960} = 11^{1062960} = 1 \mod 1065023 \end{aligned}$$

4 Modulare Arithmetik III

Berechnen Sie jeweils ohne Taschenrechner. Die Ergebnisse sollen dabei im Bereich 0,. . . , Modulus-1 liegen.

4.1 $10^{1313} \mod 11 = 10$

$$\begin{aligned}(-1)^{1313} &= 1 \mod 11\\ &= (-1)^{1313} \mod 11\\ &= -1 \mod 11 = 10\end{aligned}$$

4.2 $4 * x = 4 \mod 8$

$$\begin{aligned}(4 * x = 4) &\mod 8\\ x &= 2n + 1\end{aligned}$$

4.3 $3x + 4 = 2 \mod 5$

$$\begin{aligned}3x + 4 = 2 &\mod 5\\ x &= 1\end{aligned}$$

5 RSA Ver- und Entschlüsselung

Vervollständigen Sie zunächst das Schlüsselpaar und führen Sie zunächst eine RSA Verschlüsselung und anschließend eine Entschlüsselung mit folgenden Systemparametern durch: $p = 19, q = 43, e = 5, x = 143$ Bitte geben Sie jeweils die Chiffrate und d an. Begründen Sie kurz, wie Sie d ermittelt haben.

$$\begin{aligned}N &= 817\\ d &= 605\\ Codiert &= 668\\ Dekodiert &= 143\end{aligned}$$

d ist die erste Zahl für die gilt: $e * d \mod ((p-1) * (q-1)) = 1$

6 Stromchiffren

6.1 Gegeben sei das LFSR. Vervollständigen Sie die Tabelle.

Rückkopplungsvorschrift: $s_{i+3} = s_{i+2} + s_i \mod 2, i \geq 0$

Anfangsbelegung: $r_2 = 1$,$r_1 = 0$,$r_0 = 1$.

Ausgabefolge s_i	r_2	r_1	r_0
1	1	0	1
0	0	1	0
1	0	0	1
0	1	0	0
0	1	1	0
1	1	1	1
0	0	1	1
1	1	0	1

6.2 Bestimmen Sie die Periode.

Die Ausgabesequenz wiederholt sich nach 7 Takten.

6.3 Bestimmen Sie die Frobeniusbegleitmatrix.

$$F = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

6.4 Warum wäre das System gebrochen, wenn man z.B. $s_1, \ldots, s_3$ kennt ?

Das System wäre gebrochen da s_i auch für die Entschlüsselung verwendet wird.

6.5 Wäre das System auch gebrochen, wenn mann z.B. $s_3, \ldots, s_{130}$ kennt ?

Ja, auch hier wird s_i für Ver- und Entschlüsselung verwendet.

6.6 Nehmen Sie nun an, dass obiges LFSR wird mit dem Filter $x_1x_0 + x_2x_0 + x_2x_1 + x_2 + x_1 + x_0$ kombiniert.

6.6.1 Berechnen Sie die ersten 3 Ausgabebits ausgehend von obiger Startbelegung.

$z_0 = 1, z_1 = 1, z_3 = 1$

6.6.2 Wäre das System immer noch gebrochen, wenn man $s_1, \dots s_3$ kennt ?

Das System wäre immer noch gebrochen wenn auch der Filter bekannt ist und so alles ausgehend von $s_1, \dots s_3$ berechnet werden kann.

6.7 Warum bietet es sich an, bei der Implementierung mit der Frobeniusbeleitmatrix zu arbeiten ?

Das auslesen und verarbeiten einer Matrix ist einfacher als bei einem Textes. Es kann sofort berechnet werden und kein Text zusätzlich ausgewertet werden.

6.8 Gegeben sei nun das LFSR. Vervollständigen Sie die Tabelle.

$$F = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Ausgabefolge s_i	r_2	r_1	r_0
1	1	0	1
0	1	1	0
1	0	1	1
1	1	0	1
0	1	1	0
1	0	1	1
1	1	0	1
0	1	1	0

6.9 Welche Periode ergibt sich hier ?

Es ergibt sich eine Periode von 3.

6.10 Welche Dimension hat der von $F^i * \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ erzeugte Untervektorraum ?

Der erzeugte Untervektorraum hat eine Dimension von 2.

6.10.1 Welche Dimension ergibt sich bei 6.1 ?

Der bei 6.1 erzeugte Untervektorraum hat eine Dimension von 3.

6.10.2 Welcher Zusammenhang ergibt sich zu den Perioden ?

p = Periode
d = Dimension

$$p = 2^d - 1$$

6.11 Wie lautet die Rückkopplungsvorschrift ?

Die Rückkopplungsvorschrift lautet $s_{i+3} = s_i \mod 2, i \geq 0$

7 OpenSSL

7.1 Verschlüsseln die eine Textdatei dessen Inhalt sich etwa 50 mal wiederhohlt mittels AES-192-ECB.

7.2 Was fällt Ihnen am Geheimtext auf und wie lässt sich das Ergebnis interpretieren ?

Der Geheimtext wiederhohlt sich alle 192 Zeichen. Jedem Zeichen im Klartext wurde ein Zeichen, basierend auf der Position, im verschlüsselten Text zugeordnet.

7.3 Wie lässt sich ein solcher Geheimtext vermeiden ?

Einen Operationsmodus verwenden der bei der Verschlüsselung zusätzlich Daten aus dem vorherigem Block verwendet. Ein solcher Modus verschlüsselt gleiche Zeichen je nach Block unterschiedlich.

