Seminar IT-Sicherheit

Cryptoanalytic Attacks on RSA: Simple Elementary Attacks

Cordula Eichhorn

Betreut von

Prof. Dr. Gerd Beuster

Eingereicht am

18. September 2016

Inhaltsverzeichnis

1 Einführung **3**

2 Mathematische Grundlagen **4**
2.1 Quadratische Gleichung 4
2.1.1 a-b-c-Formel zur Lösung von allgemeinen quadratischen Gleichungen 4
2.1.2 p-q-Formel zur Lösung von quadratischen Gleichung in Normalform 4
2.2 Lineare diophantische Gleichung 4
2.3 Erweiterter euklidischer Algorithmus 6

3 Chosen Text Attacks **7**
3.1 Blinding Attack on RSA Signatures 7
3.2 Chosen Ciphertext Attack 9

4 Common Modulus Attack **10**

5 Fixed-Point Attack **12**

6 Guessing ϕ(N) Attack **14**

1 Einführung

Diese Ausarbeitung basiert auf Kapitel 6: Simple Elementary Attacks aus dem Buch **Cryptanalytic Attacks on RSA** von Song Y. Yan [8]. Es werden dabei verschiedene, kryptoanalytische Attacken auf den RSA Algortihmus betrachtet. RSA ist ein asymmetrisches Verschlüsselungsverfahren, welches 1978 von Rivest, Shamir und Adleman veröffentlicht wurde [5].

Der mit RSA verschlüsselte Chiffretext C einer Nachricht M ergibt sich wie folgt:

$$C \equiv M^e \pmod{N}$$

Um aus dem Chiffrentext wieder den Klartext M zu erhalten, wird folgende Berechnung durchgeführt:

$$M \equiv C^d \pmod{N}$$

Dabei ist $N = p \cdot q$ mit p und q zwei Primzahlen, e der Verschlüsselungsexponent und d der Entschlüsselungsexponent. Für e und d gilt $e \cdot d \equiv 1 \pmod{\phi(N)}$ (mit der Eulerschen Phi-Funktion $\phi(N) = (p-1) \cdot (q-1)$). Die vier RSA Parameter $\{d, p, q, \phi(N)\}$ bilden die RSA-Falltür. Dabei sind alle vier Parameter gleich wichtig. Ist einer von ihnen bekannt, kann die RSA-Verschlüsselung komplett gebrochen werden.

Die meisten Attacken auf RSA nutzen die mathematischen Schwächen des Algorithmus aus oder beruhen darauf, dass RSA unsachgemäß benutzt wurde. Diese Attacken werden *indirekte, algorithmische Attacken* genannt. Im Folgenden werden einfache, aber indirekte, elementare, mathematische/algorithmische Attacken betrachtet (auf englisch: *Simple Elementary Attacks*).

Einige der vorgestellten Attacken (z.B. 3.1 Blinding Attack on RSA Signatures) werden verhindert, indem an den Klartext M zufällig Ziffern angehängt werden, bevor dieser verschlüsselt wird. Diese zufälligen Ziffern sollten vor jeder Verschlüsselung unabhängig voneinander neu generiert werden. Man bezeichnet diesen Prozess auch als *Salting*. Wenn zufällige Ziffern am Anfang und Ende eines Klartextes angefügt werden, wird dies als *Padding* bezeichnet. Es ist allerdings anzumerken, dass dies eine stark vereinfachte Darstellung der beiden Prozesse ist. Details zu den Techniken finden sich z.B. im PKCS#1: RSA Cryptography Standard (s. [2]).

2 Mathematische Grundlagen

2.1 Quadratische Gleichung

Eine **quadratische Gleichung** ist eine Gleichung der Form $0 = a \cdot x^2 + b \cdot x + c$ mit der Unbekannten $x \in \mathbb{R}$, den Koeffizienten $a, b, c \in \mathbb{R}$ und $a \neq 0$. Dabei ist $a \cdot x^2$ das quadratische Glied, $b \cdot x$ das lineare Glied und c das absolute Glied. Geometrisch gesehen beschreibt sie die Nullstellen einer Parabel, welche durch eine quadratische Funktion $f(x) = a \cdot x^2 + b \cdot x + c$ gegeben ist.

