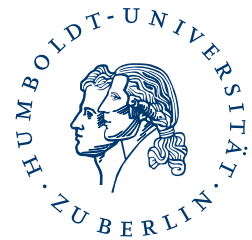
Einführung in die Kryptologie

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

SS 2022

Asymmetrische Kryptosysteme

- Diffie und Hellman hatten 1976 die Idee, dass ein Kryptosystem selbst dann sicher sein könnte, wenn der Chiffrierschlüssel k veröffentlich wird
- Natürlich darf dann der Dechiffrierschlüssel k' nicht mit vertretbarem Aufwand aus dem Chiffrierschlüssel k berechenbar sein
- Jeder Teilnehmer X kann dann ein Schlüsselpaar k_X, k'_X erzeugen und den Chiffrierschlüssel k_X veröffentlichen, während k'_X geheim bleibt
- Dies hat den großen Vorteil, dass für die Übertragung des Schlüssels k_X nur ein authentisierter (und kein sicherer) Kanal benötigt wird
- Es reicht nämlich aus, dass sich der Empfänger von der Herkunft und Originalität des Schlüssels k_X überzeugen kann
- Ein Kryptosystem heißt symmetrisch, wenn die Kenntnis des Chiffrierschlüssels gleichbedeutend mit der Kenntnis des Dechiffrierschlüssels ist, der eine also leicht aus dem anderen berechnet werden kann
- Bei einem asymmetrischen Kryptosystem darf dagegen der Chiffrierschlüssel veröffentlicht werden, da sich der Kryptotext damit nicht entschlüsseln lässt

Asymmetrische Kryptosysteme

- Symmetrische Kryptosysteme werden auch als konventionell oder als Secret-Key-Kryptosysteme bezeichnet, während man bei asymmetrischen Kryptosystemen auch von Public-Key-Kryptosystemen spricht
- Wie der Name schon sagt, sind bei einem symmetrischen Kryptosystem die Rollen von Sender und Empfänger austauschbar, da sie ein gemeinsames Geheimnis in Form des symmetrischen Schlüssels teilen
- Der Unterschied lässt sich durch folgende Analogie verdeutlichen, in der Geheiminformationen mithilfe eines Bankschließfachs übergeben werden:
  - Symmetrische Verschlüsselung: Alice und Bob sind im Besitz eines Schlüssels k für das Schließfach, welches sich mit k sowohl auf- als auch zuschließen lässt. Alice schließt die Nachricht in den Tresor ein und Bob öffnet danach das Schließfach, um die Nachricht zu lesen
  - Asymmetrische Verschlüsselung: Am Schließfach befindet sich ein Zahlenschloß, dessen Zahlenkombination k'_B nur Bob bekannt ist. Alice kennt nur die Schließfachnummer k_B, legt ihre Nachricht hinein und verdreht anschließend das Schloß. Bob kann das Schließfach mit seinem „privaten“ Schlüssel k'_B öffnen und die Nachricht entnehmen

Asymmetrische Kryptosysteme

- An dieser Analogie wird auch deutlich, warum der öffentliche Schlüssel k_B über einen authentisierten Kanal an Alice übergeben werden muss
- Andernfalls könnte sich nämlich ein Angreifer als Bob ausgeben und Alice seinen eigenen Schlüssel zusenden
- Anschließend könnte er die für Bob bestimmte Nachricht lesen (und ggf. mit k_B verschlüsselt an Bob weiterleiten) ohne dass dies bemerkt wird
- Da Alice nicht im Besitz von Bobs privatem Schlüssel k'_B ist, kann sie keine mit k_B verschlüsselten Nachrichten lesen; insbesondere auch keine, die Bob von anderen Teilnehmern erhält
- Dies hat den Vorteil, dass für jeden Teilnehmer nur ein asymmetrisches Schlüsselpaar generiert werden muss, während für die Kommunikation zwischen n Teilnehmern bis zu $\binom{n}{2}$ symmetrische Schlüssel nötig wären
- Zu beachten ist auch, dass mit Bobs Schlüsselpaar (k_B, k'_B) nur eine Nachrichtenübermittlung von Alice (oder anderen Teilnehmern) an Bob möglich ist, und für die Übermittlung von Nachrichten an Alice das Schlüsselpaar (k_A, k'_A) von Alice benutzt werden muss

