mobile business

Lecture 5

Cryptography II

Information & Communication Security (WS 2014)

Prof. Dr. Kai Rannenberg

Deutsche Telekom Chair of Mobile Business & Multilateral Security Goethe-University Frankfurt a. M.





Agenda



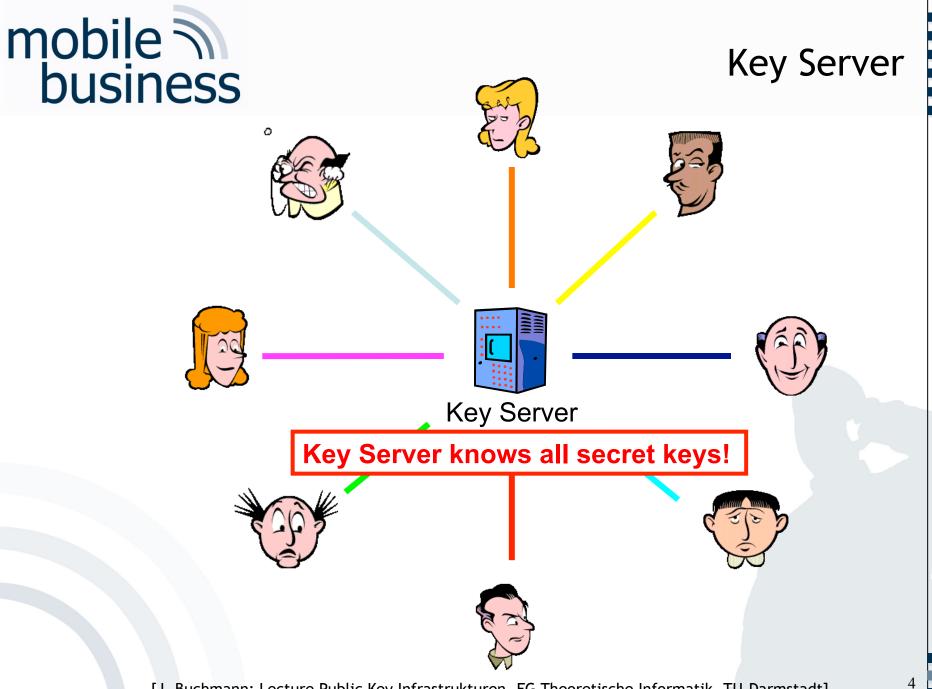
Introduction

- Classical cryptosystems
- Public key cryptography
 - General concept
 - Algorithms
 - Hybrid systems
 - Key management
 - Example: PGP

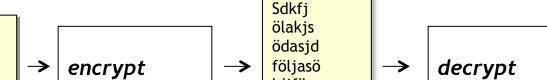
Problems of Symmetric Cryptosystems: Key Exchange

0 n*(n-1)/2 Keys Internet: ~ 1.000.000.000 Users => ~ 500.000.000.000.000 Keys

mobile business



[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]



public

ldjföas jölakj

asymmetric

Offer

Alice

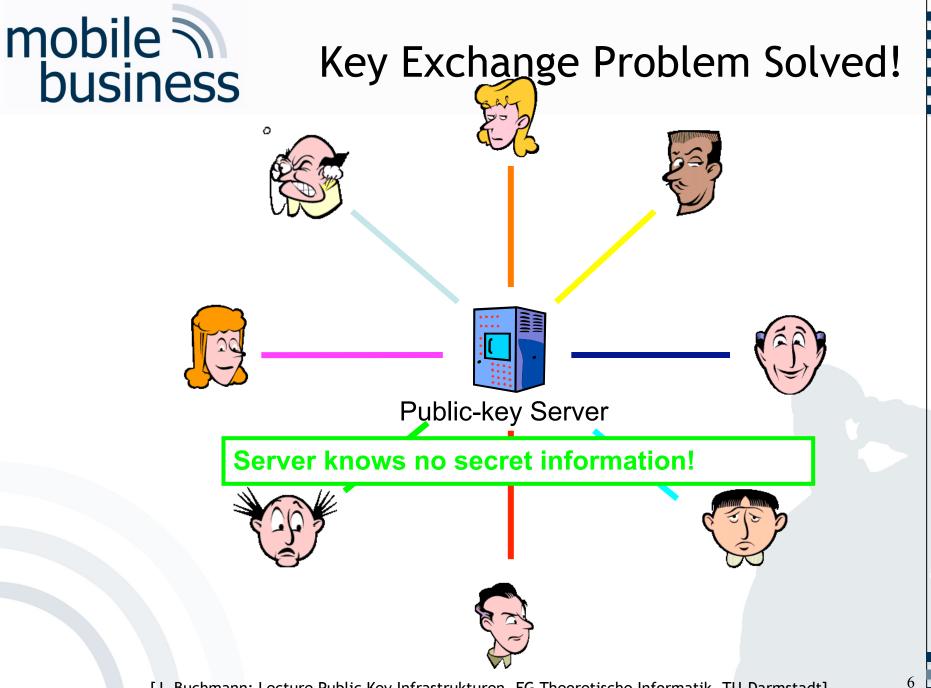
Public Key Encryption

private

Offer

Bob

->



[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]



Agenda

- Introduction
- Classical cryptosystems
- Public key cryptography
 - General concept
 - Algorithms
 - Hybrid systems
 - Key management
 - Example: PGP