Eine quadratische Gleichung ist in Normalform, wenn der Koeffizient des quadratischen Glieds den Wert 1 hat. Man schreibt sie meist in der Form $0 = x^2 + p \cdot x + q$.

2.1.1 a-b-c-Formel zur Lösung von allgemeinen quadratischen Gleichungen

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

Der Term $D = b^2 - 4 \cdot a \cdot c$ wird Diskriminante genannt. Aus ihr lässt sich schließen wie viele Lösungen es für die Gleichung gibt. Bei $D < 0$ gibt es im reellen Zahlenbereich keine Lösung für die Gleichung. Für $D = 0$ gibt es genau 1 (doppelte) Lösung, für $D > 0$ zwei Lösungen.

2.1.2 p-q-Formel zur Lösung von quadratischen Gleichung in Normalform

$$x_{1,2} = \frac{-p \pm \sqrt{p^2 - 4 \cdot q}}{2}$$

Für die Diskriminante $D = p^2 - 4 \cdot q$ der quadratischen Gleichung in Normalform gelten die gleichen Aussagen wie für die der allgemeinen quadratischen Gleichung.

2.2 Lineare diophantische Gleichung

Eine Gleichung $a \cdot x + b \cdot y = c$ mit $a, b, c \in \mathbb{Z}$ heißt **lineare diophantische Gleichung mit zwei Variablen**, wenn als Lösung nur Paare $(x, y) \in \mathbb{Z}^2$ zugelassen sind. Eine lineare diophantische Gleichung ist genau dann lösbar, wenn gilt $\text{ggT}(a, b) | c$. Zur Lösung dieser Gleichungen kann z.B. das Eulersche Reduktionsverfahren angewendet werden, welches alle ganzzahligen Lösungen liefert [3, S. 61ff].

Beispiel 2.1. nach [3, S. 63f]

Im Folgenden werden für die lineare diophantische Gleichung $3 \cdot x + 5 \cdot y = 1$ alle Lösungspaare $x, y \in \mathbb{Z}^2$ berechnet, welche die Gleichung erfüllen. Dazu wird das Eulersche Reduktionsverfahren verwendet.

Die Gleichung ist lösbar, da ggT(3, 5 = 1) und 1|1.

1 Die Gleichung wird nach der Variable mit dem kleinsten Koeffizient umgestellt (in diesem Fall x):

$$x = \frac{1 - 5 \cdot y}{3}$$

2 Die Division wird nun soweit wie möglich ausgeführt:

$$x = -y + \frac{1 - 2 \cdot y}{3}$$

3 Da die Lösung für die Gleichung ganzzahlig ist, muss auch der Bruch ganzzahlig sein. Es wird eine neue, ganzzahlige Variable a eingeführt. Durch Umformung entsteht eine neue, lineare diophantische Gleichung:

$$a = \frac{1 - 2 \cdot y}{3}$$
$$3 \cdot a = 1 - 2 \cdot y$$

Auf die neue Gleichung werden die drei Schritte wieder angewendet, bis eine Gleichung gefunden wird in der einer der Koeffizienten der Variablen den Wert 0 hat:

1
$$y = \frac{1 - 3 \cdot a}{2}$$

2
$$y = -a + \frac{1 - 1 \cdot a}{2}$$

3 Es wird eine neue Variable b eingeführt:

$$b = \frac{1 - 1 \cdot a}{2}$$
$$2 \cdot b = 1 - 1 \cdot a$$

Da nun eine Variable mit dem Koeffizienten 1 gefunden wurde, ist der Algorithmus beendet. Um eine Lösung zu erhalten, muss nun zuerst $a = 1 - 2 \cdot b$ in $y = \frac{1-3 \cdot a}{2}$

eingesetzt werden, und das Ergebnis dann in $x = \frac{1-5\cdot y}{3}$:

$$\begin{aligned} y &= \frac{1-3\cdot a}{2} = \frac{1-3\cdot(1-2\cdot b)}{2} = \frac{-2+6\cdot b}{2} \\ &= -1+3\cdot b \end{aligned}$$

$$\begin{aligned} x &= \frac{1-5\cdot y}{3} = \frac{1-5\cdot(-1+3\cdot b)}{2} = \frac{6-15\cdot b}{3} \\ &= 2-5\cdot b \end{aligned}$$

Die Lösung lautet folglich:

$$\begin{aligned} y &= -1+3\cdot b \\ x &= 2-5\cdot b \end{aligned}$$

für ein beliebiges, ganzzahliges b.