Asymmetrische Kryptosysteme

- Dass bei der Verschlüsselung kein geheimer Schlüssel benutzt wird, hat andererseits den Nachteil, dass ein asymmetrisches Kryptosystem nicht absolut sicher sein kann (siehe Übungen)
- Da der Chiffrierschlüssel k_B öffentlich bekannt ist, kann ein Gegner bei bekanntem Kryptotext nämlich alle Klartexte ausprobieren
- Damit das System dennoch sicher ist, muss E_{k_B} eine Einwegfunktion (engl. one-way function) sein, d.h. E_{k_B} darf ohne Kenntnis des privaten Schlüssels k'_B nicht effizient umkehrbar sein
- Da dies bei Kenntnis von k'_B möglich ist, spricht man von einer Falltürfunktion (engl. trapdoor one-way function)
- Da E_{k_B} zudem bijektiv ist, handelt es sich genauer um eine Falltür-permutation (engl. trapdoor one-way permutation)
- In den Übungen wird gezeigt, dass mit deterministischen Public-Key-Verfahren keine komplexitätstheoretische Sicherheit erreichbar ist
- Hierzu muss der Verzicht auf die Geheimhaltung von k_B durch Verwendung von Zufall bei der Berechnung von E_{k_B} kompensiert werden

Das RSA-System

- Das RSA-Kryptosystem wurde 1978 von Rivest, Shamir und Adleman veröffentlicht
- Während es beim Primzahlproblem nur um die Frage „Ist n prim?“ geht, muss beim Faktorisierungsproblem im Falle einer zusammengesetzten Zahl mindestens ein nicht-trivialer Faktor berechnet werden
- Genauer gesagt beruht das RSA-Verfahren darauf, dass die Primzahleigenschaft zwar effizient getestet werden kann, aber keine effizienten Faktorisierungsalgorithmen bekannt sind

Schlüsselgenerierung

Für jeden Teilnehmer X werden zwei Primzahlen p, q und zwei Exponenten e, d mit $ed \equiv_{\varphi(n)} 1$ generiert, wobei $n = pq$ und $\varphi(n) = (p-1)(q-1)$ ist

- öffentlicher Schlüssel: $k_X = (e, n)$
- privater Schlüssel: $k'_X = (d, n)$

Das RSA-System

Ver- und Entschlüsselung

- Jede Nachricht x besteht aus einer Folge $x_1, x_2, \ldots$ von Zahlen $x_i \in \mathbb{Z}_n$, die einzeln wie folgt ver- und entschlüsselt werden:
  - $\mathrm{RSA}((e,n),x) = x^e \bmod n$
  - $\mathrm{RSA}^{-1}((d,n),y) = y^d \bmod n$
- Der Schlüsselraum ist also

  $$K = \{(c,n) \mid \text{es gibt Primzahlen } p \text{ und } q \text{ mit } n = pq \text{ und } c \in \mathbb{Z}^*_{\varphi(n)}\}$$

  und

  $$S = \{((e,n),(d,n)) \in K \times K \mid ed \equiv_{\varphi(n)} 1\}$$

  ist die Menge aller zueinander passenden Schlüsselpaare
- Die Chiffrierfunktionen $\mathrm{RSA}_{(e,n)}$ und $\mathrm{RSA}^{-1}_{(d,n)}$ sind durch Wiederholtes Quadrieren und Multiplizieren effizient berechenbar

Das RSA-System

Der folgende Satz garantiert die Korrektheit des RSA-Systems

Satz

Für jedes Schlüsselpaar $((e,n),(d,n)) \in S$ und alle $x \in \mathbb{Z}_n$ gilt

$$x^{ed} \equiv_n x$$

Beweis.

