Hochschule für Technik und Wirtschaft
Studiengang Kommunikationsinformatik
Studiengang Praktische Informatik
Prof. Dr.–Ing. Damian Weber

# Kryptographie – Übung 3

**Aufgabe 1 (ElGamal–Kryptosystem)**

Implementieren Sie das ElGamal–Kryptosystem. Hierzu definiere man eine Klasse ElGamal, die als Attribute die `BigInteger`–Werte $p, g, a$ und die im folgenden beschriebenen Methoden enthält.

a) *setup* erzeuge für vorgegebene Bitlänge die 3 Parameter.

b) *encrypt* verschlüssle ein Element aus $\mathbf{Z}/p\mathbf{Z}$.

c) *decrypt* entschlüssle ein Element aus $\mathbf{Z}/p\mathbf{Z}$.

d) *sign* signiere ein Element aus $\mathbf{Z}/p\mathbf{Z}$.

e) *verify* verifiziere die Signatur für ein Element aus $\mathbf{Z}/p\mathbf{Z}$.

Für eine vorgegebene Bitlänge $b$ sollten Sie $p$ folgendermaßen wählen:

- wähle eine Primzahl $q$ der Bitlänge $b - 1$
- prüfe, ob $p = 2q + 1$ eine Primzahl ist, falls ja $\leadsto$ fertig
- falls $p$ keine Primzahl ist, so wähle ein neues $q$

Dann ergibt sich nämlich die Primfaktorzerlegung von $p - 1$ sehr einfach als $p - 1 = 2 \cdot q$ und das Erzeugerkriterium für $g$ ist ein kurzer Test mit Hilfe von zwei Potenzierungen.

Lesen Sie die Bitlänge von der Kommandozeile ein und testen Sie

- mit Hilfe der *decrypt*–Methode, ob die *encrypt*–Methode
- mit Hilfe der *verify*–Methode, ob die *sign*–Methode

korrekt arbeitet.