IT-Sicherheit

- Sicherheit vernetzter Systeme -

Kapitel 6: Asymmetrische und Hybride Kryptosysteme

Inhalt

- Asymetrische Kryptosysteme
  - RSA
  - Sicherheit von RSA
- Schlüssellängen und Schlüsselsicherheit
- Hybride Kryptosysteme
- Digitale Signatur

Wdh.: Kryptographisches System (Def.)

- Geg. zwei endliche Zeichenvorräte (Alphabete) A_1 und A_2
- Ein Kryptosystem (KS) ist gegeben durch ein Tupel

$$KS = (M, C, EK, DK, E, D)$$

1. Nicht leere endliche Menge von Klartexten $M \subseteq A_1^*$
2. nicht leere endliche Menge von Krypto- bzw. Chiffretexten $C \subseteq A_2^*$
3. der nicht leeren Menge von Verschlüsselungsschlüsseln EK
4. der nicht leeren Menge von Entschlüsselungsschlüsseln DK sowie einer Bijektion $f : EK \rightarrow DK$ mit $f(K_E) = K_D$
5. Injektives Verschlüsselungsverfahren $E : M \times EK \rightarrow C$
6. Entschlüsselungsverfahren $D : C \times DK \rightarrow M$ mit $\forall m \in M : D(E(m, K_e), K_d)) = m$

Asymmetrisches Kryptosystem, Def.

- $KS = (M, C, EK, DK, E, D)$
- Schlüsselpaare müssen leicht (effizient) zu erzeugen sein, und es muss gelten:
  - $K_E = f(K_D), mit\ K_E \in EK,\ K_D \in DK$
  - $\forall m \in M: \quad D(E(m, K_E), K_D) = m$
  - K_E kann öffentlich bekannt gemacht werden
- Ver- und Entschlüsselungsfunktion E und D effizient zu berechnen
- K_D aus K_E nicht mit vertretbarem Aufwand berechenbar; Umkehrfunktion von f schwer zu berechnen; (Stichwort: **Einwegfunktionen**)
- Kryptosystem zur digitalen Signatur geeignet falls zus. gilt:
  - $\forall m \in M: \quad E(D(m, K_D), K_E) = D(E(m, K_E), K_D) = m$

RSA

- Benannt nach den Erfindern: Rivest, Shamir, Adleman (1978)
- Sicherheit basiert auf dem Faktorisierungsproblem:
  - Geg. zwei große Primzahlen p und q (z.B. 200 Dezimalstellen):
  - n=pq ist auch für große Zahlen einfach zu berechnen,
  - aber für gegebenes n ist dessen Primfaktorzerlegung sehr schwierig
- Erfüllt alle Anforderungen an asymmetrisches Kryptosystem
- In USA patentiert (Patent ist 2000 ausgelaufen)
- Große Verbreitung, verwendet in:
  - SSL (Secure Socket Layer)
  - PEM (Privacy Enhanced Mail)
  - PGP (Pretty Good Privacy)
  - GNUpg
  - ....

RSA Mathematische Grundlagen

- Restklassenarithmetik
- Restklassenring modulo m
- Multiplikatives Inverses
- Eulersche φ Funktion
- kleiner Fermatscher Satz
- Einweg-Funktion (One way function)
- Einweg-Funktion mit Falltür (Trapdoor one way function)

Vgl. Tafel

RSA Funktionsweise

1. Wähle große Primzahlen p und q und berechne $n = pq$
   n wird als RSA-Modul bezeichnet
2. Wähle $d \in [0, n-1]$ so dass d relativ prim ist zu $\varphi(n)$
   - D.h. $ggT(\varphi(n), d) = 1$
   - $\varphi(n) = (p-1)(q-1)$

   Wähle d so dass gilt: $\max(p,q) < d < \varphi(n) - 1$
3. Wähle $e \in [0, n-1]$ mit $e \times d \equiv 1 \bmod \varphi(n)$
   d.h. e ist das multiplikative Inverse modulo $\varphi(n)$ zu d
4. Öffentlicher Schlüssel (e, n)
5. Geheimer Schlüssel (d, n)
6. Verschlüsseln eines Klartextes $m \in [0, n-1]$

$$E(m) = m^e \bmod n = c$$

7. Entschlüsseln:

$$D(c) = c^d \bmod n$$

Erfüllt RSA Anforderungen an asym. Kryptosystem?