- Sei $n = pq$ und sei z eine natürliche Zahl mit $ed = z\varphi(n) + 1$
- Wir zeigen $x^{ed} \equiv_p x$ (die Kongruenz $x^{ed} \equiv_q x$ folgt analog und beide Kongruenzen zusammen implizieren $x^{ed} \equiv_n x$)
- Wegen $\varphi(n) = (p-1)(q-1)$ und wegen $x^{p-1} \equiv_p 1$ für $x \not\equiv_p 0$ folgt

$$x^{ed} = x^{z\varphi(n)+1} = x^{z(p-1)(q-1)}x = \left(x^{p-1}\right)^{z(q-1)}x \equiv_p x$$

□

Praktische Durchführung

- Bestimmung von p und q:
  - Man wählt zufällig eine Zahl x der Form $30z$ und der gewünschten Größe (z. B. $x \in I = (10^{1000}, 10^{1001})$) und führt einen Primzahltest für die Zahlen $x+1$, $x+7$, $x+11$, $x+13$, $x+17$, $x+19$, $x+23$, $x+29$, $x+30+1$, $x+30+7$, ... durch, bis eine Primzahl p gefunden ist
  - Wegen $\pi(I)/|I| \approx 1/(\ln p)$ und da nur 8 von 30 Zahlen getestet werden, sind hierzu ungefähr $(8/30) \ln p$ Primzahltests durchzuführen (bei 500-stelligen Dezimalzahlen sind das ca. 300 Tests)
- Bestimmung von d:
  - d soll teilerfremd zu $\varphi(n) = (p-1)(q-1)$ sein
  - Dies trifft z. B. auf jede Primzahl $d > \max\{p, q\}$ zu
- Bestimmung von e:
  - Da $\mathrm{ggT}(d, \varphi(n)) = 1$ ist, liefert der erweiterte euklidische Algorithmus das multiplikative Inverse $e = d^{-1} \bmod \varphi(n)$

Praktische Durchführung

- Komplexität der Ver- und Entschlüsselung:
  - Im Vergleich zu symmetrischen Verfahren wie z.B. 3DES oder AES ist RSA mindestens um den Faktor 100 langsamer
  - Daher wird RSA meist nur dazu benutzt, um einen symmetrischen Schlüssel (auch Sitzungsschlüssel genannt) auszutauschen
  - Damit lassen sich dann auch große Datenmengen chiffrieren und dechiffrieren (hybride Verschlüsselung)

Kryptoanalytische Betrachtungen

- Es ist klar, dass das RSA-Verfahren gebrochen ist, falls dem Gegner die Faktorisierung des Moduls n gelingt
- In diesem Fall kann er $\varphi(n)$ und damit auch den privaten Dechiffrierexponenten aus dem öffentlichen Exponenten e berechnen
- Umgekehrt lässt sich n bei Kenntnis von $\varphi(n)$ wie folgt faktorisieren:
  - Sei $n = pq$ (mit $p, q \in \mathcal{P}$; o.B.d.A. sei $p > q$)
  - Wegen

$$\varphi(n) = (p-1)(q-1) = (p-1)(n/p - 1) = -p + n + 1 - n/p$$

    erhalten wir die Gleichung $p - \underbrace{(n + 1 - \varphi(n))}_{=:c} + n/p = 0$

  - Diese führt auf die quadratische Gleichung $p^2 - cp + n = 0$ mit den beiden Lösungen

$$p, q = \frac{c \pm \sqrt{c^2 - 4n}}{2}$$

Kryptoanalytische Betrachtungen

- Natürlich sollte q hinreichend groß sein, da n sonst mittels $\pi(q) \leq q$ Probedivisionen faktorisiert werden kann (Sieb des Eratosthenes)
- Zudem sollten die Primfaktoren p und q nicht zu nahe beieinander liegen, da q sonst ausgehend von $\lfloor\sqrt{n}\rfloor$ gefunden werden kann:
  - Sei $a = \frac{p+q}{2}$ das arithmetische und $\sqrt{n}$ das geometrische Mittel von p und q (o.B.d.A. sei $p > q$)
  - Wegen
    $$4a^2 = (p+q)^2 = p^2 + 2n + q^2 = 4n + \underbrace{p^2 - 2n + q^2}_{(p-q)^2>0} > 4n$$
    ist $a^2 > n$ und daher folgt $q < \sqrt{n} < a < p$
  - Der Primteiler q kann also ausgehend von $\lfloor\sqrt{n}\rfloor$ nach höchstens $\sqrt{n} - q < a - q = b := \frac{(p-q)}{2}$ Schritten gefunden werden
- Im Fall $p > 2q$ ist der Aufwand hierfür jedoch proportional zu
  $$\sqrt{n} - q = \sqrt{pq} - q > \sqrt{2}q - q = (\sqrt{2} - 1)q > q/3$$

