

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 veröffentlicht 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 **Falltürpermutation** (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: $k'_X = (d, n)$

Ver- und Entschlüsselung

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

- 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 $\text{RSA}_{(e,n)}$ und $\text{RSA}_{(d,n)}^{-1}$ sind durch **Wiederholtes Quadrieren und Multiplizieren** effizient berechenbar

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

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



- 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 $x + 1, x + 7, x + 11, x + 13, x + 17, x + 19, x + 23, x + 29, x + 30 + 1, x + 30 + 7, \dots$ 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 $\text{ggT}(d, \varphi(n)) = 1$ ist, liefert der erweiterte euklidische Algorithmus das multiplikative Inverse $e = d^{-1} \bmod \varphi(n)$

- 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 beispielsweise 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

- 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**).

- 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 $k = \text{kgV}(p - 1, q - 1)$
- Die effiziente Faktorisierung von n bei Kenntnis eines Vielfachen v von k beruht auf folgendem Lemma

Lemma

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

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 Faktorisierungsalgorithmus für RSA-Module n umwandeln lässt

Sicherheit des privaten RSA-Schlüssels

Algorithmus $MRT(n)$

```

1 sei  $n - 1 = \sum_{i=0}^r e_i \cdot 2^i$  mit  $e_r = 1$ 
2 guess randomly  $a \in [n - 1]$ 
3  $z := a$ 
4 for  $i := r - 1$  downto 0 do
5    $y := z$ 
6    $z := z^2 \bmod n$ 
7   if  $z \equiv_n 1 \wedge y \not\equiv_n \pm 1$  then
8     return(„zusammengesetzt“)
9   if  $e_i = 1$  then  $z := z \cdot a \bmod n$ 
10 if  $z \not\equiv_n 1$  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 $\text{ggT}(z, n) > 1$
- Für Carmichaelzahlen gilt $z \equiv_n a^{n-1} \equiv_n 1$ für alle $a \in \mathbb{Z}_n^*$
- Daher gilt für sie die Äquivalenz $z \not\equiv_n 1 \Leftrightarrow \text{ggT}(z, n) > 1$ und sie können auch in Zeile 11 faktorisiert werden
- Ersetzen wir nun $n - 1$ durch v , so gilt für alle RSA-Module n die Äquivalenz $z \not\equiv_n 1 \Leftrightarrow \text{ggT}(z, n) > 1$, da $z \equiv_n a^v \equiv_n 1$ für alle $a \in \mathbb{Z}_n^*$ ist
- Daher können RSA-Module nach dieser Modifikation sowohl in Zeile 8 als auch in Zeile 11 faktorisiert werden

- Damit erhalten wir folgenden Las-Vegas Algorithmus *RSA-Factorize*

Algorithmus $\mathcal{MRT}(n)$

```

1 sei  $n - 1 = \sum_{i=0}^r e_i \cdot 2^i$  mit  $e_r = 1$ 
2 guess randomly  $a \in [n - 1]$ 
3  $z := a$ 
4 for  $i := r - 1$  downto 0 do
5    $y := z$ 
6    $z := z^2 \bmod n$ 
7   if  $z \equiv_n 1 \wedge y \not\equiv_n \pm 1$  then
8     return(„zusammengesetzt“)
9   if  $e_i = 1$  then  $z := z \cdot a \bmod n$ 
10 if  $z \not\equiv_n 1$  then
11   return(„zusammengesetzt“)
12 else return(„prim“)

```

RSA-Factorize(n, v)

