RSA Sicherheit

Das RSA Problem ist, zu (n,e) und $c \in \mathbb{Z}/n\mathbb{Z}$ ein $m \in \mathbb{Z}/n\mathbb{Z}$ mit $m^e = c$ zu berechnen (also e-te Wurzeln in $\mathbb{Z}/n\mathbb{Z}$ zu ziehen, oder die RSA Funktion zu invertieren).

Angriffe auf das RSA Problem

- durch Faktorisieren von n.
- bei kleinem d.
- bei kleinem e.

Außerdem: Partielle Information über Klartext aus Chiffretext erhaltbar, Homomorphieeigenschaft.

RSA: Kleines d, kleines e

Thm (Wiener): Gilt $q < p < 2q$ und $d < n^{1/4}/3$, so kann d aus n,e (mit $e < \phi(n)$) effizient berechnet werden.

Man vermutet, daß $d < n^{1/2}$ bereits ausreicht.

Thm (Hastad): Seien n_i teilerfremd und $g_i \in \mathbb{Z}[x]$ normiert mit $s = \max_i \deg(g_i)$, für $1 \le i \le k$. Es gebe ein eindeutiges $m < \min_i n_i$ mit $g_i(m) = 0 \bmod n_i$. Ist $k > s$, so kann m effizient aus den n_i und g_i berechnet werden.

Anwenden mit $g_i(m) = f_i(m)^{e_i} - c_i \bmod n_i$.

- Speziell $e_i = e$ und lineare f_i (z.B. festes Padding, $f_i(m) = m + b_i$).
- Im allgemeinen kann auf die Bedingung g_i normiert verzichtet werden.

RSA: Kleines e

Franklin-Reiter Related Message Angriff.

- Annahme: $m_2 = f(m_1)$ für ein $f \in \mathbb{Z}/n\mathbb{Z}[x]$.
- Sei $c_i = m_i^e \bmod n$. Dann ist m_1 Nullstelle von $g(x) = f(x)^e - c_2$ und $h(x) = x^e - c_1$ modulo n.
- In vielen Fällen ist $s(x) = \gcd\{g(x), h(x)\}$ linear, so daß Absolutkoeffizient m_1 ergibt.
- Laufzeit quadratisch in $e \deg(f)$.

Ist $e = 3$ und $f(x) = ax + b \in \mathbb{Z}/n\mathbb{Z}[x]$ mit $a, b \neq 0 \bmod n$, so können wir entweder n faktorisieren oder es gilt $\deg(s(x)) = 1$.

(Koeff-vergleich $\Rightarrow g \nmid h$, $h \nmid g \bmod n$. Mod p und q ist $\deg(s) \in \{1,3\}$ da h lin. und quad. Faktor besitzt. Zusammen folgt $\deg(s) = 3$ modulo n und der Leitkoeffizient von s ist $\notin (\mathbb{Z}/n\mathbb{Z})^\times$, oder $\deg(s) = 1$ modulo n.)

RSA: Kleines e

Coppersmith Short Pad Angriff.

Thm: Sei $r = \lfloor \log_2(n)/e^2 \rfloor$ und m eine Nachricht mit $\le \log_2(n) - r$ Bits. Sei $m_i = 2^r m + r_i$ für $i \in \{1,2\}$, wobei $0 \le r_i < 2^r$. Aus den $c_i = m_i^e \bmod n$ läßt sich m effizient berechnen.

Bew: Sei $g_1(x,y) = x^e - c_1$ und $g_2(x,y) = (x+y)^e - c_2$. Für $y = r_2 - r_1$ haben g_1 und g_2 die gemeinsame Nullstelle m_1. Also ist $r_2 - r_1$ Nullstelle der Resultante $h(y) = \mathrm{res}_x(g_1, g_2) \in \mathbb{Z}/n\mathbb{Z}[y]$. Es gilt $\deg(h) \le e^2$. Außerdem $|r_2 - r_1| < 2^r < n^{1/e^2}$. Daher ist $r_2 - r_1$ eine kleine Nullstelle und kann mit dem Satz von Coppersmith ausgerechnet werden. Danach Franklin-Reiter mit $f(x) = x + r_2 - r_1$ anwenden. $\square$

Angriff möglich für $e = 3$ (Padlänge $< 1/9$ der Nachrichtenlänge), nicht aber für $e = 2^{16} + 1$ bei den gegenwärtigen Größen von n.