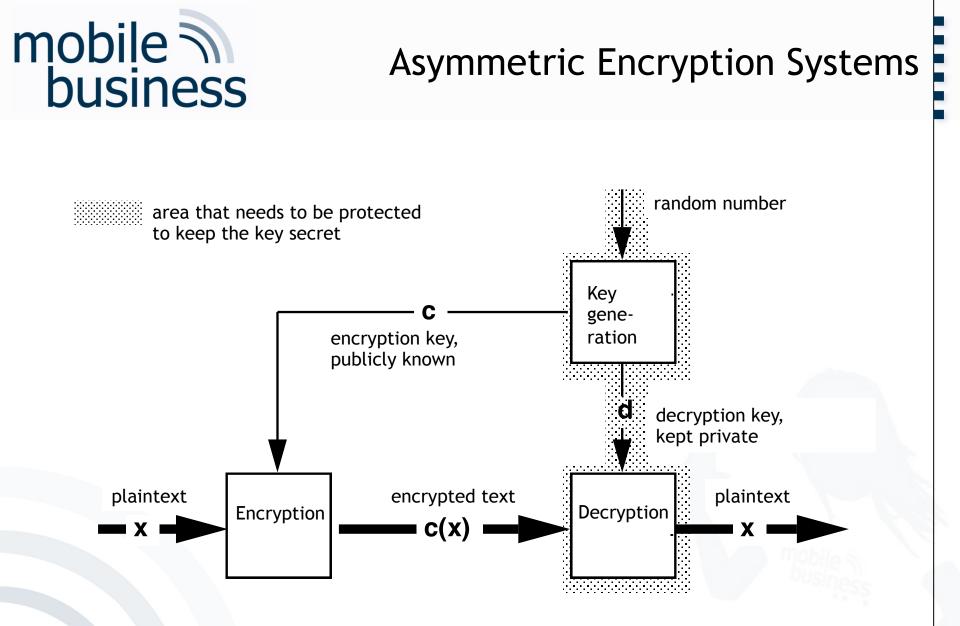




mobile business

Concept of Asymmetric Encryption Systems

- Use of 'corresponding' key pairs instead of one key:
 - Public key is solely for encryption.
 - Encrypted text can only be decrypted with the corresponding **private** (undisclosed) key.
- Deriving the private key from the public key is hard (practically) impossible).
- The public key can be distributed freely, even via insecure ways (e.g. directory (public key crypto system)).
- Messages are encrypted via the public key of the addressee.
- Only the addressee holds the private key for decoding (and has to manage the relation between the private and the public key).



box with slot, access to messages only with a key



- Introduction
- Classical cryptosystems
- Public key cryptography
 - General concept
 - Algorithms
 - Hybrid systems
 - Key management
 - Example: PGP



mobile business

Asymmetric Encryption Systems: Examples

RSA

- Rivest, Shamir, Adleman, 1978
- Based on the assumption that the factorization of the product of two (big) prime numbers (p*q) is "difficult" (product is the public key)
- Key lengths often 1024 bit; recommended 2048 or 4096 bit

Diffie-Hellman

- Diffie, Hellman, 1976
- First patented algorithm with public keys
- Allows the exchange of a secret key
- Based on the "difficulty" of calculating discrete logarithms in a finite field



- Introduction
- Classical cryptosystems
- Public key cryptography
 - General concept
 - Algorithms
 - RSA
 - Hybrid systems
 - Key management
 - Example: PGP

Agenda



RSA Encryption

- To encrypt a message *M*, using a public key (*e*,*n*), proceed as follows (*e* and *n* are a pair of positive integers):
 - First represent the message as an integer between 0 and *n*-1 (break long messages into a series of blocks, and represent each block as such an integer).
 - Then encrypt the message by raising it to the *e*th power modulo *n*.
 - The result (the ciphertext C) is the remainder of M^e divided by n.
 - The encryption key is thus the pair of positive integers (*e*,*n*).

[RSA78]



- To decrypt the ciphertext, raise it to another power d, again modulo n.
- The decryption key is the pair of positive integers (*d*,*n*).
- Each user makes his encryption key public, and keeps the corresponding decryption key private.



• $C \equiv E(M) \equiv M^e \pmod{n}$, for a message M

M = D(C) = C^d (mod n), for a ciphertext C

[RSA78]



Choosing the Keys (I)

- You first compute *n* as the product of two primes *p* and *q*.
- n=p*q
- These primes are very large "random" primes.
- Although you will make *n* public, the factors *p* and *q* will be effectively hidden from everyone else due to the enormous difficulty of factoring *n*.
- This also hides the way, how *d* can be derived from *e*.



Choosing the Keys (II)

- You then choose an integer *d* to be a large, random integer which is relatively prime to (*p-1*)*(*q-1*).
- That is, check that d satisfies:
 - The greatest common divisor of d and (p-1)*(q-1) is 1.
 - gcd(d,(p-1)*(q-1))=1



- The integer *e* is finally computed from *p*,*q*, and *d* to be the "multiplicative inverse" of *d*, modulo (*p*-1)*(*q*-1).
- Thus we have $e^*d \equiv 1 \pmod{(p-1)^*(q-1)}$.



