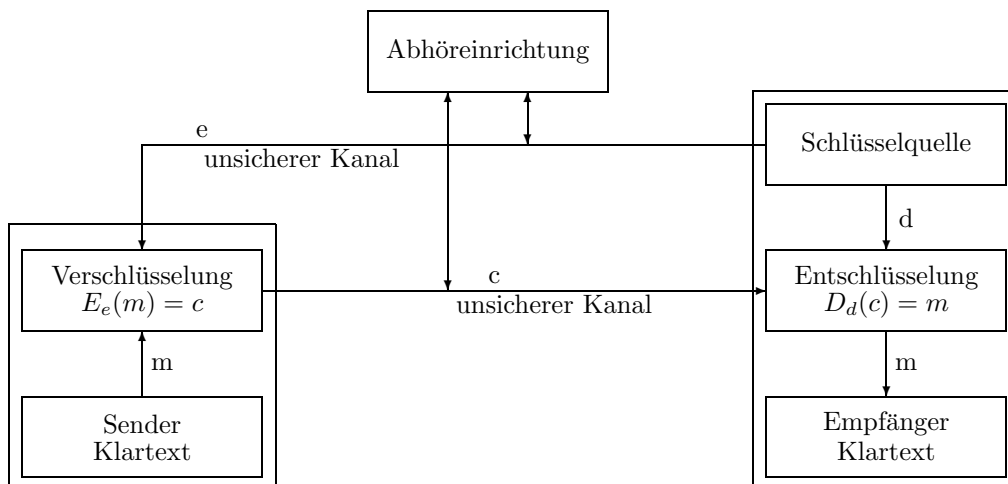


Konzepte von Betriebssystem-Komponenten:  
Schwerpunkt Sicherheit  
Grundlagen:  
Asymmetrische Verschlüsselung, Digitale Signatur

Rudi Pfister  
Rudi.Pfister@informatik.stud.uni-erlangen.de

## Public-Key-Verfahren

### Prinzip der Public-Key-Verfahren



Bei der Public-Key-Verschlüsselung erzeugt der Empfänger einen Chiffrierschlüssel  $e$  und einen Dechiffrierschlüssel  $d$ . Der Chiffrierschlüssel  $e$  wird öffentlich bekannt gegeben, so dass der Sender einer Nachricht den Klartext  $m$  mit dem  $e$ -Schlüssel des Empfängers verschlüsseln kann. Dazu bedient er sich des Algorithmus  $E_e(m) = c$ . Der Empfänger der chiffrierten Nachricht benutzt den Algorithmus  $D_d(c) = m$ , dazu benötigt er seinen privaten Schlüssel  $d$ . Zwar beinhalten die Schlüssel  $e$  und  $d$  Anweisungen, wie die zueinander inversen Vorgänge des Chiffrierens und Dechiffrierens auszuführen sind, aber sie sind so konstruiert, dass es nicht möglich ist mit realistischen Mitteln und in angemessener Zeit  $d$  aus  $e$  abzuleiten. Daher darf sowohl die verschlüsselte Nachricht  $c$ , als auch der Chiffrierschlüssel  $e$  über einen unsicheren Kanal transportiert werden.

## Public-Key-Algorithmen

Die Algorithmen, die zur asymmetrischen Verschlüsselung eingesetzt werden, beruhen auf nichtdeterministischen Problemen mit exponentiell anwachsender Lösungszeit. Sie sind dadurch charakterisiert, dass sich die Richtigkeit von Lösungen schnell prüfen lässt, aber es sehr zeitraubend ist systematisch nach Lösungen zu suchen.

Infolgedessen eignen sie sich zur Konstruktion von „Falltür-Funktionen“, d.h. Funktionen, die leicht zu berechnen sind, deren Inverses ohne weitere Informationen, jedoch kaum berechnet werden können.

