IT-Sicherheit

- Sicherheit vernetzter Systeme -

Kapitel 6: Asymmetrische und hybride Kryptosysteme

Inhalt

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

Wdh.: Kryptosystem: Asymmetrische Verfahren

- Jeder Partner besitzt **Schlüsselpaar** aus
  - persönlichem, **geheim** zu haltenden **Schlüssel** (*private key*)
    (wird NIE übertragen)
  - und **öffentlich** bekannt zu gebenden **Schlüssel** (*public key*)
    (kann über unsichere und öffentliche Kanäle übertragen werden)
- Protokoll:
  1. Alice und Bob erzeugen sich Schlüsselpaare: (k_e^A, k_d^A) (k_e^B, k_d^B)
  2. Öffentliche Schlüssel (k_e^A, k_e^B) werden geeignet öffentlich gemacht
  3. Alice will m an Bob senden; dazu benutzt sie Bobs öffentlichen Schlüssel
     $$c = e(m, k_e^B)$$
  4. Bob entschlüsselt die Nachricht mit seinem privaten Schlüssel:
     $$m = d(c, k_d^b) = d(e(m, k_e^b), k_d^b)$$
- Beispiele: RSA, DSA, ElGamal, ...

Asymmetrische Kryptosysteme: Zielsetzung

- Effizienz / Performanz:
  - Schlüsselpaare sollen „einfach“ zu erzeugen sein.
  - Ver- und Entschlüsselung soll „schnell“ ablaufen.
- Veröffentlichung von k_e darf keine Risiken mit sich bringen
- Privater Schlüssel k_d darf nicht „einfach“ aus k_e ableitbar sein
  - D.h. Funktion f mit $f(k_d) = k_e$ soll nicht umkehrbar sein („Einwegfunktion“)

- Einsatz zur Verschlüsselung:
  - Alice schickt Nachricht m mit Bobs Public Key verschlüsselt an Bob
  - Bob entschlüsselt den empfangenen Chiffretext mit seinem privaten Schlüssel

- Einsatz zur elektronischen Signatur:
  - Alice verschlüsselt ein Dokument mit ihrem privaten Schlüssel
  - Bob entschlüsselt das Dokument mit Alices öffentlichem Schlüssel

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
- 1983 (nur) in USA patentiert (im Jahr 2000 ausgelaufen)
- Große Verbreitung, verwendet in:
  - SSL (Secure Socket Layer)
  - PEM (Privacy Enhanced Mail)
  - PGP (Pretty Good Privacy)
  - gpg (GNU Privacy Guard)
  - SSH
  - ....

Überblick über den Ablauf von RSA

- Erzeugung eines Schlüsselpaars
- Verschlüsselung
- Entschlüsselung

RSA: Erzeugung eines Schlüsselpaars

- Randomisierte Wahl von zwei ähnlich großen, unterschiedlichen Primzahlen, p und q
- n = pq ist sog. RSA-Modul
- Euler'sche Phi-Funktion gibt an, wie viele positive ganze Zahlen zu n teilerfremd sind: $\Phi(n) = (p-1)(q-1)$
- Wähle teilerfremde Zahl e mit $1 < e < \Phi(n)$
  d.h. der größte gemeinsame Nenner von e und $\Phi(n)$ = 1
  - Für e wird häufig 65537 gewählt: Je kleiner e ist, desto effizienter ist die Verschlüsselung, aber bei sehr kleinen e sind Angriffe bekannt.
  - Der öffentliche Schlüssel besteht aus dem RSA-Modul n und dem Verschlüsselungsexponenten e.
- Bestimme Zahl d als multiplikativ Inverse von e bezüglich $\Phi(n)$

$$d = e^{-1} \bmod \Phi(n)$$

  - Berechnung z.B. über den erweiterten Euklidischen Algorithmus
  - n und d bilden den privaten Schlüssel; d muss geheim gehalten werden

RSA: Ver- und Entschlüsselung

- Alice kommuniziert ihren öffentlichen Schlüssel (n,e) geeignet an Bob (Ziel hier: Authentizität von Alice, nicht Vertraulichkeit!)

