Einführung in die Kryptologie

Johannes Köbler

Institut für Informatik
Humboldt-Universität zu Berlin

SS 2020

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 (anstelle eines sicheren) 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

- Symmetrisches 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

- 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, für die Übermittlung an Alice jedoch 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. die inverse Funktion $D_{k'_B}$ darf ohne Kenntnis des privaten Schlüssels k'_B nicht effizient berechenbar sein
- Da die Kenntnis von k'_B dies dennoch ermöglicht, 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 Wegfall des geheimen Chiffrierschlüssels durch Zufall kompensiert wird (siehe Abschnitt über probabilistische Kryptosysteme)

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 wird durch eine Folge $x_1, x_2, \ldots$ von Zahlen $x_i \in \mathbb{Z}_n$ dargestellt, 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

Ver- und Entschlüsselung

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

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

$\square$

Praktische Durchführung

Bestimmung von p und q: Man wählt zufällig eine Zahl x der Form $30z$ und der verlangten Größenordnung (z. B. $x \in I = [10^{500}, 10^{501})$) 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. Diese Bedingung wird z. B. von jeder Primzahl größer als $\max\{p, q\}$ erfüllt

Bestimmung von e: Da $\mathrm{ggT}(d, \varphi(n)) = 1$ ist, liefert der erweiterte euklidische Algorithmus das multiplikative Inverse e von d modulo $\varphi(n)$

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

Kryptoanalytische Betrachtungen

- Es ist klar, dass das RSA-Verfahren gebrochen ist, falls es dem Gegner gelingt, den Modul n zu faktorisieren
- In diesem Fall kann er $\varphi(n)$ und damit auch den privaten Dechiffrierexponenten aus dem öffentlichen Exponenten e berechnen
- Bei Kenntnis von $\varphi(n)$ lassen sich p und q wie folgt berechnen:
  - Sei $n = pq$ (mit $p, q \in \mathcal{P}$; $p > q$)
  - 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

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

- Die Primfaktoren p und q sollten nicht zu nahe beieinander liegen, da n sonst leicht faktorisiert werden kann
- Sei $p > q$
- Dann gilt $q < \sqrt{n} < a < p$, wobei $a = \frac{(p+q)}{2}$ das arithmetische Mittel von p und q ist
- Sei $b = \frac{(p-q)}{2}$ die Entfernung zwischen a und q
- Ist nun $p - q$ klein, so ist auch $\lfloor\sqrt{n}\rfloor - q < a - q = b$ klein
- Daher kann q ausgehend von $\lfloor\sqrt{n}\rfloor$ nach höchstens b Schritten gefunden werden
- Um dies zu verhindern, genügt es, $p > 2q$ zu wählen, da dann

  $$\sqrt{n} - q = \sqrt{pq} - q > \sqrt{2}q - q > q/3$$

  ist

- Mit dem Verfahren der Differenz der Quadrate lässt sich q sogar in $a - \lceil\sqrt{n}\rceil$ 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 $n = 124\,711$ ist z.B. $\lceil\sqrt{n}\rceil = 353$ und bereits für $a = 356$ 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
- Der Aufwand für die Suche 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:

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

- Im Fall $p \geq 2q$ gilt wegen $b = (p-q)/2 = (p+q)/6 + (p-2q)/3$ $\geq (p+q)/6 = a/3$ (also $3b/a \geq 1$),

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

- Daher bringt dieser Angriff in diesem Fall keinen nennenswerten Vorteil

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 Faktorisierung des Moduls n bekannt ist
- 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 der folgende Satz zeigt, erfordert die Bestimmung des geheimen Schlüssels dagegen den gleichen Aufwand wie das Faktorisieren von n
- Bei Kenntnis von d kann nämlich leicht ein Vielfaches $v = ed - 1$ von $k = \mathrm{kgV}(p - 1, q - 1)$ bestimmt und somit n faktorisiert werden
- Zunächst beweisen wir jedoch folgendes Lemma, auf welchem die Faktorisierung von n bei Kenntnis von v beruht

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 ist $\mathrm{ggT}(y+z, m)$ ein nicht-trivialer Faktor 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 $1 < \mathrm{ggT}(y+z, m) < m$ $\square$