Kryptoanalytische Betrachtungen

- Mit dem Verfahren der Differenz der Quadrate (auch Faktorisierungsmethode von Fermat genannt) lassen sich a und b (und damit p und q) sogar in $a - \lfloor\sqrt{n}\rfloor$ Schritten finden
- Wegen $n = pq = (a+b)(a-b) = a^2 - b^2$ genügt es nämlich, eine Zahl $a > \sqrt{n}$ zu finden, so dass $a^2 - n = b^2$ eine Quadratzahl ist
- Für $n = 124\,711 \approx 353{,}1^2$ reichen bespielsweise 3 Schritte:
  - Bereits für $a = \lfloor\sqrt{n}\rfloor + 3 = 356$ ist

    $$a^2 - n = 126\,736 - 124\,711 = 2025 = 45^2$$

    eine Quadratzahl, woraus wir die beiden Faktoren $p = a + 45 = 401$ und $q = a - 45 = 311$ erhalten
  - Eine Suche nach q ausgehend von $\lfloor\sqrt{n}\rfloor = 353$ würde dagegen $354 - 311 = 43$ Schritte benötigen

Kryptoanalytische Betrachtungen

- Der Aufwand für die Suche nach a ausgehend von $\lfloor\sqrt{n}\rfloor$ ist proportional zur Differenz $a - \sqrt{n}$
- Diese lässt sich wegen $\sqrt{x-y} \leq \sqrt{x} - \frac{y}{2\sqrt{x}}$ wie folgt abschätzen:

$$a - \sqrt{n} \;=\; a - \sqrt{a^2 - b^2} \;\geq\; b^2/2a$$

- Ist $p \geq 2q$, so folgt

$$b \;=\; (p-q)/2 \;=\; (p+q)/6 + \underbrace{(p-2q)/3}_{\geq 0} \;\geq\; (p+q)/6 \;=\; a/3,$$

also $3b/a \geq 1$, und somit

$$a - \sqrt{n} \;\geq\; b^2/2a \;=\; 3b/a \cdot b/6 \;\geq\; b/6 \;\geq\; q/12$$

- Daher ist dieser Angriff im Fall $p \geq 2q$ auch nicht deutlich effizienter

Kryptoanalytische Betrachtungen

- Für die Teilnehmer sollten verschiedene Module $n = pq$ gewählt werden
- Wir werden später sehen, dass sich n bei Kenntnis eines Schlüsselpaares $(e, n), (d, n)$ mit $ed \equiv_{\varphi(n)} 1$ effizient faktorisieren lässt
- Aus Effizienzgründen wird der Verschlüsselungsexponent e meist klein gewählt
- Kleinere Werte als z.B. die vierte Fermat-Zahl $2^{16} + 1 = 65537$ sollte man jedoch nicht verwenden, da dies zu Angriffsmöglichkeiten führt
- Wird etwa dieselbe Nachricht an mehrere Empfänger gesendet, kann eine Dechiffrierung mithilfe des Chinesischen Restsatzes möglich sein (Angriff von Hastad, siehe Übungen)
- Auch die Wahl des Entschlüsselungsexponenten d sollte nicht zu klein ausfallen
- Beträgt die Bitlänge von d weniger als ein Viertel der Bitlänge von n, kann d unter Umständen mit einem auf Kettenbrüchen basierenden Verfahren effizient berechnet werden (Angriff von Wiener).

