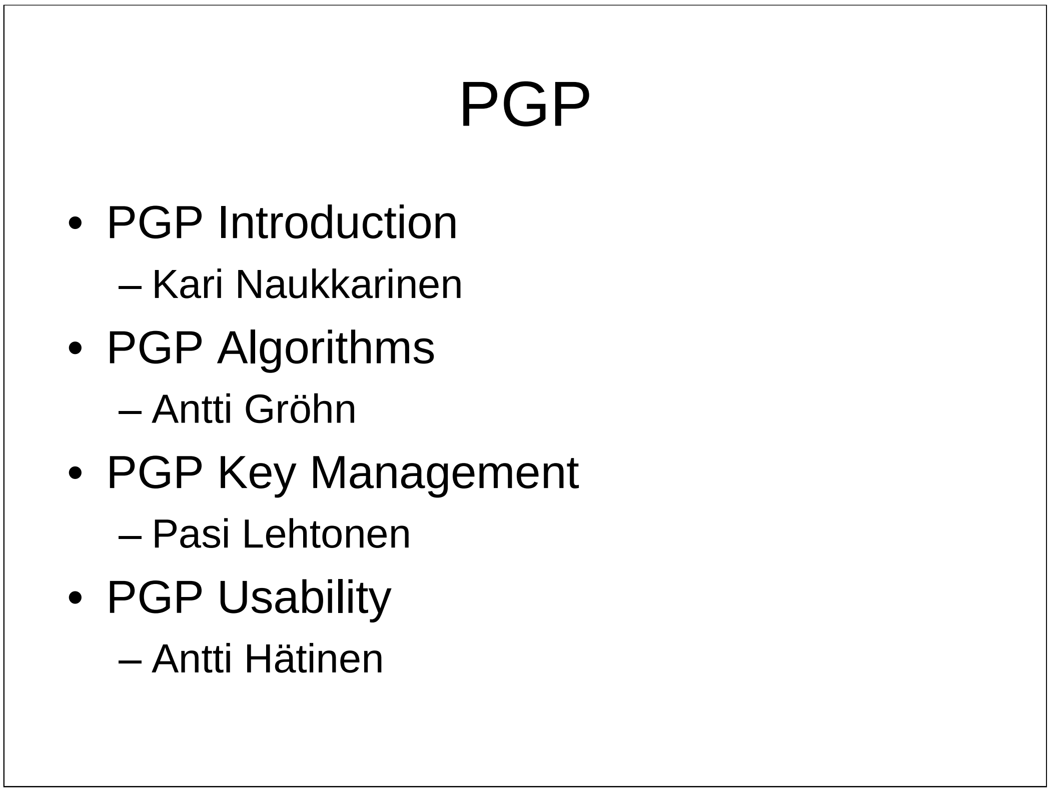


PGP

- PGP Introduction
  - Kari Naukkarinen
- PGP Algorithms
  - Antti Gröhn
- PGP Key Management
  - Pasi Lehtonen
- PGP Usability
  - Antti Hätinen