Simplified Example (I)





Private (d,n)



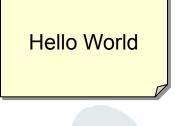
- Let p=7 and q=11.
- Then n=77.
- Alice chooses d=53, so e=17.
- gcd(d,(p-1)*(q-1)) =
 gcd(53,(7-1)*(11-1)) =
 gcd(53,60) = 1
- e*d mod (p-1)*(q-1) = 901 mod 60 = 1

Based on [Bi05]



Simplified Example (II)

- Bob wants to send the message "Hello World" to Alice.
- Each plaintext character is represented by a number between 00(A) and 25 (Z).
- Therefore, we have the plaintext as:
 07 04 11 11 14 26 22 14 17 11 03



Based on [Bi05]

Bob



Simplified Example (III)

- Using Alice's public key the ciphertext is:
 - 07¹⁷ mod 77 = 28
 - 04¹⁷ mod 77 = 16
 - 11¹⁷ mod 77 = 44

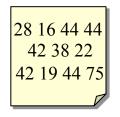


Or 28 16 44 44 42 38 22 42 19 44 75



Bob





- Alice decrypts the ciphertext by calculating:
 - 28⁵³ mod 77 = 07
 - 16⁵³ mod 77 = 04
 - 44⁵³ mod 77 = 11



75⁵³ mod 77 = 03
Or: 07 04 11 11 14 26 22 14 17 11 03 = "Hello World"

Based on [Bi05]



- Introduction
- Classical cryptosystems
- Public key cryptography
 - General concept
 - Algorithms
 - Hybrid systems
 - Key management
 - Example: PGP





Performance of Public Key Algorithms

| Algorithm | Performance* | Performance compared to Symmetric encryption (AES) |
|-----------------|--------------|---|
| RSA (1024 bits) | 6.6 s | Factor 100 slower |
| RSA (2048 bits) | 11.8 s | Factor 180 slower |

Disadvantage: Complex operations with very big numbers

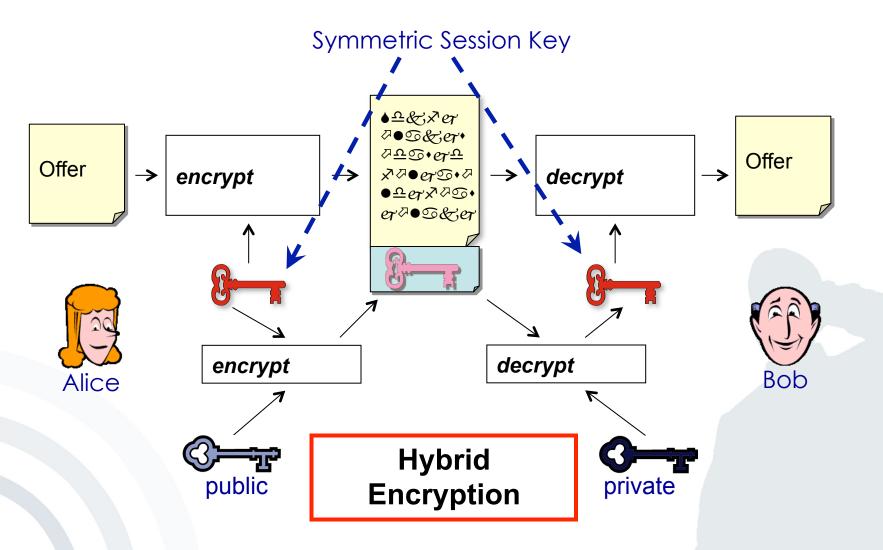
\Rightarrow Algorithms are very slow.

* Encryption of 1 MB on a Pentium 2.8 GHz, using the FlexiProvider (Java)

[J. Buchmann: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]



Solution: Hybrid Systems



[based on: J. Buchmann 2005: Lecture Public Key Infrastrukturen, FG Theoretische Informatik, TU-Darmstadt]

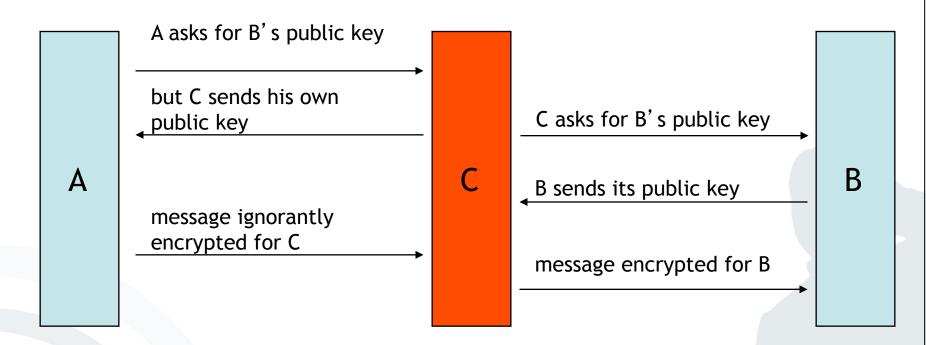


- Introduction
- Classical cryptosystems
- Public key cryptography
 - General concept
 - Algorithms
 - Hybrid systems
 - Key management
 - Example: PGP

Agenda



"Man in the middle attack"



Keys are certified: a 3rd person/institution confirms (with its digital signature) the affiliation of the public key to a person.



Certification of Public Keys

- B can freely distribute his own public key.
- But: Everybody (e.g. C) could distribute a public key and claim that this one belongs to B.
- If A uses this key to send a message to B, C will be able to read this message!
- Thus:

How can A decide if a public key was really created and distributed by B without asking B directly?