- Bob möchte Nachricht M verschlüsselt an Alice übertragen:
  - Nachricht M wird als Integer-Zahl m aufgefasst, mit $0 < m < n$
    d.h. Nachricht m muss kleiner sein als das RSA-Modul *n*
  - Bob berechnet Ciphertext $c = m^e \pmod{n}$
  - Bob schickt c an Alice

- Alice möchte Ciphertext c entschlüsseln
  - Alice berechnet hierzu $m = c^d \pmod{n}$
  - Aus Integer-Zahl m kann Nachricht M rekonstruiert werden.

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**:
   - Ausprobieren aller möglichen Schlüssel
   - Entspricht Zerlegung von n in die Faktoren p und q
   - Dauert bei großen p und q mit heutiger Technik hoffnungslos lange

2. **Chosen-Ciphertext Angriff (Davida 1982)**:
   - Angreifer möchte Ciphertext c entschlüsseln, also $m = c^d \pmod n$ berechnen
   - Angreifer kann einen Ciphertext c‘ vorgeben und bekommt m‘ geliefert
   - Angreifer wählt $c' = s^e c \pmod n$, mit Zufallszahl s
   - Aus der Antwort $m' = c'^d \pmod n$ kann $m = m's^{-1}$ rekonstruiert werden.

RSA-Sicherheit / mögliche Angriffe

3. **Timing-Angriff:** [Kocher 1995]
   - Überwachung der Laufzeit von Entschlüsselungsoperationen
   - Über Laufzeitunterschiede kann privater Schlüssel ermittelt werden
   - Gegenmaßnahme: Blinding; Alice berechnet statt $c^d \pmod n$ mit einmaliger Zufallszahl *r*

     $$(r^e c)^d \pmod n = r c^d \pmod n$$

     und multipliziert das Ergebnis mit der Inversen von r.
   - Folge: Dauer der Entschlüsselungsoperationen hängt nicht mehr direkt nur von *c* ab, Timing-Angriff scheitert.

4. **Angriffe auf Signaturen** (vgl. spätere Folien zur dig. Signatur)
   - Multiplikativität von RSA $m^e r^e = (mr)^e$ erlaubt die Konstruktion gültiger Signaturen für ein Dokument, das aus korrekt signierten Teildokumenten zusammengesetzt ist.

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: $\sqrt{L(N)}$
- Angriffe werden ggf. einfacher:
  - Wenn die Anzahlen 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.
- Vgl. [Boneh 1999]: Twenty Years of attacks on the RSA cryptosystem, `http://crypto.stanford.edu/~dabo/pubs/papers/RSA-survey.pdf`

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-/softwarebasierter Angriff, oder
- Angriff mit dedizierter Hardware
  - Angriff mit integrierter Schaltung (ASIC, application specific integrated circuit)
  - Angriff mit programmierbarer integrierten Schaltung (FPGA, field programmable gate array)
  - GPGPU (General-purpose computing on graphics processing units)

