Diffie-Hellman, ElGamal und DSS

- Vortrag von David Gümbel am 28.05.2002

Übersicht

- Prinzipielle Probleme der sicheren Nachrichtenübermittlung
- 'Diskreter Logarithmus'-Problem
- Diffie-Hellman
- ElGamal
- DSS / DSA
- Vergleich
- Praxisbeispiel

Prinzipielle Probleme der sicheren Nachrichtenübermittlung

- Unsicherer Kanal
  - Abhören und Manipulieren möglich
    - Symmetrische Verfahren ungeeignet, da sicherer Kanal für Schlüsselübergabe erforderlich
    - Authentizität der Nachricht nicht gesichert
      Lösung: Signatur (RSA et al.)
    - Bei asymmetrischen Verfahren: Schlüsselaustausch problematisch
      Lösung: Diffie-Hellman Schlüsselaustausch, basierend auf 'diskreter-Logarithmus'-Problem

'Diskreter Logarithmus'-Problem

- Gegeben: F_p^* multiplikative Gruppe des endlichen Körpers F_p
- Beliebiges Basiselement g ? F_p^*
- Finde zu y ? F_p^* ein x so daß $y = g^x$

Diskreter Logarithmus und RSA

- RSA basiert auf dem Problem, große Zahlen zu faktorisieren (IFP)
- IFP und DLP: 'Ein-Weg'-Funktionen
- Sicherheit eines Algorithmus meist nicht beweisbar
- DLP und IFP sind relativ gut erforscht
- Daher können die auf DLP und IFP basierenden Algorithmen als relativ sicher gelten
- "Based on the best attacks known, RSA at 1024 bits, DSA and Diffie-Hellman at 1024 bits, and elliptic curves at about 170 bits give comparable levels of security." [M. J. Wiener, 1998]

Diffie-Hellman Schlüsselaustausch

- Basierend auf DLP
- Verfahren:
  - Alice wählt g ? F_p^* und (geheim) a
  - Sendet (g, g^a) über unsicheren Kanal an Bob
  - Bob sendet g^b
- Schlüssel ist nun $g^{ab} = (g^a)^b = (g^b)^a$
- 'Diffie-Hellman-Problem': Finde g^{ab} aus g, g^a, g^b

Diffie-Hellman Schlüsselaustausch

- Einsatzgebiete:
  - GnuPG, PGP, SSH
  - Standards: OpenPGP, S/MIME
  - Sichere Netzwerke
- Implementierungen:
  - Kryptographische Bibliotheken (OpenSSL, JCE, ...)
  - SSH
  - IPSec

Diffie-Hellman Schlüsselaustausch

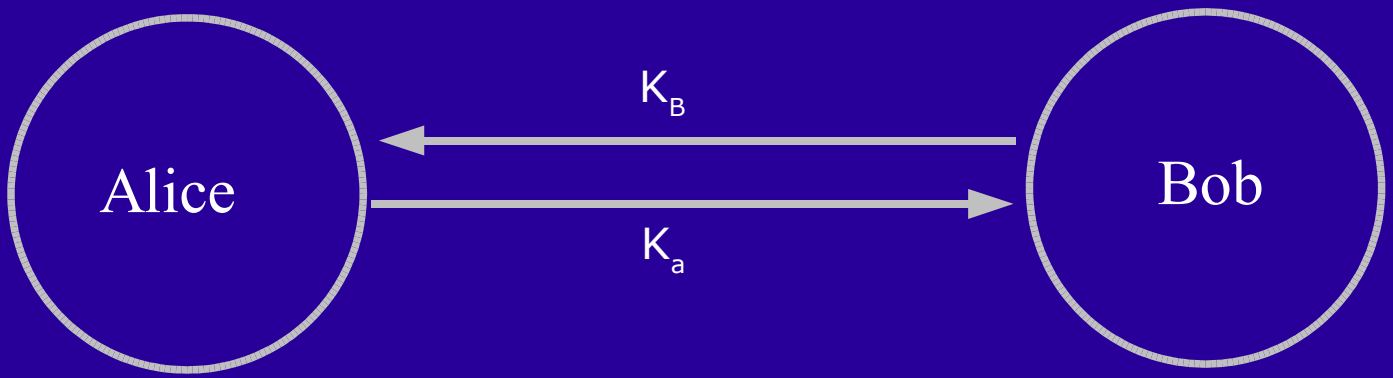
- Vorteile:
  - Schlüsselaustausch sicherer als bei z.B. RSA ('Session'-Keys)
  - Leicht zu implementieren:

```
#!/usr/local/bin/perl -- -export-a-crypto-system-sig Diffie-Hellman-2-lines
($g,$e,$m)=@ARGV,$m||die"$0 gen exp mod\n";print`echo "16dio1[d2%Sa2/d0<X+d
*La1=z\U$m%0]SX$e"[$g*]\EszlXx+p|dc`
```