Sicherheit des privaten RSA-Schlüssels

- Wie wir gesehen haben, ist das RSA-System gebrochen, falls die Primfaktoren des Moduls n bekannt sind
- RSA ist daher höchstens so schwer zu brechen wie n zu faktorisieren
- Dagegen ist nicht bekannt, ob auch umgekehrt aus einem effizienten Algorithmus, der bei Eingabe von (e, n) und y einen Klartext x mit $x^e \equiv_n y$ berechnet, ein effizienter Faktorisierungsalgorithmus für n gewonnen werden kann
- Es ist also nach heutigem Kenntnisstand nicht ausgeschlossen, dass RSA leichter zu brechen ist als n zu faktorisieren
- Wie wir nun zeigen werden, erfordert die Berechnung von d aus (e, n) jedoch den gleichen Aufwand wie das Faktorisieren von n
- Wegen $ed \equiv_{\varphi(n)} 1$ ist $v = ed - 1$ nämlich ein Vielfaches von $\varphi(n)$ und damit von $k = \mathrm{kgV}(p-1, q-1)$
- Die effiziente Faktorisierung von n bei Kenntnis eines Vielfachen v von k beruht auf folgendem Lemma

Sicherheit des privaten RSA-Schlüssels

Lemma

- Sei $m \geq 1$ und seien y, z zwei Lösungen der Kongruenz $x^2 \equiv_m a$ mit $y \not\equiv_m \pm z$
- Dann sind $\mathrm{ggT}(y+z, m)$ und $\mathrm{ggT}(y-z, m)$ nicht-triviale Faktoren von m

Beweis.

- Wegen $y^2 \equiv_m z^2$ existiert ein $t \in \mathbb{Z}$ mit

$$(y+z)(y-z) = y^2 - z^2 = tm$$

- Da m also das Produkt $(y+z)(y-z)$ teilt, aber wegen $y \not\equiv_m \pm z$ keiner der beiden Faktoren $y+z$ und $y-z$ durch m teilbar ist, müssen sich die Faktoren von m auf $y+z$ und $y-z$ verteilen
- Daraus folgt $1 < \mathrm{ggT}(y+z, m), \mathrm{ggT}(y-z, m) < m$ □

Sicherheit des privaten RSA-Schlüssels

- Um nun n bei Kenntnis eines Vielfachen v von $k = \text{kgV}(p-1, q-1)$ zu faktorisieren, überlegen wir, wie sich der Miller-Rabin-Primzahltest in einen entsprechenden Faktorisierungsalgorithmus umwandeln lässt

Algorithmus $\mathcal{MRT}(n)$

```
1  sei n − 1 = Σ_{i=0}^{r} e_i · 2^i mit e_r = 1
2  guess randomly a ∈ [n − 1]
3  z := a
4  for i := r − 1 downto 0 do
5     y := z
6     z := z^2 mod n
7     if z ≡_n 1 ∧ y ≢_n ±1 then
8        return(„zusammengesetzt“)
9     if e_i = 1 then z := z · a mod n
10 if z ≢_n 1 then
11    return(„zusammengesetzt“)
12 else return(„prim“)
```

- Falls MRT die Eingabe n in Zeile 8 als zusammengesetzt erkennt, wird auch ein nicht-trivialer Teiler $\text{ggT}(y+1, n)$ von n gefunden
- In Zeile 11 gelingt dies dagegen nur im Fall $\text{ggT}(z, n) > 1$
- Wegen $z \equiv_n a^{n-1}$ ist dies zu $\text{ggT}(a, n) > 1$ äquivalent und daher sehr unwahrscheinlich
- Ersetzen wir aber $n-1$ durch v, so gilt $z \equiv_n a^v \equiv_n 1$ für alle $a \in \mathbb{Z}_n^*$ (d.h. $z \not\equiv_n 1 \Leftrightarrow \text{ggT}(z, n) > 1$)

Sicherheit des privaten RSA-Schlüssels

- Wir werden beweisen, dass nach Ersetzung von $n-1$ durch v ein Primteiler von n in Zeile 8 mit hoher Wahrscheinlichkeit gefunden wird
- Damit erhalten wir folgenden Las-Vegas Algorithmus $\mathcal{RSA}$-$\mathcal{F}actorize$

Algorithmus $\mathcal{MRT}(n)$

```
1  sei n − 1 = Σ_{i=0}^{r} e_i · 2^i mit e_r = 1
2  guess randomly a ∈ [n − 1]
3  z := a
4  for i := r − 1 downto 0 do
5     y := z
6     z := z^2 mod n
7     if z ≡_n 1 ∧ y ≢_n ±1 then
8        return(„zusammengesetzt“)
9     if e_i = 1 then z := z · a mod n
10 if z ≢_n 1 then
11    return(„zusammengesetzt“)
12 else return(„prim“)
```