Sicherheit des privaten RSA-Schlüssels

Um nun n bei Kenntnis von d zu faktorisieren, betrachten wir folgenden Las-Vegas Algorithmus RSA-Factorize, der durch eine leichte Modifikation aus dem Miller-Rabin Primzahltest hervorgeht

$MRT(n)$, n ungerade

sei $\sum_{i=0}^{r} e_i \cdot 2^i$, $e_r = 1$,
  die Binärdarstellung von $n-1$
guess randomly
  $a \in \{1, \ldots, n-1\}$
$z := a$
for $i := r-1$ **downto** 0 **do**
  $y := z$
  $z := z^2 \bmod n$
  if $z \equiv_n 1 \wedge y \not\equiv_n \pm 1$ **then**
    return(zusammengesetzt)
  if $e_i = 1$ **then** $z := z \cdot a \bmod n$
if $z \not\equiv_n 1$ **then**
  return(zus. gesetzt)
else return(prim)

RSA-Factorize(n, v)

sei $\sum_{i=0}^{r} e_i \cdot 2^i$, $e_r = 1$,
  die Binärdarstellung von v
guess randomly
  $a \in \{1, \ldots, n-1\}$
$z := a$
for $i := r-1$ **downto** 0 **do**
  $y := z$
  $z := z^2 \bmod n$
  if $z \equiv_n 1 \wedge y \not\equiv_n \pm 1$ **then**
    return(ggT$(y+1, n)$)
  if $e_i = 1$ **then** $z := z \cdot a \bmod n$
if ggT$(z, n) > 1$ **then**
  return(ggT(z, n))
else return(?)

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 $v = ed - 1 = 4224$
- Damit führt RSA-Factorize für $a = 174$, $a' = 111$ und $a'' = 137$ die auf der nächsten Folie angegebenen Werte z_i, z_i' bzw. z_i''

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	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 $\text{ggT}(103 + 1, 221) = 13$ $\triangleleft$

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 RSA-Factorize(n, v) mit Wahrscheinlichkeit größer $1/2$ einen Primfaktor von n aus

Beweis.

- Es ist klar, dass jede Ausgabe von RSA-Factorize in Zeile 13 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 10 gezeigt ist
- Wir schätzen nun die Wahrscheinlichkeit ab, dass die Faktorisierung von n nicht gelingt und RSA-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. $i \le j$
- 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

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

- RSA-Factorize findet bei Wahl einer Basis $a \in \mathbb{Z}_n^* \setminus S(n)$ wegen $a^u \not\equiv_n 1$ und $a^{2^t u} \not\equiv_n -1$ für alle $t \ge 0$, aber $a^{2^m u} \equiv_n a^v \equiv_n 1$ einen Primfaktor
- Daher gilt $F(n) \subseteq S(n)$ und es folgt

  $$\Pr[\text{RSA-Factorize}(n, v) = ?] \le \sigma(n)/(n-1)$$

  wobei $\sigma(n) = \|S(n)\|$ ist
- Um $\sigma(n)$ zu berechnen, betrachten wir für $t \ge 0$ die Funktionen $\alpha(n) = \|\{a \in \mathbb{Z}_n^* \mid a^u \equiv_n 1\}\|$ und $\alpha_t(n) = \|\{a \in \mathbb{Z}_n^* \mid a^{2^t u} \equiv_n -1\}\|$
- Wir beweisen nun eine Reihe von Behauptungen, aus denen $\sigma(n)/(n-1) \le 1/2$ folgt

Sicherheit des privaten RSA-Schlüssels

Behauptung 1.
Es gilt $\mathrm{ggT}(2^t u, p-1) = 2^{\min(t,i)} u_1$ und $\mathrm{ggT}(2^t u, q-1) = 2^{\min(t,j)} u_2$

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

Sicherheit des privaten RSA-Schlüssels

Behauptung 2. $\alpha(n) = u_1 u_2$

- Mit dem chinesischen Restsatz folgt

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

- Sei nun g ein Erzeuger von $\mathbb{Z}_p^*$. Dann gilt

$$g^{ku} \equiv_p 1 \quad \Leftrightarrow \quad ku \equiv_{p-1} 0$$

- Dies zeigt $\beta(n) = \mathrm{ggT}(u,\ p-1) \stackrel{\text{Beh. 1}}{=} u_1$
- Analog folgt $\gamma(n) = u_2$. $\square$