PGP

Pretty Good Privacy

49845K

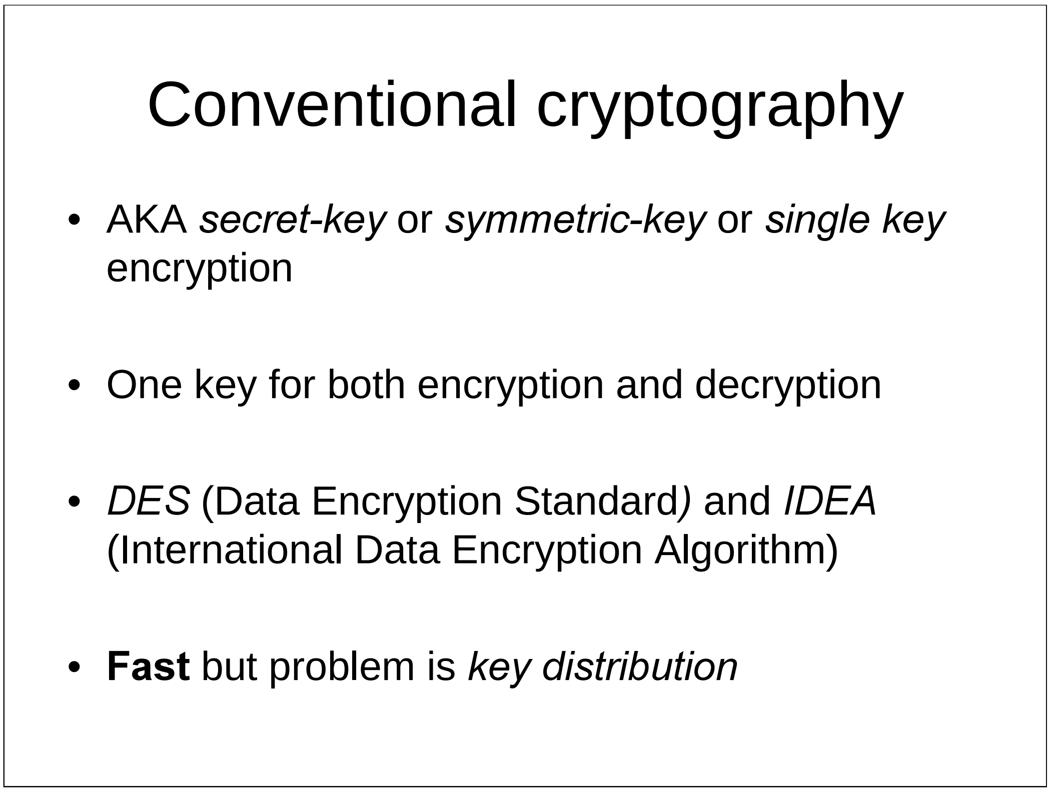
Kari Naukkarinen
knaukkar@cc.hut.fi

Basics

- Originally created by Phil Zimmerman in 1991
- Allows people to exchange files or e-mail with *privacy, authentication, and convenience*
- Public key encryption program
- Available for many different platforms, including Windows, Unix, MS-Dos, OS/2, Macintosh etc.

Terminology

- Cryptography
  - Science of using mathematics to encrypt and decrypt data
- Cryptanalysis
  - Science of analyzing and breaking secure communication
- Encryption and decryption
  - The method of hiding plaintexts substance is called ***encryption***
  - Reverting encrypted text, or *ciphertext* to original plaintext is called ***decryption***.