2.3 Erweiterter euklidischer Algorithmus

Mit dem **euklidischen Algorithmus** kann man den größten gemeinsamen Teiler zweier natürlichen Zahlen $\text{ggT}(a,b)$ berechnen. Der erweiterte euklidische Algorithmus berechnet zusätzlich dazu zwei Zahlen $x, y \in \mathbb{Z}$ für die gilt: $a\cdot x + b\cdot y = \text{ggT}(a,b)$.

3 Chosen Text Attacks

Chosen Text Attacks, zu deutsch ausgewählte Text Attacken, können durch zufällige Paddingtechniken vermieden werden. Im Folgenden wird zunächst *Blinding Attack on RSA Signatures* vorgestellt, die darauf basiert, dass der Angreifer einen Klartext auswählen kann (gehört daher auch zu den sogenannten *chosen plaintext attacks*). Anschließend folgt die *Chosen Ciphertext Attack*, die - wie ihr Name schon vermuten lässt - darauf beruht, dass der Angreifer den Chiffretext wählen bzw. nach eigenen Wünschen verändern kann.

3.1 Blinding Attack on RSA Signatures

Diese einfache elementare Attacke zielt auf RSA-Signaturen ab und nutzt dabei die Selbst-Reduzierbarkeit (self-reducibility) von RSA aus. Sie gelingt allerdings nur, wenn auf Padding, bzw. Salting verzichtet wird. Werden zufällige Paddingtechniken eingesetzt, ist die Attacke sofort zunichtegemacht.

Angenommen von Bob ist (e, N, M) bekannt und die Cryptoanalystin, die Angreiferin, Eve möchte die digitale Signatur S erfahren. Das heißt

$$\{e, N, m \equiv S^e \pmod N\} \xrightarrow[\text{forging } S]{\text{find}} \{S\}$$

Seien (e, N) und (d, N) Bobs öffentlicher und privater Schlüssel. Bobs Signatur S einer Nachricht $M \in \mathbb{Z}_N^*$(mulitplikative Gruppe $\mathbb{Z}_N^* = \{a \in \mathbb{Z}_N : ggT(a, N) = 1\}$) wird wie folgt berechnet:

$$S \equiv M^d \pmod N$$

Angenommen die Cryptoanalystin Eve möchte Bobs Signature S finden, bzw. fälschen. Dann wäre ihre Vorgehensweise:

1 Eve wählt eine zufällige Zahl $r \in \mathbb{Z}_N^*$ und berechnet

$$M' \equiv r^e \cdot M \pmod N$$

2 Eve bittet nun Bob die zufällige Nachricht M' zu signieren. Diese sieht - so wie es sein sollte - wie ein zufälliger Hashwert aus.

3 Ist Bob bereit die Nachricht M' zu signieren bekommt Eve

$$S' \equiv (M')^d \pmod N$$

4 Eve gelangt anschließend einfach an Bobs Signatur S:

$$S \equiv \frac{S'}{r} \pmod N$$

Das folgt aus

$$
\begin{aligned}
S^e &\equiv \left(\frac{S'}{r}\right)^e && \pmod N \\
&\equiv \frac{(S')^e}{r^e} && \pmod N && \mid\ S' \equiv (M')^d \\
&\equiv \frac{\left((M')^d\right)^e}{r^e} && \pmod N && \mid\ e \cdot d = 1 \\
&\equiv \frac{M'}{r^e} && \pmod N && \mid\ M' \equiv r^e \cdot M \\
&\equiv \frac{r^e \cdot M}{r^e} \\
&\equiv M && \pmod N
\end{aligned}
$$

Eve hat erfolgreich Bobs gültige Signatur gefälscht ohne dass sie seinen privaten Exponenten d kannte. Bob kann die Fälschung nicht entdecken, da $M \equiv S^e \pmod N$.

