Hoe je het cryptosysteem RSA soms kunt kraken

Benne de Weger
Technische Universiteit Eindhoven

1 Inleiding

1.1 RSA

RSA is een veelgebruikt cryptografisch systeem, bijvoorbeeld voor het beveiligen van internetverkeer. Het is een asymmetrisch systeem, d.w.z. versleutelen gebeurt met een publieke sleutel, ontsleutelen met de bijbehorende privé-sleutel. RSA maakt gebruik van eenvoudige getaltheorie en kan dan ook goed behandeld worden in een keuzeonderwerp Cryptografie bij Wiskunde D[1]. In de Vakantiecursus 2008 heeft Lenny Taelman [T] de basis van RSA besproken. Een recent Nederlandstalig boek dat een inleiding geeft tot de wiskunde van de asymmetrische cryptografie is [W1], zie ook [B, hst. 9], [Ke, par. 13.5].

De publieke sleutel – de naam zegt het al – mag iedereen weten. De privé-sleutel zal de eigenaar strict geheim willen houden. Zo kan iedereen iets versleutelen wat alleen de eigenaar van het sleutelpaar weer leesbaar kan maken. Omdat de publieke sleutel publiek is, maar wel samenhangt met de privé-sleutel, kun je je afvragen of de publieke sleutel niet genoeg informatie biedt om de privé-sleutel uit af te kunnen leiden. Dat zou natuurlijk niet moeten. Dat dat inderdaad praktisch onmogelijk is, is gebaseerd op het feit dat het ontbinden van grote getallen in priemfactoren een erkend (maar onbewezen) moeilijk probleem is. Verder moet de privé-sleutel op een veilige plek bewaard worden, bijvoorbeeld op smartcard, omdat goede smartcards niet zomaar uit te lezen zijn.

RSA heeft een goede reputatie als een zeer veilig systeem. Maar onder bepaalde omstandigheden is RSA wel degelijk te kraken. Gelukkig zijn deze omstandigheden makkelijk te vermijden, maar je wilt toch graag weten waar je aan toe bent. In deze cursus zullen enkele technieken om onder voorwaarden RSA te kunnen kraken geïllustreerd worden aan de hand van voorbeelden.

[1] Lesmateriaal over cryptografie voor Wiskunde D is o.m. te vinden bij het Steunpunt Wiskunde D van de TU/e, `http://www.win.tue.nl/wiskunded`.

1.2 Soms te kraken

Na kort RSA besproken te hebben in hoofdstuk 2, gaan we als eerste kijken naar zwakke sleutels. In Euclides van juni/juli 2009 is daar ook over geschreven [W2], en in deze cursus gaan we daar dieper op in. Met name de zogenaamde "privé-exponent" moet niet te klein gekozen worden. We laten in hoofdstuk 3 zien hoe kettingbreuken een rol spelen bij het kraken van RSA als deze privé-exponent te klein is, en in hoofdstuk 4 zien we hoe een geavanceerdere techniek daarin nog wat verder komt. Deze techniek is het zoeken van kleine nulpunten van polynomen in twee variabelen.

Vervolgens gaan we in hoofdstuk 5 uit van de omstandigheid dat een deel van de privé-sleutel gelekt is. Dat zou bijvoorbeeld kunnen door het stroomgebruik van de smartcard te meten terwijl die berekeningen doet met de privé-sleutel. Als een voldoende groot deel van de privé-sleutel (bijvoorbeeld van de privé-exponent, of van de priemfactoren van het te ontbinden getal) bekend is, kan de rest berekend worden. Ook hier gaat dat met het zoeken van kleine nulpunten van polynomen in twee variabelen.

Tenslotte kijken we in hoofdstuk 6 naar een situatie waarbij expres een foutje in de privé-sleutel geïntroduceerd wordt. Dat kan bijvoorbeeld door een smartcard even in de magnetron te leggen. We laten zien hoe de originele privé-sleutel kan lekken door de zo mishandelde smartcard een berekening te laten doen.

2 RSA

2.1 Een sleutelpaar

Een RSA-sleutelpaar wordt als volgt gemaakt.

- Maak twee willekeurige priemgetallen p en q, en bereken hun product $n = pq$. Deze n noemen we de *modulus*.
- Bereken $\phi(n) = (p-1)(q-1)$, en kies een getal e met $3 \leq e < \phi(n)$ dat geen delers met $\phi(n)$ gemeen heeft. Deze e noemen we de *publieke exponent*.
- Bereken het getal d, de *privé-exponent* genaamd, met de eigenschap dat

(1) $$\boxed{ed \equiv 1 \pmod{\phi(n)}}.$$

- De publieke sleutel bestaat uit de modulus n en de publieke exponent e, en de privé-sleutel uit de (publieke!) modulus n en de privé-exponent d.

Enkele opmerkingen hierbij:

- De priemgetallen moeten groot genoeg zijn, tenminste 150 cijfers, om factorisatie echt moeilijk te laten zijn. Het is niet moeilijk zulke grote priem-

getallen te maken. Zie [B, par. 8.2], [Ke, par. 13.2] of [W1, par. 3.1, 3.3]. Het is aan te bevelen om p en q evenveel cijfers te laten hebben.

- De methode om e onderling ondeelbaar met $\phi(n)$ te kiezen en om d te berekenen is het *uitgebreide algoritme van Euclides*. Zie [B, par. 3.4], [Ke, par. 7.4, 7.5] of [W1, par. 1.6].
- De getallen p, q en $\phi(n)$ zijn niet meer nodig als de publieke en de privé-sleutel eenmaal berekend zijn. Het is verstandig deze getallen te vernietigen, want de privé-exponent kan er eenvoudig uit berekend worden.
- In plaats van eerst e kiezen en dan d berekenen kun je net zo goed eerst d kiezen, en dan e berekenen zodat (1) geldt.

We gaan er van uit dat n en e publiek bekend zijn, dus ook bij een tegenstander die de privé-sleutel (d.w.z. de privé-exponent d) wil achterhalen. Zo'n tegenstander kan proberen om n in factoren te ontbinden, dan kan zij namelijk $\phi(n)$ uitrekenen en met behulp daarvan kan ze uit e ook d berekenen.

2.2 Versleutelen en ontsleutelen

RSA kan gehele getallen versleutelen als die > 1 en $< n - 1$ zijn. Als m zo'n getal is (de 'klare tekst', die niet in verkeerde handen mag vallen), dan wordt het geheimschrift c (de 'cijfertekst', die iedereen mag zien) berekend met behulp van de publieke sleutel (n, e):

(2) $\boxed{c \equiv m^e \pmod{n}}$.

De eigenaar van de privé-sleutel (n, d) is de enige die het geheimschrift c weer kan terugvertalen naar de klare tekst m. Dit ontsleutelen gaat als volgt:

(3) $\boxed{m \equiv c^d \pmod{n}}$.

