Datensicherheit durch Kryptographie

- Dr. Michael Hortmann
- Fachbereich Mathematik,
  Universität Bremen

- T-Systems

- Michael.Hortmann@gmx.de

Kryptographie:

Klassisch:

Wissenschaft und Praxis
der **Datenverschlüsselung**

Nur bestimmte Empfänger sollen eine Nachricht lesen können.

Steganographie

Die **Existenz** einer Nachricht soll verborgen werden.

Beispiel:
versteckte Nachrichten in Bild- oder Ton Dateien

Cäsar Chiffre

ABCDEFGHIKLMNOPQRSTUVXYZ
DEFGHIKLMNOPQRSTUVXYZABC

VENIVIDIVICI
ZHQMZMGMZMFM

Verschlüsselung: $x \to x+3 \mod 25$

Entschlüsselung: $x \to x-3 \mod 25$

Klassische Anwendungen

- Militär
- Diplomatie

Beispiel: **Enigma** Chiffriermaschine

Verschlüsselung des Funkverkehrs für die deutsche U-Boot Flotte im 2. Weltkrieg

Gebrochen von **Alan Turing**

Moderne Anwendungen

Email-Verschlüsselung: **PGP, S/MIME**
SSL Verschlüsselung im Web-Browser
IPSEC VPN

Digitale Signaturen (Signaturgesetz ‘97)
ec-Karten Geheimzahl

Digitales Geld, E-Commerce
Digitale Wahlen, E-Government

Moderne Entwicklung

Public Key Kryptographie 1976

Diffie Hellman
Rivest Shamir Adleman **RSA**

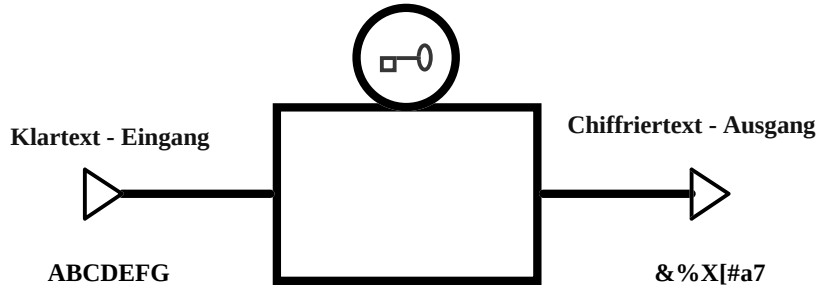
Sicherheit der heutigen Public Key Verfahren beruht auf der praktischen **Unlösbarkeit** des

Faktorisierungs Problems
Diskreten Logarithmus Problems

Quantenalgorithmen machen diese Probleme effektiv lösbar!
(Peter Shor 1994)

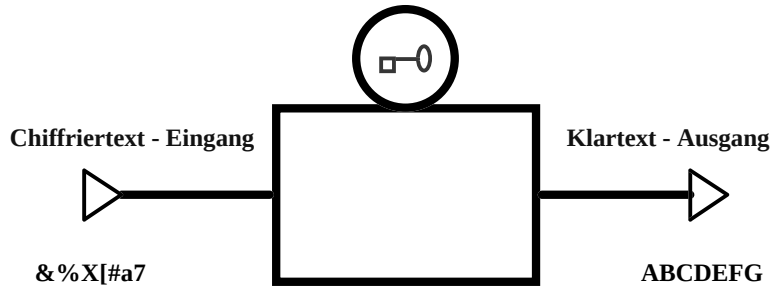
Klassische Chiffriermaschine

Alice verschlüsselt



Alice und Bob müssen denselben Schlüssel besitzen

Bob entschüsselt



Problem bei klassischen Verfahren:

Schlüsselmanagement

- Verteilung der Schlüssel
- Anzahl der Schlüssel

Schlüssel:

Enigma: Verdrahtungsschema

DES: 56 Bit

AES 256 Bit

Public Key Idee

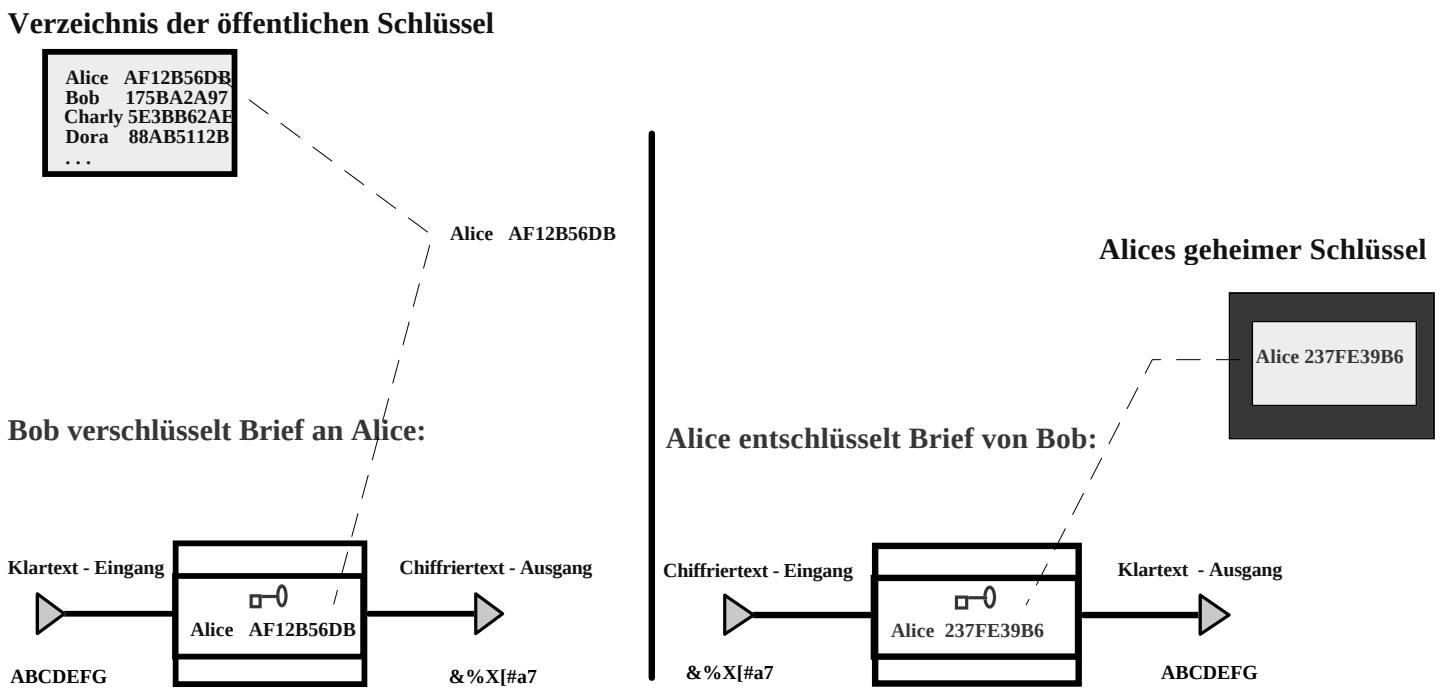
Der Schlüssel für die Entschlüsselung ist verschieden vom Schlüssel für die Verschlüsselung und läßt sich aus diesem nicht berechnen.

Der Schlüssel für die **Verschlüsselung** kann also in einem öffentlichen Verzeichnis stehen! Er heißt deshalb

Öffentlicher Schlüssel

Der Schlüssel für die **Entschlüsselung** ist von jedem Anwender geheimzuhalten:

Geheimer/Privater Schlüssel


Verzeichnis der öffentlichen Schlüssel
Alice AF12B56DB
Bob 175BA2A97
Charly 5E3BB62AE
Dora 88AB5112B
. . .
Alice AF12B56DB
Alices geheimer Schlüssel
Alice 237FE39B6
Bob verschlüsselt Brief an Alice:
Alice entschlüsselt Brief von Bob:
Klartext - Eingang
Chiffriertext - Ausgang
Alice AF12B56DB
ABCDEFG
&%X[#a7
Chiffriertext - Eingang
Klartext - Ausgang
Alice 237FE39B6
&%X[#a7
ABCDEFG

Das **RSA Public Key System**

Basis Fakt:

Es ist einfach, zufällige große Primzahlen p,q zu finden. (p,q 512 Bit)

Es ist praktisch unmöglich, diese Faktoren wieder aus dem Produkt **n=pq** zurückzugewinnen (Gilt nicht mehr für Quantencomputer!)

Modulare Arithmetik

$$Z_n = \{0,1,2,\ldots,n-1\}$$

Addition und Multiplikation werden „modulo n“ definiert:

Man addiert bzw. multipliziert zunächst „normal“, dividiert dann das Ergebnis durch n und nimmt als Ergebnis der Operation den Divisionsrest.

Beispiel

n=11

Operationen in Z_{11}

$7 + 6 = (13) = 2$

$7*6 = (42) = 9$

$7 = -4$

Potenzieren in Z_{11}

$$3^{10} = ((3^2)^2\ 3)^2 =$$
$$= ((-2)^2\ 3)^2 = 1$$

Durch diesen Gruppierungstrick wird erreicht, daß die Potenz auch für hundertstellige Exponenten errechenbar ist.

Erweiterter Euklidischer Algorithmus:

$x \in Z_n$ x,n teilerfremd

Man kann das Inverse von x in Z_n berechnen:

$y \in Z_n : xy = 1$

Beispiel: $5 \cdot 17 = 1$ in Z_{21}

RSA Formel, Satz von Euler

p,q seien Primzahlen
$n=pq$

$m=(p-1)(q-1)$
e teilerfremd zu m
d: $d \cdot e = 1$ in Z_m.