```

1 sei  $v = \sum_{i=0}^r e_i \cdot 2^i$  mit  $e_r = 1$ 
2 guess randomly  $a \in [n - 1]$ 
3  $z := a$ 
4 for  $i := r - 1$  downto 0 do
5    $y := z$ 
6    $z := z^2 \bmod n$ 
7   if  $z \equiv_n 1 \wedge y \not\equiv_n \pm 1$  then
8     return( $\text{ggT}(y + 1, n)$ )
9   if  $e_i = 1$  then  $z := z \cdot a \bmod n$ 
10 if  $\text{ggT}(z, n) > 1$  then
11   return( $\text{ggT}(z, n)$ )
12 else return(„?“)

```

- Bevor wir beweisen, dass *RSA-Factorize*(n, v) mit hoher Wahrscheinlichkeit einen Primteiler von $n = pq$ findet, betrachten wir ein Beispiel

Beispiel

- Für $n = 221 = 13 \cdot 17$ ist $\varphi(221) = 12 \cdot 16 = 192$ und $\text{kgV}(12, 16) = 48$
- 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 *RSA-Factorize* 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 $z''_i = z_i(a'')$

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					

- *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 *RSA-Factorize* den Faktor $\text{ggT}(118 + 1, 221) = 17$ und im zweiten den Faktor $\text{ggT}(103 + 1, 221) = 13$

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 $\mathcal{RSA}\text{-Factorize}(n, v)$ mit Wahrscheinlichkeit größer $1/2$ einen Primfaktor von n aus

Beweis.

- Es ist klar, dass jede Ausgabe von $\mathcal{RSA}\text{-Factorize}$ 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 $\mathcal{RSA}\text{-Factorize}$ ein Fragezeichen ausgibt

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 *RSA-Factorize* 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

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

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

Beweis von Behauptung 1

- Wegen

$$k = \text{kgV}(p - 1, q - 1) = \text{kgV}(2^i u_1, 2^j u_2) = 2^{\max(i, j)} \text{kgV}(u_1, u_2)$$

und $k \mid v = 2^m u$ folgt $u_1 \mid u$ und $u_2 \mid u$

- Da u ungerade ist, folgt somit

$$\text{ggT}(2^t u, p - 1) = \text{ggT}(2^t u, 2^i u_1) = 2^{\min(t, i)} u_1$$

und

$$\text{ggT}(2^t u, q - 1) = \text{ggT}(2^t u, 2^j u_2) = 2^{\min(t, j)} u_2$$



Behauptung. Es gilt

- 1 $\text{ggT}(2^t u, p - 1) = 2^{\min(t, i)} u_1$ und $\text{ggT}(2^t u, q - 1) = 2^{\min(t, j)} u_2$
- 2 $\alpha(n) = u_1 u_2$
- 3 $\alpha_t(n) = 2^{2t} u_1 u_2$ für $t = 0, \dots, i - 1$ 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 $\alpha(p) = \text{ggT}(u, p - 1) = u_1$ ist
- Analog folgt $\alpha(q) = u_2$ □

Behauptung. Es gilt

- 1 $\text{ggT}(2^t u, p - 1) = 2^{\min(t, i)} u_1$ und $\text{ggT}(2^t u, q - 1) = 2^{\min(t, j)} u_2$
- 2 $\alpha(n) = u_1 u_2$
- 3 $\alpha_t(n) = 2^{2t} u_1 u_2$ für $t = 0, \dots, i - 1$ 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

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

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 $\text{ggT}(2^t u, p-1) \stackrel{\text{Beh. 1}}{=} 2^t u_1$ 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 $t = 0, \dots, i-1$ und $\alpha_t(p) = 0$ für alle $t \geq i$ ist
- Analog folgt $\alpha_t(q) = 2^t u_2$ für $t = 0, \dots, j-1$ und $\alpha_t(q) = 0$ für alle $t \geq j$



Behauptung. Es gilt

- 1 $\text{ggT}(2^t u, p - 1) = 2^{\min(t, i)} u_1$ und $\text{ggT}(2^t u, q - 1) = 2^{\min(t, j)} u_2$
- 2 $\alpha(n) = u_1 u_2$
- 3 $\alpha_t(n) = 2^{2t} u_1 u_2$ für $t = 0, \dots, i - 1$ 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

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

- Als nächstes gehen wir der Frage nach, wie sicher einzelne Bits einer (binär kodierten) mit RSA verschlüsselten Klartextnachricht sind
- Wir zeigen, dass es einem Angreifer nicht möglich ist, das Paritätsbit des Klartextes zu ermitteln, außer wenn es ihm gelingt, RSA vollständig zu brechen
- Die Paritätsbits der Klartexte sind also genau so sicher wie der gesamte Klartext
- Hierzu transformieren wir jeden effizienten Algorithmus, der aus dem Kryptotext y und dem öffentlichen Schlüssel (e, n) die Parität des Klartextes x berechnet, in einen effizienten Algorithmus, der den gesamten Klartext x effizient berechnet
- Wir werden später sehen, dass sich andere partielle Informationen über den Klartext sehr wohl aus dem zugehörigen Kryptotext gewinnen lassen

Sicherheit partieller Klartextinformationen

- Für $x, y \in \mathbb{Z}_n$ mit $y \equiv_n x^e$ sei

$$\text{clear-parity}(y) = \text{parity}(x) = \begin{cases} 1 & \text{falls } x \text{ ungerade} \\ 0 & \text{falls } x \text{ gerade} \end{cases}$$

und

$$\text{clear-half}(y) = \text{half}(x) = \begin{cases} 0 & \text{falls } 0 \leq x < n/2 \\ 1 & \text{falls } n/2 \leq x < n \end{cases}$$

- Wegen

$$2x \bmod n = \begin{cases} 2x & \text{half}(x) = 0 \\ 2x - n & \text{sonst} \end{cases}$$

folgt $\text{half}(x) = \text{parity}(2x \bmod n)$

- Daher lässt sich $\text{clear-half}(y)$ auf $\text{clear-parity}(y)$ reduzieren:

$$\text{clear-half}(y) = \text{half}(x) = \text{parity}(2x \bmod n) = \text{clear-parity}(2^e y \bmod n)$$

- Für die Bits $b_1 b_2 \dots$ der Binärdarstellung von $x/n = \sum_{i=1}^{\infty} b_i 2^{-i}$ gilt

$$\begin{aligned} 2^{i-1}x &= n(2^{i-2}b_1 + \dots + b_{i-1} + b_i/2 + b_{i+1}/4 + \dots) \\ &\equiv_n n(b_i/2 + b_{i+1}/4 + \dots) \end{aligned}$$

- Daher berechnet sich die Bitfolge b_i , $i = 1, 2, \dots$ zu

$$\begin{aligned} b_i &= \text{half}(2^{i-1}x \bmod n) = \text{parity}(2^i x \bmod n) \\ &= \text{clear-parity}(2^{ie} y \bmod n) \end{aligned}$$

Sicherheit partieller Klartextinformationen

- Setzen wir $z_i = n \sum_{j=1}^i b_j 2^{-j}$, so gilt für alle $i > k := \log_2 n$

$$0 \leq x - z_i = n \sum_{j=i+1}^{\infty} b_j 2^{-j} \leq n \sum_{j=i+1}^{\infty} 2^{-j} = n/2^i < 1$$

und somit $x = \lceil z_k \rceil$

- Daher lässt sich x mit Orakelfragen an clear-parity wie folgt unter Berechnung der Bits b_i für $i = 1, 2, \dots, k$ bestimmen:

```

1  z := 0
2  for i := 1 to  $\lceil \log_2 n \rceil$  do
3    y :=  $2^e y \bmod n$ 
4     $b_i := \text{clear-parity}(y)$ 
5    if  $b_i$  then z := z +  $n2^{-i}$ 
6  output  $\lceil z \rceil$ 

```


Beispiel

- Sei $n = 1457$, $e = 779$ und $y = 722$
- Falls das Orakel clear-parity die in der Tabelle angegebenen Antworten $b_i = \text{clear-parity}(y_i)$ für die Kryptotexte $y_i = 2^{ie}y \bmod n$ zurückgibt, erhalten wir die folgenden Werte $z_i = n \sum_{j=1}^i b_j 2^{-j}$, $i = 1, \dots, 11$

i	1	2	3	4	5	6	7	8	9	10	11
y_i	1136	847	1369	1258	1156	826	444	408	1320	71	144
b_i	1	0	1	0	1	1	1	1	1	0	0
$n2^{-i}$	728,5	364,3	182,1	91,1	45,5	22,8	11,4	5,7	2,8	1,4	0,7
z_i	728,5	728,5	910,6	910,6	956,2	978,9	990,3	996	998,8	998,8	998,8
x_i	541	1082	707	1414	1371	1285	1113	769	81	162	324

- Der gesuchte Klartext ist also $x = \lceil z_{11} \rceil = \lceil 998,8 \rceil = 999$
- Dass dieser tatsächlich die vorgegebene Paritätsbitfolge (b_i) generiert, lässt sich durch Berechnung der zu den Kryptotexten y_i gehörigen Klartexte $x_i = 2^i x \bmod n$ verifizieren (siehe letzte Tabellenzeile)

Quadratische Reste

- Als nächstes betrachten wir das Problem, Lösungen für eine quadratische Kongruenzgleichung zu bestimmen
- Zuerst wollen wir herausfinden, ob überhaupt Lösungen existieren

Definition

- Ein Element $a \in \mathbb{Z}_m^*$ heißt **quadratischer Rest modulo m** (kurz: $a \in \text{QR}_m$), falls ein $x \in \mathbb{Z}_m^*$ mit $x^2 \equiv_m a$ existiert
- Die Menge $\text{QNR}_m := \mathbb{Z}_m^* \setminus \text{QR}_m$ enthält alle **quadratischen Nichtreste modulo m**
- Für eine Primzahl $p > 2$ und eine Zahl $a \in \mathbb{Z}$ heißt

$$\mathcal{L}(a, p) = \left(\frac{a}{p} \right) = \begin{cases} 1, & a \bmod p \in \text{QR}_p \\ -1, & a \bmod p \in \text{QNR}_p \\ 0, & \text{sonst} \end{cases}$$

das **Legendre-Symbol von a modulo p**

- Die quadratische Kongruenz $x^2 \equiv_m a$ besitzt also für ein $a \in \mathbb{Z}_m^*$ genau dann eine Lösung, wenn $a \in \text{QR}_m$ ist
- Da mit $a, b \in \text{QR}_m$ auch $ab \in \text{QR}_m$ ist, bildet QR_m eine Untergruppe von \mathbb{Z}_m^*
- Wie das folgende Lemma zeigt, kann die Lösbarkeit von $x^2 \equiv_m a$ im Fall $m = p$ prim effizient entschieden werden

Quadratische Reste

Lemma

- Sei $a \in \mathbb{Z}_p^*$, $p > 2$ prim, und sei g ein beliebiger Erzeuger von \mathbb{Z}_p^*
- Dann sind die folgenden drei Bedingungen äquivalent:
 - 1) $a \in \text{QR}_p$
 - 2) $a^{(p-1)/2} \equiv_p 1$
 - 3) $\log_{p,g}(a)$ ist gerade

Beweis.

1) \Rightarrow 2): Ist $a \in \text{QR}_p$, d. h. $b^2 \equiv_p a$ für ein $b \in \mathbb{Z}_p^*$, so folgt mit dem Satz von Fermat

$$a^{(p-1)/2} \equiv_p b^{p-1} \equiv_p 1$$

2) \Rightarrow 3): Gilt $a \equiv_p g^k$ für ein ungerades $k = 2 \cdot j + 1$, so folgt

$$a^{(p-1)/2} \equiv_p g^{k(p-1)/2} \equiv_p g^{(p-1)j} g^{(p-1)/2} \equiv_p g^{(p-1)/2} \equiv_p -1 \not\equiv_p 1$$

3) \Rightarrow 1): Ist $a \equiv_p g^k$ für $k = 2j$, so folgt $a \equiv_p (g^j)^2$, also $a \in \text{QR}_p$ □

- Somit zerfällt \mathbb{Z}_p in die drei Teilmengen QR_p , QNR_p und $\mathbb{Z}_p \setminus \mathbb{Z}_p^* = \{0\}$
- Die beiden Teilmengen QR_p und QNR_p enthalten jeweils $(p-1)/2$ Elemente
- Zudem ist das Produkt ab von $a, b \in \mathbb{Z}_p^*$ genau dann in QR_p , wenn $a, b \in QR_p$ oder $a, b \in QNR_p$ sind
- Als weitere Folgerung erhalten wir folgende Formel zur effizienten Berechnung des Legendre-Symbols

Quadratische Reste

Satz (Eulers Kriterium)

Für alle $a \in \mathbb{Z}$ und $p > 2$ prim gilt

$$a^{(p-1)/2} \equiv_p \left(\frac{a}{p}\right)$$

Beweis.

- Es ist klar, dass diese Kongruenz im Fall $a \equiv_p 0$ gilt
- Nach obigem Lemma gilt sie auch im Fall $a \bmod p \in \text{QR}_p$, da dann $a^{(p-1)/2} \equiv_p 1 = \left(\frac{a}{p}\right)$ ist
- Es bleibt also der Fall, dass $a \bmod p \in \text{QNR}_p$ ist
 - Da das Polynom $x^2 - 1$ in \mathbb{Z}_p höchstens zwei Nullstellen hat und neben $x = 1$ nach dem Satz von Fermat auch $a^{(p-1)/2} \bmod p$ eine Nullstelle ist, muss $a^{(p-1)/2} \equiv_p \pm 1$ sein
 - Daraus folgt nun $a^{(p-1)/2} \equiv_p -1$, da im Fall $a^{(p-1)/2} \equiv_p 1$ die Zahl $a \bmod p$ in QR_p und somit nicht in QNR_p wäre □

Korollar

Für alle $a, b \in \mathbb{Z}$ und $p > 2$ prim gilt

- $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2} = \begin{cases} 1, & p \equiv_4 1 \\ -1, & p \equiv_4 3 \end{cases}$
- $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{b}{p}\right)$