- Nachteile:
  - Middleperson-Angriff möglich

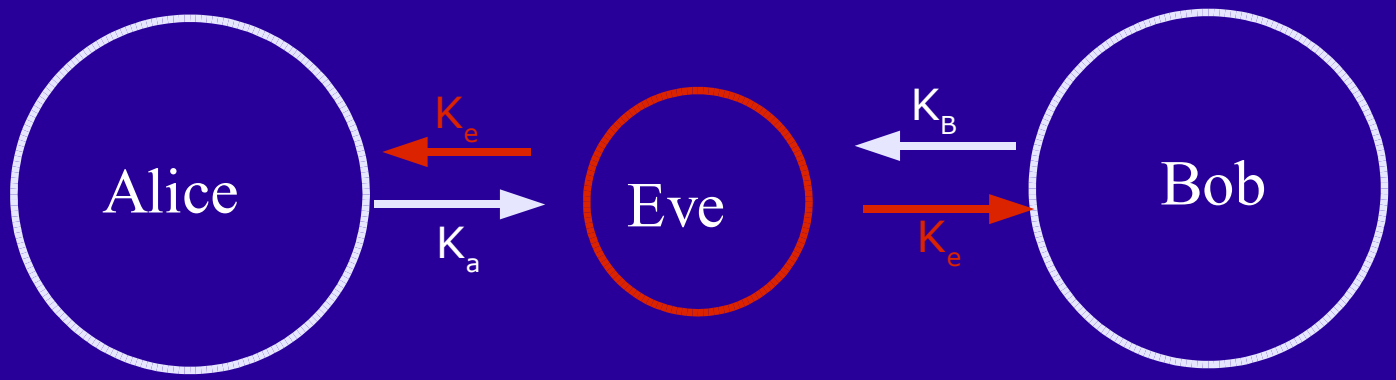
Middleperson-Angriff

- Alice und Bob wollen kommunizieren
- daher Schlüsselaustausch i.d.R. über unsicheres Medium



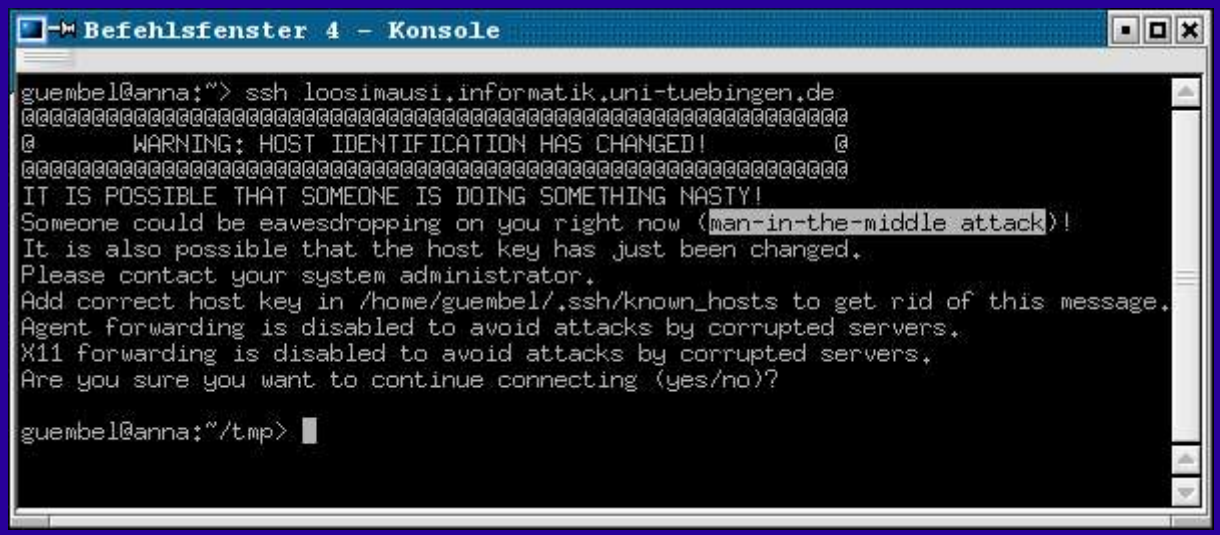
Middleperson-Angriff

- Angreiferin Eve hat Kontrolle über den Nachrichtenkanal
- Eve sendet eigenen Public Key als fremden an Alice bzw. Bob