Beispiel 3.1.
Es seien $e = 23, N = 143, M = 7$ Eve bekannt. Sie möchte Bobs Signatur S herausfinden. Sie wählt $r = 42$ und berechnet

$$
\begin{aligned}
M' &\equiv r^e \cdot M \pmod N \\
&\equiv 76
\end{aligned}
$$

Bob signiert Eve nun die Nachricht M' mit seinem privaten Exponenten $d = 47$ und Eve erhält

$$
\begin{aligned}
S' &\equiv (M')^d \pmod N \\
&\equiv 32
\end{aligned}
$$

Nun gelangt Eve an Bobs gültige Signatur S:

$$
\begin{aligned}
S &\equiv \frac{S'}{r} \pmod N \\
&\equiv 28
\end{aligned}
$$

Für Bobs Signatur S gilt:

$$
\begin{aligned}
S &\equiv M^d \pmod N \\
&\equiv 28
\end{aligned}
$$

Dies entspricht dem von Eve berechneten Wert, Eve hat erfolgreich Bobs Signatur gefälscht.

3.2 Chosen Ciphertext Attack

Diese ausgewählte Chiffretext-Attacke ähnelt der *Blinding Attack on RSA Signatures* (3.1). Sie benutzt ebenfalls das Prinzip der Multiplikation mit einer zufälligen Zahl $r \in \mathbb{Z}_N^*$. Allerdings hat es die Angreiferin Eve nicht auf die Signatur S von Bob, sondern auf den Klartext M einer verschlüsselten Nachricht $C \equiv M^e \pmod N$ von Bob an Alice abgesehen. Die Vorgehensweise ist:

1 Die Angreiferin Eve fängt einen Chiffretext C von Bob an Alice ab.

2 Eve wählt eine zufällige Zahl $r \in \mathbb{Z}_N^*$, berechnet

$$\widetilde{C} \equiv C \cdot r^e \pmod N$$

und sendet $\widetilde{C}$ an Alice.

3 Alice entschlüsselt den Chiffretext $\widetilde{C}$

$$\begin{aligned} \widetilde{M} &\equiv \widetilde{C}^d && \pmod N && \mid \widetilde{C} \equiv C \cdot r^e \\ &\equiv (C \cdot r^e)^d && \pmod N && \mid C \equiv M^e \\ &\equiv (M^e \cdot r^e)^d && \pmod N \\ &\equiv M^{e \cdot d} \cdot r^{e \cdot d} && \pmod N && \mid e \cdot d = 1 \\ &\equiv M \cdot r && \pmod N \end{aligned}$$

4 Gelangt Eve nun auf irgendeine Weise an $\widetilde{M} \equiv M \cdot r$, kann sie die ursprüngliche Nachricht M berechnen:

$$M \equiv r^{-1} \cdot \widetilde{M} \pmod N$$

Beispiel 3.2.
Es sei $e = 23, N = 143$ Eve bekannt. Sie fängt nun den Chiffretext $C = 2$ von Bob an Alice ab. Mit der zufällig gewählten Zahl $r = 42$ berechnet sie

$$\begin{aligned} \widetilde{C} &\equiv C \cdot r^e \pmod N \\ &\equiv 83 \end{aligned}$$

und schickt $\widetilde{C}$ an Alice. Diese entschlüsselt den Chiffretext und erhält:

$$\begin{aligned} \widetilde{M} &\equiv \widetilde{C}^d \pmod N \\ &\equiv 8 \end{aligned}$$

Gelangt Eve nun auf an $\widetilde{M}$, berechnet sie M mit $r^{-1} = 126$

$$\begin{aligned} M &\equiv r^{-1} \cdot \widetilde{M} \pmod N \\ &\equiv 7 \end{aligned}$$

Damit hat Eve erfolgreich den Klartext M wiederhergestellt, denn

$$\begin{aligned} C &\equiv M^e \pmod N \\ &\equiv 2 \end{aligned}$$

4 Common Modulus Attack

Ähnlich wie die vorangegangenen Attacken wird diese Attacke ermöglicht, wenn RSA nicht angemessen verwendet wird. Allerdings ermöglicht hier nicht mangelndes Padding einen Angriff, sondern die Verwendung eines gemeinsamen Modulus N. Der Angreifer braucht dann keinen der vier RSA Parameter $\{d, p, q, \phi(N)\}$ zu wissen, um die Verschlüsselung zu brechen.