Enkele opmerkingen hierbij:

- Als je een leesbare tekst bestaande uit letters een leestekens wilt versleutelen zul je die eerst moeten coderen in een getal, bv. met de ASCII-code. In de praktijk wordt RSA overigens vooral gebruikt voor het versleutelen van sleutels, en dat zijn toch al getallen.
- Er zijn efficiënte technieken om te machtsverheffen modulo n. Zie [B, par. 8.2], [Ke, par. 11.3] of [W1, par. 2.2].
- Er moet natuurlijk gegarandeerd zijn dat ontsleutelen inderdaad het versleutelen ongedaan maakt. Dus de uitkomst van de berekening (3) moet de oorspronkelijke m weer opleveren. Deze garantie wordt gegeven door de Stelling van Euler en het verband (1) tussen e en d. De Stelling van Euler zegt dat $a^{\phi(n)} \equiv 1 \pmod{n}$ (tenzij a en n een deler gemeen hebben,

wat in de praktijk niet zal voorkomen). Uit (1) volgt dan dat er een geheel getal k is met $ed = 1 + k\phi(n)$, en daarmee volgt met (2)

$$c^d \equiv (m^e)^d = m^{ed} = m^{1+k\phi(n)} = m \cdot \left(m^{\phi(n)}\right)^k \equiv m \cdot 1^k = m \pmod{n}.$$

2.3 RSA-CRT

Een veelgebruikte variant van RSA is RSA-CRT. Zie [W1, par. 4.6]. Het idee is om, in plaats van modulo n, afzonderlijk modulo p en modulo q te gaan werken. Dat kan alleen bij het ontsleutelen, omdat alleen de eigenaar van de privé-sleutel beschikt over p en q. Hierbij wordt als privé-sleutel niet d opgeslagen, maar de vijf getallen p, q, d_p, d_q, u, waarbij

$$d_p \equiv d \pmod{p-1}, \quad d_q \equiv d \pmod{q-1}, \quad pu \equiv 1 \pmod{q}.$$

Het optreden van de moduli $p-1$ en $q-1$ in de definities van d_p, d_q komt door de Stelling van Euler, die nu voor priem-moduli gebruikt moet worden, en dan luidt: $a^{p-1} \equiv 1 \pmod{p}$, m.a.w. $\phi(p) = p - 1$, en net zo voor q. Ontsleutelen gebeurt dan als volgt:

$$m_p \equiv c^{d_p} \pmod{p}, \quad m_q \equiv c^{d_q} \pmod{q},$$

en $m_p \pmod{p}$, $m_q \pmod{q}$ worden dan gecombineerd tot $m \pmod{n}$, door te berekenen

$$m \equiv m_p + pu(m_q - m_p) \pmod{n}.$$

Merk op dat inderdaad

$$m \equiv m_p \pmod{p}, \quad m \equiv m_p + 1(m_q - m_p) = m_q \pmod{q}.$$

Een algemene stelling die deze reconstructie van m modulo n uit m modulo delers van n beschrijft heet de "Chinese Reststelling"[2].

Het voordeel van RSA-CRT boven gewoon RSA is dat het machtsverheffen met veel kleinere getallen gebeurt (p en q zijn maar half zo lang als n), en dat levert aanzienlijke tijdswinst op, zelfs nu voor één ontsleuteling twéémaal een machtsverheffing moet worden uitgevoerd.

3 Zwakke sleutel: te kleine privé-exponent

3.1 Factoriseren, of raden

Een eerste gedachte is dat de sleutel niet te klein mag zijn. Voor de modulus betekent dat dat die zo groot moet zijn dat bekende factorisatietechnieken

[2] Engels: "Chinese Remainder Theorem", vandaar de naam RSA-CRT.

praktisch onuitvoerbaar zijn. De stand van zaken op dit terrein is dat in januari 2010 bekend werd gemaakt [Kl] dat getallen van 768 bits, dat is 232 cijfers, nu routinematig in factoren kunnen worden ontbonden (hoewel dat bepaald geen kleine klus is). Met een ruime marge genomen betekent dat dat toch ten minste 1024 bits (309 cijfers) cijfers nodig zijn voor de modulus. Voor kritische toepassingen wordt al geruime tijd geadviseerd om 2048 bits te gebruiken.

Ook de privé-exponent mag niet makkelijk te raden zijn. Dat betekent dat deze in ieder geval niet te klein mag zijn, anders zijn alle mogelijkheden gewoon uit te proberen. Een vuistregel is dat het vooralsnog praktisch ondoenlijk is om 2^{80} berekeningen te doen. We gaan er daarom van uit dat $d > 2^{80}$.

3.2 Een benaderingsprobleem

Maar dat is niet genoeg, zo merkte Mike Wiener in 1990 op. Hij zag hoe je d eenvoudig kunt berekenen uit de publieke sleutel als $d < n^{1/4}$. Hoe dat gaat laten we nu zien.

Veronderstel dat $d \approx n^\delta$, met $\delta < \frac{1}{4}$. Uit (1) volgt dat er een gehele k is zodat

(4) $\boxed{ed = 1 + k\phi(n)}$.

Nu is het zo dat als d essentieel kleiner is dan n, het getal e met grote waarschijnlijkheid ongeveer even veel cijfers heeft als n. Omdat

$$\phi(n) = (p-1)(q-1) = n - (p+q) + 1$$

even veel cijfers heeft als n, volgt uit (4) dat k ongeveer even veel cijfers als d zal hebben. We zeggen: ook $k \approx n^\delta$, en $e \approx n$. We kijken nog iets nauwkeuriger naar $\phi(n)$: dit getal ligt zelfs tamelijk dicht bij n, want $n - \phi(n) = p + q - 1$, en p en q hebben beide ongeveer de helft van het aantal cijfers van n, m.a.w. $p, q \approx n^{1/2}$, omdat n immers hun product is.

Uit (4) leiden we nu af:

$$\left|\frac{e}{n} - \frac{k}{d}\right| = \frac{1}{nd}\,|ed - kn| = \frac{1}{nd}\,|1 + k(\phi(n) - n)| \approx \frac{1}{nd}k(p+q),$$

en met de schattingen $d, k \approx n^\delta$ en $p, q \approx n^{1/2}$ vinden we

(5) $\boxed{\left|\frac{e}{n} - \frac{k}{d}\right| \approx n^{-1/2}}$.

De breuk $\frac{e}{n}$ bestaat helemaal uit publieke informatie. De breuk $\frac{k}{d}$ is bij de tegenstander niet bekend, maar we zien hier wel dat deze breuk dicht bij het bekende getal $\frac{e}{n}$ moet liggen. Op zich is dat niet bijzonder, want er zijn heel veel van zulke breuken. Maar de meeste daarvan hebben grote teller en noemer.

