CWI

Probabilistische aspecten bij public-key crypto (i.h.b. RSA)

Herman te Riele, CWI Amsterdam

Nationale Wiskunde Dagen

Noordwijkerhout, 31 januari 2015

CWI

Overzicht

- Binair exponentiëren
- RSA
- Factorisatie-algoritmen en rekentijd-schattingen
- Genereren van RSA-sleutels
- Valkuil voor RSA
- Mogelijke aanvallen op RSA
- De cycling attack
- Stelling van Pocklington voor priemgetallen
- Maurer's generator van priemgetallen
- Conclusies

CWI

Binair modulair exponentiëren

Cruciale rekentruc!

Gegeven natuurlijke getallen m, n, e, bereken $m^e \bmod n$. Schrijf de exponent e als een binair getal $e = b_0 b_1 \ldots b_l$ met $b_0 = 1$.
Bereken nu achtereenvolgens (exponenten binair geschreven):

$c_0 := m^{b_0} \bmod n = m \bmod n$,

$c_1 := m^{b_0 b_1} \bmod n = c_0 \times c_0 \times m^{b_1} \bmod n$,

$c_2 := m^{b_0 b_1 b_2} \bmod n = c_1 \times c_1 \times m^{b_2} \bmod n$,

$\ldots$,

$c_l := m^{b_0 b_1 b_2 \ldots b_l} \bmod n = c_{l-1} \times c_{l-1} \times m_l^b$.

Aantal rekenstappen: $l = \lfloor {}^2\log(e) \rfloor$.

CWI

Binair modulair exponentiëren, vb.

$3^{61} \bmod 101$, $61 = 32 + 16 + 8 + 4 + 1 = 111101_2$,
dus we berekenen achtereenvolgens:
$3^{1_2}, 3^{11_2}, \ldots, 3^{111101_2} \bmod n$ als volgt:
3^{1_2}, $b_0 = 1$: 3,
3^{11_2}, $b_1 = 1$: $3^2 \times 3 \bmod 101 = 27$,
3^{111_2}, $b_2 = 1$: $27^2 \times 3 \bmod 101 = 66$,
3^{1111_2}, $b_3 = 1$: $66^2 \times 3 \bmod 101 = 39$,
3^{11110_2},$b_4 = 0$: $39^2 \bmod 101 = 6$,
3^{111101_2}, $b_5 = 1$: $6^2 \times 3 \bmod 101 = 7$.
Aantal rekenstappen: $\lfloor {}^2\log(61) \rfloor = 5$.

CWI

RSA sleutel

Iedere deelnemer krijgt een openbare en een geheime sleutel, als volgt:
Kies twee grote priemgetallen, p en q, en bereken hun product $n = pq$, de modulus.
Kies een getal e met

$$1 < e < n \quad \text{en} \quad \gcd(e, (p-1)(q-1)) = 1.$$

Bereken met behulp van de algoritme van Euclides het getal d waarvoor geldt:

$$ed \equiv 1 \bmod (p-1)(q-1).$$

De openbare sleutel is nu het paar (n, e) en de geheime sleutel het paar (n, d) (of andersom).
De priemgetallen p en q moeten geheim gehouden worden, maar mogen ook "weggegooid" worden.

CWI

RSA-vercijfering en -ontcijfering

Stel dat Alice een geheime boodschap wil sturen aan Bob. Alice neemt Bobs openbare sleutel (n, e) uit de public-key lijst, vercijfert haar boodschap m door $c = m^e \bmod n$ te berekenen en stuurt c aan Bob.
Bob ontcijfert c met behulp van zijn geheime sleutel (n, d) door $c^d \bmod n$ te berekenen.

Vanwege de volgende Stelling van Euler:

$$\text{als } \gcd(m, n) = 1, \quad \text{dan is } m^{\phi(n)} \equiv 1 \bmod n,$$

geldt $c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{k\phi(n)+1} \equiv m \bmod n$
(want $\phi(n) = \phi(pq) = (p-1)(q-1)$).
Als m zo was gekozen dat $1 < m < n$ dan geldt $c^d \bmod n = m$, en heeft Bob zo de boodschap m van Alice ontcijferd.
Wie de factoren p en q van n kent, kan op eenvoudige wijze d berekenen en dan ook m uit c.