- Als weiteres Korollar aus Eulers Kriterium erhalten wir eine Methode, quadratische Kongruenzgleichungen im Fall $p \equiv_4 3$ effizient zu lösen
- Im Fall $p \equiv_4 1$ ist dagegen kein effizienter deterministischer Lösungsalgorithmus bekannt
- Allerdings gibt es hierfür effiziente probabilistische Algorithmen (z.B. von Tonelli und Shanks)

Quadratische Reste

Korollar

- Sei $p > 2$ prim, dann besitzt die quadratische Kongruenzgleichung $x^2 \equiv_p a$ für jedes $a \in \text{QR}_p$ in \mathbb{Z}_p genau zwei Lösungen
- Im Fall $p \equiv_4 3$ sind dies $\pm a^k \pmod p$ (für $k = (p+1)/4$), wovon nur $a^k \pmod p$ ein quadratischer Rest ist

Beweis.

- Da $a \in \text{QR}_p$ ist, existiert ein $b \in \mathbb{Z}_p^*$ mit $b^2 \equiv_p a$
- Mit b ist auch $-b$ Lösung von $x^2 \equiv_p a$ mit $-b \not\equiv_p b$ (p ist ungerade)
- Da \mathbb{Z}_p ein Körper ist, existieren keine weiteren Lösungen
- Im Fall $p \equiv_4 3$ liefert Eulers Kriterium für $k = (p+1)/4$ die Kongruenz

$$(a^k)^2 = a^{(p+1)/2} = a^{(p-1)/2} \cdot a \equiv_p a$$

- Da mit a auch $a^k \pmod p \in \text{QR}_p$ ist, folgt

$$\left(\frac{-a^k}{p}\right) = \left(\frac{-1}{p}\right) \cdot \left(\frac{a^k}{p}\right) = (-1)^{(p-1)/2} \left(\frac{a^k}{p}\right) = -1$$

- Also ist $-a^k \pmod p$ ein quadratischer Nichtrest



Das Rabin-System

- Wie das RSA-Verfahren beruht das Rabin-System darauf, dass es zwar effiziente Algorithmen für das Testen der Primzahleigenschaft gibt, effiziente Faktorisierungsalgorithmen aber nicht bekannt sind
- Im Gegensatz zum RSA-Verfahren, von dem nicht bekannt ist, dass es nur durch Faktorisierung des Moduls n gebrochen werden kann, erfüllt das Rabin-System diese Bedingung
- Ähnlich wie RSA verwendet das Rabin-System als Falltürfunktion eine Polynomfunktion $E_k(x) = x(x + e) \bmod n$, wobei $n = pq$ das Produkt zweier großer Primzahlen ist
- Um die Dechiffrierung zu erleichtern, wählt jeder Teilnehmer ein Primzahlpaar p, q mit $p \equiv_4 q \equiv_4 3$, während für e eine beliebige Zahl in \mathbb{Z}_n gewählt werden kann
- Wir werden später sehen, dass der Parameter e keine kryptografische Relevanz hat und daher auf Null gesetzt werden kann

Das Rabin-System

- Der öffentliche Schlüssel ist $k = (e, n)$, wobei $n = pq$ das Produkt zweier Primzahlen p, q mit $p \equiv_4 3$ $q \equiv_4 3$ und $e \in \mathbb{Z}_n$ ist
- Der private Schlüssel ist $k = (p, q)$
- Der Klartextraum ist $M = \mathbb{Z}_n = \{0, \dots, n-1\}$
- Die Verschlüsselungsfunktion ist

$$E_{(e,n)}(x) = x(x + e) \bmod n = y$$

- Zur Entschlüsselung eines Kryptotextes $y \in \{0, \dots, n-1\}$ muss Bob die quadratische Kongruenzgleichung $x(x + e) \equiv_n y$ lösen
- Diese ist äquivalent zu der Kongruenz

$$\underbrace{(x + 2^{-1}e)^2}_{x'} \equiv_n \underbrace{y + (2^{-1}e)^2}_{y'}$$

(**quadratische Ergänzung**), wobei $2^{-1} = (n+1)/2$ das multiplikative Inverse zu 2 modulo n ist

- Setzen wir also $x' = x + 2^{-1}e$ und $y' = y + (2^{-1}e)^2$, so genügt es, alle Lösungen x'_i der Kongruenz $(x')^2 = y'$ zu bestimmen
- Aus diesen lassen sich die zugehörigen Klartext-Kandidaten $x_i = x'_i - 2^{-1}e \pmod n$ berechnen
- Im Fall $y' \equiv_n 0$ (d.h. $\text{ggT}(y', n) = n$) gibt es nur eine Lösung $x' = 0$
- Im Fall $\text{ggT}(y', n) \in \{p, q\}$ gibt es zwei Lösungen (dieser Fall ist unwahrscheinlich und würde dem Gegner die Faktorisierung von n ermöglichen)
- Im verbliebenen Fall $\text{ggT}(y', n) = 1$, also $y' \in \mathbb{Z}_n^*$, hat die Kongruenz $(x')^2 = y'$ vier Lösungen für x' (der Satz auf der nächsten Folie zeigt, wie sich diese bei Kenntnis von p und q effizient bestimmen lassen)
- Das Rabin-System erfüllt also nicht die Bedingung der eindeutigen Dechiffrierbarkeit
- Wir werden jedoch sehen, dass sich der Klartextraum auf eine geeignete Teilmenge von \mathbb{Z}_n^* einschränken lässt, so dass diese Bedingung erfüllt ist

Das Rabin-System

Satz

- Sei $n = pq$ für Primzahlen p, q mit $p \equiv_4 3$ $q \equiv_4 3$
- Dann besitzt die quadratische Kongruenz $x^2 \equiv_n a$ für jedes $a \in \text{QR}_n$ genau vier Lösungen in \mathbb{Z}_n^* , wovon genau eine ein quadratischer Rest ist (d.h. QR_n enthält genau $\varphi(n)/4$ Elemente)

Beweis.

- Mit $x^2 \equiv_n a$ besitzen wegen $n = pq$ auch die beiden Kongruenzen $x^2 \equiv_p a$ und $x^2 \equiv_q a$ Lösungen, und zwar jeweils genau zwei:

$$u_1 = a^{(p+1)/4} \bmod p \in \text{QR}_p \quad u_2 = -a^{(p+1)/4} \bmod p \in \text{QNR}_p$$