Het probleem om benaderingsbreuken $\frac{T}{N}$ te vinden van een gegeven reëel getal α is een klassiek probleem, waarbij de afstand $\left|\alpha - \frac{T}{N}\right|$ afgewogen wordt tegen de grootte van teller en noemer. Als je met breuken $\frac{T}{N}$ steeds dichter bij α wilt komen, zul je steeds grotere tellers en noemers moeten nemen. Dit leidt tot de volgende definitie.

Definitie. *De vereenvoudigde breuk* $\frac{T}{N}$ *wordt* beste benadering *van* α *genoemd als iedere breuk die dichter bij* α *ligt een noemer groter dan* N *heeft.*

3.3 Kettingbreuken

Er is een fraaie manier om de beste benaderingen van een gegeven getal α te vinden. Dat gaat met de kettingbreuk van α. Met de notatie $\lfloor x \rfloor$ bedoelen we het gehele deel van x, oftewel afronden naar beneden.

Laat $a_0 = \lfloor \alpha \rfloor$. Bereken dan $\alpha_1 = \dfrac{1}{\alpha - a_0}$, zodat $\alpha = a_0 + \dfrac{1}{\alpha_1}$. Nu is $\alpha_1 > 1$, en we nemen dan $a_1 = \lfloor \alpha_1 \rfloor$. Bereken dan $\alpha_2 = \dfrac{1}{\alpha_1 - a_1}$, zodat $\alpha = a_0 + \dfrac{1}{a_1 + \frac{1}{\alpha_2}}$. Zo gaan we door. We krijgen dan de *kettingbreuk* van α: een schrijfwijze van α met behulp van een rij gehele getallen $a_0, a_1, a_2, a_3, \ldots$:

$$\alpha = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{\cdots}}}}.$$

Omdat dit een typografische nachtmerrie wordt, is de notatie

$$\alpha = [a_0, a_1, a_2, a_3, \ldots]$$

gebruikelijker. Als α rationaal is, dan zal op een gegeven moment een α_k geheel worden, en dan stopt de kettingbreukontwikkeling. Een voorbeeld:
$\alpha = \dfrac{23}{16} = 1 + \dfrac{7}{16}$, dus $a_0 = 1$, en $\alpha_1 = \dfrac{16}{7} = 2 + \dfrac{2}{7}$, dus $a_1 = 2$, en
$\alpha_2 = \dfrac{7}{2} = 3 + \dfrac{1}{2}$, dus $a_2 = 3$, en $\alpha_3 = \dfrac{2}{1} = 2$, dus $a_3 = 2$, en het proces stopt.
Er volgt $\dfrac{23}{16} = 1 + \dfrac{1}{2 + \frac{1}{3 + \frac{1}{2}}}$, oftewel $\dfrac{23}{16} = [1, 2, 3, 2]$.

Een ander voorbeeld:
$\alpha = \pi = 3.14159\ldots = 3 + 0.14159\ldots$, dus $a_0 = 3$, en
$\alpha_1 = \dfrac{1}{0.14159\ldots} = 7.06251\ldots = 7 + 0.06251\ldots$, dus $a_1 = 7$, en
$\alpha_2 = \dfrac{1}{0.06251\ldots} = 15.99659\ldots = 15 + 0.99659\ldots$, dus $a_2 = 15$, en

$\alpha_3 = \dfrac{1}{0.99659\ldots} = 1.00341\ldots = 1 + 0.00341\ldots$, dus $a_3 = 1$, en

$\alpha_4 = \dfrac{1}{0.00341\ldots} = 292.63459\ldots = 292 + 0.63459\ldots$, dus $a_4 = 292$, enzovoorts.

Er volgt $\pi = 3 + \dfrac{1}{7 + \frac{1}{15 + \frac{1}{1 + \frac{1}{292 + \frac{1}{\ldots}}}}}$, oftewel $\pi = [3, 7, 15, 1, 292, \ldots]$.

Een eindige kettingbreuk levert een rationaal getal op[3]. Bijvoorbeeld, enig rekenen geeft: $[3, 7, 15, 1] = \dfrac{355}{113}$. Het valt op dat $\dfrac{355}{113} = 3.14159292\ldots$ een heel goede benadering van $\pi = 3.14159265\ldots$ is[4]. De theorie van de kettingbreuken verklaart dit. Eerst geven we de breuken die je krijgt door een kettingbreuk ergens af te kappen een naam.

Definitie. *Laat α de kettingbreukontwikkeling $\alpha = [a_0, a_1, a_2, \ldots]$ hebben. Noteer $\dfrac{T_i}{N_i} = [a_0, a_1, a_2, \ldots, a_i]$ voor de achter a_i afgekapte kettingbreuk, met $\dfrac{T_i}{N_i}$ de vereenvoudigde breuk. Deze breuken $\dfrac{T_i}{N_i}$ noemen we* convergenten *van α. De getallen $a_0, a_1, a_2, \ldots$ heten* wijzergetallen *van α.*

De centrale stelling uit de kettingbreuktheorie is de volgende. Zie bijvoorbeeld [B, hst. 14], [Ke, par. 7.6, 15.6, 15.7] voor bewijzen.

Stelling.

(a) *Voor de convergenten $\dfrac{T_i}{N_i}$ van α geldt*

$$\boxed{\frac{1}{(a_{i+1}+2)N_i^2} < \left|\alpha - \frac{T_i}{N_i}\right| < \frac{1}{a_{i+1}N_i^2}}. \tag{6}$$

(b) *Een convergent van α is altijd een beste benadering van α, en andersom: een beste benadering van α is altijd een convergent van α.*

Ongelijkheid (6) doet precies wat we willen: het afwegen van de afstand van de breuk $\dfrac{T_i}{N_i}$ tot α tegen de grootte van de noemer. Een direct gevolg van ongelijkheid (6) is dat $\lim\limits_{i\to\infty} \dfrac{T_i}{N_i} = \alpha$, in woorden: de rij van de convergenten van α convergeert inderdaad naar α.

Een andere interessante observatie op grond van (6) is dat als je een extra groot wijzergetal a_{i+1} tegenkomt, de eraan voorafgaande convergent $\dfrac{T_i}{N_i}$ een

[3] In het rationale geval is overigens het kettingbreukalgoritme essentieel hetzelde als het algoritme van Euclides voor het berekenen van de ggd van teller en noemer.

[4] De Chinees Zu Chongzhi wist dat al in de 5e eeuw. De eerste Europese vermelding is uit 1585, van Adriaan Anthoniszoon. Archimedes (3e eeuw v. Chr.) wist al wel dat $\dfrac{22}{7} = [3, 7]$ een goede benadering van π is.

extra goede benadering van α is. Dat zagen we bij $\dfrac{355}{113}$ als benadering van π al optreden, kennelijk vanwege het bijzonder grote wijzergetal 292 in de kettingbreuk van π.