RSA: Kleines e

Partial Key Exposure Angriff.

Sei b die Bitlänge von n und gelte $b = 0 \bmod 4$ und $e < n^{1/2}$.

Thm (Coppersmith): Mit Hilfe der $b/4$ unteren oder oberen Bits von p kann n effizient faktorisiert werden.

Thm (Boneh, Durfee, Fraenkel): Mit Hilfe der $b/4$ unteren Bits von d kann ganz d in Zeit $e \log_2(e)$ berechnet werden.

Bew: Es gibt k mit $ed - k(n-p-q+1) = 1$. Aus $d < \phi(n)$ folgt $0 < k \leq e$. Mit $q = n/p$ folgt $(ed)p - kp(n-p+1) + kn = p \bmod 2^{b/4}$. Der Wert $ed \bmod 2^{b/4}$ ist bekannt. Daher Gleichung in k und p. Für jedes k nach $p \bmod 2^{b/4}$ lösen und mit Thm Coppersmith testen. Es gibt höchstens $e \log_2(e)$ viele Lösungen. $\square$

RSA: Kleines e

Die Hälfte der oberen Bits von d ist im wesentlichen öffentlich bekannt:

- Wieder $ed - k(n-p-q+1) = 1$ mit $0 < k \leq e$.
- $d' = \lfloor (kn+1)/e \rfloor$.
- $|d - d'| \leq k(p+q)/e \leq O(kn^{1/2}/e)$.
- Damit ist d' eine gute Approximation an d.

Nicht jede Nachricht wird gut verschlüsselt:

- Kleines m, so daß $m^e < n$.
- Dann keine Reduktion $\bmod n$ und m durch e-te Wurzel in $\mathbb{Z}$ bestimmt.

Jacobi Symbol

Sei p ungerade Primzahl und $a, n \in \mathbb{Z}$ mit $n \geq 1$ ungerade. Dann

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{wenn } a = 0 \bmod p. \\ 1 & \text{wenn a ein Quadrat in } (\mathbb{Z}/p\mathbb{Z})^\times \text{ ist.} \\ -1 & \text{sonst.} \end{cases}$$

$\left(\frac{a}{n}\right) = \prod_{p \text{ prim}} \left(\frac{a}{p}\right)^{v_p(n)}$. Damit multiplikativ in a und n.

$\left(\frac{a}{n}\right) = \left(\frac{a + \lambda n}{n}\right)$ für alle $\lambda \in \mathbb{Z}$.

Für $\gcd\{a, n\} > 1$ ist $\left(\frac{a}{n}\right) = 0$.

Quadratisches Reziprozitätsgesetz

Thm: Seien $b, n \in \mathbb{Z}$ mit $b, n \geq 0$ beide ungerade.

$\left(\frac{b}{n}\right) = (-1)^{(b-1)(n-1)/4} \left(\frac{n}{b}\right)$, $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$, $\left(\frac{-1}{n}\right) = (-1)^{(n-1)/2}$.

Der Satz erlaubt es, Jacobisymbole effizient zu berechnen:

1. Ersetze b durch $b \bmod n$. Wenn $b = 0$, dann Ergebnis 0.
2. $b = 2^{v_2(b)} b'$. Benutze Multiplikativität und zweite Formel.
3. Vertausche b' und n entsprechend der ersten Formel.
4. Wiederhole ab 1.

RSA Partielle Information

Sei $c = m^e \bmod n$. Da e ungerade ist, gilt $\left(\frac{c}{n}\right) = \left(\frac{m}{n}\right)^e = \left(\frac{m}{n}\right)$.
Man kann also aus dem Chiffretext ein Bit Information über den Klartext erhalten.

Setze $f(c) = 0$ für $0 \le m < n/2$ und $f(c) = 1$ für $n/2 < c \le n-1$.
Können wir $f(c)$ effizient berechnen, so auch m:

- Berechne $h_i = f(c(2^e)^i)$ für $0 \le i \le \log_2(n)$. Es gilt $h_i = f((m2^i)^e)$.
- Weiter $h_i = 0 \Leftrightarrow m2^i \bmod n \in [0, n/2) \Leftrightarrow m2^i - jn \in [0, n/2) \Leftrightarrow$
  $$m \in \cup_{j=0}^{2^i-1} [nj/2^i, n(j+1/2)/2^i).$$
- $\#[nj/2^i, n(j+1/2)/2^i) \cap \mathbb{Z} = 1$ für $i = \lfloor \log_2(n) \rfloor$.
- $h_i = 0$ linkes Intervall, $h_i = 1$ rechtes Intervall. Dann $i \leftarrow i+1$, rekursiv wiederholen mit halber Intervallbreite. Ist binäre Suche, liefert m.

