

Anleitung: RSA knacken mit *CrypTool*

1. Starte *CrypTool* und rufe im Menü *Einzelverfahren* → *RSA-Kryptosystem* → *RSA-Demo...* auf!

Wähle diesmal den zweiten Radioknopf „Zur Verschlüsselung von Daten...“!

In diesem Fall können nur der RSA-Modul N und der öffentliche Schlüssel e eingegeben werden (e ist wieder auf $2^{16}+1$ voreingestellt, das kann aber geändert werden).

RSA mit privatem und öffentlichem Schlüssel -- oder nur mit öffentlichem Schlüssel

Wählen Sie 2 Primzahlen p und q . Die Zahl $N = pq$ ist der öffentliche RSA-Modul, und $\phi(N) = (p-1)(q-1)$ ist die Eulersche Phi-Funktion. Der öffentliche Schlüssel e ist teilerfremd zu $\phi(N)$. Daraus wird der geheime Schlüssel $d = e^{-1} \pmod{\phi(N)}$ berechnet.

Zur Verschlüsselung von Daten oder zur Verifikation einer Signatur genügt es, dass Sie die öffentlichen RSA-Parameter angeben: den RSA-Modul N und den öffentlichen Schlüssel e .

Faktorisierungsangriff:
Sie können mit Hilfe der Faktorisierung versuchen, den öffentlichen RSA-Modul N in seine Primfaktoren p und q zu faktorisieren. RSA-Modul faktorisieren...

RSA-Parameter:

RSA-Modul N : (öffentlich)

$\phi(N) = (p-1)(q-1)$: (geheim)

Öffentlicher Schlüssel e :

Geheimer Schlüssel d :

Parameter aktualisieren

2. Gib einen RSA-Modul N ein!

3. Klicke nun den Knopf „RSA-Modul faktorisieren“! Wenn die eingegebene Zahl ein gültiger RSA-Modul (und nicht zu groß) ist, werden die beiden Primfaktoren p und q in dem Faktorisierungsfenster von *CrypTool* gefunden. Falls N keine Semi-primzahl ist, wird eine entsprechende Fehlermeldung ausgegeben.

Faktorisieren einer Zahl

Algorithmen zur Faktorisierung:

- Brute-Force
- Brent
- Pollard
- Williams
- Lenstra
- Quadratisches Sieb

Eingabe:

Geben Sie die zu faktorisierende Zahl ein:

Faktorisierung (schrittweise):
Durch das Anklicken des Buttons "Weiter" wird initial die Zahl im Eingabefeld und dann jeweils die nächste zusammengesetzte Zahl im Feld "Produktdarstellung" in zwei Faktoren zerlegt.

Weiter

Faktorisierungsergebnis:
Die Faktorisierung wird in dem Format $\langle z_1^{a_1} * z_2^{a_2} * \dots * z_n^{a_n} \rangle$ dargestellt. Zusammengesetzte Zahlen sind rot markiert.

Letzte Faktorisierung durch: Brute Force 2 Faktoren gefunden in 0,030 Sekunden.

Produktdarstellung der Faktorisierung:

Details

Schließen

Erfolgreiche Faktorisierung
des RSA-Moduls $N = 32442353$.

Nach Schließen dieses Fensters werden die Primfaktoren p und q im normalen Fenster vom *RSA-Demo* eingetragen, außerdem werden sogleich $\phi(N)$ und der geheime Schlüssel d berechnet. Man hat damit wieder ein voll funktionsfähiges RSA-System, obwohl nur der öffentliche Schlüssel bekannt war!

Merke:

**Die Sicherheit von RSA wird ganz wesentlich
von der Schlüssellänge des gewählten RSA-Moduls bestimmt!**

Forschungsauftrag

Wieviel Bit muss ein RSA-Modul haben, damit er nicht mehr innerhalb von wenigen Sekunden mit *CrypTool* in der oben beschriebenen Weise geknackt werden kann?

Hinweis: Wie kann ich z. B. einen RSA-Modul mit 64 Bit erzeugen? Dafür werden zwei Primfaktoren p und q mit jeweils 32 Bit Länge benötigt. Man gebe also im Fenster „Primzahlen generieren...“ als Untergrenze jeweils 2^{31} und als Obergrenze 2^{32} ein. Nach den Klicks auf „Primzahlen generieren“ und danach „Primzahlen übernehmen“ hat man zwei Primzahlen p und q mit 32 Bit Länge und einen Modul $N = p \cdot q$ mit 64 Bit Länge erzeugt!