$\mathcal{RSA}$-$\mathcal{F}actorize(n, v)$

```
1  sei v = Σ_{i=0}^{r} e_i · 2^i mit e_r = 1
2  guess randomly a ∈ [n − 1]
3  z := a
4  for i := r − 1 downto 0 do
5     y := z
6     z := z^2 mod n
7     if z ≡_n 1 ∧ y ≢_n ±1 then
8        return(ggT(y + 1, n))
9     if e_i = 1 then z := z · a mod n
10 if ggT(z, n) > 1 then
11    return(ggT(z, n))
12 else return(„?“)
```

Sicherheit des privaten RSA-Schlüssels

Beispiel

- Für $n = 221 = 13 \cdot 17$ ist $\varphi(221) = 12 \cdot 16 = 192$ und $\mathrm{kgV}(12, 16) = 48$
- Falls der Gegner zu $(e, n) = (25, 221)$ den privaten Schlüssel $(d, n) = (169, 221)$ bestimmen kann, erhält er $v = ed - 1 = 4224$
- Bei Eingabe von $n = 221$ und $v = 4224$ berechnet $\mathcal{RSA}$-$\mathcal{F}actorize$ bei Wahl der Basen $a = 174$, $a' = 111$ und $a'' = 137$ die auf der nächsten Folie angegebenen Werte $z_i = z_i(a)$, $z'_i = z_i(a')$ und $z''_i = z_i(a'')$

Sicherheit des privaten RSA-Schlüssels

Beispiel (Fortsetzung)

i	e_i	c_i	$z_i = 174^{c_i}$	$(z_i)^2$	$z_i' = 111^{c_i}$	$(z_i')^2$	$z_i'' = 137^{c_i}$	$(z_i'')^2$
12	1	1	174	220	111	166	137	205
11	0	2	220	1	166	152	205	35
10	0	4	1	1	152	120	35	120
9	0	8	1	1	120	35	120	35
8	0	16	1	1	35	120	35	120
7	1	33	$1 \cdot 174 = 174$	220	$120 \cdot 111 = 60$	64	$120 \cdot 137 = 86$	103
6	0	66	220	1	64	118	**103**	1
5	0	132	1	1	**118**	1		
4	0	264	1	1				
3	0	528	1	1				
2	0	1056	1	1				
1	0	2112	1	1				
0	0	4224	1					

- $\mathcal{RSA}$-*Factorize* gelingt also die Faktorisierung von $n = 221$ bei Wahl von $a = 174$ nicht, wohl aber bei Wahl von $a' = 111$ und $a'' = 137$
- Im ersten Fall findet $\mathcal{RSA}$-*Factorize* den Faktor $\mathrm{ggT}(118 + 1, 221) = 17$ und im zweiten den Faktor $\mathrm{ggT}(103 + 1, 221) = 13$ ◁

Sicherheit des privaten RSA-Schlüssels

Satz

- Sei $n = pq$ ($p, q \geq 3$ prim) und $v > 0$ ein Vielfaches von $k = \text{kgV}(p-1, q-1)$
- Dann gibt $\mathcal{RSA}\text{-}\mathit{Factorize}(n, v)$ mit Wahrscheinlichkeit größer $1/2$ einen Primfaktor von n aus

Beweis.

- Es ist klar, dass jede Ausgabe von $\mathcal{RSA}\text{-}\mathit{Factorize}$ in Zeile 11 ein nichttrivialer Faktor von n sein muss
- Mit obigem Lemma folgt

$$y \not\equiv_n \pm 1, y^2 \equiv_n 1 \quad \Rightarrow \quad \text{ggT}(y+1, n) \in \{p, q\},$$

  womit auch die Korrektheit jeder Ausgabe in Zeile 8 gezeigt ist
- Wir schätzen nun die Wahrscheinlichkeit ab, dass die Faktorisierung von n nicht gelingt und $\mathcal{RSA}\text{-}\mathit{Factorize}$ ein Fragezeichen ausgibt

Sicherheit des privaten RSA-Schlüssels

Beweis (Fortsetzung)