- Es sind zwei Sachverhalte zu zeigen:

1. Der Chiffrentext kann auch wieder entschlüsselt werden:

$$D(E(m)) = m$$

2. Das Verfahren eignet sich zur digitalen Signatur

$$D(E(m)) = E(D(m))$$

- Beweis vgl. Tafel

Einschub: Nomenklatur für kryptologische Verfahren

- Für Verschlüsselungsverfahren wird künftig die folgende Notation verwendet:

A_p	Öffentlicher (public) Schlüssel von A
A_s	Geheimer (secret) Schlüssel von A
A_p{m}	Verschlüsselung der Nachricht m mit dem öffentlichen Schlüssel von A
A_s{m} oder A{m}	Von A erstellte digitale Signatur von m
S[m]	Verschlüsselung von m mit dem **symmetrischen** Schlüssel S

RSA Sicherheit / mögliche Angriffe

1. **Brute force**: Testen aller möglichen Schlüssel
2. **Chosen-Cipertext Angriff**: (vgl. Tafel)
3. **Mathematische Angriffe**:
   - Faktorisierung von n;
   - direkte Bestimmung von $\varphi(n)$ ohne Faktorierung;
   - direkte Bestimmung von d ohne Bestimmung von $\varphi(n)$
4. **Timing Angriff:** Info: [Koch 96, Kali 96]
   - Überwachung der Laufzeit von Entschlüsselungsoperationen
   - Über Laufzeitunterschiede kann privater Schlüssel ermittelt werden
   - Analogie: Einbrecher ermittelt Kombination eines Tresors mit Hilfe der mitgehörten Stellzeiten am Zahlenschloß
5. **Angriffe auf Signaturen** (vgl. spätere Folien zur dig. Signatur)
   - Existentielle Fälschung
   - Multiplikativität von RSA

RSA: Chosen-Ciphertext Angriff

Vgl. Tafel

RSA Mathematische Angriffe

- Mathematische Angriffe lassen sich auf Faktorisierung zurückführen
- Schnellster bekannter Algorithmus General Number Field Sieve (GNFS), vgl. [Silv 01]
  - Laufzeitkomplexität: $L(N) = e^{(c+o(1))\bullet\sqrt[3]{\log(N)}\bullet\sqrt[3]{\log(\log(N))^2}}$
  - Speicherplatzkomplexität: $\sqrt{L(N)}$
- Faktorisierung wird einfacher falls:
  - Anzahl der Ziffern von p und q große Unterschiede aufweisen (z.B. |p| = 10 und |q| = 120)
  - Falls $d < 1/3 \bullet \sqrt[4]{n}$ kann d leicht berechnet werden
  - Die ersten m/4 Ziffern oder die letzten m/4 Ziffern von p oder q sind bekannt

Inhalt

- Asymetrische Kryptosysteme
  - RSA
  - Sicherheit von RSA
- Schlüssellängen und Schlüsselsicherheit
- Hybride Kryptosysteme
- Digitale Signatur

Wie lang muss ein sicherer Schlüssel sein? Einflussfaktoren

- Symmetrisches oder Asymmetrisches Verfahren ?
- Algorithmus
- PC / Software basierter Angriff
- Angriff mit integrierter Schaltung (ASIC, application specific integrated circuit)
- Angriff mit pogrammierbarer integrierten Schaltung (FPGA, field programmable gate array)
- GPGPU (General-purpose computing on graphics processing untis)
- Kosten und Ressourcenbedarf

Angriffe auf symmetrische Kryptosysteme

- Brute-Force Angriff
  - Durchsuchen des gesamten Schlüsselraums
  - Im Mittel ist halber Schlüsselraum zu durchsuchen
- Referenzzahlen; Größenordnungen (gerundet)