Eine Verfahren und deren zugrundeliegenden mathematischen Probleme:

- **RSA** - Bezieht seine Sicherheit aus dem Faktorisierungsproblem, d.h. Problem der Zerlegung grosser Zahlen in ihre Primfaktoren.
- **Rabin** - Beruht auf dem Problem, Quadratwurzeln modulo  $n = p*q$  zu bestimmen; entspricht dem Faktorisierungsproblem. Dieses Verfahren war das erste, dessen Sicherheit bewiesen werden konnte.
- **ElGamal** - Problem der Berechnung diskreter Logarithmen, gilt als ähnlich schwierig wie das Faktorisierungsproblem.
- **McEliece** - Basiert auf fehlerkorrigierenden Codes, deren Entschlüsselung NP-vollständig ist. In der Praxis kaum verwendet, da die empfohlene Schlüssellänge zu lang ist.
- **Merkle-Hellmann Untersummen** - Nutzt das Untersummenproblem aus, jedoch wurde die Angreifbarkeit der meisten Varianten nachgewiesen.
- **Goldwasser-Micali Verfahren** - Rechnen mit quadratischen Resten modulo  $m$ , jedes Bit wird einzeln verschlüsselt, was dem Text sehr aufbläht.
- **Elliptische Kurven** - Bereits genannte Verfahren werden auf elliptische Kurven über endlichen Körpern angewendet.

## Sicherheit, Schlüssellänge, Effizienz

Da die mathematischen Probleme, die den Verschlüsselungsverfahren zugrunde liegen (wie z.B. die Faktorisierung von Zahlen), nach heutigem Kenntnisstand schwer zu lösen sind gelten die Public-Key-Verfahren bei ausreichender Schlüssellänge als sicher. Jedoch ist nicht bewiesen, dass es keine einfacheren Algorithmen, d.h. solche mit deutlich kürzerer Laufzeit, gibt. Auch muss die Schlüssellänge bei asymmetrischen Verfahren deutlich länger sein als bei symmetrischen, da öffentlicher und privater Schlüssel auseinander abgeleitet werden. Auch sind die Algorithmen zur asymmetrischen Verschlüsselung deutlich langsamer, als die Algorithmen der symmetrischen Verschlüsselung, weswegen die asymmetrische Verschlüsselung oft nur zur Übertragung symmetrischer Schlüssel eingesetzt wird.

# Digitale Signaturen

## Prinzip einer digitalen Signatur

Derjenige, der seine Nachrichten signieren möchte, erzeugt sich einen öffentlichen und einen privaten Schlüssel. Beim signieren einer Nachricht wird aus dem privaten Schlüssel und der Nachricht eine Signatur erzeugt und an die Nachricht angehängt. Der Empfänger kann anhand der Nachricht, der Signatur und des öffentlichen Schlüssels prüfen, ob die Nachricht wirklich vom angegebenen Sender stammt und Unterwegs nicht verändert wurde.

## Algorithmen für digitale Signaturen

- **DSA** - Das meist verbreitete Signaturverfahren, seine Sicherheit beruht auf dem Problem der Berechnung diskreter Logarithmen.
- **GOST** - Standardverfahren in Russland, ähnlich wie DSA.
- **Ong-Schnorr-Shamir** - Das Verfahren benutzt Polynome modulo  $n$ . Die meisten Varianten des Verfahrens gelten als unsicher.
- **ESIGN** - Gilt als genauso sicher wie RSA oder DSA und ist schneller.
- Weiterhin können einige **asymmetrische Verschlüsselungsverfahren** wie z.B. RSA durch Vertauschen von Chiffrieren und Dechiffrieren zur digitalen Signatur verwendet werden.

## Signieren mit DSA

### Globale öffentliche Parameter

Primzahl  $p$  (512 bis 1024 Bits Länge)

Primteiler  $q$  von  $(p - 1)$  (160 Bits)

$$g = h^{(p-1)/q} \text{ modulo } p$$

### Privater Schlüssel

$x$  Zufallszahl mit  $0 < x < q$

### Öffentlicher Schlüssel

$$y = g^x \text{ modulo } p$$

### Geheime Zufallszahl pro Nachricht

$k$  Zufallszahl mit  $0 < k < q$

### Signieren der Nachricht $m$

$$r = (g^k \text{ modulo } p) \text{ modulo } q$$

$$s = (k^{-1}(H(m) + xr)) \text{ modulo } q, \text{ wobei } H \text{ eine Ein-Weg-Hash-Funktion ist}$$

$r$  und  $s$  bilden die Signatur, die zusammen mit der Nachricht übermittelt werden.