Als α tot op voldoende cijfers achter de komma bekend is, zeg tot op b cijfers, dan zijn de convergenten met noemers tot aan ongeveer $b/2$ cijfers heel snel te berekenen. De te gebruiken formules zijn

$$\begin{cases} T_{-2}=0, & T_{-1}=1, & T_i = a_i T_{i-1} + T_{i-2} \\ N_{-2}=1, & N_{-1}=0, & N_i = a_i N_{i-1} + N_{i-2} \end{cases} \qquad \text{voor } i = 0,1,2,\ldots$$

(zie de genoemde literatuur voor de afleiding ervan). Gebruik makend van $a_i \geq 1$ voor $i \geq 1$ is meteen in te zien dat $N_i \geq F_{i+1}$, waarbij F_i het i-e Fibonacci-getal is. Dat impliceert dat de noemers N_i exponentieel hard groeien, want dat doen de Fibonacci-getallen F_i al.

Voor de toepassing is het onderstaande gevolg van deze stelling heel nuttig.

Stelling. *Als* $\dfrac{T}{N}$ *een breuk is die voldoet aan*

$$\boxed{\left|\alpha - \frac{T}{N}\right| < \frac{1}{2N^2}}, \tag{7}$$

dan is $\dfrac{T}{N}$ *een convergent van* α.

Bewijs: Laat $\dfrac{T'}{N'}$ een breuk zijn die dichter bij α ligt dan $\dfrac{T}{N}$. Dan is

$$\left|\frac{T'}{N'} - \frac{T}{N}\right| \leq \left|\frac{T'}{N'} - \alpha\right| + \left|\alpha - \frac{T}{N}\right| < 2\left|\alpha - \frac{T}{N}\right| < \frac{1}{N^2},$$

en omdat $T'N - TN'$ geheel is en niet 0, volgt

$$1 \leq |T'N - TN'| = NN'\left|\frac{T'}{N'} - \frac{T}{N}\right| < NN'\frac{1}{N^2} = \frac{N'}{N}.$$

Hier staat dat $N' > N$, dus iedere breuk die dichter bij α ligt dan $\dfrac{T}{N}$ heeft een grotere noemer. Dus is $\dfrac{T}{N}$ een beste benadering van α, dus een convergent. □

3.4 Het kraken

Nu gaan we deze theorie toepassen op ons probleem om de privé-exponent d te achterhalen als deze erg klein is. Dan hebben we uitdrukking (5), en hierbij is $\alpha = \dfrac{e}{n}$ bekend. Ongelijkheid (7) met $\dfrac{T}{N} = \dfrac{k}{d}$ zegt dat $\dfrac{k}{d}$ een convergent is als

$n^{-1/2} < \dfrac{1}{2d^2}$. Als we de factor 2 voor het gemak even verwaarlozen (we zijn op wel meer plekken een beetje slordig geweest in de afschattingen), dan zien we dat dit het geval is als $d < n^{1/4}$, dus als $\delta < 1/4$. Dat betekent dan dat we alleen maar de convergenten van de kettingbreuk van $\dfrac{e}{n}$ hoeven uit te rekenen tot de noemers groter worden dan $n^{1/4}$. Dat zijn er maar een paar, want ze groeien exponentieel. Die kandidaten voor $\dfrac{k}{d}$ testen we dan of ze voldoen. En mocht δ echt een stukje kleiner zijn dan $\dfrac{1}{4}$, dan zegt (6) bovendien dat we een extreem groot wijzergetal zullen tegenkomen, en weten we doorgaans meteen welke convergent we moeten hebben.

3.5 Een voorbeeld

U zult begrijpen dat we geen voorbeeld nemen met een modulus van 300 cijfers, maar een veel kleinere. Voor de methode maakt dat niet uit, voor de rekentijd wel, maar met moderne computers en de juiste software blijft het binnen een fractie van een seconde.

Laten we $n = 205320043521075746592613$ en $e = 70760135995620281241019$ nemen. Stel we vermoeden dat de bijbehorende d echt een stukje kleiner dan $n^{1/4}$ gekozen was. Dan berekenen we de kettingbreuk van $\dfrac{e}{n}$, deze blijkt te beginnen als $[0, 2, 1, 9, 6, 54, 5911, 1, 5, 1, 1, \ldots]$. Het grote wijzergetal 5911 suggereert om $\dfrac{k}{d} = [0, 2, 1, 9, 6, 54]$ uit te proberen, dat is $k = 3304$, $d = 9587$. Dat uittesten van een kandidaat $\dfrac{k}{d}$ gaat als volgt: in de gelijkheid (4) weten we nu alle getallen behalve $\phi(n)$, en die kunnen we er dus uit oplossen:

$$\begin{aligned}\phi(n) = \frac{ed-1}{k} &= (70760135995620281241019 \cdot 9587 - 1)/3304 \\ &= 205320043519979308794688.\end{aligned}$$

Dit is een geheel getal, en dat zegt op zich al iets, want bij 'verkeerde' convergenten zal hier vaak een niet-geheel getal uitkomen. Nu weten we

$$p + q = n + 1 - \phi(n) = 1096437797926,$$

en ook wisten we

$$pq = n = 205320043521075746592613.$$

Uit deze twee vergelijkingen kunnen we (met de *abc*-formule) kijken of we gehele p en q vinden. Dat gaat inderdaad goed: $p = 239635170197$ en $q = 856802627729$, en deze kloppen.

3.6 Priemfactoren dicht bij elkaar

We gaan nog iets nauwkeuriger kijken naar n als benadering van $\phi(n)$. De fout $n - \phi(n)$ is immers de term die in (5) de hoofdverantwoordelijke is voor de afschatting $n^{-1/2}$. Dit kunnen we ietsje beter doen, door niet n maar $n + 1 - 2\sqrt{n}$ te nemen als benadering van $\phi(n)$, en dus niet de kettingbreuk van $\frac{e}{n}$ te bekijken, maar die van $\frac{e}{n+1-2\sqrt{n}}$. Immers,

$$n+1-2\sqrt{n}-\phi(n) = p+q-2\sqrt{pq} = (\sqrt{p}-\sqrt{q})^2 = \frac{(p-q)^2}{\left(\sqrt{p}+\sqrt{q}\right)^2} \approx \frac{(p-q)^2}{4\sqrt{n}}.$$

Deze afschatting komt in de plaats van $n - \phi(n) = p + q - 1 \approx 2\sqrt{n}$. En in plaats van (5) vinden we nu

$$\left|\frac{e}{n+1-2\sqrt{n}} - \frac{k}{d}\right| \approx \frac{(p-q)^2}{n^{3/2}}.$$

Als p en q dicht bij elkaar liggen, zeg $|p - q| \approx n^\beta$ voor een $\frac{1}{4} < \beta < \frac{1}{2}$, betekent dit dat de eis van $d > n^{1/4}$ navenant verscherpt zal moeten worden tot $d > n^{3/4-\beta}$ (als $\beta \leq \frac{1}{4}$ dan is er een veel simpeler methode, zie [W2, deel 1]). Als de priemgetallen onafhankelijk van elkaar willekeurig worden gekozen is de kans dat dit optreedt overigens astronomisch klein.