- Keys get certified, i.e. a third person/institution confirms with its (digital) signature the affiliation of a public key to entity B.
- Public Key Infrastructures (PKIs)



Certification of Public Keys

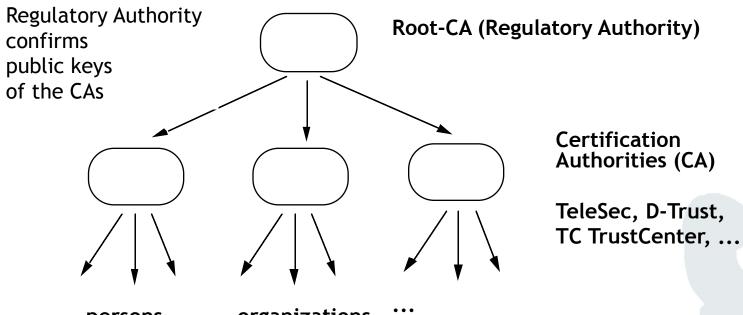
Three types of organization for certification systems (PKIs?):

- Central Certification Authority (CA)
 - A single CA, keys often integrated in checking software
 - eExample: older versions of Netscape (CA = Verisign)
- Hierarchical certification system
 - CAs which in turn are certified by "higher" CA
 - Examples: PEM, TeleTrust, infrastructure according to Signature Law
- Web of Trust
 - Each owner of a key may serve as a CA.
 - Users have to assess certificates on their own.
 - Example: PGP (but with hierarchical overlay system)



Hierarchical Certification of Public Keys

(Example: German Signature Law)

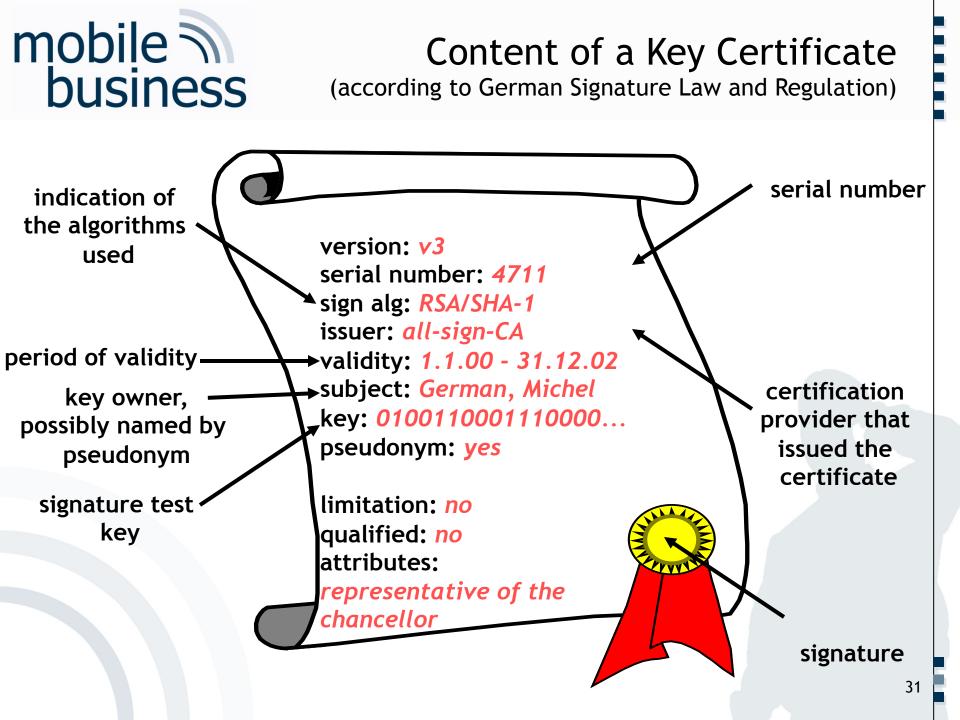


persons

organizations

The actual checking of the identity of the key owner takes place at so called Registration Authorities (e.g. notaries, bank branches, T-Points, ...)

Security of the infrastructure depends on the reliability of the CAs.







- Reliable identification of persons who apply for a certificate
- Information on necessary methods for fraud resistant creation of a signature
- Provision for secure storage of the private key
 - at least Smartcard (protected by PIN)
- Publication of the certificate (if wanted)
- Barring of certificates
- If necessary issuing of time stamps
 - for a fraud resistant proof that an electronic document has been at hand at a specific time



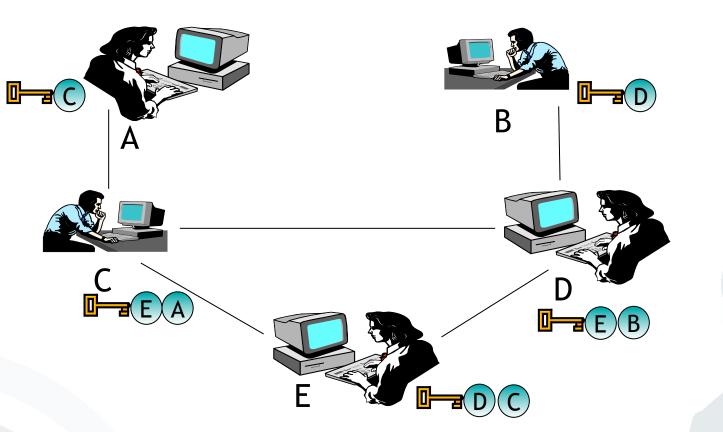
- Checking of the following items by certain confirmation centers (BSI, TÜVIT, ...)
 - Concept of operational security
 - Reliability of the executives and of the employees as well as of their know-how
 - Financial power for continuous operation
 - Exclusive usage of licensed technical components according to SigG and SigV
 - Security requirements as to operating premises and their access controls
- Possibly license of the regulation authority

mobile business Web of Trust "Introducer" David Bob knows David and has received David's public key by David himself Alice lets David sign her public key Bob can verify Alice's key Alice sends the signed on the basis of David's key to Bob signature Bob encrypts his message to Alice Bob Alice with the received key 5

- Each user can act as a "CA".
- Mapping of the social process of creation of trust
- Keys are "certified" through several signatures.
- Expansion is possible by public key servers and (hierarchical) CAs.



Web of Trust Example



Web of Trust:

- Certification of the public keys mutually by users
- Level of the mutual trust is adjustable.