Middleperson-Angriff

- Praktisch leicht durchführbar
- Für übliche Protokolle wie SSH und SSL: Toolsammlung dsniff automatisiert Angriff



ElGamal

- 'erweiterte' Variante von Diffie-Hellman
- Schlüsselgenerierung:
  - F_q endlicher Körper, g ? F_q^*
  - Wähle a, 0 < a < q-1
  - Privater Schlüssel: Exponent a
  - Öffentlicher Schlüssel: g^a

ElGamal

- Verschlüsselung:
  - Nachricht P
  - Wähle Zufallszahl k
  - Sende (g^k, Pg^{ak})
- Entschlüsselung:
  - Teile Pg^{ak} durch $g^{ak} = (g^k)^a$

ElGamal

- Vorteile:
  - Ähnlich unkompliziert wie Diffie-Hellman
  - Diffie-Hellman-Problem und ElGamal äquivalent
- Nachteile:
  - Vergrößert Nachrichtenlänge
- Einsatz z.B. bei GnuPG
- Schlüssellängen: 1024 Bit und mehr (bis 4096 Bit in GnuPG)

DSA- Digital Signature Algorithm

- Als DSS (Digital Signature Standard) amerikanischer Standard
- 1991 vorgeschlagen, 1994 vom NIST verabschiedet
- Basiert ebenfalls auf DLP
- Verwendung z.B. in GnuPG für Signaturen

DSA- Digital Signature Algorithm

- Vorbereitungen für Signatur:
  - Wähle Primzahl q (Größenordnung: 160 Bit)
  - Wähle Primzahl p = 1 mod q (Größenordnung: 512 bis 1024 Bit)
  - Wähle einen Erzeuger der zyklischen Untergruppe von F_p^* der Ordnung q
  - Wähle Zufallszahl x, 0 < x <q als privaten Schlüssel
  - Setze öffentlichen Schlüssel $y = g^x \bmod p$

DSA- Digital Signature Algorithm

- Signieren:
  - Bilde Hashwert H, $0 < H < q$ der Nachricht
  - Wähle Zufallszahl k gleicher Größenordnung, berechne $r = (g^k \bmod p) \bmod q$
  - Finde s mit $sk = H + xr \bmod q$
  - Signatur: (r, s)
- Verifizieren:
  - Berechne $u = s^{-1}H \bmod q$, $t = s^{-1}r \bmod q$
  - Prüfe: $(g^u y^t \bmod q) = r$