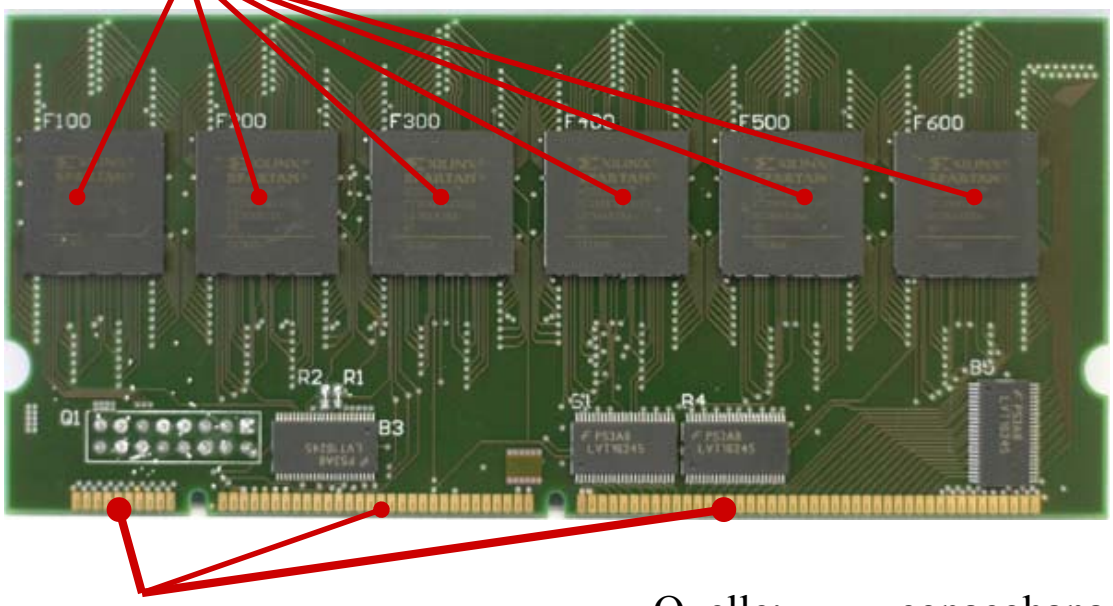
Tag: 86400
Jahr 31.536.000

	Größenordnu
Sekunden in einem Jahr	$3 * 10^7$
Alter des Universums in Sekunden	$4 * 10^{17}$
Schlüsselraum bei 64 Bit Schlüssellänge	$2 * 10^{19}$
Masse des Mondes [kg]	$7 * 10^{22}$
Masse der Erde [kg]	$6 * 10^{24}$
Masse der Sonne [kg]	$2 * 10^{30}$
Schlüsselraum bei 128 Bit Schlüssellänge	$3 * 10^{38}$
Anzahl Elektronen im Universum	10^{77} - 10^{79}

PC versus FPGA basierter Angriff

- FPGA Implementierung Copacobana (www.copacobana.org)

6x Spartan 3 FPGA (xc3s1000, FT256 packaging)



Connection to backplane (64-bit data bus)