In Z_n gilt: $$(x^e)^d = (x^e)^d = x^{ed} = x$$

Interpretation:

$x \to y = x^e$ Verschlüsselung

$y \to x = y^d$ Entschlüsselung

(n,e) öffentlicher Schlüssel
(n,d) geheimer Schlüssel

Der geheime Exponent *d* ist aus *(n,e)* nur berechenbar, wenn man die Faktorisierung von n kennt.

Diese ist aber für hinreichend große *p,q* nicht berechenbar.

Damit sind alle Forderungen an ein Public Key System erfüllt.

- p:=nextprime(34271); 34273
- q:=nextprime(58927); 58937

- e:=257; 257
- d:=(1/e) mod ((p-1)*(q-1)); 1878386177

- n:=p*q; 2019947801

- y:=powermod(27,e,n); 1708925226

- powermod(y,d,n); 27

Um das RSA Verfahren zu brechen, muß $n=pq$ faktorisiert werden.

Dazu reicht es, die Ordnung eines invertierbaren Elements in Z_n berechnen zu können.

$$ord_n(x) = min\{k|x^k = 1\}$$

Beispiel: $ord_{15}\ 2 = 4$

Algorithmus:

1. Wähle ein zufälliges Element y in Z_n. Berechne mit dem **Shor Algorithmus** $r = ord_n(y)$. Wiederhole diesen Schritt, bis sich ein gerades r ergibt.

2. Es ist $(y^{r/2})\,(y^{r/2}) = y^r = 1$ in Z_n
   also ist n ein Teiler von $(y^{r/2}-1)\,(y^{r/2}+1)$

3. Wenn $ggT(y^{r/2}-1,n)$ kein Faktor von n, gehe zurück zu 1.

Beispiel:

Faktorisierung von $n = 15$:

1. $y=2, r = ord_{15}\ 2 = 4$

2. $ggT(y^{r/2}-1,n) = ggT(2^2-1,15) = 3$

Also ist 3 ein Faktor von 15.

Elemente des Shor Algorithmus

1. Quantenalgorithmus für Fouriertrans-formation.

2. Quantenalgorithmus zur Schätzung der Phase eines Quantenzustands.

Literatur:

Michael A. Nielsen, Isaac L. Chuang
Quantum Computation and
Quantum Information
Cambridge University Press

Bruce Schneier
Applied Cryptography
Wiley

Digitale Signatur:

Nachricht m

Inhaber des geheimen Schlüssels berechnet Signatur s:

$$m \rightarrow s = m^d$$

Signaturprüfer prüft:

$$m = ? \, s^e$$

Public Key Infrastruktur

Trustcenter signiert Zugehörigkeit
von Namen zu öffentlichen Schlüsseln
(Zertifikate, Standard X.509)

Signaturgesetz 1997, 2001
EU-Signaturrichtlinie 2000

Zertifizierte **kryptographische Chipkarten**

Bundesamt für Sicherheit in der Informationstechnik (**BSI**)

Diskreter Logarithmus

p sei eine Primzahl

Es gibt ein Element g von Z_p so daß alle von 0 verschiedenen Elemente von Z_p als Potenzen von g darstellbar sind.

Logarithmus Problem:
Gegeben x, berechne n, so daß $x=g^n$.

Beispiel:

Z_{23}
$g=7$

1	2	3	4	5	6	7	8	9	10	11
7	3	21	9	17	4	5	12	15	13	22

12	13	14	15	16	17	18	19	20	21	22
16	20	2	14	6	19	18	11	8	10	1

Um das Logarithmusproblem

$6 = 7^n$

in Z_{23} zu lösen und $n=16$ zu finden, muß man prinzipiell $n = 1,2,3$ ... ausprobieren, bis man auf die Lösung $n=16$ stößt.

Für hinreichend große Moduli (1024 Bit) ist dies mit klassischen Methoden nicht möglich.

Diffie Hellman Schlüsselaustausch

Schlüsselparameter
öffentlich: Primzahl p, Erzeuger g,
geheim: Zufallszahl k

öffentlicher Schlüssel
$x=g^k$

Falls p 1024 Bit, kann k aus x nicht berechnet werden

B will Nachricht an A senden.

B beschafft sich sich A‘s Parameter p,g,x, erzeugt Zufallszahl l, berechnet $y=x^l$, benutzt y als Schlüssel eines klassischen Chiffrier-systems und stellt der chiffrierten Nachricht g^l voran.

B kann den Schlüssel via $(g^l)^k = (g^k)^l = x^l$ berechnen. Ein Abhörer müßte ein DLP lösen.