- Introduction
- Classical cryptosystems
- Public key cryptography
 - General concept
 - Algorithms
 - Hybrid systems
 - Key management
 - Example: PGP





Protection of Email Example PGP

- PGP = Pretty Good Privacy
- De facto-Standard for freely accessible email encryption systems on the Internet
- First implementation by Phil Zimmermann
- Long trial against Phil Zimmermann because of suspicion of violation of export clauses
- In U.S. free version in cooperation with MIT (agreement with RSA because of then patent)
- Meanwhile commercialized: www.pgp.com
- Gnu Privacy Guard (GPG): non-commercial Open Source variant (OpenPGP, RFC2440)

mobile business

PGP: Encrypt Message

| Klausur MC1 - Nachricht - Microsoft Word | | Frage hier eine | zeben - X | |
|--|--|-----------------|----------------|------|
| Datei Bearbeiten Ansicht Einfügen Format Extras Tabelle Fenster ? | 204 [2] | Frage nier eing | Jepen • * | |
| | | | | |
| A Nur Text → Courier New → 10 → F K U = = = = ↓ = ↓ = ↓ = ↓ = ↓ = ↓ | ∃ • <u>⊿</u> • <u>A</u> • • | | | |
| Endgültige Version enthält Markups 🔹 Anzeigen 🔹 🌚 🌚 🗣 🌚 👻 🏙 🔹 🎽 | | | | |
| 📁 PGPtray - Key Sel | ection Dialog | | | ?) |
| 🖃 Senden 🛛 Konten 🔻 🕕 🔹 🔯 🎭 🧜 🦊 🤻 🏥 Oglionen 👻 I | | Lindiality . | Gine | |
| An Jan Muntermann; Drag users from this list Drag users from this list | to the Recipients list dreas.albers@m-lehrstuhl.de> | Validity | Size 2048/1024 | |
| Cc | | <u> </u> | 3096/1024 | |
| Betreff: Klausur MC1 | och@M-ceniscani.ae> | | 1024 | |
| Hallo Jan, Fitsch@dfki.uni-sb.d | | | 1024 | - |
| Anbei meine Aufgaben für die MC1 Klausur: | | <u> </u> | 1024 | |
| Fritsch@pfsparc01.p | | - | 1024 | |
| heiko rossnagel heiko.rossnagell | | ä | 1024 | |
| universitaet frankfurt direkt: +49-69-7 | ∝ eiko.rossnagel@m-lehrstuhl.de> | | 1024/1024 | |
| | i.Rannenberg@m-lehrstuhl.de> | | 2048 | |
| | i Rannenberg@m-lebrstubl.de> | | 2048/1024 | - |
| Recipients | | Validity | Size | |
| | iko.rossnagel@m-lehrstuhl.de> | | 2048/1024 | _ |
| | unterma@wiwi.uni-frankfurt.de> | <u> </u> | 1024 | |
| | | | | |
| | | | | |
| | | 1 | | |
| | OK | Ca | incel | Help |
| | | | | |
| Conventional Encr | yption | | | |
| | | | | |
| | | | | |
| | | | | |