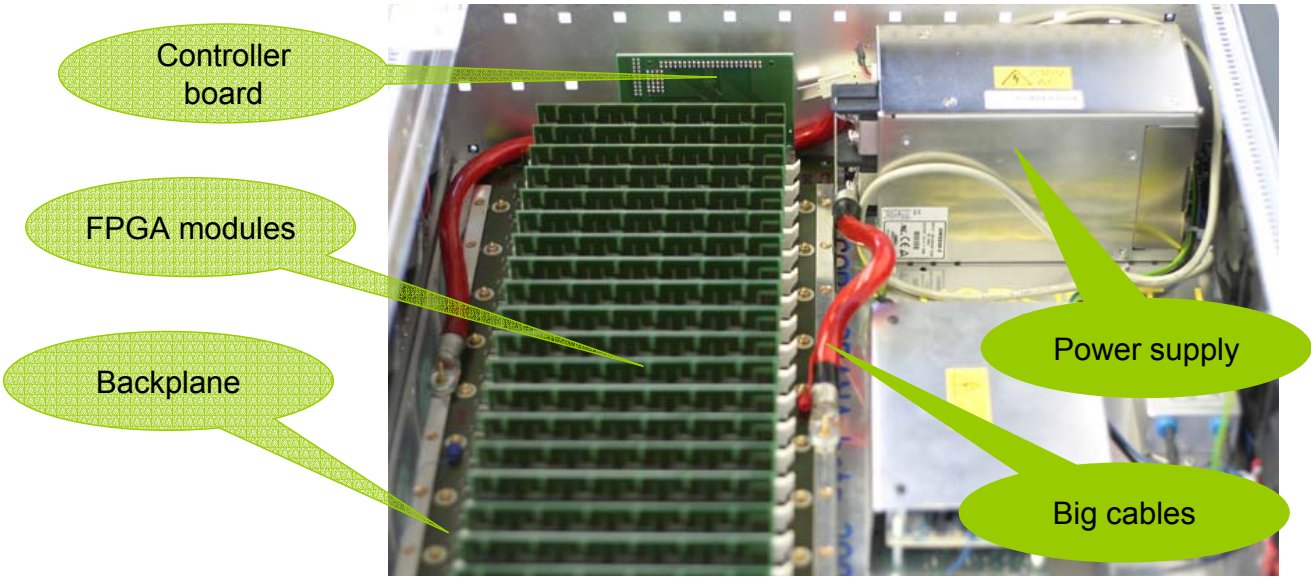
Quelle: www.copacobana.org

Copacobana

[Pelzl 2006]

Copacobana: Innenleben

- 20 Module pro Maschine mit 120 FPGAs



[Pelzl 2006]

DES: Brute Force Angriff

- Pentium 4; 3 GHz: ca. $2 * 10^6$ Schlüssel/s
- Copacabana;
  - 2006: $4{,}793 * 10^{10}$ Schlüssel/s
  - 2007: $6{,}415 * 10^{10}$ Schlüssel/s

Schlüssellänge [Bit]	#Schlüssel	durchschnittliche Zeit [s]		
		2006	2007	Pentium PC
40	$1{,}1 * 10^{12}$ ($5{,}5 * 10^{11}$)	11,5	8,6	274.878 3,18 d
56	$7{,}2 * 10^{16}$	751.680 8,7 d	561.600 6,5 d	$1{,}8 * 10^{10}$ 571 Jahre
128	$3{,}4 * 10^{38}$	$3{,}55 * 10^{27}$ $1{,}12 * 10^{20}$ J.	$2{,}65 * 10^{27}$ $8{,}4 * 10^{19}$ J	$8{,}5 * 10^{31}$ $2{,}6 * 10^{24}$ J.

DES Brute Force Angriff; Kosten

- Ziel: DES im Mittel in 8,7 Tagen brechen (Stand 2006)
- Dafür rd. 24.000 Pentium (a 150 €) erforderlich:
  3,6 Mio €
- Copacabana: **9.000 €**

AES: Brute Force

- Schlüssellänge 128 Bit
- FPGA Implementierung schaffen 22 Gb/s Durchsatz (2007)
- $22 * 2^{30} / 128 = 1{,}845 * 10^{8}$ Schlüssel/s

Schlüssellänge	Zeit [s]
128	$9{,}2 * 10^{29}$

RSA Schlüssellängen

- RSA Challange: Belohnung für das Brechen von RSA Schlüsseln, z.B. durch Faktorisierung (2007 eingestellt)

Dezimalstellen	Bits	Datum	Aufwand	Algorithmus
100	332	April 1991	7 Mips Jahre	Quadratisches Sieb
110	365	April 1992	75 Mips J.	
120	398	Juni 1993	830 Mips J.	
129	428	April 1994	5000 Mips J.	
130	431	April 1996	1000 Mips J.	General Number Field Sieve (GNFS)
140	465	Februar 1999	2000 Mips J.	
155	512	August 1999	8000 Mips J.	
160	530	April 2003	k.A.	GNFS(Lattice Sieve)
174	576	Dez. 2003	k.A.	GNFS(Lattice/Line Sieve)
193	640	Nov. 2005	30 2,2 GHz Opteron J.*	GNFS

* Halb so lange wie für RSA-200 Challange

RSA Schlüssellängen

- RSA Challange: Belohnung für das brechen von RSA Schlüsseln, z.B. durch Faktorisierung (2007 eingestellt)

Dezimalstellen	Bits	Datum	Aufwand	Algorithmus
100	332	April 1991	7 Mips Jahre	Quadratisches Sieb
110	365	April 1992	75 Mips J.	
120	398	Juni 1993	830 Mips J.	
129	428	April 1994	5000 Mips J.	
130	431	April 1996	1000 Mips J.	
140	465	Feb. 1999	2000 Mips J.	
155	512	August 1999	8000 Mips J.	
160	530	April 2003	k.A.	GNFS(Lattice Sieve)
174	576	Dez. 2003	k.A.	GNFS(Lattice/Line Sieve)
193	640	Nov. 2005	30 2,2 GHz Opteron J.*	GNFS

d.h. 512 Bit Schlüssel ist nicht mehr sicher!