- Kosten und Ressourcenbedarf

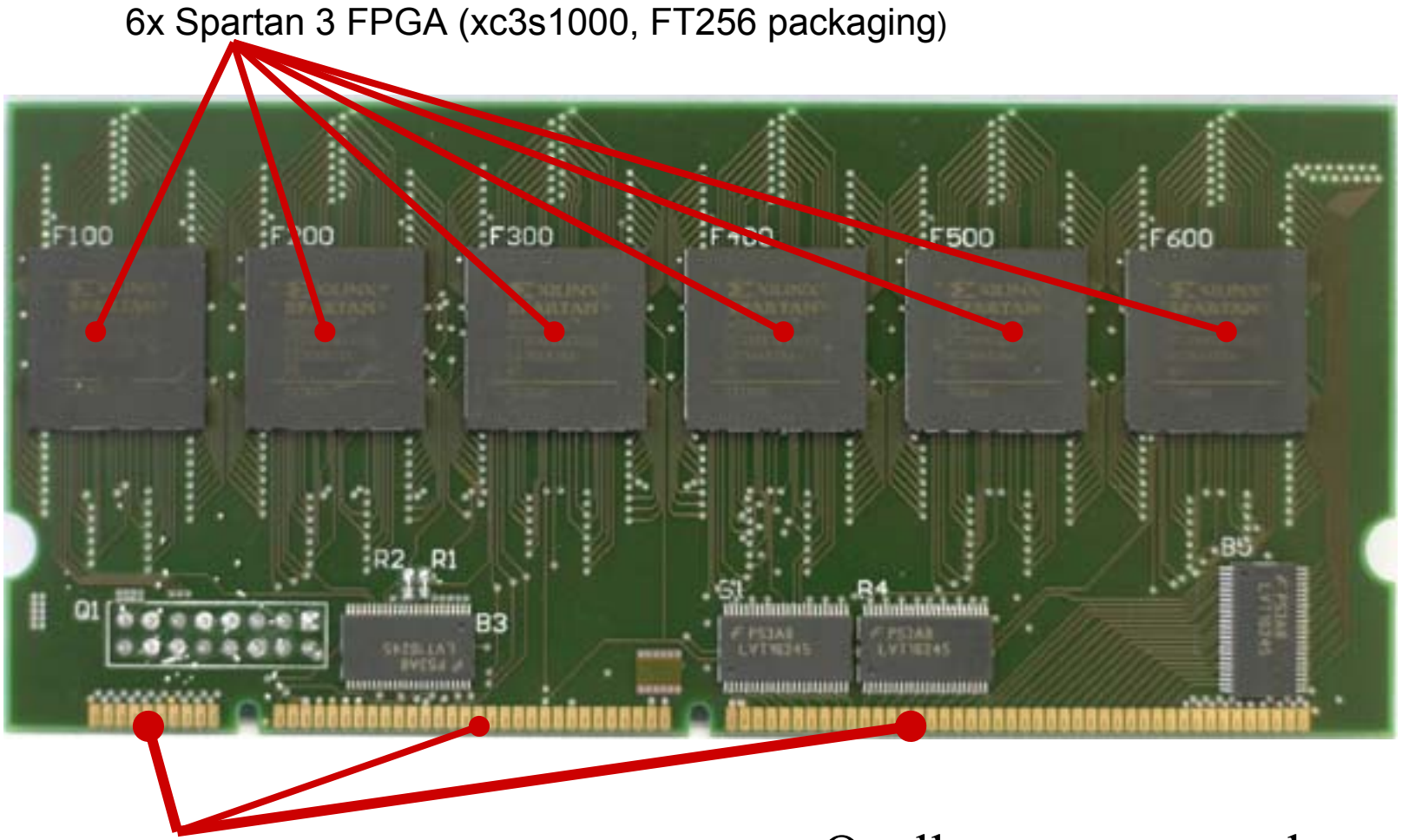
Angriffe auf symmetrische Kryptosysteme

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

	Größenordnung
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)