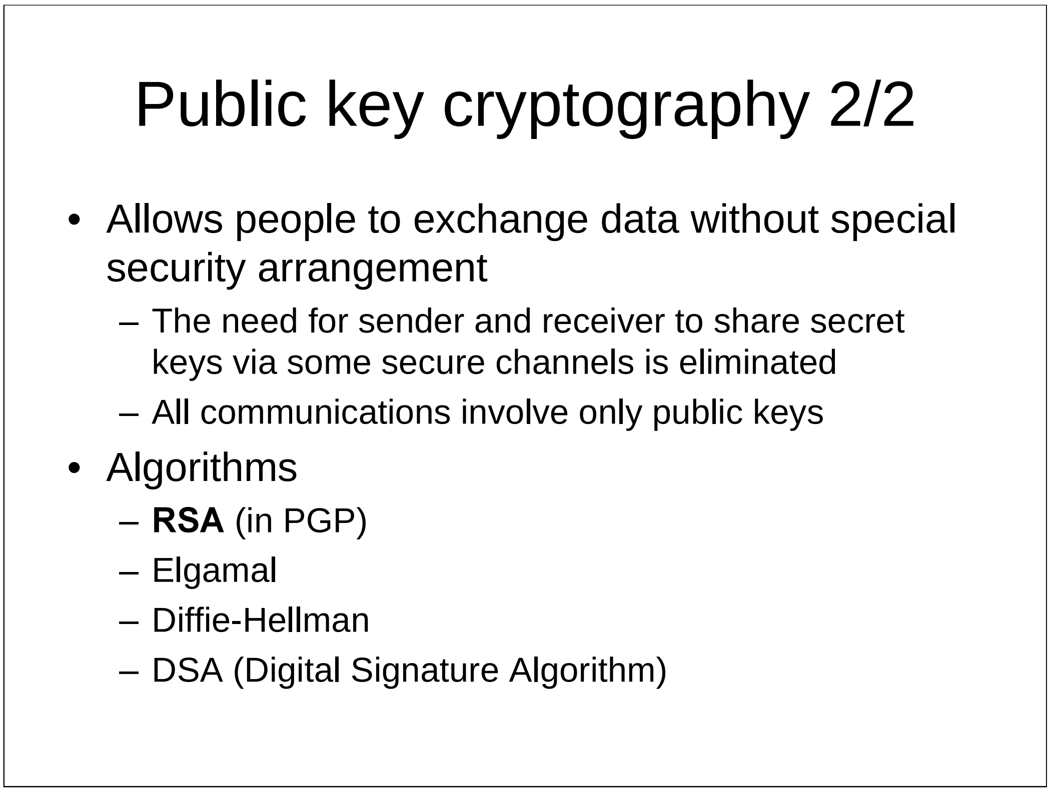


Conventional cryptography

- AKA *secret-key* or *symmetric-key* or *single key* encryption
- One key for both encryption and decryption
- *DES* (Data Encryption Standard*)* and *IDEA* (International Data Encryption Algorithm)
- **Fast** but problem is *key distribution*

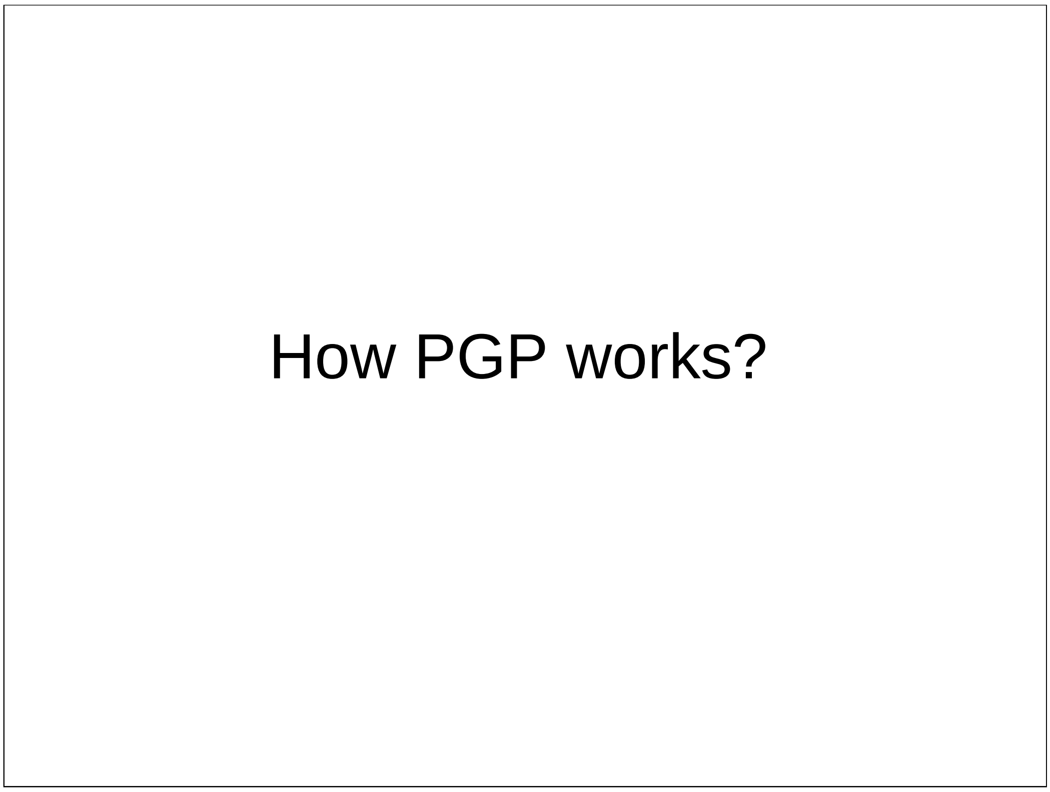
Public key cryptography 1/2

- Introduced by Whitfield Diffie and Martin Hellman in 1975
- Asymmetric method that uses *pair* of keys
  - ***public key*** encrypts data
  - ***private / secret key*** for decryption
- Public keys are published to the world
  - Anyone can use a copy of your public key and encrypt information only you can read
- Private keys are kept in secret
  - Protected by pass phrase