* Halb so lange wie für RSA-200 Challange

Schlüsselsicherheit symmetrisch vs. asymmetrisch

- Verschiedene Institutionen geben Vergleiche heraus Bits of Security (äquiv. Schlüssellänge symmetrischer Verfahren)
  - NIST (National Institute of Standards and Technology) 2007:

Bits of Security	80	112	128	192	256
Modullänge (pq)	1024	2048	3072	7680	15360

  - NESSIE (New European Schemes for Signatures, Integrity and Encryption) (2003)

Bits of Security	56	64	80	112	128	160
Modullänge (pq)	512	768	1536	4096	6000	10000

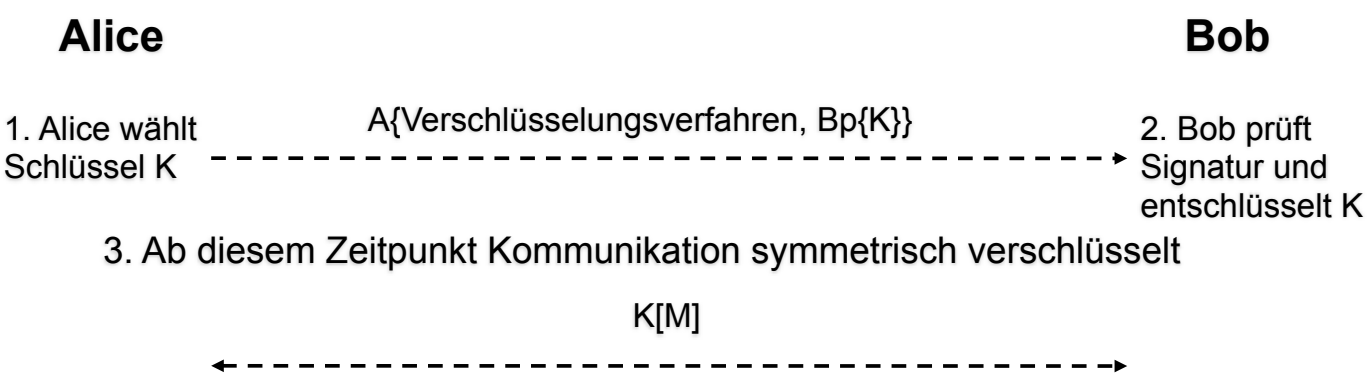
Empfohlene Schlüssellängen

- Quelle: ECRYPT 2006: Report on Algorithms and Keysizes
- Minimale Schlüssellängen:

Angreifer	Budget	Hardware	min. Anzahl Bits of Security
„Hacker“	0	PC	52
	< $ 400	PC(s)/FPGA	57
	0	„Malware“	60
Small organization	$ 10k	PC(s)/FPGA	62
Medium organization	$ 300k	FPGA/ASIC	67
Large organization	$ 10M	FPGA/ASIC	77
Intelligence agency	$ 300M	ASIC	88