Behauptung 3.
Für $t = 0, \dots, i-1$ ist $\alpha_t(n) = 2^{2t} u_1 u_2$ und für $t \geq i$ ist $\alpha_t(n) = 0$

- Mit dem chinesischen Restsatz folgt zunächst

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

- Sei nun g ein Erzeuger von $\mathbb{Z}_p^*$. Dann gilt

$$g^{k2^t u} \equiv_p -1 \quad \Leftrightarrow k2^t u \equiv_{p-1} (p-1)/2$$

- Da $\mathrm{ggT}(2^t u, p-1) \overset{\text{Beh. 1}}{=} 2^t u_1$ genau dann ein Teiler von $(p-1)/2 = 2^{i-1} u_1$ ist, wenn $t \leq i-1$ ist, folgt $\beta_t(n) = 2^t u_1$ für $t = 0, \dots, i-1$ und $\beta_t(n) = 0$ für alle $t \geq i$
- Analog folgt $\gamma_t(n) = 2^t u_2$ für $t = 0, \dots, j-1$ und $\gamma_t(n) = 0$ für alle $t \geq j$ und damit die Behauptung $\square$

Behauptung 4. Es gilt $\sigma(n) \le \varphi(n)/2$

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

$$\begin{aligned}
\sigma(n) &= u_1 u_2 + \sum_{t=0}^{i-1} 2^{2t} u_1 u_2 = u_1 u_2 (1 + \sum_{t=0}^{i-1} 2^{2t}) \\
&= u_1 u_2 (1 + (2^{2i} - 1)/3) = u_1 u_2 (2^{2i} + 2)/3 \\
&\le u_1 u_2 (2^{i+j} + 2^{i+j-1})/3 = \varphi(n)(1 + 2^{-1})/3 = \varphi(n)/2
\end{aligned}$$ □

- Wegen $\varphi(n) = n - p - q + 1 < n - 1$ folgt

$$\sigma(n)/(n-1) \le \varphi(n)/2(n-1) < 1/2$$

womit nun auch der Satz bewiesen ist ■

Sicherheit partieller Klartextinformationen

- Als nächstes gehen wir der Frage nach, wie sicher einzelne Bits der Klartextnachricht sind
- Falls es möglich wäre, aus dem Kryptotext y und dem öffentlichen Schlüssel (e, n) die Parität des Klartextes x effizient zu bestimmen, so könnte auch der gesamte Klartext x effizient berechnet werden
- Das letzte Bit des Klartextes ist also genau so sicher wie der gesamte Klartext
- Einem Angreifer ist es daher nicht möglich, das letzte Bit des Klartextes zu ermitteln, außer wenn es ihm gelingt, RSA vollständig zu brechen
- Wir werden später sehen, dass andere Eigenschaften des Klartextes sehr wohl durch den zugehörigen Kryptotext preisgegeben werden

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

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

  und

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

- Wegen

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

  gilt dann $(2x \bmod n) \equiv_2 \text{half}(x)$ und somit $\text{half}(x) = \text{parity}(2x \bmod n)$

- Daher lässt sich die Berechnung von klartext-half(y) auf die Berechnung von klartext-parity(y) reduzieren:

$$\begin{aligned} \text{klartext-half}(y) &= \text{half}(x) = \text{parity}(2x \bmod n) \\ &= \text{klartext-parity}(2^e y \bmod n) \end{aligned}$$

Sicherheit partieller Klartextinformationen

- Wir stellen die Zahl x/n wie folgt dar

$$x/n = \sum_{i=1}^{\infty} b_i 2^{-i}$$

- Dann gilt

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

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

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

Sicherheit partieller Klartextinformationen

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

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

  und somit $x = \lceil z_{\lceil \log_2 n \rceil} \rceil$

- Daher lässt sich x mit Orakelfragen an klartext-parity wie folgt unter Berechnung der Bits b_i für $i = 1, 2, \ldots, \lceil \log_2 n \rceil$ bestimmen:

```
z := 0
for i := 1 to ⌈log₂ n⌉ do
   y := 2^e y mod n
   b_i := klartext−parity(y)
   if b_i then z := z + n2^(−i)
output ⌈z⌉
```

