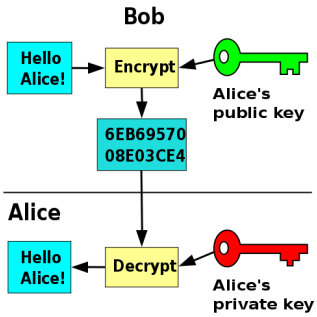
Einführung

1. Symmetrische und asymetrische Verschlüsselung

Die Cäsar-Verschlüsselung wie auch die Vigenère-Verschlüsselung sind sogenannte *symmetrische* Verschlüsselungsverfahren. Das bedeutet, dass der Schlüssel für die Verschlüsselung der selbe ist wie für die Entschlüsselung.
Der Vorteil hiervon ist, dass diese sehr einfach und schnell funktioniere. Der Hauptnachteil ist, dass der Schlüssel auf beiden Seiten der selbe sein muss und dieser Schlüssel auf einem sicheren Weg ausgetauscht werden muss.
Diesen sicheren Weg gibt es aber im Internet für gewöhnlich nicht!
Um dieses Dilemma zu umgehen nutzt man eine *asymmetrische* Verschlüsselung. Der Schlüssel für die Verschlüsselung ist dabei nicht identisch zum Schlüssel zur Entschlüsselung. Man spricht dabei auch oft von *Public-Key-Verschlüselung.*

2. Private- und Public-Key



Zur Ver- und Entschlüsselung benötigt man ein Schlüsselpaar. Will **Bob** nun eine verschlüsselte Nachricht an **Alice** schicken, so braucht Bob hierfür den *Public-Key* von Alice.
Diesen Public-Key kann aber auch gefahrlos über einen unsicheren Weg übertragen werden, da eine mit diesem Schlüssel codierte Nachricht nur mit dem *Private-Key* wieder entschlüsselt werden kann!
Bob holt sich also den Public-Key von Alice, verschlüsselt damit seine Nachricht und verschickt diese dann. Alice kann mit ihrem geheimen Private-Key die Nachricht wieder entschlüsseln.
Selbst wenn nun ein Angreifer die Übertragung des Public-Keys und die verschlüsselte Nachricht abfängt, so kann er diese nicht entschlüsseln.

3. RSA

Das RSA-Verfahren basiert darauf, dass es selbst für Hochleistungsrechner sehr schwierig ist, große Zahlen in ihre Primfaktoren zu zerlegen, der umgekehrte Weg – also die Multiplikation zweier Primzahlen – jedoch sehr schnell zu berechnen ist.

Beispiel 1:

Versuche einmal selbst (auch mit Taschenrechner) die Zahl 629 zu faktorisieren.
Anschließend versuche es mit der Zahl 17 699 629. Es wird dir nicht gelingen!
Die umgekehrte Rechnung: multipliziere die beiden Primzahlen 3 259 und 5 431 miteinander. Diese Rechnung kannst du sehr schnell durchführen.

3.1 Modulo-Rechnung

Eine weitere Rechenart ist die *Modulo*-Rechnung. Diese kennst du bereits seit der Grundschule!
Modulo bedeutet nichts anderes als dass man den Rest berechnet, der bei einer ganzzahligen Division übrig bleibt.

Beispiel 2: Modulo

$\frac{20}{6} = 3$ Rest 2, deshalb ist $20 \mod 6 = 2$ $\frac{43}{5} = 8$ Rest 3, deshalb ist $43 \mod 5 = 3$

3.2 Erzeugung eines Schlüsselpaares

Zuerst wählt Alice sich zwei Primzahlen. Für eine echte Verschlüsselung wäht man natürlich zwei möglichst große Zahlen, hier als Beispiel arbeiten wir mit Zahlen bis 100.
Alice wählt $p = 17$ und $q = 11$ Diese beiden Zahlen muss Alice geheim halten!
Zusätzlich wählt sie eine weitere (Prim-)Zahl $e = 7$. Diese sollte teilerfremd sein zu $\big((p-1)\cdot(q-1)\big)$.

3.2.1 privater Schlüssel

Den eigentlichen **privaten Schlüssel** d berechnet Alice anschließend mit der Formel

$$1 = (e \cdot d) \mod \big((p-1)\cdot(q-1)\big)$$

also

$$1 = (7 \cdot d) \mod (160)$$

Alice findet so $d = 23$, für welches die Gleichung stimmt.
Der private Schlüssel besteht dann aus den beiden Zahlen $d = 23$ und $N = p \cdot q = 187$.

3.2.2 öffentlicher Schlüssel

Der öffentliche Schlüssel besteht dann aus den beiden Zahlen $e = 7$ und $N = 187$. Diesen Schlüssel bekommt nun Bob um seine Nachricht damit verschlüsseln zu können.

3.2.3 Verschlüsselung

Zur Verschlüsselung wandelt man die Daten zuerst in eine Zahl M um, beispielsweise mit dem ASCII-Code. Bob will beispielsweise den Buchstaben **X** verschicken, im ASCII-Code entspricht dieser dem Wert $M = 88$.
Zur Verschlüsselung rechnet Bob jetzt

$$C = M^e \mod N$$

also

$$C = 88^7 \mod 187$$

$$C = 40867559636992 \mod 187 = \mathbf{11}$$

Bob verschickt nun also diesen Verschlüsselten Wert $C = 11$ an Alice.

3.2.4 Entschlüsselung

Alice kann mit ihrem privaten Schlüssel nun die Nachricht wieder wie folgt entschlüsseln:

$$M = C^d \mod N$$

also

$$M = 11^{23} \mod 187$$

$$M = 8954302432552373722465 31 \mod 187 = \mathbf{88}$$

und hat damit die Nachricht wieder entschlüsselt.

3.2.5 Warum?

Warum diese Rechnung funktioniert ist sehr komplex und basiert auf dem **Satz von Euler**.
Warum dieses System so schwer zu knacken ist liegt daran, dass man bei dem Schlüssel zwar das Produkt der Primzahlen p und q hat, jedoch ist es nahezu unmöglich für genügend große Primzahlen hierzu das passende d zu berechnen sofern man nicht die Primzahlen selbst hat!