$$v_1 = a^{(q+1)/4} \bmod q \in \text{QR}_q \quad v_2 = -a^{(q+1)/4} \bmod q \in \text{QNR}_q$$
- Mit dem Chinesischen Restsatz lässt sich für jedes Paar $(i, j) \in [2] \times [2]$ eine Lösung x_{ij} des folgenden Systems bestimmen

$$x \equiv_p u_i$$

$$x \equiv_q v_j$$

Das Rabin-System

Beweis (Fortsetzung).

- Die Kongruenz $x^2 \equiv_n a$ kann nicht mehr als diese vier Lösungen haben, da sonst für mindestens eine der beiden Kongruenzen $x^2 \equiv_p a$ und $x^2 \equiv_q a$ mehr als zwei Lösungen existieren würden

- Wegen

$$x_{ij} \in \text{QR}_n \Rightarrow \exists s : s^2 \equiv_n x_{ij} \Rightarrow s^2 \equiv_p u_i \wedge s^2 \equiv_q v_j \Rightarrow u_i \in \text{QR}_p \wedge v_j \in \text{QR}_q$$

können $x_{1,2}, x_{2,1}, x_{2,2}$ keine quadratischen Reste modulo n sein

- Da aber u_1 und v_1 quadratische Reste modulo p bzw. q sind, gibt es Zahlen $s \in \mathbb{Z}_p^*$ und $t \in \mathbb{Z}_q^*$ mit $s^2 \equiv_p u_1$ und $t^2 \equiv_q v_1$
- Folglich erfüllt die Lösung $w \in \mathbb{Z}_n^*$ des Systems

$$x \equiv_p s$$

$$x \equiv_q t$$

die Kongruenzen

$$w^2 \equiv_p s^2 \equiv_p u_1 \equiv_p x_{1,1} \quad \text{und} \quad w^2 \equiv_q t^2 \equiv_q v_1 \equiv_q x_{1,1}$$

und somit $w^2 \equiv_n x_{1,1}$, d.h. $x_{1,1} \in \text{QR}_n$



Das Rabin-System

- Als weitere für die Kryptografie interessante zahlentheoretische Funktionen erhalten wir somit für jedes $n = pq$, wobei p, q Primzahlen mit $p \equiv_4 3$ $q \equiv_4 3$ sind, die **diskrete Quadratfunktion** $x \mapsto x^2 \bmod n$, deren Einschränkung auf QR_n nach vorigem Satz eine Permutation auf QR_n ist
- Die zugehörige Umkehrfunktion $x \mapsto \sqrt{x} \bmod n$ ist ebenfalls eine Permutation auf QR_n und heißt **diskrete Quadratwurzelfunktion** auf QR_n
- Wir werden später sehen, dass sich diese Wurzelfunktionen nur bei Kenntnis der Primfaktoren p und q von n effizient berechnen lassen
- Ohne Kenntnis der Faktoren von n lässt sich nicht einmal effizient entscheiden, ob eine gegebene Zahl $a \in \mathbb{Z}_n^*$ in QR_n ist oder nicht
- Aus diesem Grund können wir den Klartextraum des Rabin-Systems auch nicht einfach auf die Menge QR_n einschränken, um die Chiffrierfunktion injektiv zu machen

Beispiel

- Wählen wir $p = 7$, $q = 11$ und $e = 2$, so erhalten wir
 - den öffentlichen Schlüssel $k = (e, n) = (2, 77)$ und
 - den privaten Schlüssel $k' = (p, q) = (7, 11)$
- Um den Klartext $x = 12$ zu verschlüsseln, wird der Kryptotext

$$y = E(k, x) = 12(12 + 2) \bmod 77 = 14$$
 berechnet
- Da $2^{-1}e = 2^{-1} \cdot 2 = 1$ ist, kann dieser durch Lösen der Kongruenz

$$(x + 1)^2 \equiv_{77} y + 1^2 = 15$$
 entschlüsselt werden
- Hierzu löst der legale Empfänger zunächst die beiden Kongruenzen

$$u^2 \equiv_7 15 \equiv_7 1 \text{ und } v^2 \equiv_{11} 15 \equiv_7 4$$
 zu $u_{1,2} = \pm 1^2 = \pm 1$ (wegen $\frac{p+1}{4} = 2$) und
 $v_{1,2} = \pm 4^3 \bmod 11 = \pm 2 \bmod 11$ (wegen $\frac{q+1}{4} = 3$)

Beispiel (Fortsetzung)

- Hierzu löst der legale Empfänger zunächst die beiden Kongruenzen
$$u^2 \equiv_7 15 \equiv_7 1 \text{ und } v^2 \equiv_{11} 15 \equiv_7 4$$
zu $u_{1,2} = \pm 1^2 = \pm 1$ (wegen $\frac{p+1}{4} = 2$) und $v_{1,2} = \pm 4^3 \bmod 11 = \pm 2 \bmod 11$ (wegen $\frac{q+1}{4} = 3$)
- Mit dem Chinesischen Restsatz lassen sich $u_{1,2}$ und $v_{1,2}$ zu den vier Lösungen $x'_{ij} = 57, 64, 13$ und 20 zusammensetzen
- Diese führen auf die vier Klartextkandidaten 12, 19, 56 und 63



Das Rabin-System

- Da auch ein Angreifer die Kongruenz $x(x + e) \equiv_n y$ in die Kongruenz $(x')^2 \equiv_n y'$ mit $x' = x + 2^{-1}e$ und $y' = y + (2^{-1}e)^2$ überführen kann, können wir e auch gleich auf Null setzen
- Zudem können wir die Anzahl der Klartextkandidaten von vier auf zwei reduzieren, wenn wir den Klartextrraum von $M = \mathbb{Z}_n$ auf die Menge $M' = \{1, \dots, (n-1)/2\}$ einschränken
- Um das Rabin-System zu brechen, wird ein effizienter (probabilistischer) Algorithmus \mathcal{A} benötigt, der bei Eingabe (a, n) mit $a \in \text{QR}_n$ eine Zahl $x = \mathcal{A}(a, n)$ mit $x^2 \equiv_n a$ berechnet
- Dabei können wir o.B.d.A. annehmen, dass $x \leq (n-1)/2$ ist
- Wie zeigen nun, dass sich das Problem, ein Rabin-Modul n zu faktorisieren, auf das Lösen von Kongruenzen der Form $x^2 \equiv_n a$ mit $a \in \text{QR}_n$ reduzieren lässt
- Hierzu betrachten wir folgenden Las-Vegas-Algorithmus *Rabin-Factorize*, der n unter Verwendung von \mathcal{A} faktorisiert

Rabin-Factorize(n)

