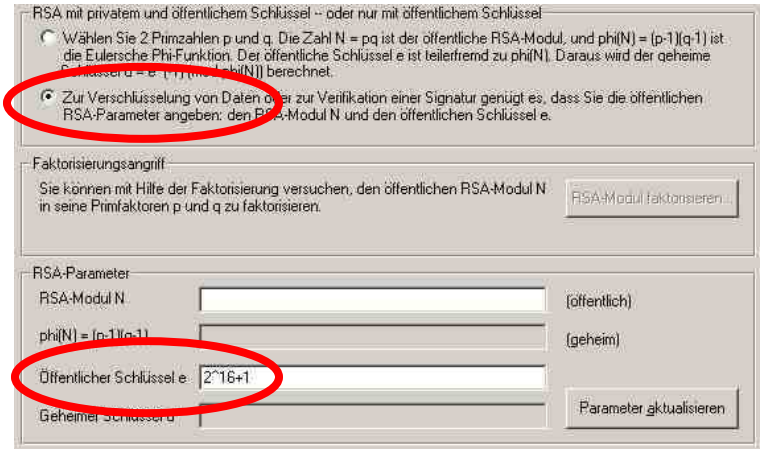
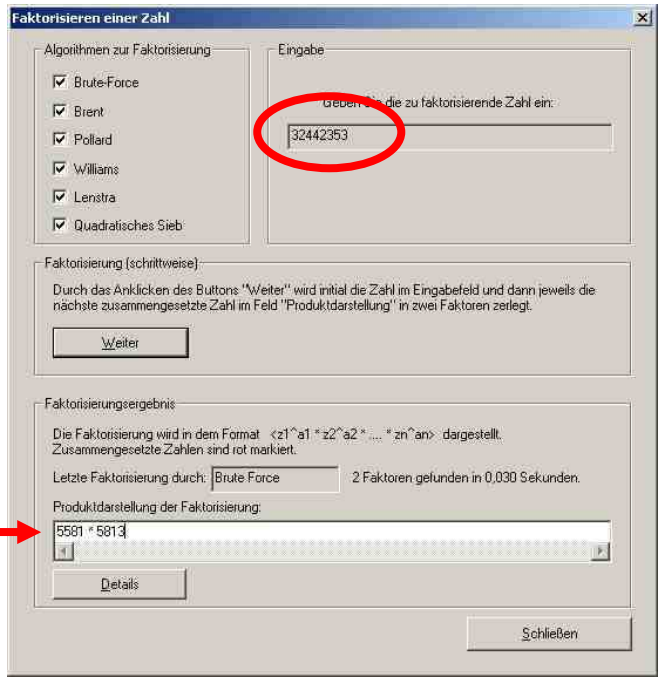
Anleitung: RSA knacken mit *CrypTool*

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).



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 Semiprimzahl ist, wird eine entsprechende Fehlermeldung ausgegeben.



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: **<u>Die Sicherheit von RSA wird ganz wesentlich</u>**

<u>von der Schlüssellänge des gewählten RSA-Moduls bestimmt!</u>

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·q mit 64 Bit Länge erzeugt!