RSA Partielle Information

Setze $g(c) = m \bmod 2$ (Parität von m).

- Es gilt $f(c) = g(c2^e \bmod n)$, denn $(2m \bmod n) \bmod 2 = 2m \bmod 2 = 0$ für $m < n/2$, und $(2m \bmod n) = (2m - n) \bmod 2 = 1$, da n ungerade.
- Es gilt $g(c) = f(c2^{-e} \bmod n)$, nach der ersten Gleichung und da 2 modulo n invertierbar ist.

Die Berechnungen von f und g sind daher polynomiell äquivalent.

Folgerung: Unter der Annahme, daß das RSA Problem schwer ist, ist also auch die Berechnung von f und g schwer.
Man kann zeigen, daß jedes individuelle Bit sicher ist.

(Es könnte aber noch sein, daß es einen effizienten Algorithmus gibt, der f und g nur mit Wahrscheinlichkeit $3/4$ korrekt berechnet ...)

RSA Homomorphieeigenschaft

Sind c_1, c_2 die Chiffretexte zu den Nachrichten m_1, m_2, so ist c_1c_2 der Chiffretext zu m_1m_2. Man kann also den Chiffretext von m_1m_2 ausrechnen, ohne m_1m_2 zu kennen.

Diese homomorphe Eigenschaft birgt Vor- und Nachteile in sich.

- Betrugsmöglichkeit (Abhilfe: Klartextraum einschränken, so daß m_1m_2 ungültig ist).
- Hilfreich bei anonymem digitalen Geld.
- Benutzt bei manchen der vorstehenden Angriffe, und beim folgenden Angriff.

RSA Meet-in-the-middle Angriff

Es gelte $m = m_1m_2$ in $\mathbb{Z}$ mit $m_1 \le 2^{b_1}$ und $m_2 \le 2^{b_2}$. Der Chiffretext sei $c = m^e \bmod n$. Es folgt $c/m_1^e = m_2^e \bmod n$.

Dies liefert folgende Strategie:

- Liste von den Werten m_2^e für alle $m_2 \le 2^{b_2}$ machen.
- Die Werte c/m_1^e für alle $m_1 \le 2^{b_1}$ berechnen und nach Kollision mit m_2^e suchen.
- Wenn Kollision, dann gilt für $m = m_1m_2$, daß $m^e = c \bmod n$ ist.
- Also ist m der Klartext.

Aufwand $\approx 2^{b_1} + 2^{b_2}$. Wahrscheinlichkeit, daß 64 Bitzahl in zwei gleichgroße Faktoren zerfällt $\approx 18\%$ (Session keys).

Angriff nur für kleine m, aber beliebige e relevant.

RSA Zusammenfassung

Man sollte die Parameter nicht so wählen, daß d eine besondere Gestalt bekommt (also Parameter zufällig wählen).

Bei kleinem e gibt es verschiedene Probleme, wenn:
- eine Nachricht mit vielen Schlüsseln verschlüsselt wird.
- korrellierte Nachrichten mit einem Schlüssel verschlüsselt werden.
- spezielle Nachrichten verschlüsselt werden.

Mit den unteren $b/4$ Bits der b Bits von d kann n faktorisiert werden. Die oberen $b/2$ Bits von d sind unwichtig.

Plain RSA Verschlüsselung gibt Information über die Nachrichten preis (Jacobi Symbol). Die Parität und das Intervall der Nachricht ($[0, n/2)$, $(n/2, n)$) sind jedoch geschützt.

Rabin Kryptosystem

Ist wie RSA aber mit $e = 2$.
- Problem: $\gcd\{e, \phi(n)\} \neq 1$, daher gibt es d nicht!
- Wie also Quadratwurzel in $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ ziehen?
- Reduktion auf Quadratwurzeln in endlichen Körpern ziehen ...