Sicherheit partieller Klartextinformationen

Beispiel

- Sei $n = 1\,457$, $e = 779$ und $y = 722$
- Falls das Orakel klartext-parity die in der Tabelle angegebenen Werte $b_i = \text{klartext-parity}(y_i)$ für die Kryptotexte $y_i = 2^{ie} y \bmod n$ liefert, ergeben sich folgende Werte für $z_i = n \sum_{j=1}^{i} b_i 2^{-i}$

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) $\triangleleft$

- 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: $a \in \mathrm{QR}_m$), falls ein $x \in \mathbb{Z}_m^*$ mit $x^2 \equiv_m a$ existiert
- Die Menge $\boldsymbol{QNR_m} := \mathbb{Z}_m^* \setminus \mathrm{QR}_m$ enthält alle ***quadratischen Nichtreste modulo*** m
- Für eine Primzahl $p > 2$ und eine Zahl $a \in \mathbb{Z}$ heißt

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

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 $a \in \mathrm{QR}_m$ ist
- Da mit $a, b \in \mathrm{QR}_m$ auch $ab \in \mathrm{QR}_m$ ist, bildet QR_m eine Untergruppe von $\mathbb{Z}_m^*$
- Wie das folgende Lemma zeigt, kann die Lösbarkeit von $x^2 \equiv_m a$ für primes m 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) $a \in \mathrm{QR}_p$
  2) $a^{(p-1)/2} \equiv_p 1$
  3) $\log_{p,g}(a)$ ist gerade

Beweis.

1) $\Rightarrow$ 2): Ist $a \in \mathrm{QR}_p$, 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 $a \in \mathrm{QR}_p$ $\square$

- 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 $a, b \in \mathrm{QR}_p$ oder $a, b \in \mathrm{QNR}_p$ sind
- Als weitere Folgerung erhalten wir folgende Formel zur effizienten Berechnung des Legendre-Symbols

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 $a \bmod p \in \mathrm{QR}_p$, da dann $a^{(p-1)/2} \equiv_p 1 = \left(\frac{a}{p}\right)$ ist
- Es bleibt also der Fall, dass $a \bmod p \in \mathrm{QNR}_p$ 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 $a \in \mathrm{QR}_p$ in $\mathbb{Z}_p$ genau zwei Lösungen
- Im Fall $p \equiv_4 3$ sind dies $\pm a^k$ mod p (für $k = (p+1)/4$), wovon nur a^k mod p ein quadratischer Rest ist

Beweis.

- Da $a \in \mathrm{QR}_p$ 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 mod $p \in \mathrm{QR}_p$ ist, folgt

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

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

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 bei 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 weiter unten sehen, dass e keine kryptografische Relevanz hat und daher auch weggelassen bzw. $e = 0$ gesetzt werden kann

Das Rabin-System

- Der öffentliche Schlüssel ist $k = (e, n)$
- Der geheime Schlüssel ist $k = (p, q)$
- Der Klartextraum ist $M = \mathbb{Z}_n = \{0, \ldots, n-1\}$
- Die Verschlüsselungsfunktion ist

$$E((e, n), x) = x(x + e) \bmod n = y$$

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

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

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

Das Rabin-System

- 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 $x_i = x_i' - 2^{-1}e \bmod n$ berechnen
- Im Fall $y' \equiv_n 0$ gibt es nur eine Lösung $x' = 0$
- Im Fall $\mathrm{ggT}(y', n) \in \{p, q\}$ gibt es zwei Lösungen (dieser Fall ist unwahrscheinlich und würde dem Gegner die Faktorisierung von n ermöglichen)
- Im verbliebenen Fall $\mathrm{ggT}(y', n) = 1$, 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 weiter unten sehen, wie man den Klartextraum auf eine geeignete Teilmenge $M'' \subseteq \mathbb{Z}_n^*$ einschränken kann, so dass diese Bedingung erfüllt ist

