Der PGP-Keyserver der Unix-AG

27. September 1996

Peter Conrad <*conrad@unix-ag.uni-kl.de*>

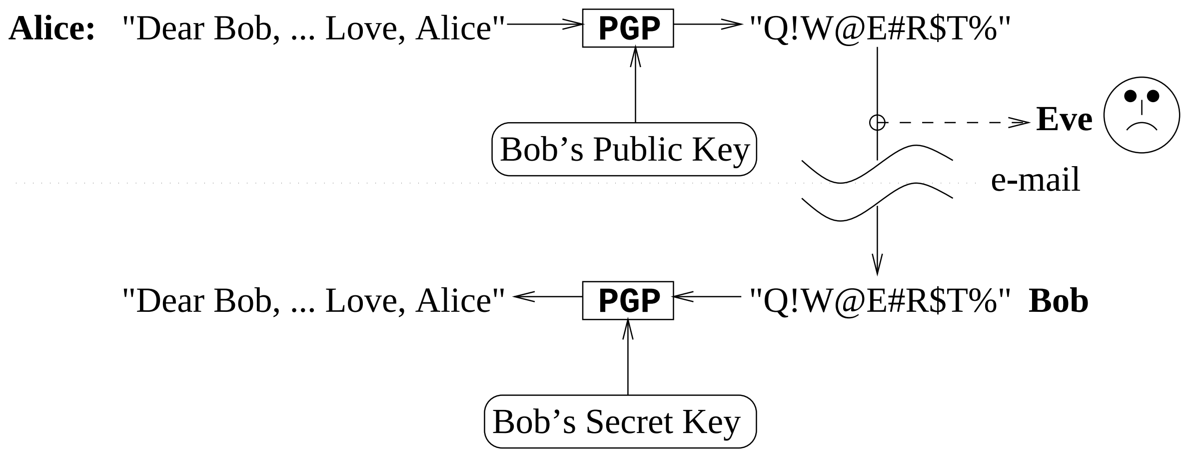
Aufbau

1. Einleitung: Prinzipien der Public-Key-Kryptographie
2. Der Server von Michael Graff <*explorer@iastate.edu*>
3. Der Server der Unix-AG (α-Version)

1 Prinzipien der Public-Key-Kryptographie

Aufgaben:

- Vertraulichkeit der Kommunikation (Verschlüsselung)

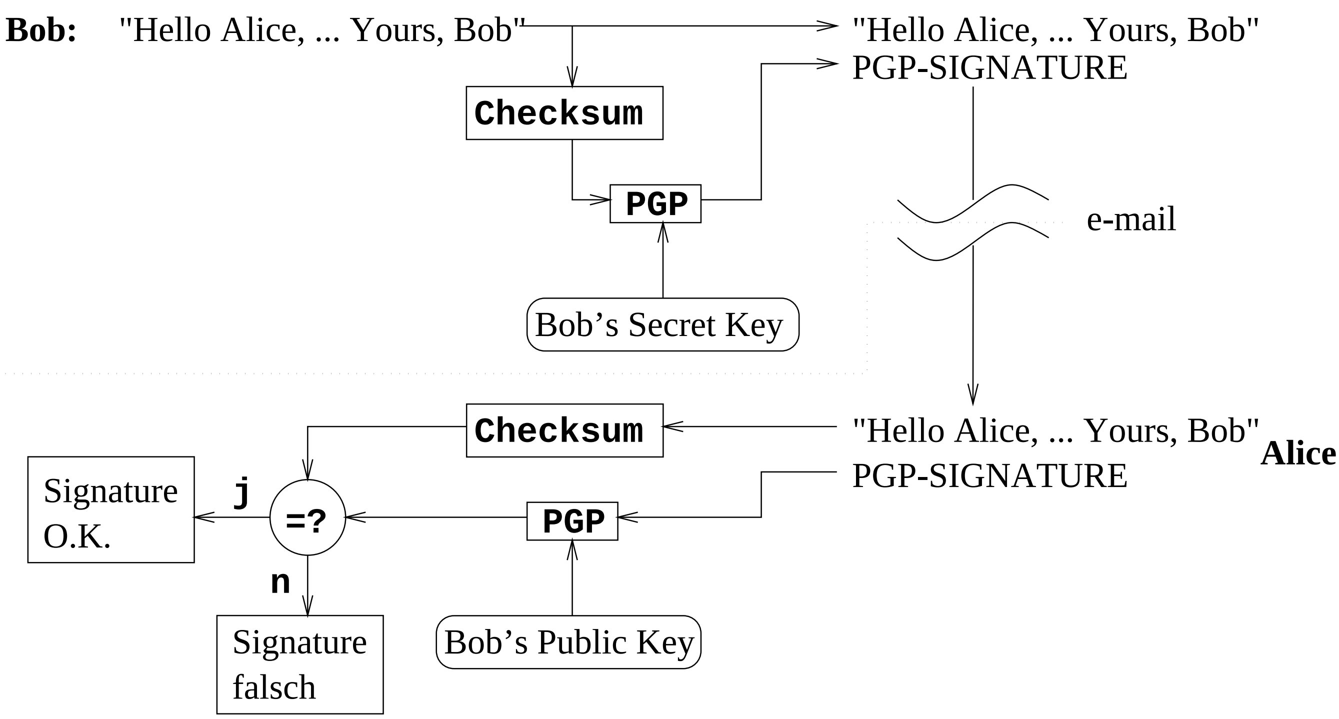


- Authentifizierung der Kommunikationspartner (elektron. Unterschriften)

Problem: Woher bekommt man den "Public Key" von Person X?