mobile business

PGP: Decrypt Message

| QCMAS/VPPIP3satAQP+LqxvxF5k4G/TAexpMLX436b1wBp6XP8pa8977c05X0 HHEsO7/tFrJFQJPBCUWouy47p4sR2F0+IXqJuJyHp5ExMGIdmQCpGXE0521jw iHEsO7/tFrJFQJPDck6ias76RBP1sq8VDrAlYopEAcqMMv2pkBuoxy03KC1Rkhi igd1YlowhVX6ZwQCAD2L9WAA97xEUBWHET6kR9n5+oafTBF+R01v6U022T055 ilkh23iQ0119Dryc/uygpcQp72HhtZ7IAjudLvi+Gaeg01WmBjY8q8G1Y61C iD0xMA91k296dhg7NpvzC1ZJ77KMtuBc9BUI8LXDDrvGLwnLtaD5+Evg1ixTu ifvQ3NiGrUEQ0HVxwjQdMtr8C09kREYLuAdD7j/05WtsAdbAVMn72PYF0IRf2 .77MitBfAbxXF0gFS7/b2LccbaR8fx6e1WNFNV07B/9qpd0g5WZVP2eQA5fbw i39jRjPE90b/HLjMPAXUHyneh9QrCXLX5qH0RNcjIYVr0yZGIk8t39059FBd i11Jd59QLiwMmXormxc0g+WR2IxAnGFX17Hy1vjkqm7gfyLxXjgeDCnjxm708J i2shuhKscfvulUg7+0a9SUA94 i2shuhkscfvulUg7+0a9SUA94 i2ncms | on: Heiko Rossnagel etreff: Klausur MC1 | An: Jan Muntermann Cc: | | |
|---|--|---|--|-----------------------|
| QQRAA5/VPPIP3setAQP+LqxvxFSk4G/TAexpHLX436biwBp6xP8pa89R7roSX PCPtray - Enter Passphrase ? HHEsO7/tFrJPQJpPBcUWouy47p4sE2F0+IXqJuJyHp5ExMGIdmQCpGXE0S2Jjw Message was encypted to the following public key(s): ISTXKUB8YJdpPnck61as78BP1sqGVDrA1VopEAeqMMw2pkBuoxy03KC1Rkhi Heiko Rossnagel@melenstuhlde>(DH/2048) IgdpIYlowhX66wQCAD2L9WAA97xEUBWHET6KR9n5+oatTBF+R01v00o22T055 Heiko Rossnagel@melenstuhlde>(DH/2048) Jan Muntermann (munterma@wiwi.unifankfut.de>(RSA/1024) Heiko Rossnagel@melenstuhlde>(RSA/1024) DP3 GEanyD1DUGP9F1XF0xFXHuBc9BUI8LX0DrvGLwnLtaD5+EvgL1XTU Heiko Rossnagel Checko.rossnagel@melenstuhlde>(RSA/1024) DD3 JAQ5Gdhg7NpvzCJ12J7xRtuBc9BUI8LX0DrvGLwnLtaD5+EvgL1XTU Fetrepassphrase for your private key: I Hide Typing 20 TOS jWCRp/Vs590g1aUtcAxd1RAj0PHyFS2EXXMC92ZVNIFM6Ktqm16EL S03/3PSPEs0b/HLjMwPAXUHyneh9QcX1X54PZCAHRS1091Y10r0y2G1ket39059FBd Inter passphrase for your private key: I Hide Typing 1JUGt90LiwMmXorxwc0gHW2ZAL8HyFEVRC5p1JaJFmrzifn2liwfuf8ZT642GBd9bP; I Hide Typing OK Cancel 1Jurc1110172/1023SU344 I Lou Jan, Anbei meine Aufgaben für die MC1 Klausur: I Abei meine Aufgaben für die MC1 Klausur: I Abei meine Aufgaben für die MC1 Klausur: I Copy to Clipboard OK | BEGIN PGP MESSAGE | | | |
| QCMAS/VPF1P3satAQP+LqxvxF5K46/TAxxpMLX43 6b1wBp6xP8p369P7ro5X0 Message was encypted to the following public key(s): Message was encypted to the following public key(s): Message was encypted to the following public key(s): Ifth2507/tF17GQJP5E/UB00v47794zF17G1HZ21X3JUdV+16seg01WmBj7Kg8G1YF01K2 Heiko Rossnagel (heiko.rossnagel@m-ehrstuhl.de> (DH/2048) Jan Muntermann <munterma@wiwi.unifrankfut.de> (R5A/1024) Heiko Rossnagel (heiko.rossnagel@m-ehrstuhl.de> (R5A/1024) DD2XEXEVEQ0ENTV2J2J2J7XRMtuBe9BU18LX0DrvGLwnLtaD5+Evg1LXTD Heiko Rossnagel (heiko.rossnagel@m-ehrstuhl.de> (R5A/1024) D0XXNA91AC96dhg7NpvzJJ2J7XRMtuBe9BU18LX0DrvGLwnLtaD5+Evg1LXTD File Folgontwini.inifrankfut.de> (R5A/1024) D0XXNA91AC96dhg7NpvzJJ2J7XRMtuB9BU18LX0DrvGLwnLtaD5+Evg1LXTD File Folgontwini.inifrankfut.de> (R5A/1024) D0XXNA91AC96dhg7NpvzJJ2LcbakGfXfileRig1D7VFNO7B/9gp40Gg5W2VP2QAS7bey File Folgontwini.inifrankfut.de> (DK 20T05JWCRp7V5590g1aUtcAxd1RAjQPHpVsFS2eXXINC9Z2VNIFM16K2T042G8d9bp OK Cancel F1JGt9QLiwMmXcorxwc0g4WR2Lx4ndFX1/Hy1vjkqm?gfyLXXjgeDCnjxm?068 OK Cancel F1SAUktScfv1Ug7+0asS1Ag4 Heilo Jan, Anbei meine Aufgaben für die MC1 Klausur: Imit Malo Jan, Anbei meine Aufgaben für die MC1 Klausur: Imit Malo Jan, Anbei meine Aufgaben für die MC1 Klausur: Imit Malo Jan, Anbei meine Aufgaben für die MC1 Klausur: Imit Malo Jan, Anbei meine Aufgaben für die MC1 Klausur: Imit Malo Jan, Anbei meine Aufgaben für die MC1</munterma@wiwi.unifrankfut.de> | Version: PGP 8.0 - not li | censed for commercial use: <u>www.pgp.com</u> | | |
| | AQCMA5/VPPIP3satAQP+Lqxvx HEsO7/tFrJFQJpPBcUWouy47 S5TXKtUB8YJdpPnck61as78RB Ag4DIY1owhVX6ZwQCAD2L9WAA Alkh23iQOII9Drye/uygpcQpT DP3GEanyDiDU6R9F1XFOvxPN AOuXNA9iAC96dhg7NpvzCJI2J AfvQ3NiGrUEQsOHVxwjQdMtr8 A77MitBfAbxXF0gFS7/b2Lccb A2oTOSjWCRp/v5s9Og1aUtcAx A39jRjPE9Ob/HLjMwPAXUHyne cr1rhf6ht7SwGgfgGW2aL8Hyi CIJGt9QLiwMmXormxcOg+WR2 JjwtR+1SkqMCXs+PzcAHDsiuG DE3huhK5cfvu1Ug7+Oa9SUAy4 JZncI3vJgkZeZr1bh+pi4dRjs END PGP MESSAGE | FSk4G/TAexpMLX436biwBp6xP8pa89R7ro5Xop4sR2F0+IXqJuJyHp5ExMGIdmQCpGXEoS2IjwP1sq8VDrAlYopEAeqMMw2pkBuoxyo3KCiRkhi Messapetee P1sq8VDrAlYopEAeqMMw2pkBuoxyo3KCiRkhi Messapetee 97xEUBWMET6kR9n5+oafTBF+R01v6UO22T055 Heiko 2HhTt2Y1AjjudLvi+GsegO1WmBjY8q8G1Y61C Messapetee Mk6Ek8hH6q237hhDNDCXkxkSjM3nJ2VuuLvXbb Jan Missapetee 7xRMtuBc9BUI8LX0DrvGLwnLtaD5+EvgL1xTu CO9kREYLuAdD7j/05WtsAdbAVMn72PYF0IRf2 aK8fx6e1VNFnV07B/9qpd0Gg5W2VP2eQA5fbw Enterp d1RAjQPHpVsFS2eXXMnC922vNIFMh6Ktqmt6Ei h9QrCX1X5qH0RNcjIYVrnQyZGIk8t39059FBd FEVRC6piJaJFmrzifnzliwfuf82Tc42GBd9bP Ix4nGFX17Hy1vjkqm7gfyLxXjgeDCnjxm708J Text Viewer Hallo Jan, | ge was encrypted to the following public key(s) : Rossnagel <heiko.rossnagel@m-lehrstuhl.de> (DH/20) untermann <munterma@wiwi.uni-frankfurt.de> (RSA/10) bassphrase for your private key : OK</munterma@wiwi.uni-frankfurt.de></heiko.rossnagel@m-lehrstuhl.de> | 024) I Hide Typing |
| | | | v | |
| | | | | |
| | | Copy to Clipboard | OK | |
| | | | | |