Das Rabin-System

Satz

- Sei $n = pq$ für Primzahlen p, q mit $p \equiv_4 q \equiv_4 3$
- Dann besitzt die quadratische Kongruenz $x^2 \equiv_n a$ für jedes $a \in \mathrm{QR}_n$ genau vier Lösungen, wovon genau eine ein quadratischer Rest ist

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

$$u_1 = a^{(p+1)/4} \bmod p \in \mathrm{QR}_p \qquad u_2 = -a^{(p+1)/4} \bmod p \in \mathrm{QNR}_p$$
$$v_1 = a^{(q+1)/4} \bmod q \in \mathrm{QR}_q \qquad v_2 = -a^{(q+1)/4} \bmod q \in \mathrm{QNR}_q$$

- 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
$$x_{ij} \in \mathrm{QR}_n \Rightarrow \exists s : s^2 \equiv_n x_{ij} \Rightarrow s^2 \equiv_p u_i \wedge s^2 \equiv_q v_j \Rightarrow u_i \in \mathrm{QR}_p \wedge v_j \in \mathrm{QR}_q$$
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. $x_{1,1} \in \mathrm{QR}_n$ $\square$

Das Rabin-System

- Als weitere für die Kryptografie interessante zahlentheoretische Funktionen erhalten wir somit für jedes $n = pq$, wobei p, q Primzahlen mit $p \equiv_4 q \equiv_4 3$ sind, die ***diskrete Quadratfunktion*** $x \mapsto x^2 \bmod n$, die nach vorigem Satz eine Permutation auf QR_n ist
- Ihre Umkehrfunktion $x \mapsto \sqrt{x} \bmod n$ heißt ***diskrete Quadratwurzelfunktion*** auf QR_n
- Wir werden später sehen, dass sich diese Funktion nur bei Kenntnis der Primfaktoren p und q von n effizient berechnen lässt
- 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

Das Rabin-System

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

  $$(x + 1)^2 \equiv_{77} y + 1 = 15$$

  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 $v_{1,2} = \pm 4^3 \bmod 11 = \pm 2$ (wegen $\frac{q+1}{4} = 3$)

Beispiel (Fortsetzung)

- 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 $\triangleleft$

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 Klartextraum von $M = \mathbb{Z}_n$ auf die Menge $M' = \{1, \dots, (n-1)/2\}$ einschränken
- Es ist klar, dass das System gebrochen ist, sobald n in seine Primfaktoren p, q zerlegt werden kann
- Wie wir gleich sehen werden, sind für Zahlen n von dieser Bauart das Faktorisierungsproblem und das Problem, eine Lösung der quadratischen Kongruenz $x^2 \equiv_n a$ für ein gegebenes $a \in \mathrm{QR}_n$ zu finden, äquivalent
- Um das Rabin-System zu brechen, wird ein effizienter Algorithmus A benötigt, der bei Eingabe (a, n) mit $a \in \mathrm{QR}_n$ eine Zahl $c = A(a, n)$ mit $c^2 \equiv_n a$ berechnet
- Dabei können wir o.B.d.A. annehmen, dass $c \leq (n-1)/2$ ist
- Unter Verwendung von A erhalten wir nun folgenden probabilistischen Faktorisierungsalgorithmus Rabin-Factorize

Das Rabin-System

Rabin-Factorize(n)

```
repeat forever
    guess randomly x ∈ {1, ..., (n−1)/2}
    if ggT(x, n) > 1 then
        return(ggT(x, n))
    a := x^2 mod n
    z := A(a, n)
    if z ≢_n ±x then
        return(ggT(x + z, n))
```

Satz

- Der Algorithmus Rabin-Factorize gibt bei Eingabe $n = pq$, p und q prim mit $p \equiv_4 q \equiv_4 3$, 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 $\mathrm{ggT}(x{,}n)$ im Fall $\mathrm{ggT}(x{,}n)>1$ ein Primfaktor von n ist
- Nach dem Lemma auf Folie 319 gilt dies auch für die Ausgabe in Zeile 7
- Um die Laufzeit von Rabin-Factorize abzuschätzen, sei X die Zufallsvariable, die die Wahl von x in der Menge $M' = \{1, \ldots, \frac{n-1}{2}\}$ beschreibt
- Zudem sei p_1 die Wahrscheinlichkeit, dass dem Algorithmus in einem Schleifendurchlauf die Faktorisierung von n gelingt
- Sei $x' \neq x$ die neben x zweite Lösung in $M' \cap \mathbb{Z}_n^*$ der $x^2 \equiv_n a$ $\square$

Das Rabin-System

Beweis (Fortsetzung).