```
1 repeat forever
2   guess randomly  $x \in \{1, \dots, \frac{n-1}{2}\}$ 
3   if  $\text{ggT}(x, n) > 1$  then
4     return( $\text{ggT}(x, n)$ )
5    $a := x^2 \bmod n$ 
6    $x' := \mathcal{A}(a, n)$ 
7   if  $x' \neq x$  then
8     return( $\text{ggT}(x + x', n)$ )
```

Satz

- Der Algorithmus *Rabin-Factorize* gibt bei Eingabe von $n = pq$ mit $p \neq q$ prim einen Primfaktor von n aus
- Die Wahrscheinlichkeit, dass er die repeat-Schleife mehr als t -mal durchläuft, ist kleiner als 2^{-t}

Beweis.

- Es ist klar, dass $\text{ggT}(x, n) \in \{p, q\}$ in Zeile 4 gilt und nach dem Lemma auf Folie 325 gilt $\text{ggT}(x + x', n) \in \{p, q\}$ in Zeile 8
- Um die Laufzeit von *Rabin-Factorize* abzuschätzen, sei \mathcal{X} die Zufallsvariable, die die Wahl von x in der Menge $\{1, \dots, \frac{n-1}{2}\}$ beschreibt
- Zudem sei α die Wahrscheinlichkeit, dass dem Algorithmus in einem Schleifendurchlauf die Faktorisierung von n gelingt
- Für $a \in \mathbb{QR}_n$ seien x_a und x'_a die beiden Lösungen der Kongruenz $x^2 \equiv_n a$ in der Menge $\{1, \dots, \frac{n-1}{2}\}$
- Dann gilt

$$\alpha = \underbrace{\Pr[\text{ggT}(\mathcal{X}, n) > 1]}_{1 - \varphi(n)/(n-1) =: \beta} + \underbrace{\Pr[\mathcal{X} \in \mathbb{Z}_n^* \text{ und } \mathcal{X} \neq \mathcal{A}(\mathcal{X}^2 \bmod n, n)]}_{\gamma}$$

Beweis (Fortsetzung).

- Dann gilt

$$\alpha = \underbrace{\Pr[\text{ggT}(\mathcal{X}, n) > 1]}_{1 - \varphi(n)/(n-1) =: \beta} + \underbrace{\Pr[\mathcal{X} \in \mathbb{Z}_n^* \wedge \mathcal{X} \neq \mathcal{A}(\mathcal{X}^2 \bmod n, n)]}_{\gamma}$$

mit

$$\begin{aligned} \gamma &= \sum_{a \in \text{QR}_n} \left(\underbrace{\Pr[\mathcal{X} = x_a \wedge \mathcal{A}(a, n) = x'_a]}_{\Pr[\mathcal{X} = x_a] \cdot \Pr[\mathcal{A}(a, n) = x'_a | \mathcal{X} = x_a]} + \underbrace{\Pr[\mathcal{X} = x'_a \wedge \mathcal{A}(a, n) = x_a]}_{\Pr[\mathcal{X} = x'_a] \cdot \Pr[\mathcal{A}(a, n) = x_a | \mathcal{X} = x'_a]} \right) \\ &= \sum_{a \in \text{QR}_n} \left(\underbrace{\Pr[\mathcal{X} = x_a]}_{2/(n-1)} \cdot \Pr[\mathcal{A}(a, n) = x'_a] + \underbrace{\Pr[\mathcal{X} = x'_a]}_{2/(n-1)} \cdot \Pr[\mathcal{A}(a, n) = x_a] \right) \\ &= 2|\text{QR}_n|/(n-1) = \varphi(n)/2(n-1) = (1 - \beta)/2 \end{aligned}$$

- Somit ist $\alpha = \beta + \gamma = (\beta + 1)/2 > 1/2$ und die Wahrscheinlichkeit, dass *Rabin-Factorize* mehr als t Schleifendurchläufe ausführt, ist $(1 - \alpha)^t < 2^{-t}$



Um eine eindeutige Dechiffrierung zu erhalten, erweitern wir das Legendre-Symbol zum Jacobi-Symbol

Definition

- Das **Jacobi-Symbol** ist für alle a und alle ungeraden $m = p_1^{e_1} \cdots p_r^{e_r} \geq 3$ durch

$$\mathcal{J}(a, m) = \left(\frac{a}{m}\right) = \left(\frac{a}{p_1}\right)^{e_1} \cdots \left(\frac{a}{p_r}\right)^{e_r}$$

definiert, wobei $p_1 < \cdots < p_r$ die Primfaktoren von m sind

- Ein Element $a \in \text{QNR}_m$ mit $\left(\frac{a}{m}\right) = 1$ heißt **quadratischer Pseudorest modulo m** (kurz: $a \in \widetilde{\text{QR}}_m$)

- Man beachte, dass im Gegensatz zum Legendre-Symbol die Eigenschaft $\left(\frac{a}{m}\right) = 1$ für ein $a \in \mathbb{Z}_m^*$ nicht immer mit $a \in \text{QR}_m$ gleichbedeutend ist
- Zum Beispiel gibt es in \mathbb{Z}_n^* ($n = p \cdot q$ für Primzahlen p und q mit $p \equiv_4 1$ $q \equiv_4 3$) wie wir gesehen haben, genau $\varphi(n)/4$ quadratische Reste und $3\varphi(n)/4$ quadratische Nichtreste
- Dagegen gilt nur für die Hälfte aller $a \in \mathbb{Z}_n^*$ die Gleichung $\left(\frac{a}{n}\right) = -1$
- Folglich gibt es in diesem Fall genau so viele quadratische Reste wie quadratische Pseudoreste
- Interessanterweise ist das Jacobi-Symbol auch ohne Kenntnis der Primfaktorzerlegung des Moduls effizient berechenbar
- Der Algorithmus basiert auf den folgenden beiden Sätzen, die wir ohne Beweis angeben

Satz (Quadratisches Reziprozitätsgesetz, Gauß)

- Seien $m, n > 2$, ungerade und teilerfremd
- Dann gilt

$$\left(\frac{n}{m}\right) \cdot \left(\frac{m}{n}\right) = (-1)^{(m-1) \cdot (n-1)/4} = \begin{cases} 1, & m \equiv_4 1 \vee n \equiv_4 1 \\ -1, & \text{sonst} \end{cases}$$

Satz

Für ungerades m gilt

$$\left(\frac{2}{m}\right) = (-1)^{\frac{m^2-1}{8}} = \begin{cases} 1, & m \equiv_8 1 \vee m \equiv_8 7 \\ -1, & \text{sonst} \end{cases}$$

Das Rabin-System

Korollar

Für a und m mit $\text{ggT}(a, m) = 1$ und $m \geq 3$ ungerade lässt sich $\left(\frac{a}{m}\right)$ in Zeit $O(n^3)$ berechnen (n ist die Länge der Eingabe)

Beweis.

Dies folgt, ähnlich wie beim euklidischen Algorithmus, aus den folgenden Gleichungen:

$$\left(\frac{a}{m}\right) = \begin{cases} 1, & a = 1 \\ \left(\frac{m \bmod a}{a}\right) (-1)^{(a-1)(m-1)/4}, & a \neq 1 \text{ ungerade} \\ \left(\frac{b}{m}\right), & a = 2^{2k}b, b \text{ ungerade} \\ \left(\frac{b}{m}\right) (-1)^{(m^2-1)/8}, & a = 2^{2k+1}b, b \text{ ungerade} \end{cases} \quad \square$$

Beispiel. Das Jacobi-Symbol von 73 modulo 83 ist

$$\left(\frac{73}{83}\right) = \left(\frac{10}{73}\right) \underbrace{(-1)^{82 \cdot 72/4}}_{=1} = \left(\frac{2}{73}\right) \left(\frac{5}{73}\right) = \left(\frac{3}{5}\right) \underbrace{(-1)^{72 \cdot 4/4}}_{=1} = \left(\frac{2}{3}\right) = -1$$

Das Rabin-System

- Schränken wir nun den Klartextrraum weiter auf die Teilmenge $\widetilde{M} = \{x \in M' \mid \left(\frac{x}{n}\right) = 1\}$ von $M' = \{1, \dots, (n-1)/2\}$ ein, so erhalten wir den Kryptotextrraum $\widetilde{C} = \{x^2 \bmod n \mid x \in \widetilde{M}\} = \text{QR}_n$
- Nun existiert zu jedem Kryptotext $y \in \text{QR}_n$ genau ein Klartext $x \in \widetilde{M}$:
 - Wir wissen bereits, dass die Kongruenz $x^2 \equiv_n y$ genau eine Lösung in QR_n hat, nämlich $x_1 = \sqrt{y} \bmod n$
 - Mit x_1 ist auch $x_2 = n - x_1$ eine Lösung von $x^2 \equiv_n y$ und wegen

$$\left(\frac{-\sqrt{y}}{n}\right) = \left(\frac{-\sqrt{y}}{p}\right) \left(\frac{-\sqrt{y}}{q}\right) = (-1) \left(\frac{\sqrt{y}}{p}\right) (-1) \left(\frac{\sqrt{y}}{q}\right) = \left(\frac{\sqrt{y}}{n}\right)$$
 sind x_1, x_2 die einzigen Lösungen von $x^2 \equiv_n y$ in \mathbb{Z}_n mit $\left(\frac{x}{n}\right) = 1$
 - Da aber M' nur eine der beiden Lösungen x_1, x_2 enthält, ist $x^2 \equiv_n y$ in \widetilde{M} eindeutig lösbar