PGP-Certification of Keys

- Certification of public keys by users: "Web of Trust"
- Differentiation between 'validity' and 'trust'
 - 'Trust': trust that a person / an institution signs keys only if their authenticity has really been checked
 - 'Validity': A key is valid for me if it has been signed by a person / an institution I trust (ideally by myself)
- Support through key servers
 - Collection of keys
 - Allocation of 'validity' and 'trust' remains task of the users.
- Path server: finding certification paths between keys

mobile business

PGP: Key Management

| PGPkeys ile <mark>Edit View Keys Server Groups Help</mark> | | | | | l | |
|---|----------|-------|-----------|--|--------------------------|------|
| × 🖷 🌮 😼 🔎 📚 🎘 😓 🏷 🛃 | | | | | | |
| Keys | Validity | Trust | Size | Description 🔺 | | |
| 🕣 🍉 Andreas Albers <andreas.albers@m-lehrstuhl.de></andreas.albers@m-lehrstuhl.de> | 0 | | 2048/1024 | DH/DSS public key | | |
| 🕣 🖙 Elvira Koch <elvira.koch@m-lehrstuhl.de></elvira.koch@m-lehrstuhl.de> | 0 | | 3096/1024 | DH/DSS public key | | |
| 🖅 🖙 fritsch@fsinfo.cs.uni-sb.de | • | | 1024 | RSA legacy public key | | |
| 🗄 🙀 Heiko Rossnagel <heiko.rossnagel@m-lehrstuhl.de></heiko.rossnagel@m-lehrstuhl.de> | | 87777 | 2048/1024 | | | |
| 🗄 🙀 Heiko Rossnagel <heiko.rossnagel@m-lehrstuhl.de></heiko.rossnagel@m-lehrstuhl.de> | | | 1024/10 | othar Fritsch <fritsch@klammer@< td=""><td></td><td>?</td></fritsch@klammer@<> | | ? |
| 🗄 🗺 Jan Muntermann <munterma@wiwi.uni-frankfurt.de></munterma@wiwi.uni-frankfurt.de> | • | | 1024 | General Subkeys | | |
| 🗄 📴 Kai Rannenberg <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | • | | 2048/102 | Control Subkeys | | |
| 🕣 🖙 Kai R. Rannenberg 2048 <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | • | | 2048 | ID: 0xFED07240 | | |
| 🖃 🖙 Lothar Fritsch <fritsch@klammeraffe.org></fritsch@klammeraffe.org> | | | 4096/102 | Type: DH/DSS | | |
| 🗄 📧 Lothar Fritsch <fritsch@klammeraffe.org></fritsch@klammeraffe.org> | • | | | Size: 4096/1024 | | |
| 🗄 🚾 Lothar Fritsch <lothar.fritsch@m-lehrstuhl.de></lothar.fritsch@m-lehrstuhl.de> | • | | | | | |
| | | | | Created: 15.01.2004 | | |
| - 🗸 fritsch@fsinfo.cs.uni-sb.de | | | | Expires: 15.01.2006 | | |
| 🏒 Jan Muntermann <munterma@wiwi.uni-frankfurt.de></munterma@wiwi.uni-frankfurt.de> | | | | Cipher: CAST | | |
| 🗸 Andreas Albers <andreas.albers@m-lehrstuhl.de></andreas.albers@m-lehrstuhl.de> | | | | 🔽 Enabled | | |
| 🕀 📧 Lothar Fritsch <lothar.fritsch@whatismobile.de></lothar.fritsch@whatismobile.de> | 0 | | | | | |
| 🛨 🖙 Stefan Figge <stefan.figge@m-lehrstuhl.de></stefan.figge@m-lehrstuhl.de> | | | 2048/10: | Fingerprint | | |
| | | | | 6075 14A6 1248 5A4A 7E18 6 | 6187 AE57 9E4D FED0 7240 | |
| key(s) selected | | | | , | | |
| | | | | | 🔽 Hexadecimal | |
| | | | | Trust Model | | |
| | | | | Invalid Valid | Untrusted Truste | ed . |
| | | | - | | Close H | lelp |