- Dann gilt

$$p_1 = \underbrace{\Pr[\mathrm{ggT}(X, n) > 1]}_{\Pr[X \notin \mathbb{Z}_n^*] =: \alpha} + \underbrace{\Pr[X \in \mathbb{Z}_n^* \text{ und } X \not\equiv_n \pm A(X^2, n)]}_{\beta}$$

mit

$$\begin{aligned} \beta &= \sum_{x \in M' \cap \mathbb{Z}_n^*} \Pr[X = x \wedge A(X^2, n) = x'] \\ &= \sum_{x \in M' \cap \mathbb{Z}_n^*} \Pr[A(X^2, n) = x'] \underbrace{\Pr[X = x \mid A(X^2, n) = x']}_{1/2} \\ &= \Pr[A(X^2, n) \in M' \cap \mathbb{Z}_n^*]/2 = \Pr[X \in \mathbb{Z}_n^*]/2 = (1 - \alpha)/2 \end{aligned}$$

- Somit ist $p_1 = \alpha + \beta = (\alpha + 1)/2 > 1/2$
- Die Wahrscheinlichkeit, dass Rabin-Factorize mehr als t Schleifendurchläufe ausführt, ist also $(1 - p_1)^t < 2^{-t}$ $\square$

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
- Ist zwar $\left(\frac{a}{m}\right) = 1$, aber $a \in \mathrm{QNR}_m$ kein quadratischer Rest modulo m, so heißt a ***quadratischer Pseudorest modulo*** m (kurz: $a \in \widetilde{\mathrm{QR}}_m$)

- 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 $a \in \mathrm{QR}_m$ gleichbedeutend ist
- Zum Beispiel gibt es in $\mathbb{Z}_n^*$ ($n = p \cdot q$ für Primzahlen p und q mit $p \equiv_4 q \equiv_4 3$) 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 $\left(\frac{a}{m}\right) = -1$
- 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

Das Rabin-System

Satz (Quadratisches Reziprozitätsgesetz, Gauß)

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

$$\left(\frac{n}{m}\right) \cdot \left(\frac{m}{n}\right) = (-1)^{(m-1)\cdot(n-1)/4}$$

Satz

Für ungerades m gilt

$$\left(\frac{2}{m}\right) = (-1)^{\frac{m^2-1}{8}}$$

- Man beachte, dass $\frac{m^2-1}{8}$ genau dann gerade ist, wenn $m \equiv_8 1$ oder $m \equiv_8 7$ gilt
- Zudem ist $(m-1)\cdot(n-1)/4$ genau dann gerade, wenn $m \equiv_4 1$ oder $n \equiv_4 1$ gilt

Das Rabin-System

Korollar

Seien a und m gegeben mit $m \geq 3$ ungerade und $\mathrm{ggT}(a, m) = 1$; dann lässt sich $\left(\frac{a}{m}\right)$ durch einen Algorithmus der Zeitkomplexität $O(n^3)$ berechnen

Beweis.

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

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

□

Beispiel. Das Jacobi-Symbol von 73 modulo 83 ist

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

◁

Das Rabin-System

- Schränken wir nun den Klartextraum weiter auf die Teilmenge $M'' = \{x \in M' \mid (\frac{x}{n}) = 1\}$ der Menge $M' = \{1, \ldots, (n-1)/2\}$ ein, so erhalten wir als Kryptotextraum die Menge QR_n
- Zudem existiert zu jedem $y \in \mathrm{QR}_n$ genau ein Klartext $x \in M''$:
  - Wir wissen bereits, dass y genau eine Wurzel $\sqrt{y}$ in QR_n hat
  - Neben $x' = \sqrt{y} \bmod n$ ist $x'' = -\sqrt{y} \bmod n$ wegen