### Verifizieren

$$w = s^{-1} \text{ modulo } q$$

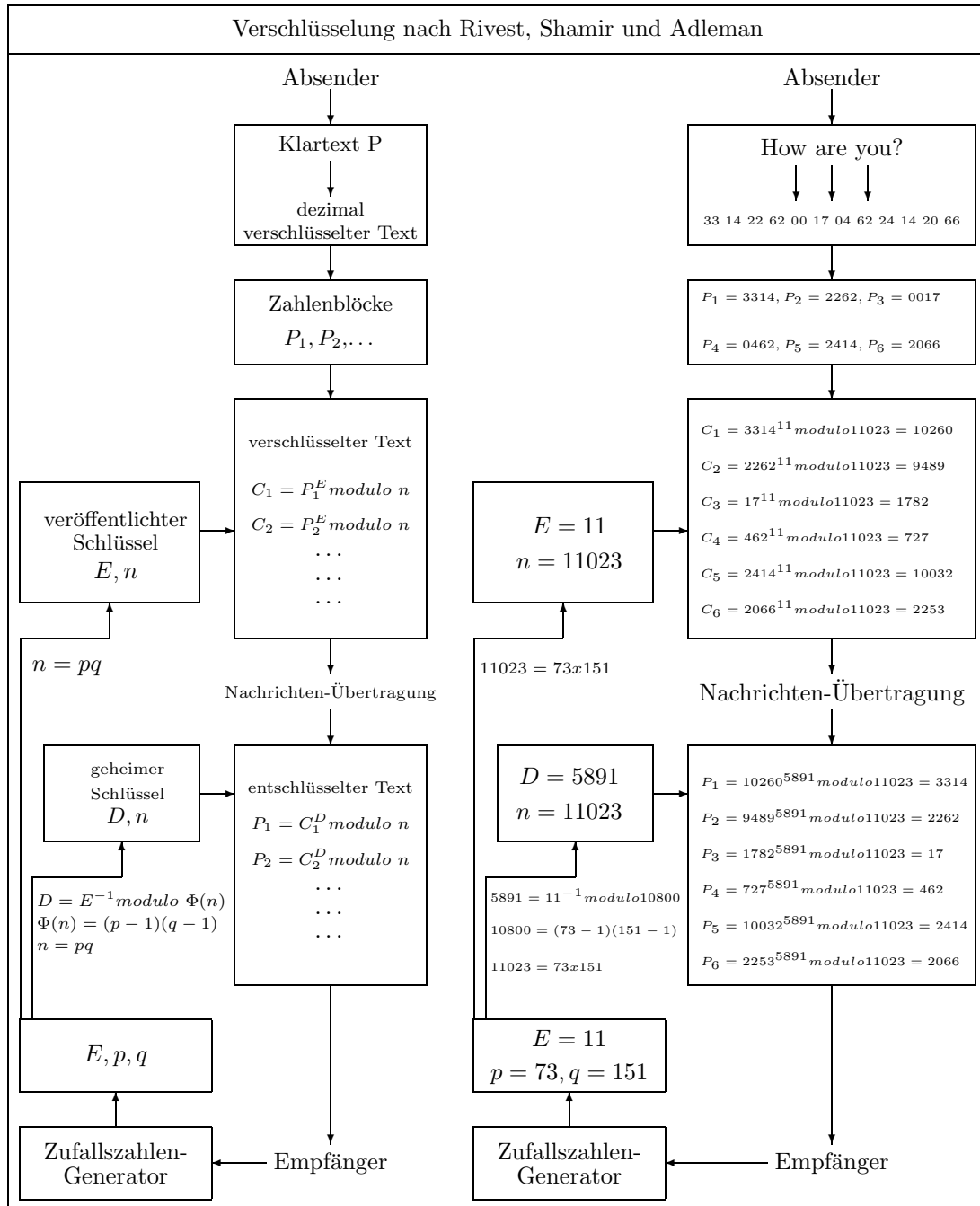
$$u_1 = (H(m) * w) \text{ modulo } q$$

$$u_2 = (rw) \text{ modulo } q$$

$$v = ((g^{u_1} * y^{u_2}) \text{ modulo } p) \text{ modulo } q$$

Wenn  $v = r$ , dann ist die Nachricht verifiziert.