4 Zwakke sleutel: te kleine privé-exponent - geavanceerd

4.1 Een polynoom met een klein nulpunt

Je kunt je afvragen waarom je eigenlijk een kleine privé-exponent zou willen: kun je hem niet altijd gewoon groot nemen? Ja, dat kan. Maar een kleine exponent kan wel voordelen hebben. De rekentijd van een machtsverheffing modulo n hangt sterk af van de grootte van de exponent: afhankelijk van de implementatie kan de rekentijd omhoog gaan met de derde macht van het aantal cijfers van de exponent. Als je het ontsleutelen op een langzame (want goedkope) processor als die van een smartcard moet doen, kan een kleine privé-exponent d dus best nuttig zijn. Hier is dus na Wiener's resultaat meer onderzoek naar gedaan. Daar is uitgekomen dat je nog meer moet oppassen: zelfs voor $\delta < 0.292$ (dus $d < n^{0.292}$) blijkt RSA onveilig te zijn. We zullen de gebruikte techniek aan de hand van een voorbeeld illustreren. Dit is werk van Dan Boneh en Glenn Durfee uit 2000, gebaseerd op ideeen van Don Coppersmith.

We nemen dezelfde n, e als in paragraaf 3.5: $n = 20532004352107574659261 3$ en $e = 70760135995620281241019$. Dat de bijbehorende d al gevonden was moet

u even vergeten. Het gaat erom dat we diezelfde d nu met een heel andere techniek ook kunnen vinden. Dat die nieuwe techniek ook voor grotere d dan $n^{1/4}$ werkt is aangetoond, maar dat kunnen we nu niet laten zien.

Kijk weer naar vergelijking (4), en wel in de vorm

$$ed = 1 + k(n + 1 - (p + q)).$$

Deze vergelijking nemen we nu modulo e. Het prettige daarvan is dat de onbekende d dan opeens er niet meer toe doet. De onbekenden zijn nu k en $p + q$, en die zijn samen een nulpunt $\pmod e$ van het polynoom

$$\boxed{f(x, y) = xy - (n + 1)x - 1},$$

want $f(k, p + q) \equiv 0 \pmod e$. Wat hierbij prettig is is dat zowel k als $p + q$ relatief klein zijn t.o.v. de grootste coëfficiënt van het polynoom f. In ons voorbeeld nemen we als bovengrens voor k nu $X = 10^4$, en als bovengrens voor $p + q$ nemen we $Y = 10^{12}$.

Het idee van Don Coppersmith was nu tweeledig:

- als een polynoom $g(x, y)$ een nulpunt $(x_0, y_0) \pmod M$ heeft, dus M is een deler van $g(x_0, y_0)$, en $g(x, y)$ heeft *zulke kleine* coëfficienten en het nulpunt (x_0, y_0) is *zo klein* t.o.v. de *zo grote* modulus M dat $|g(x_0, y_0)| < M$ waar blijkt te zijn, dan moet (x_0, y_0) dus *zelfs zonder de modulus* al een nulpunt van $g(x, y)$ zijn, oftewel dan is $g(x_0, y_0) = 0$;
- er zijn trucs om uit een polynoom $f(x, y)$ met *grote* coëfficienten een polynoom met *kleinere* coëfficienten te maken dat *hetzelfde nulpunt* $\pmod M$ heeft.

Het eerste idee kan precies gemaakt worden. De volgende stelling is een speciaal geval van een resultaat van Nick Howgrave-Graham.

Stelling (Howgrave-Graham)**.** *Laat $h(x, y)$ een polynoom zijn met maximaal 4 termen en gehele coëfficiënten. Laat de gehele getallen x_0, y_0 voldoen aan $h(x_0, y_0) \equiv 0 \pmod M$, en aan $|x_0| \leq X$, $|y_0| \leq Y$, voor zekere gegeven M, X, Y. Als nu alle coëfficiënten van $h(xX, yY)$ in absolute waarde $< \frac{1}{2}M$ zijn, dan is $h(x_0, y_0) = 0$.*

Bewijs: Laat $h(x, y) = \sum b_{i,j} x^i y^j$. De coëfficiënten van $h(xX, yY)$ zijn dan $b_{i,j} X^i Y^j$, en dus

$$\begin{aligned} |h(x_0, y_0)| &= \left|\sum b_{i,j} x_0^i y_0^j\right| \leq \sqrt{\sum \left(b_{i,j} x_0^i y_0^j\right)^2} \\ &= \sqrt{\sum \left(b_{i,j} X^i Y^j\right)^2 \left(\frac{x_0}{X}\right)^{2i} \left(\frac{y_0}{Y}\right)^{2j}} < \sqrt{4 \cdot \left(\tfrac{1}{2}M\right)^2} = M, \end{aligned}$$

en omdat $h(x_0, y_0)$ geheel is, en deelbaar door M, moet het wel 0 zijn. $\square$

4.2 Polynomen met kleine coëfficiënten

Voor het tweede idee van Coppersmith zijn verschillende strategieën bedacht. Die kunnen we hier niet behandelen. Zie [H] en [J] voor een overzicht. Aan de hand van ons concrete voorbeeld kunnen we wel een idee geven hoe het werkt.

Als modulus kiezen we $M = e^2$, en we gaan eerst maar eens een aantal polynomen opschrijven waarvan we zeker weten dat $(x_0, y_0) = (k, p+q)$ een nulpunt van $f(x,y) \pmod{e^2}$ is, zonder dat we k en $p+q$ kennen. Namelijk:

$$\begin{array}{l|l|l} g_{0,0}(x,y) = e^2 & g_{2,0}(x,y) = e^2x^2 & h_{1,0}(x,y) = e^2y \\ g_{1,0}(x,y) = e^2x & g_{1,1}(x,y) = e\ x\ f(x,y) & h_{1,1}(x,y) = e\ yf(x,y) \\ g_{0,1}(x,y) = e\quad f(x,y) & g_{0,2}(x,y) = \quad\ f(x,y)^2 & h_{1,2}(x,y) = \quad yf(x,y)^2 \end{array}$$

De volgende tabel bevat de coëfficiënten van deze polynomen:

	1	x	xy	x^2	x^2y	x^2y^2	y	xy^2	x^2y^3
$g_{0,0}$	e^2								
$g_{1,0}$		e^2							
$g_{0,1}$	$-e$	$-e(n+1)$	e						
$g_{2,0}$				e^2					
$g_{1,1}$		$-e$		$-e(n+1)$	e				
$g_{0,2}$	1	$2(n+1)$	-2	$(n+1)^2$	$-2(n+1)$	1			
$h_{1,0}$							e^2		
$h_{1,1}$			$-e(n+1)$				$-e$	e	
$h_{1,2}$			$2(n+1)$		$(n+1)^2$	$-2(n+1)$		-2	1

Deze polynomen zijn natuurlijk niet zomaar gekozen, daar zit een strategie achter, die o.a. tot de bovenstaande driehoekige vorm leidt.