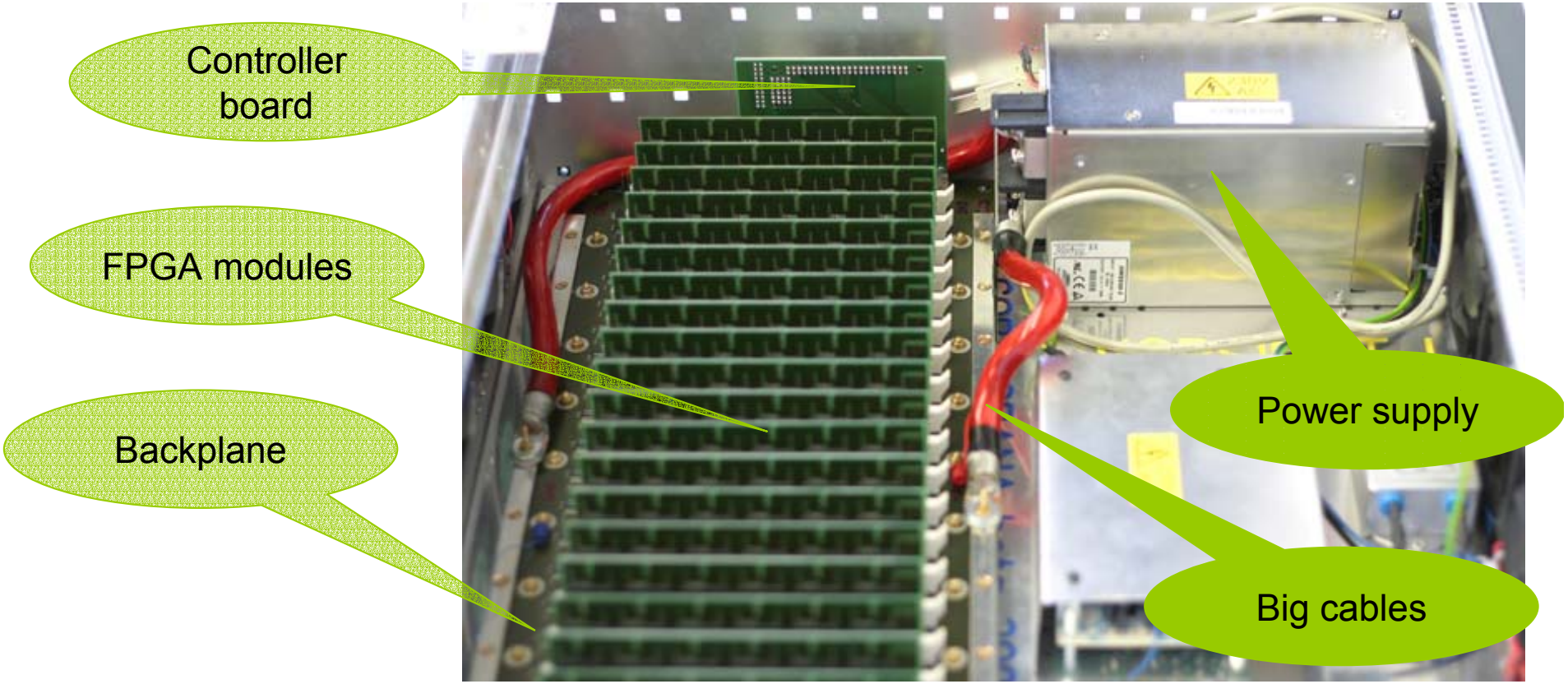
Quelle: www.copacobana.org

Copacobana

[Pelzl 2006]

Copacobana: Innenleben

- 20 Module pro Maschine mit 120 FPGAs



[Pelzl 2006]

DES: Brute Force Angriff (Stand: 2007)

- Pentium 4; 3 GHz: ca. $2 * 10^6$ Schlüssel/s
- Copacobana;
  - 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 (Stand 2006)

- Ziel: DES im Mittel in 8,7 Tagen brechen
- Dafür rd. 24.000 Pentium (je 150 €) erforderlich: **3,6 Mio €**
  Stromverbrauch: ca. 60 Watt pro CPU (~1,4 MW gesamt)
- Copacobana: **9.000 €**
  Stromverbrauch: 600 Watt (insgesamt)

AES: Brute Force

- Schlüssellänge 128 Bit
- FPGA Implementierung schafft 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 Challenge: Belohnung für das Brechen von RSA Schlüsseln, z.B. durch Faktorisierung

