# VIS-Übungsblatt 7

Hennings, Regorz, Röder, Budde, Warrelmann · WiSe 2017

## 1 Kryptografie

### a)

Der Angreifer könnte relativ einfach mit einer *Meet-in-the-middle attack* die Schlüssel $k_1$ und $k_2$ heraus finden:

$$\begin{aligned} C &= E_{k_2}\left(E_{k_1}(P)\right) \\ \Leftrightarrow D_{k_2}(C) &= D_{k_2}\left(E_{k_2}\left(E_{k_1}(P)\right)\right) \\ \Leftrightarrow D_{k_2}(C) &= E_{k_1}(P) \end{aligned}$$

Hierbei wird der bekannte Klartext P mit allen möglichen Belegungen für $k_1$ verschlüsselt. Beide Werte für $k_1$ und v werden in einer Tabelle gespeichert $[[\{value : k_{1i}\}, \{v_i\}], ...]$. Das selbe geschieht für die bekannte Chiffre C aus P. Diese wird wiederum mit allen möglichen Belegungen für $k_2$ entschlüsselt und auch hier werden die Werte in eine Tabelle geschrieben $[[\{value : k_{2j}\}, \{w_j\}], ...]$ . Gilt nun $v_i = w_j$ für ein bestimmtes Paar i und j, dann sind $k_{1i}$ und $k_{2j}$ die gesuchten Schlüssel.
Der Aufwand beträgt: $2^{56} + 2^{56} = 2 \cdot 2^{56} = 2^{57}$

Der Sicherheitsgewinn wäre nur 1 Bit gegenüber der "versprochenen" Verdopplung ($2 \cdot 56 = 112$).

### b)

Die erwartete Sicherheit eines 3-DES mit 2 Schlüsseln beträgt

$$2^{56} \cdot 2^{56} = 2^{112} \text{ Bit},$$

da der obige Angriff nun nicht mehr möglich ist und Chosen-Plaintext-Angriffe nicht angenommen werden.

Die erwartete Sicherheit eines 3-DES mit 3 Schlüsseln gegenüber Exhaustive-Search betrüge:

$$2^{56} \cdot 2^{56} \cdot 2^{56} = 2^{168} \text{ Bit}$$

Mittels einer abgewandelten *Meet-in-the-middle attack* verringert sie diese jedoch auf 112 Bit. Die ersten beiden Stufen $E(k_1) \rightarrow D(k_2)$ bzw. $E(k_1) \rightarrow E(k_2)$ werden für sämtliche möglichen $k_1$ und $k_2$ durchgeführt (Aufwand: $2^{2 \cdot 56} = 2^{112}$) und in einer Map (SubCipher $\rightarrow (k_1, k_2)$) gespeichert. Da offensichtlich $E_{k_1}\left(D_{k_2}(P)\right) = D_{k_3}(C)$ und $E_{k_1}\left(E_{k_2}(P)\right) = D_{k_3}(C)$ gelten, muss nun noch für alle möglichen $k_3$ $D_{k_3}(C)$ gebildet werden und nach einer Kollision mit einem Eintrag in der vorher angelegten Map gesucht werden (Aufwand: $2^{56}$). Insgesamt beläuft sich der Aufwand somit auf $2^{112} + 2^{56} \approx 2^{112}$.

Angenommen, dass Chosen-Plaintext-Angriffe nicht möglich sind, liefert 3-DES mit 3 Schlüsseln somit ungefähr dieselbe Sicherheit wie 3-DES mit 2 Schlüsseln, auf das derselbe Angriff möglich ist. Aufgrund der verschiedenen Verknüpfungen von $k_1$ mit $k_2$ lässt sich jedoch dabei kein Vorteil daraus ziehen, dass nur zwei verschiedene Schlüssel existieren.

## 3 Cipher-Block-Chaining-Modus

### a)

Zwei identische Klartexte, die lediglich einen anderen Initialisierungsvektor haben, sind nur in der Länge identisch. Der Vektor wird genutzt, um eine vom Schlüssel unabhängige Randomisierung zu erreichen.

### b)

Auch hier unterscheidet sich das Chiffrat stark voneinander, da Folgeblöcke vom Vorgänger abhängen. Ändert sich also ein Bit im ersten Block, hat dies Auswirkungen auf alle folgenden. Die wird unter anderem durch moderne Blockchiffren begünstigt, die selbst für zwei ansonten identische Klartexte, die sich lediglich in einem Bit unterscheiden und mit demselben Schlüssel zwei sehr unterschiedliche Schlüsseltexte erzeugen.