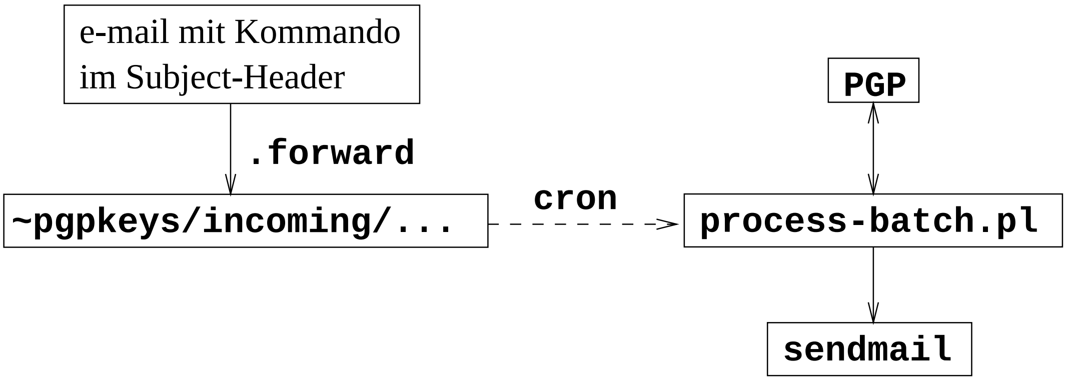
Antwort: sicherheitshalber aus möglichst vielen verschiedenen unahängigen Quellen, z. B.

- per `finger` X@*host.where.ever.org*
- auf Diskette (z. B. per Post, oder bei persönlichem Treffen)
- per email
- von einem **Keyserver**


Bob:
"Hello Alice, ... Yours, Bob"
Checksum
PGP
Bob's Secret Key
"Hello Alice, ... Yours, Bob"
PGP-SIGNATURE
e-mail
"Hello Alice, ... Yours, Bob"
PGP-SIGNATURE
Alice
Checksum
PGP
Bob's Public Key
=?
j
n
Signature O.K.
Signature falsch

2 Der Server von Michael Graff

Übersicht:



Kommandos:

- `INDEX` *substr*

  liefert eine Liste aller Keys, die *substr* in ihrer ID haben, analog zu `pgp -kv`

- `VERBOSE INDEX` *substr*

  liefert eine Liste aller Keys und Unterschriften, die *substr* in ihrer ID haben, analog zu `pgp -kvv`

- `GET` *substr*

  liefert den ersten Key, dessen User-ID *substr* enthält

- `MGET` *regexp*

  liefert alle Keys, deren User-IDs *regexp* matchen

- `LAST` *days*

  liefert alle Keys, die in den letzten *days* Tagen geändert wurden

- `ADD`

  fügt einen Key zur Datenbasis hinzu (Key im Body der Mail)

Spezialkommando:

- `INCREMENTAL`

  dient der Synchronisation mit anderen Servern,
  Mail-Header enthält `X-Keyserver-Sent:` Zeilen für Server, die diesen Key bereits "gesehen" haben

3 Der Server der Unix-AG

Nachteile des alten Servers:

- verwendet PGP → elend langsam, speziell bei `ADD` und `INCREMENTAL`, bedingt durch
  - ineffizienten Zugriff auf Keyring
  - langsame math. Routinen beim Signature-Checking
  - unnötigen Overhead (z. B. Trust-Parameter)

→ Design-Goals bei unserem Server:

- weniger I/O → Keyring komplett im RAM halten und regelmäßig mit Platte synchronisieren
- schnellere Signature-Checks durch
  - effizientere math. Routinen (GNU `libgmp`)
  - effiziente Indizierung der Keys
- weniger Overhead durch Weglassen unnötiger Berechnungen

→ das Programm PGP kann nicht mehr verwendet werden
Kommunikation über TCP-Socket (calzone:5440)
Kommandos: `INDEX`, `GET`, `MGET`, `ADD`, `LAST`, (`INCREMENT`)
Interfacing mit anderen Diensten "leicht" möglich, z. B. in Perl:

- WWW-Interface: `http://calzone.unix-ag.uni-kl.de:8087/de/keyserv/pgp.html`
- email-Interface (`mixmaster@unix-ag.uni-kl.de`, derzeit nur für Synchronisation mit anderen Servern)

Nachteile:

- PGP wird nicht mehr verwendet → evtl. Probleme mit PGP '96
- grösserer Aufwand sowohl bei Soft- als auch bei Hardware