Angenommen Bob sendet Alice zwei verschlüsselte Chiffretexte C_1 und C_2:

$$C_1 \equiv M_1^{e_1} \pmod{N_1}$$
$$C_2 \equiv M_2^{e_2} \pmod{N_2}$$

in denen der $ggT(e_1, e_2) = 1$, d.h. der größter gemeinsamer Teiler von e_1 und e_2 ist 1.

Das folgende Theorem zeigt, dass, wenn ein gemeinsamer Modulus N verwendet wird, die Angreiferin Eve den unverschlüsselten Klartext M ohne Faktorisierung von N, oder ohne dass sie eine der Falltürinformationen $(d, p, q\phi(N))$ kennt, wiederherstellen kann.

Theorem 4.1.
Es seien $N_1 = N_2$ und $M_1 = M_2$ und $e_1 \neq e_2$ und $ggT(e_1, e_2) = 1$, so dass

$$C_1 \equiv M^{e_1} \pmod{N}$$
$$C_2 \equiv M^{e_2} \pmod{N}$$

Dann kann M wiederhergestellt werden in polynomieller Zeit, also

$$\{[C_1, e_1, N], [C_2, e_2, N]\} \overset{\mathcal{P}}{\Rightarrow} \{M\}$$

Beweis 4.1.
Da $ggT(e_1, e_2) = 1$, gilt $e_1 \cdot x + e_2 \cdot y = 1$ mit $x, y \in \mathbb{Z}$. Die Lösung für x, y kann in Polynomialzeit berechnet werden, z.B. mit dem erweiterten Euklidischen Algorithmus (siehe 2.3). Es folgt

$$\begin{aligned} C_1^x \cdot C_2^y &\equiv (M^{e_1})^x \cdot (M^{e_2})^y \\ &\equiv M^{e_1 x + e_2 y} \\ &\equiv M \pmod{N} \end{aligned}$$

Damit ist gezeigt, dass der Klartext M in polynomieller Zeit wiederhergestellt werden kann.

Beispiel 4.1.
Es seien

$$\begin{aligned} e_1 &= 3 \\ e_2 &= 5 \\ N &= 2369 \\ M &= 515. \end{aligned}$$

Dann ergeben sich die folgenden beiden Chiffretexte:

$$\begin{aligned} C_1 &\equiv M^{e_1} \\ &\equiv 1442 \pmod{N} \\ C_2 &\equiv M^{e_2} \\ &\equiv 721 \pmod{N} \end{aligned}$$

Anschlißend bestimmen wir x und y in der Gleichung

$$3x + 5y = 1$$

Dazu kann der Erweiterte Euklidische Algorithmus (2.3) verwendet werden oder das Eulersche Reduktionsverfahren zur Lösung von diophantischen Gleichungen (2.2). Es ergeben sich:

$$\begin{aligned} x &= 2 \\ y &= -1 \end{aligned}$$

Anschließend wird M wie folgt berechnet:

$$\begin{aligned} M &\equiv C_1^x \cdot C_2^y \\ &\equiv 1442^2 \cdot 721^{-1} \\ &\equiv 2884 \\ &\equiv 515 \pmod{N} \end{aligned}$$

Es konnte also der Klartext M wiederhergestellt werden, ohne dass N faktorisiert oder eine der Falltürinformationen $d, p, q, \phi(N)$ benutzt werden mussten.

Es ist leicht ersichtlich, dass diese Attacke verhindert wird, wenn kein gemeinsamer Modulus benutzt wird.