# RSA - Beispiel für ein asymmetrisches Verschlüsselungsverfahren



Der Empfänger RSA-verschlüsselter Nachrichten erzeugt sich zunächst per Zufallszahlen-generator zwei geeignete Primzahlen  $p$  und  $q$  und eine grosse natürliche Zahl  $E$  mit  $1 < E < \Phi(n)$  und  $ggT(E, \Phi(n)) = 1$ . Die Zahlen  $p$  und  $q$  dürfen nur dem Empfänger

bekannt sein, während ihr Produkt  $n$  zusammen mit der Zahl  $E$  als öffentlicher Schlüssel bekannt gemacht wird. Der Absender einer zu verschlüsselnden Nachricht formt diese zunächst in einer Weise, die auch dem Empfänger bekannt sein muss, in eine Kette von Dezimalzahlen um und zerlegt die Kette dann in gleichlange Blöcke  $P_1, P_2, \dots$ , so dass gilt  $\forall P_i : P_i < n$ . Danach wird jeder Block in eine chiffrierte Zahl umgewandelt, indem man ihn in die  $E$ -te Potenz erhebt und dann modulo  $n$  reduziert, also alle  $C = P^E \text{ modulo } n$  bildet. Die so verschlüsselte Nachricht wird über einen unsicheren Kanal an den Empfänger geleitet. Eine Funktion mit Exponent, die modulo  $n$  enthält, lässt sich nur dann mühlos auswerten, wenn man ihren Exponenten modulo  $\Phi(n)$  berechnet, wobei  $\Phi(n) = (p-1)(q-1)$  ist. Die Zahlen  $p$  und  $q$  und damit  $\Phi(n)$  kennt aber nur der rechtmäßige Empfänger der Nachricht. Nur er kann also den Dechiffrierschlüssel  $D = E^{-1} \text{ modulo } \Phi(n)$  berechnen. Beim Dechiffrieren erhebt der Empfänger jede Zahl  $C_i$  des chiffrierten Textes in die  $D$ -te Potenz und reduziert modulo  $n$ . Da  $C_i^D \text{ modulo } n$  gleich  $(P_i^E)^D \text{ modulo } n$  oder  $P_i^{ED} \text{ modulo } n$  ist und da  $ED \text{ modulo } \Phi(n)$  gleich Eins ist, führt die Operation  $P_i^{ED} \text{ modulo } n$  wieder auf die Zahlenblöcke  $P_1, P_2, \dots$  des Klartextes. Diese Zahlen können dann wieder in die ursprüngliche Nachricht verwandelt werden.

#### **Ammerkungen zur Sicherheit von RSA**

Da, wenn die Faktorisierung von  $n$  möglich ist RSA gebrochen werden kann, sollte, um die klassischen Faktorisierungsalgorithmen abwehren zu können,  $n$  mindestens 1024 Bit lang sein (Tendenz steigend) und  $p$  und  $q$  die gleiche Bitlänge aufweisen, aber auch nicht zu dicht zusammenliegen. Weiterhin sollten  $p-1, p+1, q-1$  und  $q+1$  grosse Primfaktoren enthalten. Es ist auch noch nicht bekannt ob es einfachere Faktorisierungsalgorithmen gibt, es hat zwar seit den siebziger Jahren beachtliche Fortschritte in diesem Gebiet gegeben, aber der grosse Durchbruch, der das Ende von RSA bedeuten würde, ist noch nicht in Sicht.

## **Literatur**

- [1] Applied Cryptography, Bruce Schneier, Wiley, 1996
- [2] Handbook of Applied Cryptography, Menezes/von Oorschot/Vanstone, CRC Press, 1997
- [3] <http://home.in.tum.de/~gerold/neustefassung/gliederung.html>
- [4] <http://www-lehre.informatik.uni-osnabrueck.de/~aerpenbe/sec/book1.htm>
- [5] Die Mathematik von Public-Key-Verfahren, Martin E. Hellmann, Spektrum der Wissenschaft - Dossier: Kryptographie, 4/2001