Hybride Kryptosysteme

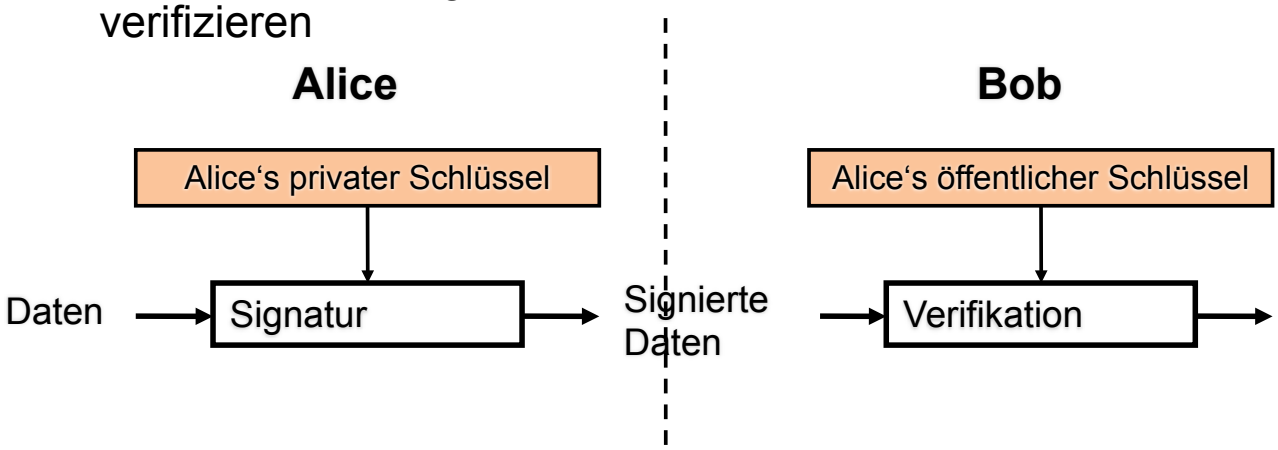
- Vereinen Vorteile von Symmetrischen und asymmetrischen Verfahren
- Asymmetrisches Verfahren zum Schlüsselaustausch
- Symmetrisches Verfahren zur Kommunikationsverschlüsselung



- Beispiele für hybride Verfahren: PGP, ssh,...

Digitale Signatur

- Alice „signiert“ Daten mit ihrem privaten Schlüssel
- Jeder kann die Signatur mit Alice‘ öffentlichem Schlüssel verifizieren



- Assymetrische Verfahren sind im Vergleich sehr langsam
- Daher i.d.R. nicht Signatur der gesamten Daten
- Lediglich kryptographischer Hash-Wert der Daten wird signiert (digitaler Fingerabdruck der Daten)

Digitale Signatur: Analogie zur Unterschrift

- Anforderungen an die (analoge) Unterschrift:

1. **Perpetuierungsfunktion:** Unterschrift kann nicht gefälscht werden und ist dauerhaft
2. **Echtheitsfunktion:** Die Unterschrift ist authentisch
3. Die Unterschrift kann **nicht wiederverwendet** werden.
4. **Abschlußfunktion:** Unterschrift kann später nicht verändert werden
5. **Beweisfunktion:** Unterzeichner kann seine Unterschrift später nicht leugnen

- Bei der Unterschrift auf Papier ist keine dieser Anforderungen vollständig erfüllt!
- Trotzdem wird die Unterschrift im Rechtsverkehr akzeptiert
- Ihre Funktion wird durch Rahmenbedingungen gesichert

Digitale Signatur: Erfüllung der Anforderungen?

1. **Perpetuierungsfunktion:** Fälschungssicher und dauerhaft
2. **Echtheitsfunktion:** authentisch
3. **Nicht wiederverwendbar**
4. **Abschlußfunktion:** nicht veränderbar
5. **Beweisfunktion:** Unterschrift nicht leugnen

- Lassen sich diese Funktionen bei der digitalen Signatur realisieren?

1. Solange privater Schlüssel geheim gehalten wird
2. Abhängig von zweifelsfreier Zuordnung des Schlüsselpaares zu einer Identität (Zertifizierung, CA)
3. Digitale Signatur „beinhaltet“ den Dateninhalt
4. vgl. 3.
5. Jeder kann Signatur bzw. Echtheit mit öffentlichem Schlüssel des Unterzeichners verifizieren

RSA Signaturangriffe

- Existentielle Fälschung:
  - Mallet wählt beliebige Zahl r mit 0 <= r < n
  - M behauptet r sei ein von Alice signiertes Dokument (z.B. abzuhebender Geldbetrag)
  - Empfänger verifiziert und falls m=A_p{r} sinnvollen Text ergibt, war der Angriff erfolgreich
- Multiplikativität von RSA
  - Mallet wählt zwei Nachrichten m1 und m2 und lässt Alice signieren
  - Mallet berechnet A{m1}*A{m2} mod n
  - Es gilt $A\{m1\} \bullet A\{m2\} \bmod n = (m1 \bullet m2)^{As} \bmod n$
  - D.h. M kann aus zwei signierten Dokumenten ein drittes, gültig signiertes Dokument erzeugen, ohne im Besitz des Schlüssels von Alice zu sein
- Gegenmaßnahme:
  Nicht das gesamte Dokument, sondern nur Hash-Wert signieren!