5 Fixed-Point Attack

Eine weitere Attacke auf RSA, die nicht darauf basiert, dass N faktorisiert wird oder eine der Falltürinformationen bekannt sind, ist die Fixed-Point Attack. Sie wird auch als *Cyclic Attack*, *Cycling Attack* oder *Superencryption Attack* bezeichnet. Diese Attacke wurde bereits 1977 entdeckt, kurz nachdem RSA veröffentlicht wurde. Allerdings halten Rivest und Silverman [6] die Wahrscheinlichkeit, dass eine zyklische Attacke von Erfolge gekrönt ist, für verschwindend gering.

Gilt bei einer RSA-Verschlüsselung für $1 \leq x < N$ dass

$$x^e \equiv x \pmod{N}$$

dann sagt man x ist ein Fixpunkt (englisch: *fixed point*) von RSA(e, N). Gilt

$$x^{e^k} \equiv x \pmod{N}, k \in \mathbb{Z}^+$$

dann ist x ein Fixpunkt der Ordnung k (*fixed point of order k*).

Theorem 5.1.
Es sei C ein Fixpunkt von RSA(e, N) mit der Ordnung k:

$$C^{e^k} \equiv C \pmod{N}, k \in \mathbb{Z}^+$$

dann gilt

$$C^{e^{k-1}} \equiv M \pmod{N}, k \in \mathbb{Z}^+$$

Beweis 5.1.
Da die RSA-Verschlüsselung $C \equiv M^e \pmod{N}$ ein Element des Nachrichtenraums $\{0, 1, 2, ..., N-1\}$ ist, muss eine Zahl (Fixpunkt) $C^{e^k} \equiv C \pmod{N}$ existieren. Aus dem selben Grund muss

$$C^{e^{k-1}} \equiv M \pmod{N}$$

gelten, da

$$\begin{array}{rrcll} & C^{e^k} & \equiv & C & \pmod{N} \quad | \; C \equiv M^e \\ \Longrightarrow & C^{e^k} & \equiv & M^e & \pmod{N} \quad | \; C^{e^k} \text{ erweitern zu } C^{e^k \cdot \frac{e}{e}} \\ \Longrightarrow & C^{e^{k-1} \cdot e} & \equiv & M^e & \pmod{N} \\ \Longrightarrow & \left(C^{e^{k-1}}\right)^e & \equiv & M^e & \pmod{N} \\ \Longrightarrow & C^{e^{k-1}} & \equiv & M & \pmod{N} \end{array}$$

Theorem 5.1 zeigt eine einfache, direkte Attacke auf RSA, indem die folgende Sequenz von Zahlen berechnet wird (Modulo N):

$$\begin{array}{ccccccc} C^e & C^{e^2} & C^{e^3} & ... & C^{e^{k-1}} & C^{e^k} \\ & & & & \Uparrow & \Downarrow \\ & & & & M & C \end{array}$$

Sobald $C^{e^k} \pmod N \equiv C$ erreicht wird, hört die Berechnung auf und die vorletzte Zahl $C^{e^{k-1}} \pmod N$ wird gewählt, welche dem Klartext M der RSA-Verschlüsselung entspricht.

Beweis 5.2.
Zur Verdeutlichung der Fixpunkt-Attacke wählen wir die folgenden Parameter:

$$\begin{aligned} e &= 23 \\ N &= 143 \\ C &= 2 \end{aligned}$$

Wir berechnen die Sequenz C^{e^k} wie folgt (nach [8] S. 165):

$e \leftarrow 23$
$N \leftarrow 143$
$C \leftarrow 2$
for k from 1 to 100 do
    $A \leftarrow C^{e^k} \pmod N$
    print(A)
    if $A = C$ then
        $M \leftarrow C^{e^{k-1}} \pmod N$
        break

Und erhalten die Sequenz:

$$\begin{array}{cccc} 85 & 28 & 7 & 2 \\ & & \Uparrow & \Downarrow \\ & & M & C \end{array}$$

Da $C \equiv 2^{23^4} \pmod{143}$ ein Fixpunkt von RSA$(23, 143)$ mit der Ordnung $k = 4$ ist, ist $M \equiv C^{23^3} \equiv 7 \pmod{143}$ der Klartext des Chiffretexts C. Dass dies tatsächlich der Fall ist, lässt sich einfach zeigen, denn

$$2^{23} \pmod{143} \equiv 2$$

In diesem Beispiel wurde keine der Falltürinformationen von RSA $(d, p, q, \phi(N))$ benutzt um den Chiffretext C zu brechen.