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-Jahre	GNFS

RSA-Schlüssellängen (Fortsetzung)

- RSA Challenge wurde 2007 eingestellt
  - rund $30.000 Preisgeld ausbezahlt
- RSA-768 wurde 2009 von Kleinjung et al. „geknackt“
  - hätte $50.000 Preisgeld eingebracht
- Bereits 2007 wurde von Kleinjung et al. die 1039. Mersenne-Zahl (1039-Bit-Zahl) faktorisiert
  - war allerdings nicht Bestandteil der RSA Challenge
- BSI Technische Richtlinie TS-02102 (2008)
  - Empfiehlt mindestens 2048 Bit lange RSA-Schlüssel
- DFN-PKI fordert ebenfalls mindestens 2048 Bit lange Schlüssel

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

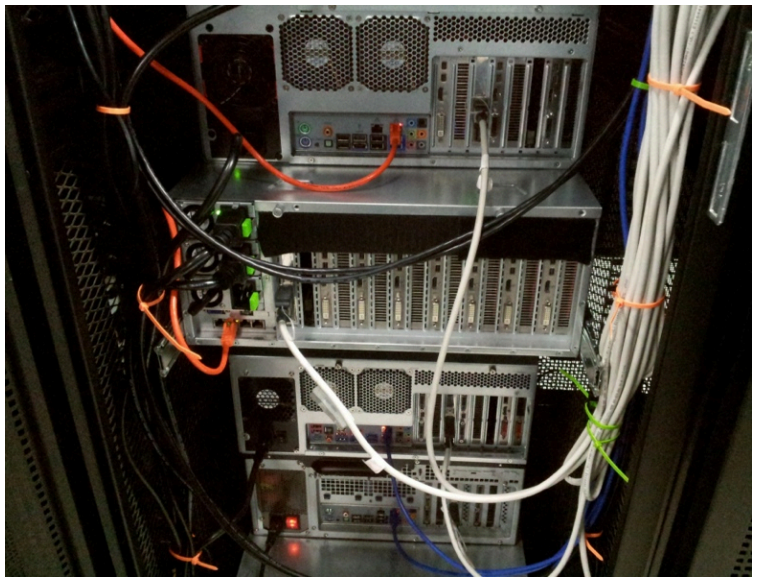
Empfohlene Schlüssellängen

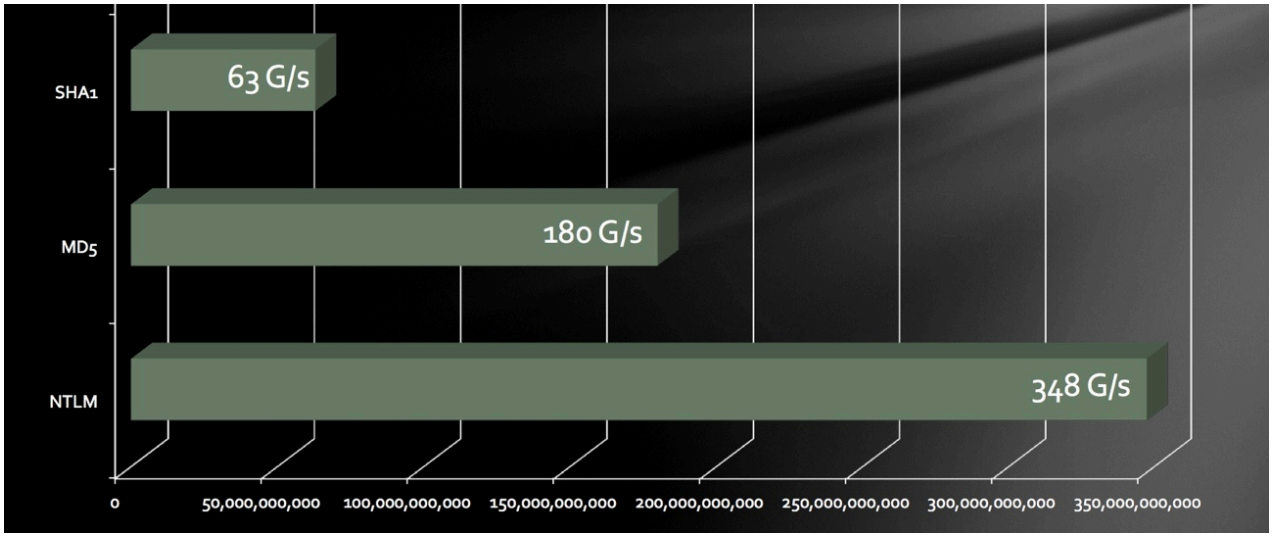
- Quelle: ECRYPT 2006: Report on Algorithms and Keysizes
- Minimale Schlüssellängen:

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

Einschub: Password-Hashes knacken (Gosney 12/2012)

- Cluster mit 25 GPUs in 5 Servern:
  - 10x HD 7970
  - 4x HD 5970 (dual GPU)
  - 3x HD 6990 (dual GPU)
  - 1x HD 5870
- Stromverbrauch: 7kW
- Beispiel: 8 Zeichen langes NTLM-Passwort in weniger als 6 Stunden geknackt

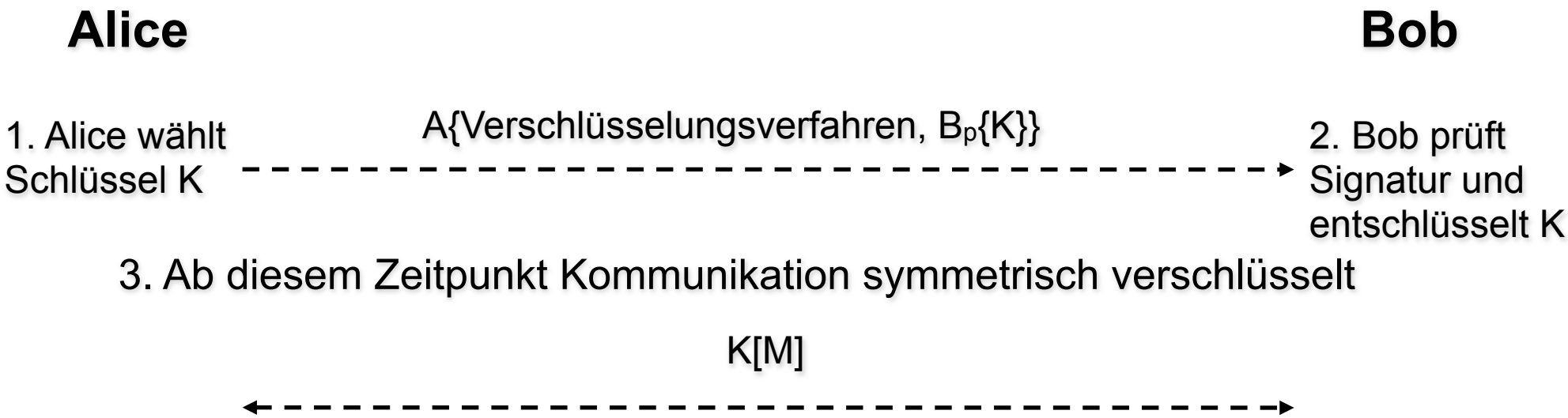




Abbildungen aus: http://passwords12.at.ifi.uio.no/Jeremi_Gosney_Password_Cracking_HPC_Passwords12.pdf