- Sei $v = 2^m u$, $p - 1 = 2^i u_1$ und $q - 1 = 2^j u_2$ mit u, u_1, u_2 ungerade und sei o. B. d. A. $i \leq j$
- Zudem sei $F(n)$ die Menge aller Basen $a \in \mathbb{Z}_n^*$, bei deren Wahl $\mathcal{RSA}$-$\mathcal{F}actorize$ ein Fragezeichen ausgibt und sei $S(n)$ die Menge

$$S(n) = \{a \in \mathbb{Z}_n^* \mid a^u \equiv_n 1 \ \vee \ \exists t \geq 0 : a^{2^t u} \equiv_n -1\}$$

- Dann liefert jede Basis $a \in \mathbb{Z}_n^* \setminus S(n)$ wegen $a^{2^m u} \equiv_n a^v \equiv_n 1$, aber $a^u \not\equiv_n 1$ und $a^{2^t u} \not\equiv_n -1$ für $t \geq 0$ einen Primfaktor von n in Zeile 8
- Da jede Basis $a \in \mathbb{Z}_n \setminus \mathbb{Z}_n^*$ einen Primfaktor in Zeile 11 liefert, folgt

$$\Pr[\mathcal{RSA}\text{-}\mathcal{F}actorize(n, v) = ?] = |F(n)|/(n-1) \leq |S(n)|/(n-1)$$

- Um $|S(n)|$ zu berechnen, betrachten wir für $t \geq 0$ die Funktionen $\alpha(n) = |\{a \in \mathbb{Z}_n^* \mid a^u \equiv_n 1\}|$ und $\alpha_t(n) = |\{a \in \mathbb{Z}_n^* \mid a^{2^t u} \equiv_n -1\}|$
- Dann gilt $|S(n)| = \alpha(n) + \sum_{t \geq 0} \alpha_t(n)$ und die folgenden Behauptungen zeigen $|S(n)|/(n-1) < 1/2$

Sicherheit des privaten RSA-Schlüssels

Behauptung. Es gilt

1. $\mathrm{ggT}(2^t u, p-1) = 2^{\min(t,i)} u_1$ und $\mathrm{ggT}(2^t u, q-1) = 2^{\min(t,j)} u_2$
2. $\alpha(n) = u_1 u_2$
3. $\alpha_t(n) = 2^{2t} u_1 u_2$ für $t = 0, \ldots, i-1$ und $\alpha_t(n) = 0$ für $t \geq i$
4. $|S(n)| \leq \varphi(n)/2 < (n-1)/2$

Beweis von Behauptung 1

- Wegen
$$k = \mathrm{kgV}(p-1, q-1) = \mathrm{kgV}(2^i u_1, 2^j u_2) = 2^{\max(i,j)} \mathrm{kgV}(u_1, u_2)$$
und $k \mid v = 2^m u$ folgt $u_1 | u$ und $u_2 | u$
- Da u ungerade ist, folgt somit
$$\mathrm{ggT}(2^t u, p-1) = \mathrm{ggT}(2^t u, 2^i u_1) = 2^{\min(t,i)} u_1$$
und
$$\mathrm{ggT}(2^t u, q-1) = \mathrm{ggT}(2^t u, 2^j u_2) = 2^{\min(t,j)} u_2$$

□

Sicherheit des privaten RSA-Schlüssels

Behauptung. Es gilt

1. $\mathrm{ggT}(2^t u, p-1) = 2^{\min(t,i)} u_1$ und $\mathrm{ggT}(2^t u, q-1) = 2^{\min(t,j)} u_2$
2. $\alpha(n) = u_1 u_2$
3. $\alpha_t(n) = 2^{2t} u_1 u_2$ für $t = 0, \ldots, i-1$ und $\alpha_t(n) = 0$ für $t \geq i$
4. $|S(n)| \leq \varphi(n)/2 < (n-1)/2$

Beweis von Behauptung 2

- Mit dem Chinesischen Restsatz folgt

$$\alpha(n) = \underbrace{|\{a \in \mathbb{Z}_p^* \mid a^u \equiv_p 1\}|}_{\alpha(p)} \cdot \underbrace{|\{a \in \mathbb{Z}_q^* \mid a^u \equiv_q 1\}|}_{\alpha(q)}$$