CWI

Factorisatiemethoden voor n

Special-purpose: *de rekentijd hangt of van bepaalde eigenschappen van de factoren van n of van n*
trial division, Pollard's rho, Pollard's $p-1$, elliptische kromme-methode, special number field sieve

General-purpose: *de rekentijd hangt alleen maar af van de grootte van n*
kettingbreuk-algoritme, kwadratische zeef, general number field sieve

Strategie: probeer eerst kleine factoren te vinden met trial division (kleine priemen $< b_1$), dan Pollard's rho ($< b_2, b_2 > b_1$), dan ECM ($< b_3, b_3 > b_2$), dan de kwadratische zeef of de general number field sieve

CWI

Rekentijd-schattingen

trial division: $\mathcal{O}(p/\log p), p|n$
Pollard's rho: $\mathcal{O}(\sqrt{p}), p|n$
Pollard's $p-1$: $\mathcal{O}(B\log(n)/\log(B)), p|n$, $p-1$ is B-smooth
ECM: $\mathcal{O}(\exp((\sqrt{2}+o(1))(\log p \log\log p)^{1/2})), p|n$

Kettingbreuk-algoritme: $\mathcal{O}(\exp((1+o(1))(\log n \log\log n)^{1/2}))$
QS: $\mathcal{O}(\exp((1+o(1))(\log n \log\log n)^{1/2}))$
NFS: $\mathcal{O}(\exp((c+o(1))((\log n)^{1/3}(\log\log n)^{2/3})))$,
SNFS: $c \approx 1,526$, GNFS: $c \approx 1,923$

CWI

Rekentijd-schattingen, vervolg

De tijd-schattingen voor ECM, CFRAC, QS, SNFS, GNFS zijn heuristisch, dus niet rigoreus, vanwege verschillende aannamen in de analyse, bv. betreffende het aantal gladde getallen geproduceerd door deze algoritmen.

De tijd-analyse van ECM is bijna rigoreus omdat die alleen maar een (redelijke) aanname doet betreffende de verdeling van gladde getallen in korte intervallen.

Er zijn wel algoritmen met rigoreuze tijd-analyses: deterministische met exponentiële tijdschattingen, bv. $\mathcal{O}(n^{1/5+o(1)})$ en probabilistische (d.w.z. ze maken random keuzes), met sub-exponentiele tijdschattingen, vergelijkbaar met ECM, maar de laatste zijn alleen van theoretisch belang omdat de $o(1)$-term zo groot is.

CWI

Genereren van RSA-sleutels

Omdat RSA in de praktijk veel wordt gebruikt is het van belang om snel RSA-sleutels te kunnen genereren, dus om snel grote priemgetallen te kunnen genereren.
Een belangrijke eis voor de veiligheid van RSA is dat de modulus n het product is van twee grote priemgetallen p en q, waarbij
$p-1$ een grote priemfactor bevat, aangeduid met r,
$p+1$ een grote priemfactor bevat, en
$r-1$ een grote priemfactor bevat.
Evenzo voor q.

Want: Pollard's $p-1$ factorisatie-algoritme is efficient als n een priemfactor p heeft met $p-1$ "glad".
Stel $p-1$ heeft een grote priemfactor r, dan is de zgn. "cycling attack" (komt verderop) kansrijk als $r-1$ glad is.
Williams heeft een factorisatie-algoritme bedacht die efficient is als n een priemfactor p heeft met $p+1$ glad.

CWI

Een valkuil voor RSA, 1

Stel dat Alice en Bob dezelfde modulus n hebben gekozen, met openbare sleutels (n, e_A) resp. (n, e_B).
Stel dat Clara zowel aan Alice als aan Bob de boodschap m (bv.: komen jullie morgen op mijn feestje?), wil sturen, onleesbaar gemaakt met RSA uiteraard.
Clara splitst de boodschap dan op in stukjes m en berekent dan $c_A = m^{e_A} \bmod n$ en $c_B = m^{e_B} \bmod n$ en stuurt c_A aan Alice en c_B aan Bob.
Is dit veilig, d.w.z., kan iemand die c_A en c_B onderschept, m vinden?

CWI

Een valkuil voor RSA, 2

Ja! Stel $f = e_A^{-1} \bmod e_B$ en $g = (e_A f - 1)/e_B$, dan geldt:

$$c_A^f (c_B^g)^{-1} = m^{e_A f} (m^{e_B g})^{-1} = m^{e_A f - e_B g} = m \bmod n.$$

Voorbeeld: $n = 309$, $e_A = 4$, $e_B = 25$, $m = 137$, $c_A = 220$, $c_B = 299$. Dan is
$f = 4^{-1} \bmod 25 = 19$ en $g = (4 \times 19 - 1)/25 = 3$ en
$220^{19} \times (299^3)^{-1} \bmod 309 = 137$.

CWI

Mogelijke aanvallen op RSA

1. factoriseer n, vind de decryptie-exponent d
2. als $m < n^{1/e}$ (kleine boodschap), dan is $m = c^{1/e}$.
3. zelfde modulus attack (valkuil)
4. cycling attack
5. timing attack

De cycling attack

Gegeven zijn een RSA modulus $n = pq$ met encryptie-exponent e, dus $c \equiv m^e \bmod n$ is de vercijferfunctie van de boodschap m.
Bereken $a_j = c^{e^j} \bmod n, j = 1, 2, \ldots$, dus $a_0 = c$ en $a_j = a_{j-1}^e \bmod n$.
Stel dat $t \in \mathbb{N}$ het kleinste natuurlijke getal is waarvoor $g = \gcd(a_t - c, n) > 1$. Als $g = p$ of $g = q$ dan hebben we n in priemfactoren ontbonden. Als zowel p als q een deler zijn van $a_t - c$, dan is $g = n$ en dan moet de voorafgaande term $a_{t-1} - c$ gelijk zijn aan m.
(Omdat de vercijferfunctie een permutatie is op de boodschap-ruimte $\{0, 1, \ldots, n-1\}$.)