Het idee is nu om lineaire combinaties van deze polynomen te kiezen, zodat de coëfficiënten zo klein mogelijk worden. De theorie hierachter is die van *roosterbasisreductie*, en het algoritme dat dit efficiënt doet is het zogenaamde LLL-algoritme. Dat kan gezien worden als een generalisatie van het algoritme van Euclides. We gaan niet op de details in, maar vermelden het resultaat: we vinden o.a.

$$\begin{aligned} p_1(x,y) &= 91910569g_{2,0}(x,y) + 63350896g_{1,1}(x,y) + 10916416g_{0,2}(x,y) \\ &= 10916416 + 23938342240568299824x \\ &\quad + 131234516261238241782836623350 09x^2 - 21832832xy \\ &\quad - 23938342240568299824x^2y + 10916416x^2y^2, \\ p_2(x,y) &= -964355g_{1,0}(x,y) - 332346g_{0,1}(x,y) \\ &\quad + 17286884292171605419691 79g_{2,0}(x,y) \\ &\quad + 17872910931668028426742 68g_{1,1}(x,y) \\ &\quad + 41064008704642407047419 6g_{0,2}(x,y) + h_{1,2} \\ &= 23517258797687464413398174770 \\ &\quad - 21457461892318384535056334741370599284123x \\ &\quad + 35649776482295032433498368412689403 47x^2 + y \\ &\quad - 23517258797687468685975463738xy \\ &\quad + 39027768456154986743754 00x^2y - 2xy^2 + 4272577288968x^2y^2 + x^2y^3. \end{aligned}$$

Duidelijk is dat inderdaad $p_1(x_0, y_0) \equiv p_2(x_0, y_0) \equiv 0 \pmod{e^2}$. De grootste coëfficiënt van $p_1(xX, yY)$ is $2.393\ldots\cdot 10^{39}$, die van $p_2(xX, yY)$ is $4.272\ldots\cdot 10^{44}$, en die zijn inderdaad kleiner dan $\frac{1}{2}e^2 = 2.503\ldots\cdot 10^{45}$. Dus weten we nu op grond van de stelling van Howgrave-Graham dat $x_0 = k$ en $y_0 = p + q$ van beide polynomen echte nulpunten zijn: $p_1(x_0, y_0) = p_2(x_0, y_0) = 0$.

4.3 Het vinden van het nulpunt

Maar hoe vinden we daaruit de waarden van x_0 en y_0?

Wat we doen is polynomen $a_1(x, y), a_2(x, y)$ zoeken zodat uit $a_1p_1 + a_2p_2$ de variabele x helemaal verdwenen is. Dat is niet zo heel moeilijk. De algemene methode is het berekenen van een zogenaamde *resultante* van p_1 en p_2. In dit geval gaat dat als volgt (in feite ook weer een generalisatie van het algoritme van Euclides). Schrijven we

$$p_1(x, y) = b_1(y) + c_1(y)x + d_1(y)x^2, \quad p_2(x, y) = b_2(y) + c_2(y)x + d_2(y)x^2,$$

dan zien we dat $p_3 = d_1p_2 - d_2p_1$ alvast geen x^2 meer bevat. En op dezelfde manier is een combinatie p_4 van p_2 en xp_3 te vinden die ook geen x^2 meer bevat. Tenslotte is op dezelfde manier een combinatie p_5 van p_3 en p_4 te vinden die ook geen x meer bevat. Het blijkt dat

$$\begin{aligned} a_1 &= -d_2(b_1d_2 - b_2d_1) + (c_2 + d_2x)(c_1d_2 - c_2d_1), \\ a_2 &= d_1(b_1d_2 - b_2d_1) - (c_1 + d_1x)(c_1d_2 - c_2d_1) \end{aligned}$$

leidt tot:

$$p_5 = a_1p_1 + a_2p_2 = (c_1d_2 - c_2d_1)(b_1c_2 - b_2c_1) - (b_1d_2 - b_2d_1)^2,$$

een uitdrukking die nog steeds hetzelfde nulpunt heeft, maar niet meer van x afhangt, alleen nog van y. Hier is dat

$$\begin{aligned} p_5(y) = a_1(x, y)p_1(x, y) + a_2(x, y)p_2(x, y) &= 45186470870881861783781526386044\backslash \\ &\quad 84590883818561242978066441050834049358921519373818309427961837356 \\ &\quad - 82424139255971828392316121222520171443544911347248067754516954088 82\backslash \\ &\quad 778504226715144612y + 37587239062664428399061109632404890728302 3250786529396290283624\backslash \\ &\quad 2513594131y^2. \end{aligned}$$

Dit polynoom heeft uiteraard y_0 als nulpunt. Het nulpunt vinden van een polynoom van slechts één variabele is kinderspel. Hier hebben we een kwadratisch polynoom, en zouden we de *abc*-formule kunnen gebruiken. Maar ook numerieke nulpuntzoekmethoden zijn bruikbaar. Hoe dan ook, hier vinden we

$$\begin{aligned} p_5(y) = 37587239062664428399061109632404890728302325078652939629028 3624\backslash \\ 2513594131 \cdot (-1096437797926 + y)^2, \end{aligned}$$

en we zien dat $y_0 = p + q = 1096437797926$. Dat geeft voldoende informatie (samen met $pq = n = 205320043521075746592613$) om p en q te berekenen. En dan is het vinden van d niet moeilijk meer.

Al deze berekeningen lijken erg ingewikkeld, door de erg grote getallen die er optreden. Maar daar moet u even doorheen prikken. In de praktijk wil je deze berekeningen doen met veel grotere getallen (bijv. n van zo'n 300 cijfers). Met een computer-algebra-pakket als Mathematica is dat niet moeilijker dan rekenen met getallen van 3 cijfers. De methode blijft exact hetzelfde, de optredende polynomen hebben dezelfde vorm, alleen grotere coëfficiënten.

5 Gedeeltelijk gelekte sleutel: factoriseren met een hint

5.1 Een polynoom met een klein nulpunt

Onder bepaalde omstandigheden kunnen delen van privé-sleutels lekken. Een techniek daarvoor is het nauwkeurig meten van het stroomgebruik van een smartcard op het moment dat die bezig is met een berekening met die privé-sleutel. Bijvoorbeeld, het programma zal wellicht iets meer werk moeten doen voor het verwerken van een bit dat 1 is dan voor een 0, en dan iets meer stroom verbruiken.

Het is o.a. daarom interessant geworden om te kijken hoeveel van de privé-sleutel moet lekken om de rest te kunnen berekenen. De techniek die behandeld is in hoofdstuk 4 kan worden aangepast voor het geval dat de privé-exponent niet klein is, maar voor een deel bekend.