- Schreiben wir nun a als g^k für einen Erzeuger g von $\mathbb{Z}_p^*$, so folgt wegen $g^{ku} \equiv_p 1 \Leftrightarrow ku \equiv_{p-1} 0$ mit Beh. 1, dass $\alpha(p) = \mathrm{ggT}(u,\ p-1) = u_1$ ist
- Analog folgt $\alpha(q) = u_2$ □

Sicherheit des privaten RSA-Schlüssels

Behauptung. Es gilt

1. $\mathrm{ggT}(2^t u, p-1) = 2^{\min(t,i)} u_1$ und $\mathrm{ggT}(2^t u, q-1) = 2^{\min(t,j)} u_2$
2. $\alpha(n) = u_1 u_2$
3. $\alpha_t(n) = 2^{2t} u_1 u_2$ für $t = 0, \ldots, i-1$ und $\alpha_t(n) = 0$ für $t \geq i$
4. $|S(n)| \leq \varphi(n)/2 < (n-1)/2$

Beweis von Behauptung 3

- Mit dem Chinesischen Restsatz folgt zunächst

$$\alpha_t(n) = \underbrace{|\{a \in \mathbb{Z}_p^* \mid a^{2^t u} \equiv_p -1\}|}_{\alpha_t(p)} \cdot \underbrace{|\{a \in \mathbb{Z}_q^* \mid a^{2^t u} \equiv_q -1\}|}_{\alpha_t(q)}$$

Sicherheit des privaten RSA-Schlüssels

Beweis von Behauptung 3

- Mit dem Chinesischen Restsatz folgt zunächst

$$\alpha_t(n) = \underbrace{|\{a \in \mathbb{Z}_p^* \mid a^{2^t u} \equiv_p -1\}|}_{\alpha_t(p)} \cdot \underbrace{|\{a \in \mathbb{Z}_q^* \mid a^{2^t u} \equiv_q -1\}|}_{\alpha_t(q)}$$

- Schreiben wir a wieder als g^k für einen Erzeuger g, so folgt wegen
  - $g^{k2^t u} \equiv_p -1 \Leftrightarrow k2^t u \equiv_{p-1} (p-1)/2$ und
  - weil $\mathrm{ggT}(2^t u, p-1) \overset{\text{Beh. 1}}{=} 2^t u_1$ die Zahl $(p-1)/2 = 2^{i-1}u_1$ genau im Fall $0 \leq t \leq i-1$ teilt,

  dass $\alpha_t(p) = 2^t u_1$ für $t = 0, \ldots, i-1$ und $\alpha_t(p) = 0$ für alle $t \geq i$ ist
- Analog folgt $\alpha_t(q) = 2^t u_2$ für $t = 0, \ldots, j-1$ und $\alpha_t(q) = 0$ für alle $t \geq j$

□

Sicherheit des privaten RSA-Schlüssels

Behauptung. Es gilt

1. $\mathrm{ggT}(2^t u, p-1) = 2^{\min(t,i)} u_1$ und $\mathrm{ggT}(2^t u, q-1) = 2^{\min(t,j)} u_2$
2. $\alpha(n) = u_1 u_2$
3. $\alpha_t(n) = 2^{2t} u_1 u_2$ für $t = 0, \ldots, i-1$ und $\alpha_t(n) = 0$ für $t \geq i$
4. $|S(n)| \leq \varphi(n)/2 < (n-1)/2$

Beweis von Behauptung 4

- Wegen $|S(n)| = \alpha(n) + \sum_{t \geq 0} \alpha_t(n)$ folgt mit obigen Behauptungen

$$\begin{aligned} |S(n)| &= u_1 u_2 + \textstyle\sum_{t=0}^{i-1} 2^{2t} u_1 u_2 = u_1 u_2 (1 + \sum_{t=0}^{i-1} 2^{2t}) \\ &= u_1 u_2 (1 + (2^{2i} - 1)/3) = u_1 u_2 (2^{2i} + 2)/3 \\ &\leq u_1 u_2 (2^{i+j} + 2^{i+j-1})/3 = \varphi(n)(1 + 2^{-1})/3 = \varphi(n)/2 \\ &= (p-1)(q-1)/2 = (n - p - q + 1)/2 < (n-1)/2 \end{aligned}$$

□