Key Server

| 🃅 PGPkeys Search Window | | | | | |
|--|----------|-------|-----------|--------------|--------------|
| Search for keys on Idap://keyserver.pgp.com 💌 where | | | | | Search |
| User ID 💌 contains 💌 Kai Rannenberg | | | | | Clear Search |
| More Choices Fewer Choices Search Pending Area | | | | | Help |
| Keys | Validity | Trust | Size | Description | |
| 🕣 🖙 Kai R. Rannenberg 2048 <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | 0 | | 2048 | RSA legacy p | oublic key |
| 🕣 🖙 Kai R. Rannenberg <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | - Ö | | 1024 | RSA legacy p | oublic key |
| 🕀 🖙 kara <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | 0 | | 2048/1024 | DH/DSS pub | lic key |
| | | | | | |
| | | | | | |
| Found 3 key(s) matching search criteria. | | | | | //, |

mobile business

mobile business

PGP: Public Key Catalogs

| <u>File E</u> o | dit <u>V</u> iew F <u>a</u> vorites | <u>T</u> ools <u>H</u> elp | |
|-----------------|-------------------------------------|---|-----------------|
| 🖨 Back | • • • 🕲 🔮 🖞 | 🖞 🔞 Search 📷 Favorites 🛞 Media 🧭 🛃 - 🎒 🕥 - 🗐 🖓 | |
| Address | bttp://blackhole.pd | :a.dfn.de:11371/pks/lookup?op=vindex&search=Kai+Rannenberg | ks ⁱ |
| - , | | | _ |
| | | | |
| Pп | blic Kev | ' Server Verbose Index ``Kai Rannenberg '' | |
| - 4 | | Server verbose much interneting | |
| m | bits/kevID | Date User ID | |
| | · • | | |
| - | | 1997/09/18 kara < <u>kara@iig.uni-freiburg.de</u> > | |
| sig cia | 0B6375FD D5CDE083 | Matthias Schunter < <u>schunter@acm.org</u> > Herbert Dowber (dewber@iig uni freiburg de) | |
| sig | 879AC041 | Herbert Damker < <u>damker@iig.uni-freiburg.de</u> > Binnit Dfitzmann 1 (nfitzblinformatik uni bildezbeim de> NO LECAL DELEUANCE | |
| sig | 8128DC75 | Birgit Pfitzmann 1 < <u>pfitzb@informatik.uni-hildesheim.de</u> > NO LEGAL RELEVANCE | |
| sig | 8128DC75 8EF041F1 | Gerhard Weck < <u>73064.2271@compuserve.com</u> > Kai R. Rannenberg 2048 <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | |
| sig sig | 2F8D5039 | Kai R. Rannenberg 2048 < <u>Raragiig.uni-freiburg.de</u> > Kai Martius <kai@imib.med.tu-dresden.de></kai@imib.med.tu-dresden.de> | - |
| - | | Holger Reif <reif@prakinf.tu-ilmenau.de></reif@prakinf.tu-ilmenau.de> | |
| sig | 5C3C4FE4 | kara <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | |
| sig | AF1FDF70 | | |
| sig | <u>49EF1D84</u> | Hannes Federrath < <u>federrath@inf.tu-dresden.de</u> > Kai R. Rannenberg <kair@microsoft.com></kair@microsoft.com> | |
| | 0B6375FD | Matthias Schunter <schunter@acm.org></schunter@acm.org> | |
| sig | AEB4BCDD | fapp2 AEB4BCDD HSK <fapp2@cam.ac.uk></fapp2@cam.ac.uk> | |
| sig | | <pre>kara <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de></pre> | |
| sig | <u>AF1FDF70</u> 044584B5 | kara < <u>karagiig.uni-freiburg.de</u> > Douglas Swiggum <swiggum@waisman.wisc.edu></swiggum@waisman.wisc.edu> | |
| sig | 04430485 | Kai Rannenberg <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | |
| sig | OCB6E63F | Martin Reichenbach <marei@iig.uni-freiburg.de></marei@iig.uni-freiburg.de> | |
| sig sig | AF1FDF70 | kara <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | |
| Sig | AT IT DT 70 | kara <kara@telematik.iig.uni-freiburg.de></kara@telematik.iig.uni-freiburg.de> | |
| siq | 0B6375FD | Mathias Schunter <schunter@acm.org></schunter@acm.org> | |
| sig | AF1FDF70 | kara <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | |
| 519 | <u>AI 11 01 70</u> | Kai R. Rannenberg <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | |
| sig | 0B6375FD | Matthias Schunter <schunter@acm.org></schunter@acm.org> | |
| sig sig | AF1FDF70 | kara <kara@iig.uni-freiburg.de></kara@iig.uni-freiburg.de> | |
| | AFTED TO | Kara (Karabity, ant-freiburg, ac/ | _1 |
| • | | | |

- Network of public-key servers:
 - pgpkeys.pca.dfn.de
 - www.cam.ac.uk.pgp.net/pgpnet/email-key-server-info.html
 - ...





PGP: Practical Attacks and Weaknesses

- Brute-Force-Attacks on the pass phrase
 - PGPCrack for conventionally encrypted files
- Trojan horses, changed PGP-Code
 - e.g. predictable random numbers, encryption with an additional key
- Attacks on the computer of the user
 - Not physically deleted files
 - Paged memory
 - Keyboard monitoring
- Analysis of electromagnetic radiation
- Non-technical attacks
- Confusion of users [WT99]





- [Bi05] Bishop, Matt: Introduction to Computer Security. Boston: Addison Wesley, 2005. pp. 113-116.
- [DH76] Diffie, Whitfield and Hellman, Martin E.: New Directions in Cryptography, IEEE Transactions on Information Theory, 1976, 22(6), pp. 644-654.
- [RSA78] Rivest, Ron L., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Communications of the* ACM, February 1978, 21(2), pp. 120-126.
- [WT99] Whitten, Alma and Tygar, J.D. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0, In: Proceedings of the 9th USENIX Security Symposium, August 1999, www.gaudior.net/alma/ johnny.pdf