Der RSA Algorithmus

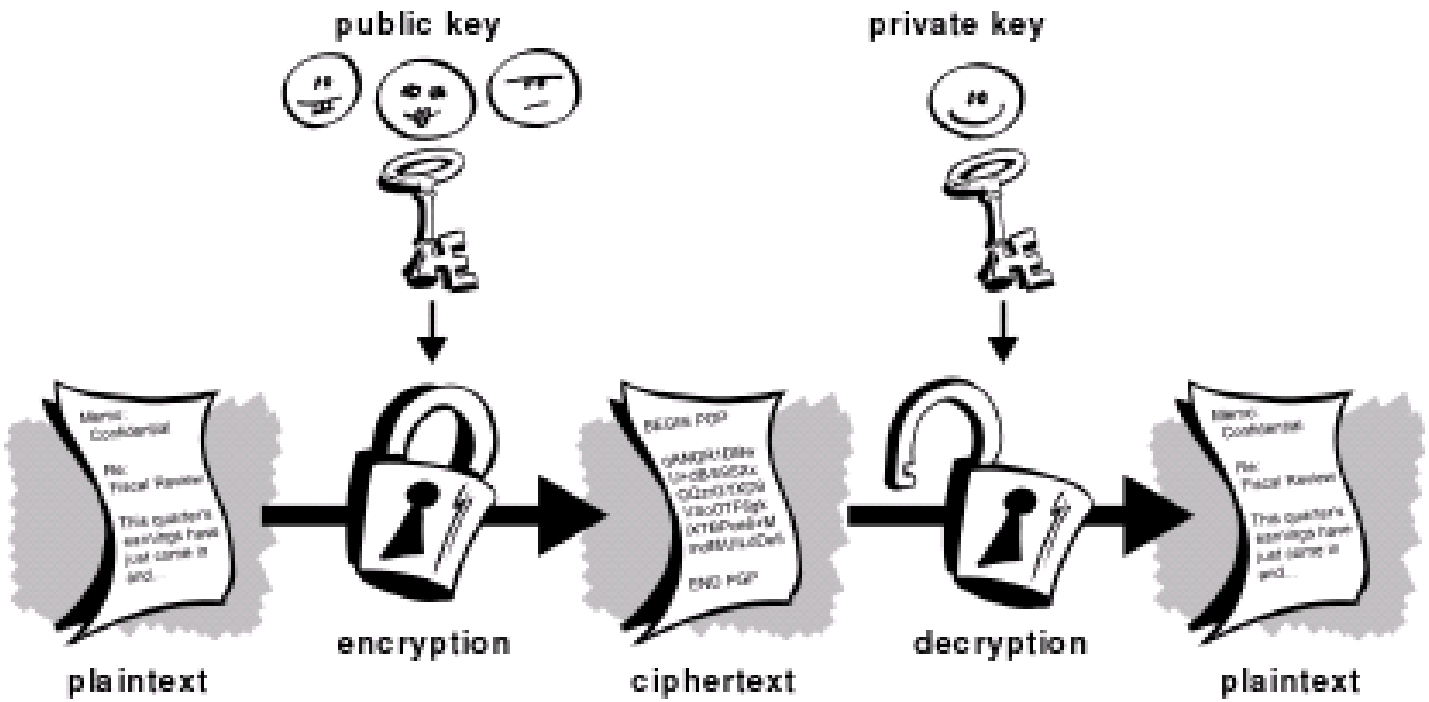
Martin Albrecht

„Die Mathematik ist die Königin der Wissenschaften und die Zahlentheorie ist die Königin der Mathematik.“ - Carl Friedrich Gauß

Das Problem und seine Lösung:

- (Patienten-)Daten im Internet können von jedem mitgelesen werden
- Dagegen kann man sich schützen, indem man seine Daten verschlüsselt
- Verschlüsselung mit Passwort: Das Passwort muss auf einem sicheren Kanal vorher übertragen werden
- Public-Key-Kryptographie löst dieses Problem

Public-Key-Kryptographie



RSA: Geschichte

- Public-Key Kryptographie wurde 1976 von Diffie und Hellman öffentlich vorgeschlagen
- 1977 entwickelten Rivest, Shamir und Adleman den ersten vollständigen, öffentlich zugänglichen Algorithmus: RSA
- Heute weiß man, dass im Britischen Geheimdienst GCHQ bereits 1970 die Public-Key Kryptographie entwickelt worden war