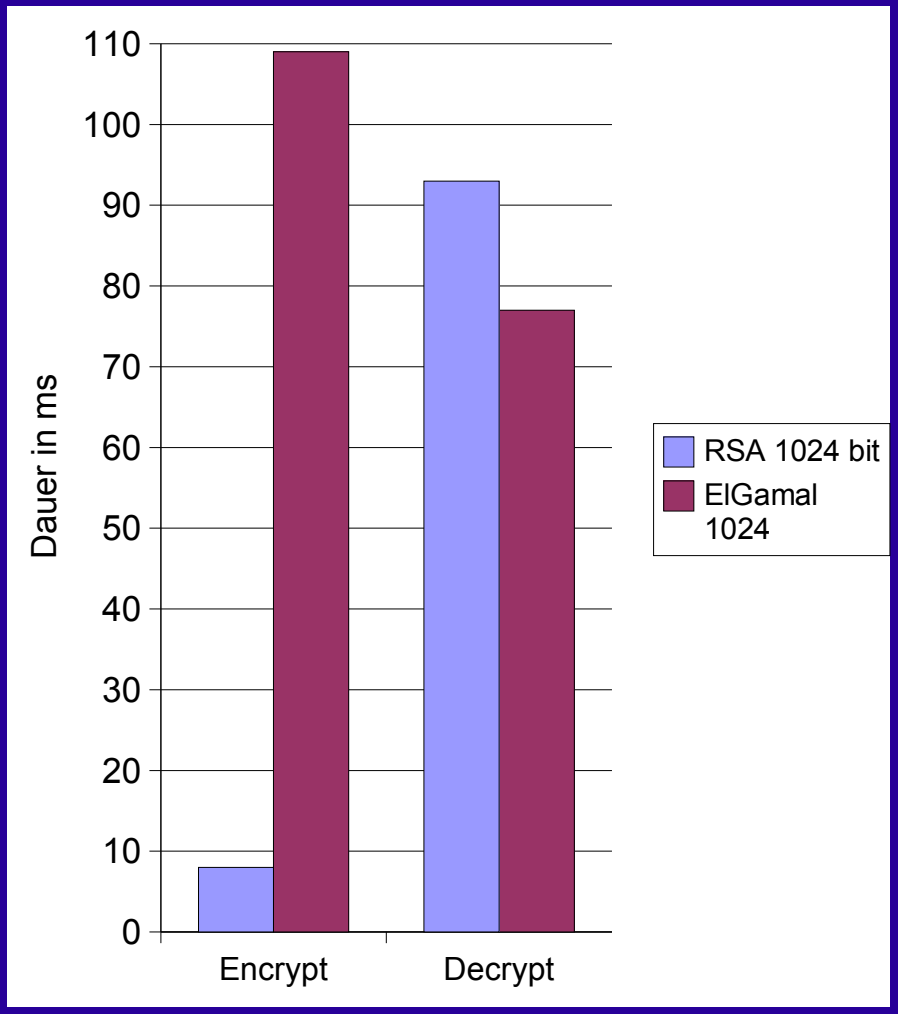
DSA - Übersicht

- Verwendung in PGP / GnuPG für Signaturen (1024 Bit)
- BSI: Sicher bis 2003 (wenn p 1024 Bit oder mehr)
- Implementiert in OpenSSL, SSH etc.
- US-Standard

Diffie-Hellman/ElGamal/DSS-Schlüssel

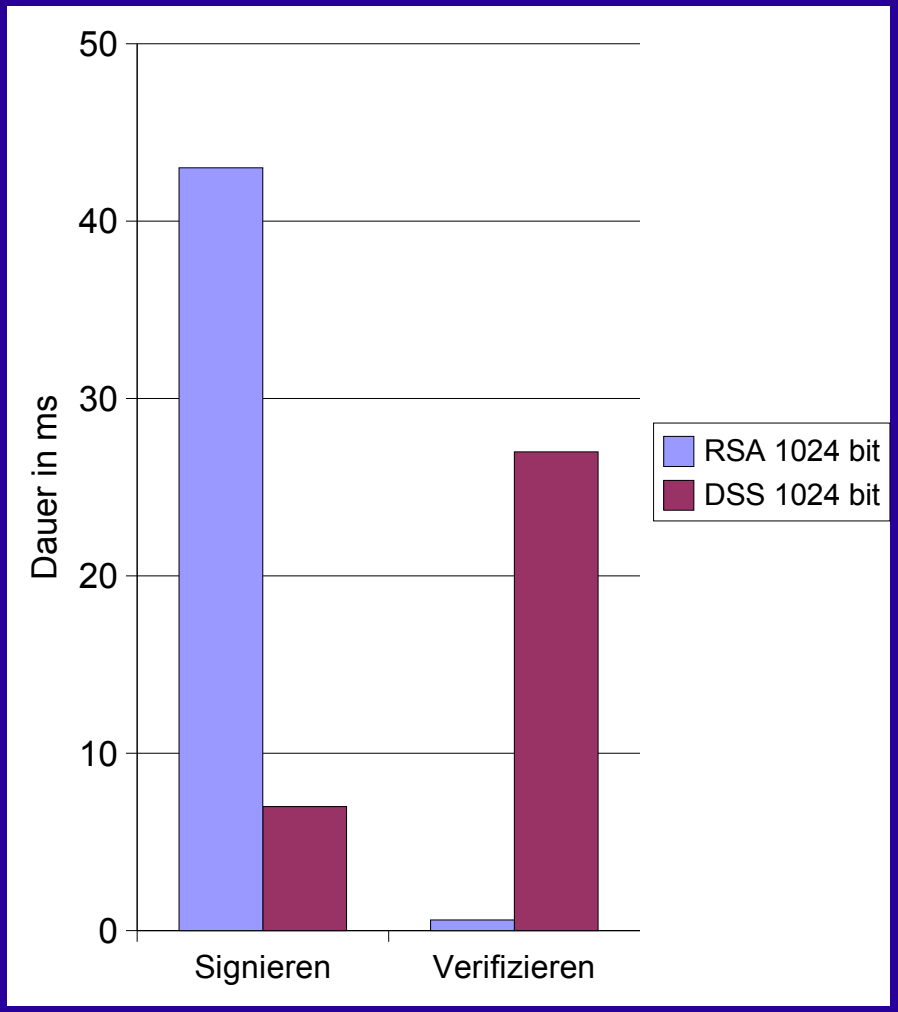
- Vorteile:
  - Schlüsselaustausch, Signatur und Verschlüsselung getrennt
  - Bruch eines Verfahrens bedeutet nicht (wie z.B. bei RSA) Bruch der anderen beiden
  - Diffie-Hellman seit 1997 ohne Patentschutz (RSA seit 2000)
- Nachteile:
  - Alle drei Systeme basieren auf dem gleichen mathematischen Problem
  - ElGamal vergrößert Nachrichtenlänge