CWI

Tests met de cycling attack

$n = pq, e = 3427, l(s)$: grootste priemfactor in s

$p(l(p-1),$ $l(l(p-1)-1)$	$q(l(q-1),$ $l(l(q-1)-1)$	t met $g = n$	$\#t$ met $g = p$ of $g = q$
701 (7,3)	1543 (257,2)	1280	67
701 (7,3)	1549 (43,7)	42	13
709 (59,29)	1523 (761,19)	11020	217
709 (59,29)	1531 (17,2)	464	35
719 (359,179)	1523 (761,19)	68020	557
719 (359,179)	1553 (97,3))	8592	225
727 (11,5)	1543 (257,2)	14080	181
727 (11,5)	1531 (17,2)	880	61
733 (61,5)	1523 (761,19)	380	94
733 (61,5)	1553 (97,3)	48	11

CWI

Stelling van Pocklington

Zij $n-1=ab, a, b \in N, b>1$ en stel dat er voor iedere priemdeler q van b een geheel getal m bestaat z.d.d. $\gcd(m^{(n-1)/q}-1, n)=1$ en $m^{n-1} \equiv 1 \bmod n$.
Dan geldt voor iedere priemdeler p van n dat $p \equiv 1 \bmod b$.
Als bovendien $b > \sqrt{n}-1$, dan is n een priemgetal.

Speciaal geval: Zij $n-1=2ab, a, b \in N, b>1$ met b een priemgetal $> a$, en stel dat er een geheel getal m bestaat z.d.d. $\gcd(m^{(n-1)/b}-1, n)=1$ en $m^{n-1} \equiv 1 \bmod n$. Dan is n een priemgetal.

CWI

Bewijs

Stel dat e de hoogste macht van q is die b deelt: notatie $q^e||b$.
Stel verder dat $c = m^{(n-1)/q^e}$ en dat p een priemdeler is van n.
Dan geldt:

$$c^{q^e} = m^{n-1} \equiv 1 \bmod p \quad \text{maar} \quad c^{q^{e-1}} = m^{(n-1)/q} \not\equiv 1 \bmod p,$$

en dit betekent dat de kleinste exponent u waarvoor
$c^u \equiv 1 \bmod p$ gelijk is aan q^e (notatie: $\mathrm{ord}_p(c) = q^e$).
Omdat ook geldt dat $m^{p-1} \equiv 1 \bmod p$ moet gelden dat $q^e|(p-1)$.
Dit geldt voor iedere priemdeler q van b, dus $b|(p-1)$ en
$p - 1 \geq b$. Omdat $b > \sqrt{n} - 1$ volgt dat $p > \sqrt{n}$, dus n is priem.

Maurer's generator van grote random priemgetallen

CWI

Genereert een random priemgetal van een gegeven bit-lengte.

1. Kies een niet te groot priemgetal b (bv. prime(random(1000))).
2. Kies een random geheel getal $a < b$.
3. Als $n := 2ab + 1$ een kleine priemdeler heeft, ga naar stap (2), anders test of n priem is m.b.v. de Stelling van Pocklington, speciaal geval. Zo niet, ga naar Stap (2).
4. Als n groot genoeg is, output n, anders: $b := n$ en ga naar stap (2).

Test van Maurer's generator

b	$a = random(b)$	$n = 2ab + 1$
7919	5752	91100177 $7 \mid n$
	7790	123378021 $3 \mid n$
	2999	47498163 $3 \mid n$
	2523	39959275 $5 \mid n$
	668	10579785 $3 \mid n$
	3093	48986935 $5 \mid n$
	2901	45946039, 7 tries
45946039	9598234	882001667390253 $3 \mid n$
	...	...
	40291817	3702498790525727, 8 tries

Test Maurer, vervolg

# bits	# tries
25	7
51	8
103	28
207	9
413	237
826	234
1653	527

totale rekentijd: 1,4 sec.

In iedere stap wordt de bitlengte van het gegenereerde priemgetal ongeveer verdubbeld. Om aan het eind exact de gewenste bitlengte te krijgen kiezen we in de laatste stap a zó dat $n = 2ab + 1$ de gewenste grootte heeft.

CWI

Conclusies

- record voor RSA-sleutel-factorisatie: 768 bits, 232 decimalen
- RSA nog redelijk veilig als $n = pq$ met p en q 512-bits priemgetallen
- toch oppassen voor valkuilen in RSA
- RSA-sleutels kunnen snel gegenereerd worden met random priemgetallen van gewenste bitsize
- leuk onderwerp voor in de klas als belangrijke toepassing van getaltheorie (?)