We zullen nu echter een wat ander voorbeeld bekijken, waarin een dergelijke techniek wordt gebruikt om $n = pq$ in factoren te ontbinden als van de priemfactoren p en q het nodige gelekt is. De techniek werkt in principe als de bovenste helft van de cijfers van p bekend is (en dus ook van q), of de onderste helft. We demonstreren de techniek nu aan dezelfde $n = 205320043521075746592613$ als hierboven, waarbij we als extra informatie (de 'hint') hebben dat 2396352 de bovenste 7 (van de 12) cijfers van p zijn, en 8568026 die van q.

We schrijven $p = p_0 + x_0$, $q = q_0 + y_0$, met $p_0 = 239635200000$, $q_0 = 856802600000$, en x_0, y_0 onbekend met beide $|x_0|, |y_0| < X = 10^5$. Nu is (x_0, y_0) een klein nulpunt van het polynoom

$$\boxed{f^*(x,y) = (p_0 + x)(q_0 + y) - n = (p_0 q_0 - n) + q_0 x + p_0 y + xy}.$$

Om technische redenen willen we graag een polynoom met constante term 1. We nemen nu een modulus $M = 10^{37}$, en berekenen c zodanig dat $c(p_0 q_0 - n) \equiv 1 \pmod{M}$, en $a \equiv cq_0 \pmod{M}$, $b = cp_0 \pmod{M}$. Dat geeft

$$\begin{aligned} c &= 47735829292983994054711929151687223 23, \\ a &= 84847864461723148181407250244398000 00, \\ b &= 90078012099704084650398076165696000 00. \end{aligned}$$

Nu nemen we

$$\boxed{f(x,y) = 1 + ax + by + cxy},$$

want $f(x_0, y_0) \equiv cf^*(x_0, y_0) = 0 \pmod{M}$.

5.2 Polynomen met kleine coëfficiënten

De strategie om een aantal polynomen op te schrijven waarvan we zeker weten dat (x_0, y_0) een nulpunt $\pmod{M}$ is, geeft nu:

$$\begin{array}{l|l|l} g_{0,0}(x,y) = X^4 \quad f(x,y) & g_{1,1}(x,y) = X^2xyf(x,y) & h_{2,1}(x,y) = Mx^2y \\ g_{1,0}(x,y) = X^3xf(x,y) & h_{2,0}(x,y) = M\ x^2 & h_{1,2}(x,y) = Mxy^2 \\ g_{0,1}(x,y) = X^3yf(x,y) & h_{0,2}(x,y) = M\ y^2 & h_{2,2}(x,y) = Mx^2y^2 \end{array}$$

De volgende tabel bevat de coëfficiënten van deze polynomen:

	1	x	y	xy	x^2	y^2	x^2y	xy^2	x^2y^2
$g_{0,0}$	X^4	aX^4	bX^4	cX^4					
$g_{1,0}$		X^3		bX^3	aX^3		cX^3		
$g_{0,1}$			X^3	aX^3		bX^3		cX^3	
$g_{1,1}$				X^2			aX^2	bX^2	cX^2
$h_{2,0}$					M				
$h_{0,2}$						M			
$h_{2,1}$							M		
$h_{1,2}$								M	
$h_{2,2}$									M

De techniek van roosterbasisreductie geeft nu een aantal polynomen met kleine coëfficiënten die lineaire combinaties zijn van bovenstaande, we selecteren daar één uit:

$$\begin{aligned} p(x,y) = {} & 34853558839845024g_{0,0}(x,y) \\ & - 29572500364518632902878965837831683033644466122567946256727g_{1,0}(x,y) \\ & - 31395392948933083309478264368411070333007955918672712382199g_{0,1}(x,y) \\ & + 532766409130722573703404011658772635023983887365524928038900684990\backslash \\ & \quad 37862430002358633401921874128924058224800200g_{1,1}(x,y) \\ & + 25091635027229353579107563644214257793162903900355703204809953177486609166h_{2,0}(x,y) \\ & + 28280345859289585817611178346362583484505908334310306340972059243462400717h_{0,2}(x,y) \\ & - 45204092071682490824457259466261991011752205482200501124149230054362384877602280830676256289704990531710269405 9h_{2,1}(x,y) \\ & - 47990539047993124717584289016595503364691998143993633257014623840121269825177630206007451121397307331807043076 3h_{1,2}(x,y) \\ & - 25432046359300241870051436398798758002499757236143986334379471470207271708999015141366521823064858514780683548 9h_{2,2}(x,y) \\ = {} & 34853558839845024000000000000000000000 \\ & - 24367047946256727000000000000000x - 156534600000000000000000000x^2 \\ & - 10495163271238219900000000000000y + 19817767412892405822480020 0xy \\ & - 70104735280720400000000x^2y - 48455040000000000000000000y^2 \\ & + 23503326169393920000000xy^2 + 124858545954864600x^2y^2 \end{aligned}$$

Nu is weer duidelijk dat inderdaad $p(x_0, y_0) \equiv 0 \pmod{M}$. De grootste coëfficiënt van $p(xX, yX)$ is $2.350\ldots \cdot 10^{37}$, en die is weliswaar net ietsje groter dan $\frac{1}{2}M = 5.000\ldots \cdot 10^{36}$, zodat aan de voorwaarden van de stelling van Howgrave-Graham niet is voldaan, maar toch blijkt dit polynoom te werken.

5.3 Het vinden van het nulpunt

De reden dat we nu maar één polynoom p hoeven te selecteren, is dat we al een ander polynoom hebben waarvan we zeker weten dat (x_0, y_0) een echt (niet-modulair) nulpunt is, namelijk f^*. We kunnen nu weer een lineaire combinatie van p en f^* zoeken die niet meer van x afhangt. Dat is hier ietsje simpeler dan in paragraaf 4.3, omdat één van de twee polynomen nu geen x^2 heeft. We schrijven dus

$$p(x, y) = b_1(y) + c_1(y)x + d_1(y)x^2, \quad f^*(x, y) = b_2(y) + c_2(y)x,$$

dan zien we dat $p_1 = d_1 x f^* - c_2 p$ geen x^2 meer bevat. De juiste combinatie p_2 van p_1 en f^* bevat dan ook geen x meer. Het blijkt dat

$$a_1 = -c_2^2, \quad a_2 = c_2 d_1 x - (b_2 d_1 - c_1 c_2)$$

leidt tot:

$$p_2 = a_1 p + a_2 f^* = c_2(b_2 c_1 - b_1 c_2) - b_2^2 d_1,$$

een uitdrukking die nog steeds hetzelfde nulpunt heeft, maar niet meer van x afhangt, alleen nog van y. Hier is dat

$$\begin{aligned} p_2(y) &= a_1(x, y)p(x, y) + a_2(x, y)f^*(x, y) \\ &= 10000000000000000000000000000000000000000 \cdot (28971678074274940917065 4 \\ &\quad - 79169533269942837 71y - 80471757777041y^2 - 409785919y^3 + 717y^4). \end{aligned}$$

Dit polynoom heeft $y_0 = 27729$ als nulpunt. Daarmee hebben we $q = q_0 + y_0 = 856802627729$ gevonden, en volgt meteen $p = n/q = 239635170197$, en is n in factoren ontbonden.