Public key cryptography 2/2

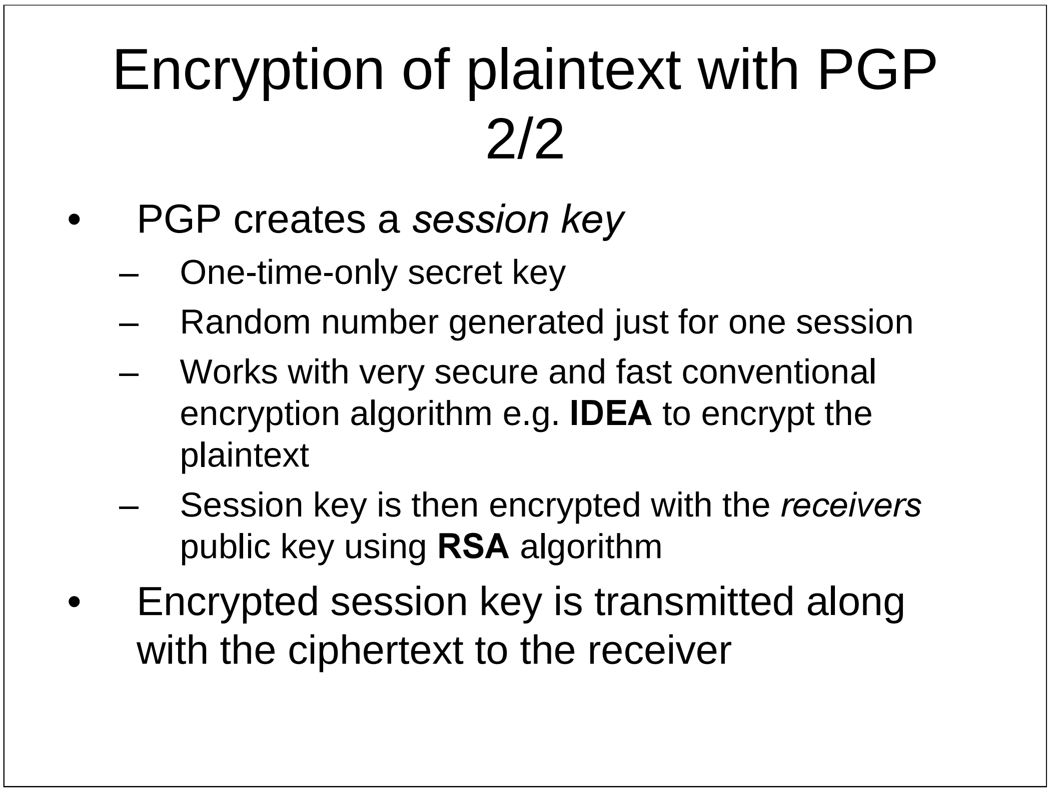
- Allows people to exchange data without special security arrangement
  - The need for sender and receiver to share secret keys via some secure channels is eliminated
  - All communications involve only public keys
- Algorithms
  - **RSA** (in PGP)
  - Elgamal
  - Diffie-Hellman
  - DSA (Digital Signature Algorithm)



How PGP works?