### c)

Eine Änderung eines Bits in einem Chiffratblock hat zur Folge, dass der entsprechende Klartext nicht mehr korrekt entschlüsselt wird und unleserlich ist (Eigenschaft moderne Blockchiffren). Der Klartext des darauffolgenden Block würde sich jedoch lediglich an derselben Stelle um ein Bit unterscheiden. Höchstwahrscheinlich ist die Nachricht damit noch immer leserlich und weist lediglich einen korrupten Buchstaben auf. Der Klartext des wiederum darauffolgenden Blocks wird nicht beeinträchtigt, weil der Bitfehler maximal eine Auswirkung auf seinen direkten Nachfolger hat. Die Blöcke vor Auftreten des Bitfehler sind ebenfalls nicht beeinträchtigt.

### d)

Der Klartext des ersten Blockes würde sich ebenfalls an derselben Stelle um ein Bit unterscheiden. Höchstwahrscheinlich ist die Nachricht damit noch immer leserlich und weist lediglich einen korrupten Buchstaben auf. Alle darauffolgenden Blöcke können problemlos entschlüsselt werden, da der IV dort nicht zur Entschlüsselung erforderlich ist.

## 6 Sicherheit des RSA Verfahrens

### a)

Das RSA-Verfahren basiert auf der Faktorisierungsannahme, die behauptet, dass es nicht effizient möglich ist, die Primfaktoren einer großen Zahl zu berechnen.
Ein Angreifer besitzt den öffentlichen Schlüssel, der beispielsweise die Werte $c = 7$ und $n = 299$ haben kann, und die verschlüsselte Nachricht, die beispielsweise $c(m) = 107$ lauten

kann. Aufgrund der Faktorisierungsannahme, kann der Angreifer die beiden Primfaktoren $p = 13$ und $q = 23$ von $n$ nicht in zumutbarer Zeit bestimmen. Dadurch ist es ihm nicht möglich die Zahl $\phi(n) = (p-1) \cdot (q-1) = (13-1) \cdot (23-1) = 264$ zu bestimmen, die er benötigen würde, um mit Hilfe des erweiterten euklidischen Algorithmus das multiplikative Inverse $d$ von $c$ in $\phi(n)$ mit $d = 151$ zu errechnen. Da der Angreifer also aufgrund der Faktorisierungsannahme mit diesem Verfahren nicht an das $d$ kommen wird, wird er nie herausfinden, dass der Inhalt der Nachricht $c(m)^d \mod n = 107^{151} \mod 299 = 42$ lautet.

### b)

Sei zunächst $m_k$ das $k$-te Zeichen des Textes, $c$ der Exponent aus der zweiten Textdatei sowie $n$ der zugehörige Modulo. Dann enthält die erste Textdatei die Einträge $e_k = m_k^c \mod n$.

Die erste Möglichkeit besteht darin, für ein beliebiges $k$ das $m \in \{0, ..., 256\}$ mit $m^c \mod n = e_k$ durch Ausprobieren zu bestimmen. Dann lässt sich durch Iterieren über $i \in \{0, ...\}$ das $i$ mit $e_k^i \mod n = m$ bestimmen. Dieses $i$ ist das $d$ des privaten Schlüssels, mit dem sich der Rest der Textdatei entschlüsseln lässt.

Die zweite Möglichkeit besteht darin, ein Dictionary $f$ mit $f(m^c \mod n) = m \ \forall m \in \{0, ..., 256\}$ zu erstellen und mithilfe dieses Dictionarys die Datei zu entschlüsseln. Diese Möglichkeit wurde im folgenden Racket-Programm implementiert:

```
1  #lang racket
2  ; ############## Loading Data ##################################
3
4  (define schluesseltexte (file->list "ciphertext.txt"))
5
6  (define publickey (car (file->list "public_key.txt")))
7  (define modulus (cadr (file->list "public_key.txt")))
8
9  ; ############## Creating Dictionany ############################
10
11 (define klartextliste (range 0 256))
12
13 (define dict (map
14   (lambda (klartext)
15     (cons klartext
16       (remainder (expt klartext publickey) modulus)))
17   klartextliste))
18
19 ; ############## Do attack ######################################
20
21 (define (finde-klartext schluesseltext)
22   (car (findf (lambda (entry)
```

```
23        (= schluesseltext (cdr entry)))
24       dict)))
25
26 (list->string (map (compose1
27     integer->char
28     finde-klartext)
29   schluesseltexte))
```

Der berechnete Klartext lautet ”Fuer die VIS-Klausur sind alle Inhalte, die in Uebung und Vorlesung behandelt wurden wichtig. Viel Erfolg! :-)”