    $$\left(\frac{x''}{n}\right) = \left(\frac{x''}{p}\right)\left(\frac{x''}{q}\right) = (-1)\left(\frac{x'}{p}\right)(-1)\left(\frac{x'}{q}\right) = \left(\frac{x'}{n}\right)$$

    die einzige Lösung von $x^2 \equiv_n y$ mit $(\frac{x}{n}) = 1$
  - Da aber genau eine dieser beiden Lösungen in M' enthalten ist, existiert in M'' genau ein x mit $x^2 \equiv_n y$
- Zwar lässt sich der Klartextraum $\mathbb{Z}_n$ problemlos auf M' einschränken, aber eine weitere Einschränkung auf M'' erscheint problematisch
- Einfacher ist es, das Bit $(\frac{x}{n})$ an den Empfänger zu übermitteln (entweder unverschlüsselt, wie es bei „Schulbuch-RSA“ der Fall ist, oder in verschlüsselter Form)

Das ElGamal-Kryptosystem

- Das System von ElGamal (1985) ist ein probabilistisches Public-key Verfahren auf der Basis des diskreten Logarithmus
- Sei p eine Primzahl und α ein Erzeuger von $\mathbb{Z}_p^*$ (p und α sind öffentlich)
- Jeder Teilnehmer X wählt als privaten Schlüssel eine Zahl $a \in \mathbb{Z}_{p-1} = \{0, \ldots, p-2\}$ und gibt $\beta = \alpha^a \bmod p$ öffentlich bekannt:
  - der öffentliche Schlüssel ist $k = (p, \alpha, \beta)$
  - der private Schlüssel ist $k' = (p, \alpha, \beta, a)$
- Um eine Nachricht $x \in \mathbb{Z}_p^* = \{1, \ldots, p-1\}$ an Bob zu senden,
  - wählt Alice zufällig eine Zahl $z \in \mathbb{Z}_{p-1}$,
  - 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 wie der Klartext
- Nach Erhalt des Kryptotextpaars (y_1, y_2)
  - 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 $x = y_2 \gamma^{-1} \bmod p$

Das ElGamal-Kryptosystem

Beispiel

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

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
- Mit Eulers Kriterium kann leicht die Parität von a aus $\beta = \alpha^a \bmod p$ und die Parität von z aus $y_1 = \alpha^z \bmod p$ bestimmt werden
- Folglich lässt sich leicht ermitteln, ob $\gamma = \beta^z$ in QR_p ist oder nicht
- Zudem ist $y_2 = \gamma x \bmod p$ genau dann in QR_p, wenn entweder $\gamma, x \in \mathrm{QR}_p$ oder $\gamma, x \in \mathrm{QNR}_p$ sind
- Nun lässt sich auch leicht ermitteln, ob x in QR_p ist oder nicht: x ist in QR_p, wenn y_2 und γ beide in QR_p oder beide in QNR_p sind

Das ElGamal-Kryptosystem

- Daraus ergibt sich unmittelbar, dass folgender Gegner $G = (X_0, X_1, V)$ einen Vorteil von $\alpha(G) = 1$ erzielen kann
- (X_0, X_1) wählen zwei Klartexte x_0 und x_1 mit $x_0 \in \mathrm{QR}_p$ und $x_1 \notin \mathrm{QR}_p$
- $V(x_0, x_1, (y_1, y_2))$ ermittelt anhand des Kryptotextes $(y_1, y_2) = E((p, \alpha, \beta), x_b)$ wie oben beschrieben das richtige Bit b
- Um diese Schwachstelle zu beheben, genügt es, von der Gruppe $\mathbb{Z}_p^*$ zur zyklischen Untergruppe $\mathrm{QR}_p = \langle \alpha^2 \rangle$ 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 $\alpha \in \mathrm{QR}_p \setminus \{1\}$ ist ein Erzeuger
- 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 Klartextraum 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 niederwertigste Bit von a leicht mit Eulers Kriterium bestimmen
- Bezeichnen wir das Bit an der Stelle $i \geq 0$ von $a = \log_{p,\alpha}(\beta)$ mit $L_i(\beta)$ (d.h. $a = (L_r(\beta) \cdots L_0(\beta))_2$ für $r = \lfloor \log_2(p-2) \rfloor$), so gilt

