Anleitung: Ver- und Entschlüsseln mit der RSA-Demo von CrypTool

1. RSA-Schlüsselpaar generieren:

Starte *CrypTool* und rufe im Menü *Einzelverfahren* \rightarrow *RSA-Kryptosystem* \rightarrow *RSA-Demo...* auf! Es erscheint ein (auf den ersten Blick etwas unübersichtliches) Fenster. Betrachte zunächst nur den oberen Ausschnitt:

 Wählen Sie 2 die Eulersche Schlüssel d = 	Primzahlen p und q. Die Zahl N = pq ist de Phi-Funktion. Der öffentliche Schlüssel e i e^(-1) (mod phi(N)) berechnet.	er öffentliche RSA-Modul, und phi(N) = (p-1)(q-1) ist ist teilerfremd zu phi(N). Daraus wird der geheime
C Zur Verschlüs BSA-Paramet	selung von Daten oder zur Verifikation eine er angeben: den RSA-Modul N und den öf	er Signatur genügt es, dass Sie die öffentlichen Ifentlichen Schlüssel e
Primzahleingabe –		
Primzahleingabe – Primzahl p	[Primzahlen generieren

Es gibt zwei Möglichkeiten, die Schlüssel (e; N) und d zu konstruieren:

- zwei Primzahlen in die Felder für p und q eintragen oder
- die Primzahlen von *CrypTool* wie folgt erzeugen lassen: Knopf "Primzahlen generieren" drücken. Ein neues Fenster erscheint mit den voreingestellten Werten:

mzahlen generieren			
Primzahlen spielen in der modernen Kry Primzahlen aus einem vorzugebenden 1	ptographie eine wichtige Rolle. Hier können Sie sich Wertebereich [Untergrenze, Obergrenze] erzeugen.		
Anzahl der zu generierenden Primzahler	ì .		
Zwei Primzahlen zufällig aus dem V	Wertebereich (den Wertebereichen) generieren		
C Alle Primzahlen in dem (für p vorge	gebenen) Wertebereich generieren		
Trennzeichen für die Ausgabe der	Primzahlen:		
Algorithmen zur Generierung	Wertebereich der Primzahlen p und g		
Miller-Rabin-Test	Unabhängig voneinander einzugeben		
C Solovay-Strassen-Test			
⊂ <u>F</u> ermat-Test	G Beide gleich (nur einen eingeben)		
Primzahl p	Primzahl q		
Untergrenze 2^7	Untergrenze 2^7		
Obergrenze 2^8	Obergrenze 2^8		
Ergebnis 211	Ergebnis 233		
Primzahlen generieren Prim	zahlen übernehmen Abbrechen		

Wir haben also zwei Primzahlen zwischen $2^7 = 128$ und $2^8 = 256$ erhalten. Wenn uns diese Zahlen nicht gefallen sollten, drücken wir ggf. mehrfach "Primzahlen generieren" und erhalten dann andere Primzahlen aus diesem Bereich, z. B. 227 und 251. *CrypTool* benutzt Pseudozufallszahlen, die stets in der gleichen Reihenfolge auftreten, es macht also Sinn, mehrmals auf den Knopf zu drücken!

Klicke nun auf "Primzahlen übernehmen". Es erscheint wieder der Ausgangsschirm, aber neben den Primzahlen p und q sind bereits der **RSA-Modul N** und **phi(N) = (p-1)(q-1)** eingetragen. Als öffentlicher Schlüssel e ist immer $2^{16}+1 = 65537$ voreingestellt¹. Wem diese Zahl nicht gefällt, kann auch hier eine andere eintragen. Diese muss aber teilerfremd zu phi(N) sein!

¹ Wer wissen will, warum gerade diese Zahl bevorzugt wird, sollte sich z. B. mit Hilfe des Windows-Taschenrechners ihre Darstellung im Dualsystem anschauen

Der zugehörige geheime Schlüssel wird ebenfalls automatisch erzeugt:

Primzahl p	211		Primzahlen generieren	
Primzahl q	233			
SA-Parameter				
RSA-Modul N	49163		(öffentlich)	
phi(N) = (p-1)(q-1)	48720		(geheim)	
Öffentlicher Schlüssel e	2^16+1		-	
Geheimer Schlüssel d	1 44273		Parameter <u>a</u> ktualisieren	
SA-Verschlüsselung mit	e / Entschlüsselung mit d			
Eingabe als 📀 Text 🥤 Zahlen		<u>O</u> ptionen für	Optionen für Alphabet und Zahlensystem	

2. Verschlüsseln:

- Gib einen Text in das untere Eingabefeld ein!
- Klicke auf "Verschlüsseln" um ihn zu verschlüsseln!



Wegen des relativ kleinen Moduls **N** wird der Text in Blöcke der Länge 1 unterteilt² und als Zahlen dargestellt (die entsprechenden ASCII-Nummern). Diese werden dann Block für Block (bei Blocklänge 1 also Zeichen für Zeichen) verschlüsselt.

3. Entschlüsseln:

Trotzdem wollen wir uns überzeugen, dass sich dieser "Geheimtext" wieder korrekt entschlüsseln lässt:

- Kopiere den Geheimtext in die Eingabezeile!
- Klicke auf den Knopf "Zahlen"! (Wenn Du das vergisst, weißt dich CrypTool darauf hin.)
- Klicke auf den Knopf "Entschlüsseln"! Ergebnis:

Eingabe als 🦳 Text 🔎 Zahlen	Optionen für Alphabet und Zahlensystem
Geheimtext in Zahlendarstellung zur Basis 10.	
1 # 07428 # 09394 # 29564 # 18504 # 35574 # 233	366 # 08293 # 20714 # 35574 # 23366 # 10710 # 37441
Entschlüsselung in den Klartext m[i] = c[i]^d (mod N)	
00068 # 00097 # 00115 # 00032 # 00105 # 00115 #	# 00116 # 00032 # 00101 # 00105 # 00110 # 00101 # 0
Ausgabetext aus der Entschlüsselung (in Blöcken der	Länge 1; das Symbol '#' dient nur als Trennzeichen).
D#a#s# #i#s#t# #e#i#n#e# #g#e	e#h#e#i#m#e# #N#a#c#h#r#i#c#h#
Klartext	

² Bei einem so kleinen Modul N handelt es sich um eine schlichte monoalphabetische Verschlüsselung, die ein Knacken per Häufigkeitsanalyse erlaubt. Bei größeren Primzahlen werden jedoch stets mehrere Zeichen in einem Block zusammengefasst.