6 Guessing ϕ(N) Attack

Die letzte hier vorgestellte Attacke beschreibt, was passiert, wenn der korrekte Wert von $\phi(N)$ erraten werden kann. In dem Fall wäre der RSA Klartext M ausgehend von dem zugehörigen Chiffretext C in Polynomialzeit berechenbar. Das heißt

$$\phi(N) \overset{\mathcal{P}}{\Longrightarrow} \{M\}$$

Zuerst zeigen wir, dass die Berechnung von $\phi(N)$ und die Faktorisierung von N, $IFP(N)$ deterministisch Polynomialzeit äquivalent sind.

Theorem 6.1 (Die Äquivalenz von $\phi(N)$ und IFP(N))**.**

$$\phi(N) \overset{\mathcal{P}}{\Longleftrightarrow} \text{IFP}(N)$$

Beweis 6.1.
Wenn $(N, \phi(N))$ bekannt ist und angenommen wird, dass N das Produkt zweier Primzahlen p und q ist, dann kann N einfach faktorisiert werden. Angenommen

$$N = pq$$

dann ist

$$\begin{aligned}\phi(N) &= (p-1)(q-1)\\ &= pq - p - q + 1\end{aligned}$$

Subtrahiert man $\phi(N)$ von der Gleichung erhält man

$$\begin{aligned}
0 &= pq - p - q + 1 - \phi(N) && \mid \text{einsetzen von } q = \tfrac{N}{p}\\
&= p \cdot \frac{N}{p} - p - \frac{N}{p} + 1 - \phi(N)\\
&= N - p - \frac{N}{p} + 1 - \phi(N) && \mid \text{ multiplizieren mit } p\\
&= p \cdot N - p^2 - N + 1 \cdot p - \phi(N) \cdot p && \mid \text{ multiplizieren mit } -1\\
&= -p \cdot N + p^2 + N - 1 \cdot p + \phi(N) \cdot p\\
&= p^2 - p \cdot N + \phi(N) \cdot p - 1 \cdot p + N\\
&= p^2 - (N - \phi(N) + 1) \cdot p + N
\end{aligned}$$

Dies ist eine quadratische Gleichung (siehe 2.1) mit $a = 1$, $b = N - \phi(N) + 1$ und $c = N$. Die beiden Wurzeln und damit Primfaktoren von N dieser Gleichung sind

$$p_{1,2} = \frac{b \pm \sqrt{b^2 - 4 \cdot N}}{2}$$

Andererseits, wenn die beiden Primfaktoren p und q von N bekannt sind, dann folgt $\phi(N) = (p-1)(q-1)$ direkt aus

$$\phi(N) = N \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right)$$

wenn

$$N = \prod_{i=1}^{k} p_i$$

Das Theorem sagt aus, dass wenn eine feindliche Angreiferin Eve $\phi(N)$ berechnen könnte, könnte sie RSA brechen da d das multiplikative Inverse von e (mod $\phi(N)$) ist. Das ist $d \equiv \frac{1}{e} \pmod{\phi(N)}$. Andererseits führt das Wissen über $\phi(N)$ zum einfachen Faktorisieren von N, da gilt, dass

$$\begin{aligned} p + q &= n - \phi(N) + 1 \\ \\ (p-q)^2 &= (p+q)^2 - 4 \cdot N \end{aligned}$$

Ersteres ergibt sich, indem $\phi(N) = (p-1) \cdot (q-1)$ ausmultipliziert und umgeformt wird. Die zweite Formel lässt sich wie folgt herleiten:

$$\begin{aligned} & p^2 + q^2 &&= p^2 + q^2 \\ \Rightarrow \quad & p^2 + q^2 &&= p^2 + q^2 - 4 \cdot N + 2 \cdot N + 2 \cdot N \\ \Rightarrow \quad & p^2 - 2 \cdot N + q^2 &&= p^2 + 2 \cdot N + q^2 - 4 \cdot N \\ \Rightarrow \quad & (p-q)^2 &&= (p+q)^2 - 4 \cdot N \end{aligned}$$

Damit lassen sich p und q berechnen:

$$\begin{aligned} p &= \frac{(p+q) + (p-q)}{2} \\ q &= \frac{(p+q) - (p-q)}{2} \end{aligned}$$

In anderen Worten: Die Berechnung von $\phi(N)$ ist nicht einfacher als die Faktorisierung von N.