Hybride Kryptosysteme

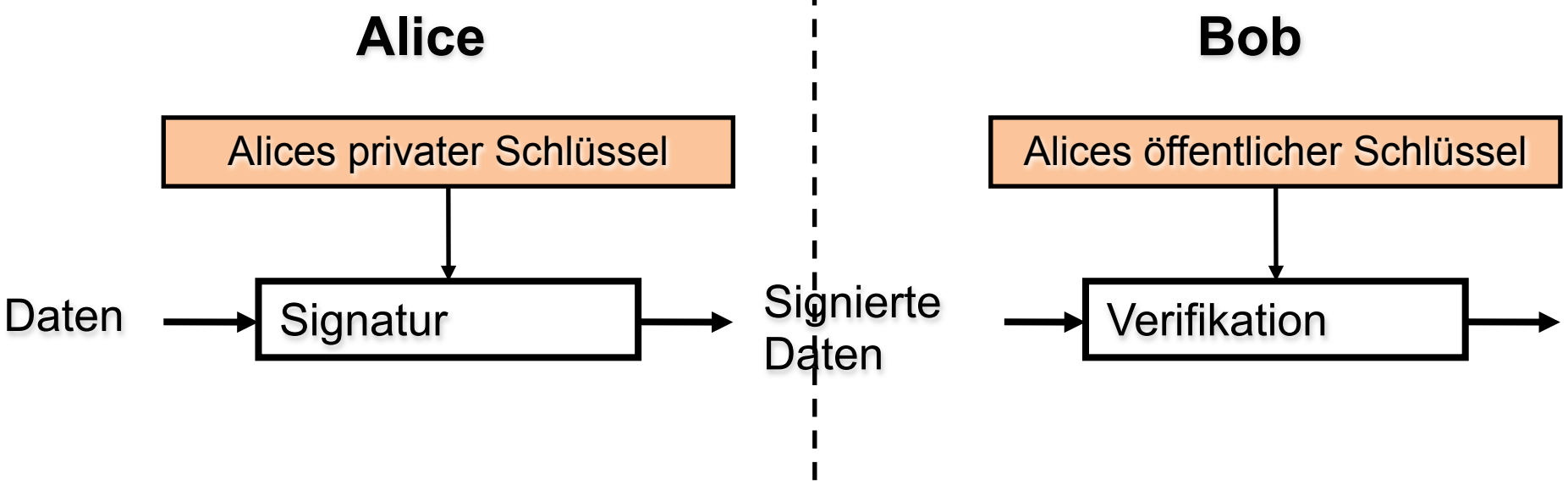
- Vereinen Vorteile von symmetrischen und asymmetrischen Verfahren
- Asymmetrisches Verfahren zum Schlüsselaustausch
- Symmetrisches Verfahren zur Kommunikationsverschlüsselung



- Beispiele für hybride Verfahren: SSL/TLS, PGP, SSH,...
  - Oftmals neuer Schlüssel pro Nachricht oder Zeiteinheit, aus K abgeleitet