Jacobisymbol gibt an, ob ein Element ein Quadrat in $\mathbb{Z}/p\mathbb{Z}$ ist, hilft aber nicht bei der Berechnung.
- Allgemeine Lösung ist, $x^2 - c$ in $\mathbb{Z}/p\mathbb{Z}[x]$ zu faktorisieren.
- Spezielle Lösung: $p = 3 \bmod 4$. Sind $a, b \in \mathbb{Z}/p\mathbb{Z}$ mit $a = b^2$, so folgt $(a^{(p+1)/4})^2 = a^{(p-1)/2}a = b^{p-1}a = a$, also $a^{(p+1)/4} = \pm b$.

Problem: In jeder Koordinate gibt es zwei Quadratwurzeln, also insgesamt vier! Welche ist der Klartext?
- Redundanz oder Regel hinzufügen, so daß immer nur eine Möglichkeit in Frage kommt (z.B. sollen selbst Quadrate sein).

Rabin Sicherheit

Thm: Wenn man Quadratwurzeln ziehen kann, dann kann man n faktorisieren.

Bew: Wähle $x \in \mathbb{Z}/n\mathbb{Z}$ zufällig, und lasse Quadratwurzel x' von x^2 ausrechnen. Dann ist $\gcd\{n, x' - x\}$ gleich p oder q mit Wahrscheinlichkeit $1/2$. Denn in $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$ gilt $x = (a, b)$ und $x' = (\pm a, \pm b)$. Wenn $x' - x$ in genau einer Koordinate Null ist, ist der ggT gleich p oder q. $\square$

Liefert gutes CPA Sicherheitsergebnis, aber unsicher unter CCA Angriff (bei RSA weder das eine, noch das andere).

Weitere Sicherheitsaspekte ähnlich wie bei RSA.

Blum-Goldwasser Kryptosystem

Ist ähnlich wie ein Streamcipher.
- Parameter: $n = pq$ mit $p = q = 3 \bmod 4$. $h > 0$ klein.
- Key stream $x_i = x_{i-1}^2 \bmod n$, x_0 zufälliges Quadrat in $\mathbb{Z}/n\mathbb{Z}$.
- Chiffretexte $c_i = m_i \oplus (x_i \bmod 2^h)$ für $0 \leq i \leq t$, zusammen mit x_{t+1}.
- Enschlüsseln über $x_i = x_{i+1}^{(p+1)/4}$.

Entschlüsseln klappt, da es genau eine Quadratwurzel von x_{i+1} gibt, die selbst ein Quadrat ist. Aus $x_{i+1} = x_{i-1}^4$ folgt $x_{i+1}^{(p+1)/4} = x_{i-1}^{p+1} = x_{i-1}^2 = x_i$.

Ist probabilistisch. IND-CPA sicher, wenn Faktorisieren schwer ist. Jedoch nicht sicher unter CCA. Kaum praktische Relevanz.

Goldwasser-Micali Kryptosystem

Schlüsselerzeugung:

- $n = pq$, $y \in \mathbb{Z}/n\mathbb{Z}$ kein Quadrat, aber mit $\left(\frac{y}{n}\right) = 1$.
- Öffentlicher Schlüssel (n, y).
- Privater Schlüssel: (p, q).

Verschlüsseln:

- $m = m_1 \dots m_t$ mit $m_i \in \{0, 1\}$.
- $x_i \in (\mathbb{Z}/n\mathbb{Z})^\times$ zufällig, $c_i = yx_i^2 \bmod n$ für $m_i = 1$, sonst $c_i = x_i^2 \bmod n$.
- Chiffretext ist $c = c_1 \dots c_t$.

Entschlüsseln:

- $e_i = \left(\frac{c_i}{p}\right)$, $m_i = 0$ für $e_i = 1$, sonst $m_i = 1$.
- Nachricht ist $m = m_1 \dots m_t$.

Goldwasser-Micali Sicherheit

Quadratisches Restproblem: Quadratische Reste von quadratischen Nichtresten mit Jacobisymbol eins unterscheiden.

Wenn man faktorisieren kann, dann kann man das quadratische Restproblem lösen. Die Umkehrung wird vermutet.

Die Sicherheit von Goldwasser-Micali Kryptosystem hängt vom quadratischen Restproblem ab.

Ist probabilistisch.
Keine praktische Bedeutung, da ineffizient.