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

- Allgemeiner kann man zeigen, dass sich im Fall $p - 1 = 2^m u$, u ungerade, die m niederwertigen Bits $L_{m-1}(\beta), \ldots, L_0(\beta)$ von a effizient berechnen lassen
- Dagegen ist die Berechnung des nächsten Bits $L_m(\beta)$ nicht effizient möglich, außer wenn alle Bits von a (und damit der diskrete Logarithmus) effizient berechenbar sind

Das ElGamal-Kryptosystem

- Wir zeigen dies für den Spezialfall $m = 1$ (d.h. $p \equiv_4 3$)
- Unter der Annahme, dass für ein gegebenes γ nicht nur $L_0(\gamma)$, sondern auch $L_1(\gamma)$ effizient berechenbar ist, ist $L_2(\beta)$ wie folgt berechenbar:
  - Zuerst setzen wir das niederwertigste Bit $L_0(\beta)$ im Fall $L_0(\beta) = 1$ auf 0, indem wir β durch $\beta\alpha^{-1}$ mod p ersetzen
  - Als nächstes berechnen wir die beiden Quadratwurzeln $\omega_{1,2}$ von β, d.h. $\omega_{1,2} = \pm\beta^{(p+1)/4}$ mod p
  - Da $L_0(\beta) = 0$ ist, erhalten wir die Binärdarstellung des diskreten Logarithmus $\log_{p,\alpha}(\omega_j)$ einer dieser beiden Wurzeln aus der Binärdarstellung von $\log_{p,\alpha}(\beta)$ durch einen Rechtsshift um eine Stelle, d.h. es gilt $L_{i+1}(\beta) = L_i(\omega_j)$ für $i = 0, \ldots, r-1$
  - Wegen $\omega_1 \in \mathrm{QR}_p$ und $\omega_2 \in \mathrm{QNR}_p$ ist $L_0(\omega_1) = 0 \neq 1 = L_0(\omega_2)$
  - Da wir nach Voraussetzung $L_1(\beta)$ effizient berechnen können, lässt sich die gesuchte Wurzel ω_j also daran erkennen, dass sie die Bedingung $L_0(\omega_j) = L_1(\beta)$ erfüllt
  - Da nach Voraussetzung auch $L_1(\omega_j)$ effizient berechenbar und $L_2(\beta) = L_1(\omega_j)$ ist, ist auch $L_2(\beta)$ effizient berechenbar

Das ElGamal-Kryptosystem

- Wir haben also das Problem, die Bits $L_r(\beta), \ldots, L_2(\beta)$ zu bestimmen, auf das Problem reduziert, die Bits $L_{r-1}(\omega_j), \ldots, L_2(\omega_j)$ zu bestimmen
- Indem wir dies wiederholen, haben wir spätestens nach $r-1$ Iterationen sämtliche Bits von a bestimmt
- Der Algorithmus auf der nächsten Folie berechnet die Bits a_i durch Fragen an $L_1(\beta)$

Beispiel

Für $p = 19$, $\alpha = 2$ und $\beta = 6$ werden folgenden Werte berechnet:

i	a_i	ω_i	δ_i	β_{i+1}
0	0	-	-	6
1	1	5	14	7
2	1	11	8	4
3	1	17	2	1

Die Ausgabe ist also $(a_3 \cdots a_0)_2 = (1110)_2 = 14$. $\triangleleft$

Das ElGamal-Kryptosystem

Berechnung von $a = \log_{p,\alpha}(\beta)$ durch Fragen an $L_1(\beta)$

$a_0 := L_0(\beta)$
$\beta_1 := \beta\alpha^{-a_0} \bmod p$
$i := 1$
while $\beta_i \neq 1$ **do**
  $a_i := L_1(\beta_i)$
  $\omega_i := \beta_i^{(p+1)/4}$
  if $a_i = 0$ **then** $\delta_i := \omega_i$ **else** $\delta_i := p - \omega_i$
  $\beta_{i+1} := \delta_i\alpha^{-a_i} \bmod p$
  $i := i + 1$
return $(a_{i-1} \cdots a_0)_2$