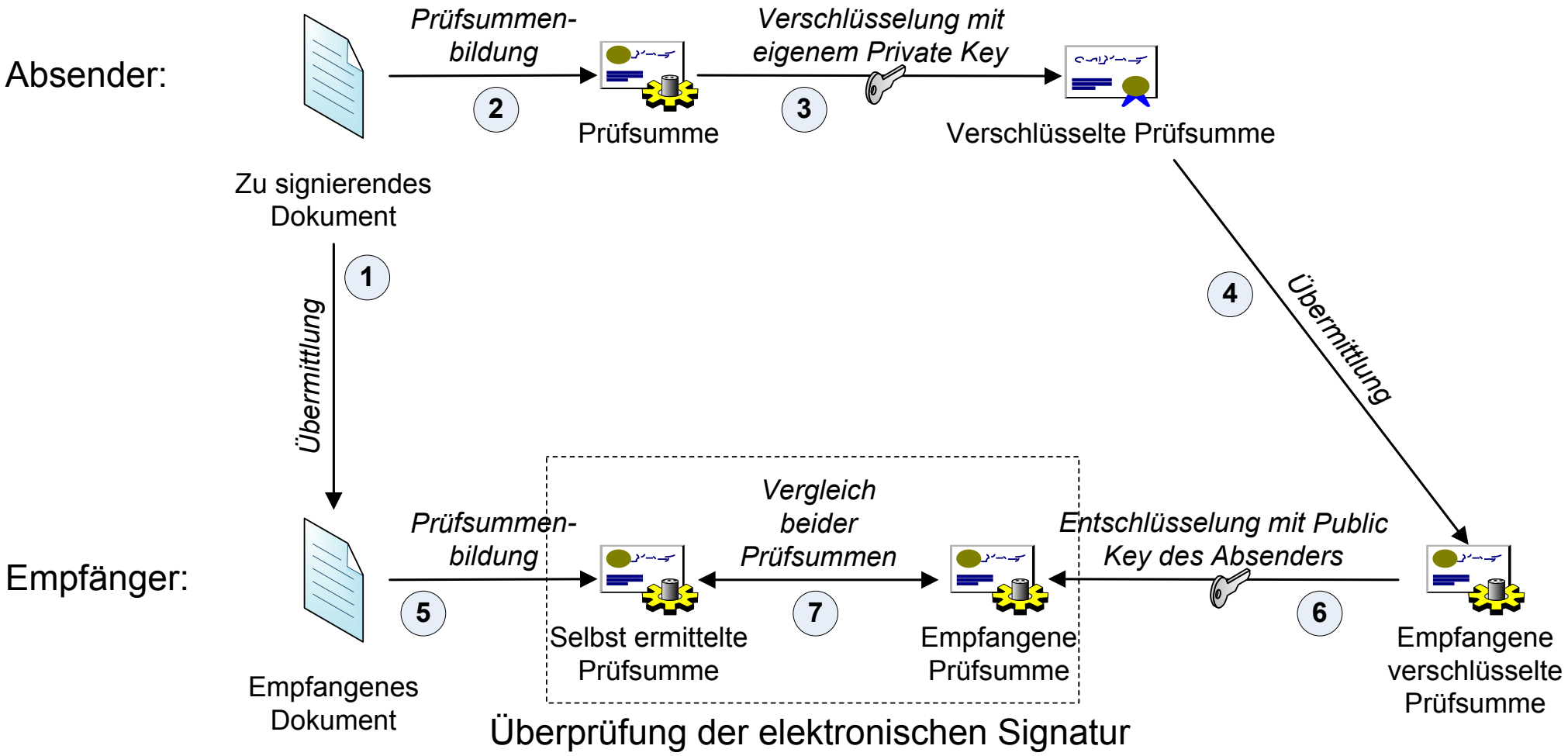
Elektronische Signatur

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



- Asymetrische 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)

Ablauf: Signatur und deren Verifikation



Elektronische Signatur: Analogie zur Unterschrift

- Zentrale Anforderungen an die (analoge) Unterschrift:

1. **Perpetuierungsfunktion:** Dokument und Unterschrift sind dauerhaft.
2. **Echtheitsfunktion:** Die Unterschrift ist authentisch.
3. Die Unterschrift kann **nicht wiederverwendet** werden.
4. **Abschlussfunktion:** Unterschrift ist räumlicher Abschluss des Dokuments; dieses kann später nicht verändert werden.
5. **Beweisfunktion:** Unterzeichner kann seine Unterschrift später nicht leugnen.

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

Elektronische Signatur: Erfüllung der Anforderungen?

1. **Perpetuierungsfunktion:** Fälschungssicher und dauerhaft
2. **Echtheitsfunktion:** Authentizität sichergestellt
3. **Wiederverwendbarkeit:** Wie gewünscht nicht gegeben
4. **Abschlussfunktion:** Nicht veränderbar
5. **Beweisfunktion:** Unterschrift ist nicht zu leugnen

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.