Performancevergleich RSA und ElGamal



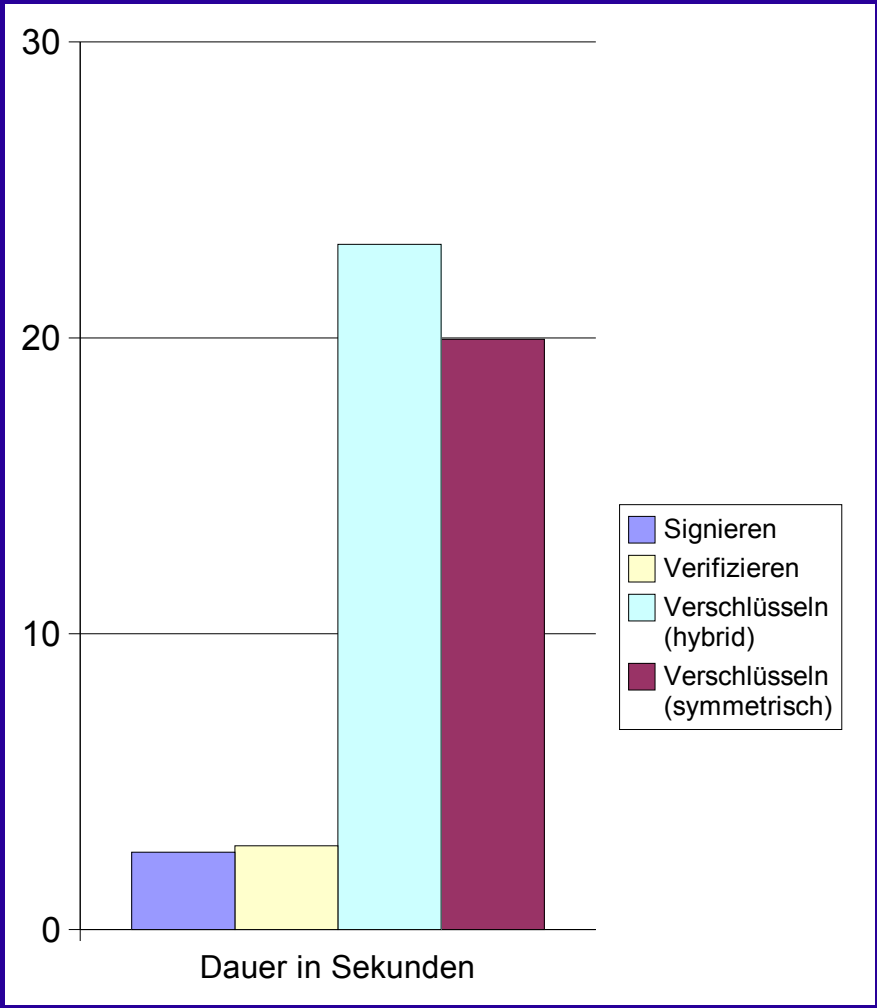
- Nachrichten werden öfter entschlüsselt als verschlüsselt
- Daher Unterschied vertretbar
- In der Praxis: Hybride Verschlüsselung
- Performance z.B. bei kryptographischer Hardware wichtig

Performancevergleich RSA und DSS



- Signaturen werden öfter überprüft als erstellt
- RSA schneller bei Verifikation
- Daher gilt RSA als das schnellere Verfahren

Praxisbeispiel GnuPG



- Verschlüsselung von 50MB Zufallsdaten auf PIII-1.1GHz
- Daten werden mit symmetrischem Verfahren (AES) verschlüsselt
- Paßwort für AES wird mit ElGamal gesichert

Noch Fragen?