- Alternativ können wir auch M' als Klartextrraum verwenden und jeden Klartext $x \in M'$ zusammen mit $\left(\frac{x}{n}\right)$ übermitteln (verschlüsselt oder unverschlüsselt wie es z.B. auch bei „Textbook-RSA“ der Fall ist)

Das ElGamal-Kryptosystem

- Das System von ElGamal (1985) ist ein probabilistisches Public-key Verfahren auf der Basis des diskreten Logarithmus
- Sei p prim und α ein Erzeuger von \mathbb{Z}_p^*
- Bob wählt als privaten Schlüssel eine Zahl $a \in \mathbb{Z}_{p-1} = \{0, \dots, p-2\}$ und gibt p , α und $\beta = \alpha^a \bmod p$ öffentlich bekannt:
 - der öffentliche Schlüssel ist $k = (p, \alpha, \beta)$ und
 - der private Schlüssel ist $k' = (p, \alpha, \beta, a)$
- Um eine Nachricht $x \in \mathbb{Z}_p^* = \{1, \dots, p-1\}$ an Bob zu senden,
 - wählt Alice zufällig eine Zahl $z \in_R \mathbb{Z}_{p-1}$,
 - berechnet den „Schlüssel“ $\gamma = \beta^z \bmod p$ und
 - sendet (y_1, y_2) mit $y_1 = \alpha^z \bmod p$ und $y_2 = \gamma x \bmod p$ an Bob
- Der Kryptotext (y_1, y_2) hat also die doppelte Länge des Klartextes
- Nach Erhalt des Kryptotextpaars $(y_1, y_2) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$
 - berechnet Bob zunächst $\gamma = y_1^a \bmod p$ (da $y_1^a \equiv_p \alpha^{za} \equiv_p \beta^z \equiv_p \gamma$)
 - und anschließend den Klartext $x = y_2 \gamma^{-1} \bmod p$

Beispiel

- Sei $p = 2579$ und $\alpha = 2$
- Bob wählt aus der Menge $\mathbb{Z}_{p-1} = \{0, \dots, 2577\}$ den privaten Schlüssel $a = 765$ und gibt die Zahl $\beta = \alpha^a \bmod p = 2^{765} \bmod p = 949$ bekannt
- Um die Nachricht $x = 1299$ an Bob zu übermitteln, berechnet Alice den zugehörigen Kryptotext (y_1, y_2) wie folgt:
 - Sie wählt zufällig eine Zahl $z \in_R \{0, \dots, 2577\}$ (z. B. $z = 853$),
 - berechnet $\gamma = \beta^z \bmod p = 949^{853} \bmod p = 2424$ und sendet
 - $(y_1, y_2) = (2^z \bmod p, \gamma x \bmod p) = (435, 2396)$ an Bob
- Nach Erhalt von $y = (435, 2396)$ berechnet Bob
 - zunächst $\gamma = y_1^a \bmod p = 435^{765} \bmod p = 2424$ und anschließend
 - den Klartext $x = y_2 \gamma^{-1} \bmod p = 2396 \cdot 2424^{-1} \bmod p = 1299$

Das ElGamal-Kryptosystem

- Es ist klar, dass das ElGamal-System gebrochen ist, falls es dem Gegner gelingt, den privaten Schlüssel $a = \log_{p,\alpha}(\beta)$ zu berechnen
- Notwendig für die Sicherheit von ElGamal ist daher, dass der diskrete Logarithmus in \mathbb{Z}_p^* nicht mit vertretbarem Aufwand zu berechnen ist
- Hierfür sollte p mindestens eine 300-stellige Dezimalzahl sein
- Im Folgenden betrachten wir verschiedene Sicherheitsaspekte von ElGamal und beginnen mit der komplexitätstheoretischen Sicherheit
- Wegen $\beta \equiv_p \alpha^a \pmod{p}$ und $y_1 \equiv_p \alpha^z \pmod{p}$ kann mit Eulers Kriterium leicht die Parität von a und von z aus β und y_1 abgeleitet werden
- Zudem ist $\gamma = \alpha^{az} \pmod{p}$ genau dann in QR_p , wenn a oder z gerade ist
- Folglich lässt sich bei Kenntnis von β und (y_1, y_2) effizient entscheiden, ob $x = y_2 \gamma^{-1} \pmod{p}$ in QR_p ist
- Dies ist genau dann der Fall, wenn y_2 und γ beide in QR_p oder beide in QNR_p sind

- Daraus ergibt sich unmittelbar, dass der folgende (deterministische) Gegner $G = (\mathcal{X}_0, \mathcal{X}_1, \mathcal{V})$ einen Vorteil von $\alpha(G) = 1$ erzielen kann
- $(\mathcal{X}_0, \mathcal{X}_1)$ wählen zwei Klartexte x_0 und x_1 mit $x_0 \in \text{QR}_p$ und $x_1 \notin \text{QR}_p$
- Dann kann $\mathcal{V}(x_0, x_1, (y_1, y_2))$ anhand von $(y_1, y_2) = E((p, \alpha, \beta), x_b)$ wie oben beschrieben das richtige Bit b ermitteln
- Um diese Schwachstelle zu beheben, genügt es, von der Gruppe $\mathbb{Z}_p^* = \langle \alpha \rangle$ zur Untergruppe $\text{QR}_p = \langle \alpha^2 \rangle$ überzugehen
- Wählen wir zudem p von der Form $p = 2q + 1$ mit p, q prim, so hat QR_p die Ordnung q , d.h. jedes Element $\alpha \in \text{QR}_p \setminus \{1\}$ erzeugt QR_p
- In diesem Fall ist wegen $p \equiv_4 3$ für jedes $x \in \mathbb{Z}_p^*$ genau eine der beiden Zahlen x und $p - x$ in QR_p
- Dies liefert eine leicht zu berechnende Bijektion zwischen QR_p und \mathbb{Z}_q , weshalb wir auch \mathbb{Z}_q anstelle von QR_p als Klartextrraum wählen können

Das ElGamal-Kryptosystem

- Als nächstes gehen wir der Frage nach, wie sicher die einzelnen Bits des privaten Schlüssels a sind
- Da a genau dann gerade ist, wenn $\beta = \alpha^a \bmod p$ ein quadratischer Rest ist, lässt sich das Paritätsbit a_0 von a leicht mit Eulers Kriterium bestimmen
- Bezeichnen wir das i -te Bit a_i der Binärdarstellung $(a_r \cdots a_0)_2$ von $a = \log_{p,\alpha}(\beta)$ mit $L_i(\beta)$, wobei $r = \lfloor \log_2(p-2) \rfloor$ ist, so gilt

$$L_0(\beta) = 0 \Leftrightarrow \beta^{(p-1)/2} \equiv_p 1$$

- Allgemeiner kann man zeigen, dass sich die m Bits $L_{m-1}(\beta), \dots, L_0(\beta)$ von a effizient berechnen lassen, falls $p-1 = 2^m u$ für ungerades u ist
- Dagegen ist $L_m(\beta)$ nicht effizient berechenbar, außer wenn a (und somit der diskrete Logarithmus von β zur Basis α) effizient berechenbar ist

Das ElGamal-Kryptosystem

- Unter der Annahme, dass L_1 effizient berechenbar ist, lässt sich L_i wie folgt auf L_{i-1} reduzieren (wir nehmen an, dass $p \equiv_4 3$, also $m = 1$ ist):
 - Ist $L_0(\beta) = 1$, ersetzen wir β durch $\beta\alpha^{-1} \bmod p$ (nun ist $L_0(\beta) = 0$)
 - Nun berechnen wir die Quadratwurzel $\omega = \beta^{(p+1)/4} \bmod p$ von β
 - Da $L_0(\beta) = 0$, also $\log_{p,\alpha}(\beta)$ gerade ist, gilt $\log_{p,\alpha}(\delta) = \log_{p,\alpha}(\beta)/2$ für $\delta = \omega$ oder $\delta = p - \omega$ und somit $L_i(\beta) = L_{i-1}(\delta)$ für $i = 1, \dots, r$
 - Wegen $\omega \in \text{QR}_p$ und $p - \omega \in \text{QNR}_p$ ist $L_0(\omega) = 0 \neq 1 = L_0(p - \omega)$
 - Daher ist $\delta = \omega$, falls $L_0(\omega) = L_1(\beta)$ ist, und sonst $\delta = p - \omega$
- Damit haben wir die Berechnung von $L_r(\beta), \dots, L_2(\beta)$ mittels einer Frage $L_1(\beta) = 0?$ auf die Berechnung von $L_{r-1}(\delta), \dots, L_2(\delta)$ reduziert
- Indem wir dies wiederholen, haben wir spätestens nach $r - 1$ Iterationen sämtliche Bits $a_i = L_i(\beta)$ von $\log_{p,\alpha}(\beta) = (a_r \cdots a_0)_2$ bestimmt
- Folgender Algorithmus berechnet die Bits $a_i = L_i(\beta)$ durch Fragen an L_1

Das ElGamal-Kryptosystem