Vorbereitung: Primzahlen

- Zahlen die nur durch sich selbst bzw. durch 1 geteilt werden können
- Es ist bewiesen, dass es unendlich viele gibt
- Beispiele: 2,3,5,7,1241421352456358230450934 77
- Zwei Zahlen *e,n* können relativ prim sein, das heißt ihr größter gemeinsamer Teiler ist 1, notiert als *gcd(e,n)=1*
- Beispiel: *gcd(4,9)=1*

Vorbereitung: Rechnen mit „Restklassen“

- Wir dividieren mit Rest und merken uns nur den Rest
- Beispiel Wochentage: Der Montag ist der erste Tag der Woche, in genau 3 Wochen ($1+21 = 22$) ist wieder Montag ($22 \equiv 1 \bmod 7$), denn $22=3*7+1$
- Ist n eine natürliche Zahl, dann ist Z_n eine „Gruppe“
- Diese Gruppe hat n Elemente
- Bespiel $Z_{15} = \{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$

Vorbereitung: „multiplikative Gruppen“

- Z_n^* ist die „multiplikative Gruppe“ zu Z_n und es gilt für jedes Element e aus Z_n^*: $gcd(e,n)=1$
- Damit hat jedes Element a ein Inverses b, so dass gilt $ab \equiv 1 \bmod n$
- Beispiel $Z_{15}^* = \{1,2,4,7,8,11,13,14\}$
- Das inverse Element b zu $a=7$ ist 13 : $7*13 \equiv 1 \bmod 15$

Vorbereitung: Anzahl der Elemente in Z_n^*

- Ist $n=p$ eine Primzahl hat die mult. Gruppe $k=p-1$ Elemente
- ist $n=pq$ das Produkt zweier Primzahlen hat die Gruppe $k=(p-1)(q-1)$ Elemente
- Es gilt für Element e aus Z_n^*: $e^k \equiv 1$

RSA: Die Variablen

- *m* Text zum Verschlüsseln (als natürliche Zahl)
- *c* Verschlüsselter Text (als natürliche Zahl)
- *n,e* öffentlicher Schlüssel
- *d* geheimer Schlüssel

RSA: Der Algorithmus

- Wähle zwei Primzahlen p und q
- Bilde $n = pq$
- Wähle ein zufälliges e, so dass e und $k=(p-1)(q-1)$ relativ prim sind: $gcd(e,k)=1$
- Berechne d, so dass gilt $ed \equiv 1 \bmod k$
- Verschlüsselung: $c \equiv m^e \bmod n$
- Entschlüsselung: $m \equiv c^d \bmod n$

Korrektheit & Angriffe

- Warum klappt das?
  - Betrachte: $c^d = (m^e)^d = m^{(ed) \bmod k} = m^1 = m$
- Wie greift man den Algorithmus an, wenn man nur e,n kennt?
  - Berechne p und q aus n, berechne $k=(p-1)(q-1)$ und ermittle d wie vorhin

RSA: Ist der Algorithmus sicher?

- Das weiß niemand! Aber alle glauben daran!
- *n* ist üblicherweise 1024-bit lang, das ist eine Zahl die 309 Stellen hat
- Die Zerlegung von großen Zahlen in ihre Primfaktoren (Faktorisierung) wird allgemein als sehr schwer angesehen
- Es gibt aber keinen Beweis dafür!

$n=pq$ mit 1024-bit

- 5268152631368340601212759188376417095342860
  6813564406643193328395662303136089462853729
  2515626776577869074786092354753420904547700
  0492201943520028969685928627352681489267989
  8651560328827232691150965427746204989304177
  3381190194737369602632983832792384216084328
  8049855154064284958518749794636058033699546
  68152727

RSA: Stand der Kunst beim Faktorisieren

- Die größte bekannte Zahl die faktorisiert wurde, hat 200 Dezimalstellen.
- „The sieving effort is estimated to have taken the equivalent of 55 years on a single 2.2 GHz Opteron CPU. The matrix step reportedly took about 3 months on a cluster of 80 2.2 GHz Opterons. The sieving began in late 2003 and the matrix step was completed in May 2005. P.“ (RSA Laboratories)

RSA: Sind 1024-bit sicher?

- Sowohl die Computer als auch die Algorithmen für die Primzahlenzerlegung werden immer besser
- Aber es gibt keinen Grund bei 1024-bit Länge für *n* zu bleiben
- mit einem *n* von 2048-bit Länge, sollte man die nächsten Jahrzehnte sicher sein.
- Es sei denn, jemand hat eine brillante Idee um RSA zu brechen ... happy hacking!