03-03-09 - RSA Kraken

RSA is een van de meest gebruikte cryptografieën die we momenteel kennen. Het idee was simpel. Je hebt een publieke sleutel en een geheime sleutel. Met de publieke sleutel kan een zender een bericht encrypten en met de geheime sleutel kan dit bericht weer worden gedecrypt. Helaas is deze manier van encryptie niet helemaal veilig. In dit artikel zal ik uitleggen hoe RSA nu precies werkt en hoe je deze eenvoudig kunt kraken.

Encrypten met RSA

Zoals ik al in de inleiding aangaf, is er een zogenaamd publiek gedeelte. Iedereen kan dit deel van de versleuteling inzien. Als ontvanger van de informatie verspreid je deze dus.

N = 143
e = 37

Dit is een voorbeeldje van zo'n publieke sleutel. Belangrijk is dat e een priemgetal is en dat N de vermenigvuldiging is van twee priemgetallen. In dit geval 13*11. Dit houdt in dat ze alleen door zichzelf deelbaar zijn of deelbaar door 1. Het getal 1 is om wiskundige redenen echter geen priemgetal. Het rijtje begint dus met 2, 3, 5, 7, 11... Naast een sleutel heb je ook een codeboek waarin staat welke letter welke numerieke waarde heeft.

c = 3

Voor de encryptie gebruiken we de volgende formule:

ne % N

In deze situatie leidt dat tot:

337 % 143

Het %-teken heet de modulo. Dit is een operator die de restwaarde van een deling uitrekent. Dus 16 % 5 = 1, want er kan drie keer een hele vijf uit en dan hou je er nog maar één over.

Om dit uit te rekenen kun je natuurlijk een rekenmachine pakken, maar bij grotere

getallen zal dit ongetwijfeld problemen op gaan leveren, omdat het simpelweg niet meer past en er dus afgerond moet worden.

Daarom gebruiken we de volgende techniek:

	1	0	0	1	0	1	= 37 binair
1	(12 * 3) % 143 = 3	32 % 143 = 9	92 % 143 = 81	(812 * 3) % 143 = 92	922 % 143 = 27	(272 * 3) % 143 = 42	% 143

We kwadrateren het voorgaande getal dus steeds en bereken de testwaarde. Wanneer er een 1 staat in de binaire reeks, moet het voorgaande getal ook nog met het grondtal vermenigvuldigd (3) worden.

We krijgen nu de 3 weer terug en als we in het codeboek kijken, zien we dat hier de [illegible]

Het laatste item uit de reeks, 42, is het resultaat van de encryptie. Deze versturen we nu naar de ontvanger.

	1	1	0	1	= 13 binair
1	42	141	53	3	% 143

1 * 120	+	0 * 37	= 120
0 * 120	+	1 * 37	= 37
1 * 120	+	-3 * 37	= 9
-4 * 120	+	13 * 37	= 1

RSA kraken

Decrypten

Maar hoe komen we nu aan deze getallen? Dat is vrij ingewikkeld om uit te leggen, maar eenvoudig te begrijpen.

We kunnen nu dus RSA encrypten en decrypten, maar het leukste is natuurlijk om [illegible]

[illegible]

[illegible] krijg je dus een hele reeks getallen.

[illegible]

[illegible] gebruiken we om de -3 uit te rekenen. 0 ? 3 * 1 = -3.

[illegible] 11 en 13, gebruikt. [illegible] krijgen we:

Om te decrypten gebruiken we een soortgelijke formule als bij het encrypten.

-4 * 120 + 13 * 37 = 1

(11-1)*(13-1) = 120

De -4 * 120 is niet belangrijk. Dit geeft alleen aan hoe vaak de modulo eraf gehaald [illegible] We weten nu namelijk dat d = 13.

Nu we dit weten, kunnen we het bericht eenvoudig decrypten.

[illegible]

Gevolgen

Om dit uit te rekenen gaan we een tabel maken:

We hebben RSA nu gekraakt, maar is deze daarom per se onveilig? RSA wordt nog steeds op grote schaal gebruikt, waaronder het bankwezen. Maar met deze methodiek zou je dat toch allemaal kunnen kraken? Ja, dat klopt! MAAR... niet met een simpele thuiscomputer. Het bovenstaande voorbeeld is een minimaal voorbeeld. 11*13? Probeer het eens met N=10850763853. Dan moet je een

machine hebben om het uit te kunnen rekenen. Hoe groter de priemgetallen zijn, hoe moeilijker het wordt om te kraken. Alleen met een kwantumcomputer zal een echt sterke RSA-encryptie te kraken zijn. Helaas zijn kwantumcomputers alleen op papier uitgevonden en niet in praktijk. Voorlopig is RSA dus nog wel veilig, alleen over een x aantal jaren zal die kwantumcomputer er echt wel komen en dan zijn de codes makkelijk te kraken. Dit betekent dat niet alleen systemen aangepast moeten worden in de toekomst, maar ook dat berichten die momenteel zijn afgeluisterd, alsnog gekraakt kunnen worden. Een pincode die door middel van RSA gekraakt is, zou dan alsnog gekraakt kunnen worden. Ik kan niet voorspellen wanneer deze kwantumcomputer op de markt komt, maar als ik de ontwikkelingen van de afgelopen 40 jaar bekijk, dan verwacht ik dat hij er wel komt de aankomende 40 jaar. Over 40 jaar is mijn pincode waarschijnlijk nog steeds hetzelfde en dan zou iemand alleen nog mijn pasje hoeven stelen.

Pincodes zouden alleen daarom al iedere tien jaar vervangen moeten worden. De miljoenennota voor volgend jaar kan prima opgeslagen worden met RSA-encryptie, omdat deze toch niet te kraken valt voordat deze publiek wordt gemaakt.

Conclusie

In het artikel heb ik besproken hoe je een bericht kunt encrypten en decrypten met RSA. Ook heb ik uitgelegd hoe een eenvoudige RSA-encryptie te kraken is. Voorlopig is RSA nog veilig, maar in de toekomst zal het te kraken zijn en we moeten dus ook voorkomen dat we data encrypten met RSA die over 40 jaar nog geheim moeten blijven.