```

1  if  $\beta^{(p-1)/2} \equiv_p 1$  then  $a_0 := 0$  else  $a_0 := 1$ 
2   $\beta_1 := \beta\alpha^{-a_0} \bmod p$ ;  $i := 1$ 
3  while  $\beta_i \neq 1$  do
4     $a_i := L_1(\beta_i)$ 
5     $\omega_i := \beta_i^{(p+1)/4}$ 
6    if  $a_i = 0$  then  $\delta_i := \omega_i$  else  $\delta_i := p - \omega_i$ 
7     $\beta_{i+1} := \delta_i\alpha^{-a_i} \bmod p$ 
8     $i := i + 1$ 
9  return $(a_{i-1} \cdots a_0)_2$ 

```

Beispiel. Für $p = 19$, $\alpha = 2$ und $\beta = 6$ ergeben sich folgende Werte:

i	a_i	ω_i	δ_i	β_{i+1}	$\log_{p,\alpha}(\omega_i)$	$\log_{p,\alpha}(\delta_i)$	$\log_{p,\alpha}(\beta_{i+1})$	$(a_i \dots a_0)_2$
0	0	-	-	6	-	-	$(1110)_2$	$(0)_2 = 0$
1	1	5	14	7	$(10000)_2$	$(111)_2$	$(110)_2$	$(10)_2 = 2$
2	1	11	8	4	$(1100)_2$	$(11)_2$	$(10)_2$	$(110)_2 = 6$
3	1	17	2	1	$(1010)_2$	$(1)_2$	$(0)_2$	$(1110)_2 = \mathbf{14}$