6 Mishandelde sleutel: de magnetron-aanval

6.1 Smartcard in de magnetron

Tenslotte beschrijven we een heel ander soort aanval op RSA. We gaan uit van de situatie dat de tegenstander een smartcard in handen heeft gekregen, die zij berekeningen kan laten doen met de privé-sleutel. Met andere woorden, als zij aan de smartcard een geheimschrift c aanbiedt, dan geeft de smartcard het ontsleutelde getal $c^d \pmod{n}$ terug, maar zonder informatie over de privé-sleutel te lekken. De tegenstander wil de privé-sleutel achterhalen. We gaan er

ook van uit dat de privé-sleutel in CRT-vorm op de smartcard ligt opgeslagen, dus de getallen p, q, d_p, d_q, u zoals in paragraaf 2.3 beschreven.

De tegenstander legt de smartcard even in de magnetron. De straling zal de inhoud van het geheugen op de chip kunnen beschadigen. We gaan er van uit dat die beschadiging alléén optreedt in het veld waar, bijvoorbeeld, d_p zich bevindt. Na de mishandeling bevat de smartcard dan d'_p in plaats van d_p, maar alle andere getallen, p, q, d_q, u, zijn nog intact.

6.2 Foute ontsleuteling lekt de sleutel

De tegenstander kiest nu een willekeurig getal m, en versleutelt dit (niet op de smartcard, maar gewoon op een PC) met de publieke sleutel tot $c \equiv m^e \pmod{n}$. Dan biedt zij c aan aan de mishandelde smartcard. Die zal nu een getal m' berekenen, als volgt:

$$m'_p \equiv c^{d'_p} \pmod{p}, \quad m_q \equiv c^{d_q} \pmod{q},$$
$$m' \equiv m'_p + pu(m_q - m'_p) \pmod{n}.$$

Alléén dit getal m' zal de smartcard teruggeven, niet de tussenresultaten of de gebruikte priemgetallen. Merk nu op dat $m' \equiv m'_p \pmod{p}$ in plaats van $m \equiv m_p \pmod{p}$, terwijl $m' \equiv m'_p + 1(m_q - m'_p) = m_q \pmod{q}$ nog wel goed is. Dus

$$m' \not\equiv m \pmod{p}, \quad m' \equiv m \pmod{q}.$$

Maar dat betekent dat $m' - m$ wel deelbaar is door q, maar niet door p. En dan is q eenvoudig te berekenen als $\text{ggd}(m' - m, n)$, en is RSA gekraakt.

6.3 Een voorbeeld

Laat $p = 97, q = 127$, dus $n = 12319$. We nemen $e = 19$, dan is $d_p = 91, d_q = 73, u = 55$. Met $m = 3507$ krijgen we $c \equiv 3507^{19} \equiv 10497 \pmod{12319}$, en een correcte ontsleuteling van c zou op de smartcard zo gaan:

$$m_p \equiv 10497^{91} \equiv 15 \pmod{97}, \quad m_q \equiv 10497^{73} \equiv 78 \pmod{127},$$
$$m \equiv 15 + 97 \cdot 55 \cdot (78 - 15) \equiv 3507 \pmod{12319}.$$

Maar stel nu dat $d_p = 91$ was veranderd in $d'_p = 90$, maar één bitje verschil. Dan zou de smartcard berekend hebben:

$$m'_p \equiv 10497^{90} \equiv 70 \pmod{97}, \quad m_q \equiv 10497^{73} \equiv 78 \pmod{127},$$
$$m' \equiv 70 + 97 \cdot 55 \cdot (78 - 70) \equiv 5793 \pmod{12319}.$$

De tegenstander ziet dat dit niet gelijk is aan $m = 3507$. Nu berekent zij

$$\text{ggd}(5793 - 3507, 12319) = \text{ggd}(2286, 12319) = 127,$$

en heeft ze een priemfactor van n gevonden, en daarmee de beschikking over de volledige privé-sleutel gekregen.

7 Tenslotte

In de standaard-tekstboeken over RSA wordt vaak uitgebreid de moeilijkheid van het factoriseren van grote getallen besproken, en maar weinig tot geen aandacht gegeven aan andere manieren om RSA te kraken. Dat is ook wel te begrijpen, want de methoden die we hierboven besproken hebben werken alleen in heel specifieke omstandigheden. Een te kleine privé-exponent is makkelijk te vermijden. De nuttige efficiëntiewinst kan ook wel op andere manieren behaald worden, bijvoorbeeld door RSA-CRT te gebruiken, in combinatie met een kleine publieke exponent e. In de praktijk wordt vaak $e = 65537$ genomen, erg klein dus, zonder verlies van veiligheid. En smartcards zijn goed te beveiligen tegen het laten lekken van sleutels door stroommetingen of door fouten in de opgeslagen sleutels. Niettemin tonen deze methoden aan dat je er bij het veilig toepassen van RSA niet bent door de modulus maar groot genoeg te kiezen. Het terrein van de cryptanalyse van RSA is de afgelopen 20 jaar een vruchtbaar onderzoeksterrein geweest, ook in Nederland, zoals o.m. blijkt uit het proefschrift [J] van Ellen Jochemsz.

Referenties

[B] Frits Beukers, *Getaltheorie voor beginners*, Epsilon Uitgaven, deel 42, 4e druk, 2008.

[H] M. Jason Hinek, *Cryptanalysis of RSA and its variants*, CRC Press, 2009.

[J] Ellen Jochemsz, *Cryptanalysis of RSA variants using small roots of polynomials*, Proefschrift, TU Eindhoven, 2007.

[Ke] Frans Keune, *Getallen – van natuurlijk naar imaginair*, Epsilon Uitgaven, deel 65, 2009.

[Kl] Thorsten Kleinjung et al., *Factorization of a 768-bit RSA modulus*, Cryptology ePrint Archive, Report 2010/006, `http://eprint.iacr.org/2010/006.pdf`.

[T] Lenny Taelman, *RSA*, in: *Wiskunde en profil – Het gezicht van de wiskunde*, Vakantiecursus 2008, CWI Syllabus 58, 2008, blz. 67–74.

[W1] Benne de Weger, *Elementaire getaltheorie en asymmetrische cryptografie*, Epsilon Uitgaven, deel 63, 2009, bijbehorende educatieve software op `http://www.win.tue.nl/~bdeweger/MCR/`.

[W2] Benne de Weger, *Zwakke sleutels bij het RSA-cryptosysteem*, Euclides 84, no. 7, juni 2009, blz. 256–260, en no. 8, juli 2009, blz. 306–309.