Beispiel 6.1.
Es sei

$$N = 143.$$

Angenommen die Angreiferin Eve weiß durch Raten, Abfangen oder eine andere Art und Weise, dass

$$\phi(N) = 120$$

gilt. Dann ist

$$\begin{aligned} b &= N - \phi(N) + 1 \\ &= 143 - 120 + 1 \\ &= 24 \end{aligned}$$

Daraus folgt, dass

$$p^2 - 24 \cdot p + 143 = 0.$$

Löst man die Gleichung, erhält man die beiden Wurzeln

$$\{p, q\} = \{11, 13\},$$

und damit die komplette Faktorisierung von N:

$$\begin{aligned} N &= 143 \\ &= 11 \cdot 13 \end{aligned}$$

Theorem 6.2.
Die RSA-Verschlüsselung ist in Polynomialzeit brechbar, wenn die Cryptoanalystin $\phi(N)$ kennt. Das bedeutet

$$\phi(N) \overset{\mathcal{P}}{\Longrightarrow} \text{RSA}(M)$$

Beweis 6.2.
Wenn $\phi(N)$ bekannt ist, dann ist $d \equiv \frac{1}{e} \pmod{\phi(N)}$, woraus folgt dass M aus $C : M \equiv C^d \pmod{N}$ hergeleitet werden kann. Wir haben also

$$\text{IFP}(N) \overset{\mathcal{P}}{\Longleftrightarrow} \phi(N) \overset{\mathcal{P}}{\Longrightarrow} \text{RSA}(M)$$

Es folgt, dass es nicht leichter ist die RSA-Verschlüsselung durch Berechnung von $\phi(N)$ zu brechen, als es ist sie durch Faktorisierung von N zu brechen. Kann jedoch jemand intelligent und effizient $\phi(N)$ raten, bzw. finden oder hat jemand $\phi(N)$ bereits auf irgendeine Weise herausgefunden, dann kann er oder sie RSA komplett ohne Faktorisierung brechen.

Literatur

[1] Dudenverlag: *DUDEN - "Basiswissen Schule"Mathe Abitur*. Mannheim, Berlin : Dudenverlag, Bibliographisches Institut F.A. Brockhaus und DUDEN PAETEC GmbH, 2007 (Basiswissen Schule). – ISBN 978–3–89818–081–8

[2] Labratories, RSA: *PKCS#1 v2.2: RSA Cryptography Standards*. https://www.emc.com/collateral/white-papers/h11300-pkcs-1v2-2-rsa-cryptography-standard-wp.pdf. Version: 2012

[3] Lang, H.W.: *Erweiterter euklidischer Algorithmus*. http://www.iti.fh-flensburg.de/lang/krypto/algo/euklid.htm. Version: 12.02.2001

[4] Menzer, Hartmut ; Althöfer, Ingo: *Zahlentheorie und Zahlenspiele*. München : Oldenbourg Wissenschaftsverlag GmbH, 2014. – ISBN 978–3–486–72030–3

[5] Rivest, R. L. ; Shamir, A. ; Adleman, L.: *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. 1978

[6] Rivest, Ronald L. ; Silverman, Robert D.: *Are 'Strong' Primes Needed for RSA*. https://people.csail.mit.edu/rivest/pubs/RS01.version-1999-11-22.pdf. Version: 1999

[7] Stellet, Jan: *Facharbeit - Lineare diophantische Gleichung*. http://jstc.de/blog/uploads/Lineare_diophantische_Gleichungen.pdf. Version: 2006

[8] Yan, Song Y. Y.: *Cryptanalytic Attacks on RSA*. New York : Springer Science+Business Media, LLC., 2008. – ISBN 978–0–387–48741–0