7.4 Erzeugen Sie ein RSA-Schlüsselpaar mit RSA-2048 & 192 Bit AES Verschlüsselung.

7.5 Geben Sie den RSA-Schlüssel aus und erläutern Sie die Ausgabe.

Private-Key: (2048 bit)
modulus:
00:ad:1c:2e:a8:...
publicExponent: 65537 (0x10001)
privateExponent:
30:7d:05:60:...
prime1:
00:de:c4:6a:16:39:49:...
prime2:
00:c6:ef:56:75:96:74:ef:...
exponent1:
7a:bf:9f:f5:3e:fb:e6:07:...
exponent2:
43:5d:af:28:98:af:b6:8c:...
coefficient:
62:4d:2d:1b:5b:78:9c:63...
writing RSA key

- Modulus n
- Öffentlicher Exponent e
- privater Exponent d
- Primzahl p
- Primzahl q
- d mod (p-1)
- d mod (q-1)
- (Inverses von q) mod p

—BEGIN RSA PRIVATE KEY—
MIIEogIBAAKCAQEArRwuqIljPOd8ailRXz+tYmM...
—END RSA PRIVATE KEY—

Key in base64 koddiert

7.6 Extrahieren Sie den öffentlichen RSA-Schlüssel und speichern Sie ihn im PEM-Format ab.

```
Enter pass phrase for rsaprivatekey.pem:
writing RSA key
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArRwuqIljPOd8ailRXz+t
YmMjObxkeTI3OyeBd9RdJnDysCZIhGIysuUn+9ZukhTwOkGELeYbblW6OhshI6eQ
RonYKyqMqvOxmoFgp3AILCLTLPCuBdoVEEWppHCOiA6DK6IPYTjIzxVXjI8DMQM1
PEri898srspRxsAEG4R6vimtSJTF8QT+BKGRL4dVRtFJy1LywvFtHKFiFKZKndZV
44EF2wimOYNx+DDqEsZXhmjwlBQxCNwTAYr9trpjMHBnfWLepS4EarhYDr6aAtf3
KFeovxDO9jNPKgWjkfQboHJpDwhwLoxmR9GndndaaEwPZQuXMF3SdwhvL5PzlLIl
7wIDAQAB
-----END PUBLIC KEY-----
```

7.7 Erstellen Sie eine Textdatei und verschlüsseln Sie diese mittels RSA.

7.8 Entschlüsseln Sie die Datei und vergleichen Sie mit der ursprünglichen Textdatei.

Inhalt der Datei ist gleich, Hashwert hat sich nicht verändert.

7.9 Erzeugen Sie sich ein weiteres RSA-Schlüsselpaar, mit dem Sie ihre Textdatei signieren und verifizieren.

RSA: Keypair generieren

Datei signieren und verifizieren

7.10 Warum ist es sinnvoll, für Verschlüsselung und Signatur zwei unterschiedliche RSA-Schlüsselpaare zu verwenden ?

- Erhält ein Angreifer einen Schlüssel kann er nur die Aktion ausführen für die der Schlüssel erstellt wurde
- Kommt ein Schlüssel abhanden muss nur dieser ausgetauscht werden
- unterschiedliche Gültigkeitszeiträume können vergeben werden

7.11 Lassen Sie sich das Zertifikat der Hochschulwebseite anzeigen und speichern Sie es im PEM-Format ab.

7.12 Geben Sie die Zertifikatskette an.

```
depth=3 C = DE, O = Deutsche Telekom AG,
OU = T-TeleSec Trust Center, CN = Deutsche Telekom Root CA 2
verify return:1
depth=2 C = DE, O = DFN-Verein,
OU = DFN-PKI, CN = DFN-Verein PCA Global - G01
verify return:1
depth=1 C = DE, ST = Niedersachsen, O = Hochschule Emden/Leer,
CN = HS-EL CA, emailAddress = pki@hs-emden-leer.de
verify return:1
depth=0 C = DE, ST = Niedersachsen, L = Emden,
O = Hochschule Emden/Leer, OU = HRZ, CN = www.hs-emden-leer.de
verify return:1
```

7.13 Was ist die Root Certification Authority (Root-CA) ?

Die Deutsche Telekom AG, erkennbar daran das der Wert für depth hier am höchsten ist.

7.14 Wandeln Sie das Zertifikat in das DER-Format um.

7.15 Geben Sie die SHA1- und MD5 Hashwerte an.

md5:

- pem: 60563c95125efbdc863c1a9f6280bf53
- der: 2f7b5a228d4824ad6b58c6e035256288

sha1:

- pem: 78a1f17660ffe9a4d25c943824693d7a85cc2877
- der: f21d0a215a3e5731815eadd06fbbeb633dcd2360

7.16 Sind diese im Zertifikat gespeichert ?

nein, diese Werte sind nicht im Zertifikat gespeichert. Der Hashwert der Datei würde sich ändern wenn ein Wert hinzugefügt wird.

7.17 Sind die jeweiligen Werte für die beiden Kodierungen PEM und DER unterschiedlich ?

PEM und DER sind komplett unterschiedliche Formate somit auch komplett unterschiedliche Hashwerte.

7.18 Geben Sie das Zertifikat im lesbaren Textformat auf dem Bildschirm aus.