Encryption of plaintext with PGP 1/2

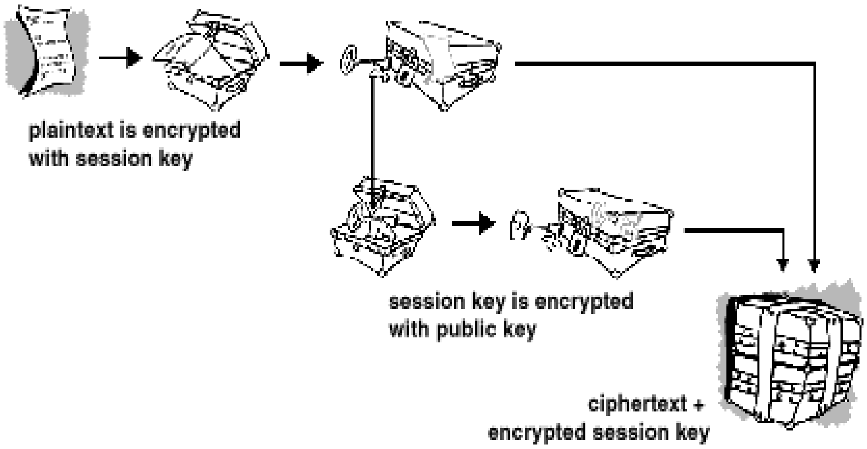
- PGP compress the plaintext
  - Strengthens cryptographic security by reducing patterns found in the plaintext ? more resistant against cryptanalysis
  - Files that are too short to compress or which doesn’t compress well aren’t compressed!!
  - Saves modem transmission time and disk space

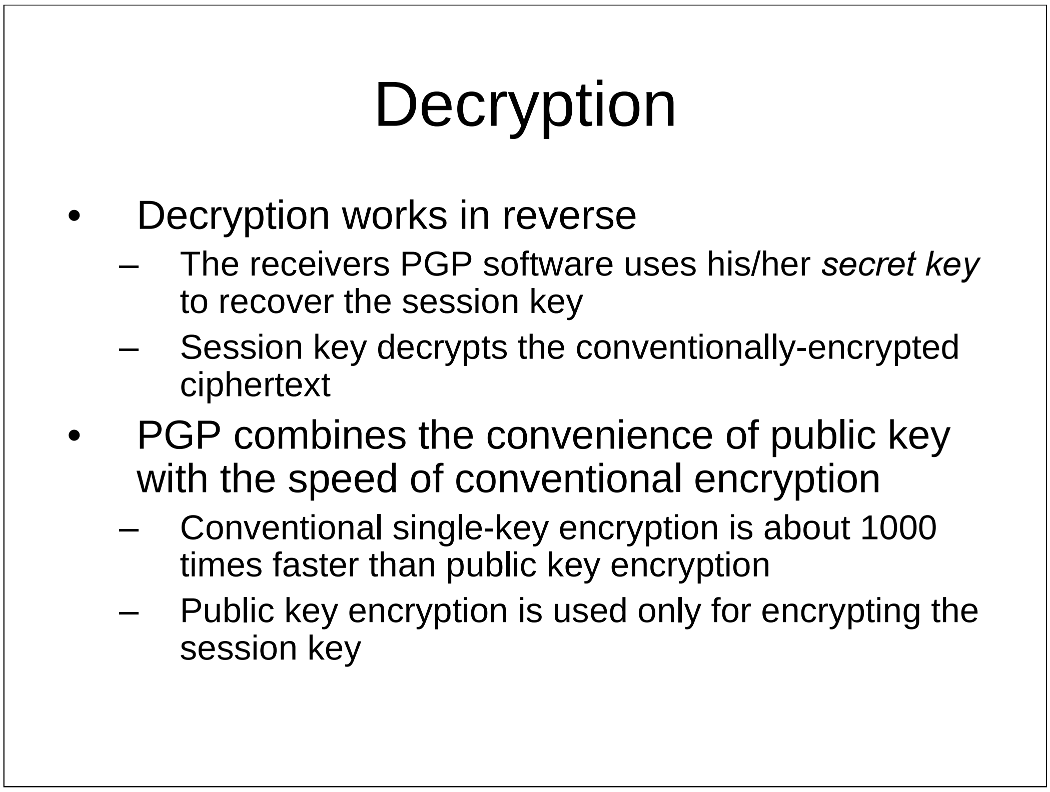


Encryption of plaintext with PGP 2/2

- PGP creates a *session key*
  - One-time-only secret key
  - Random number generated just for one session
  - Works with very secure and fast conventional encryption algorithm e.g. **IDEA** to encrypt the plaintext
  - Session key is then encrypted with the *receivers* public key using **RSA** algorithm
- Encrypted session key is transmitted along with the ciphertext to the receiver

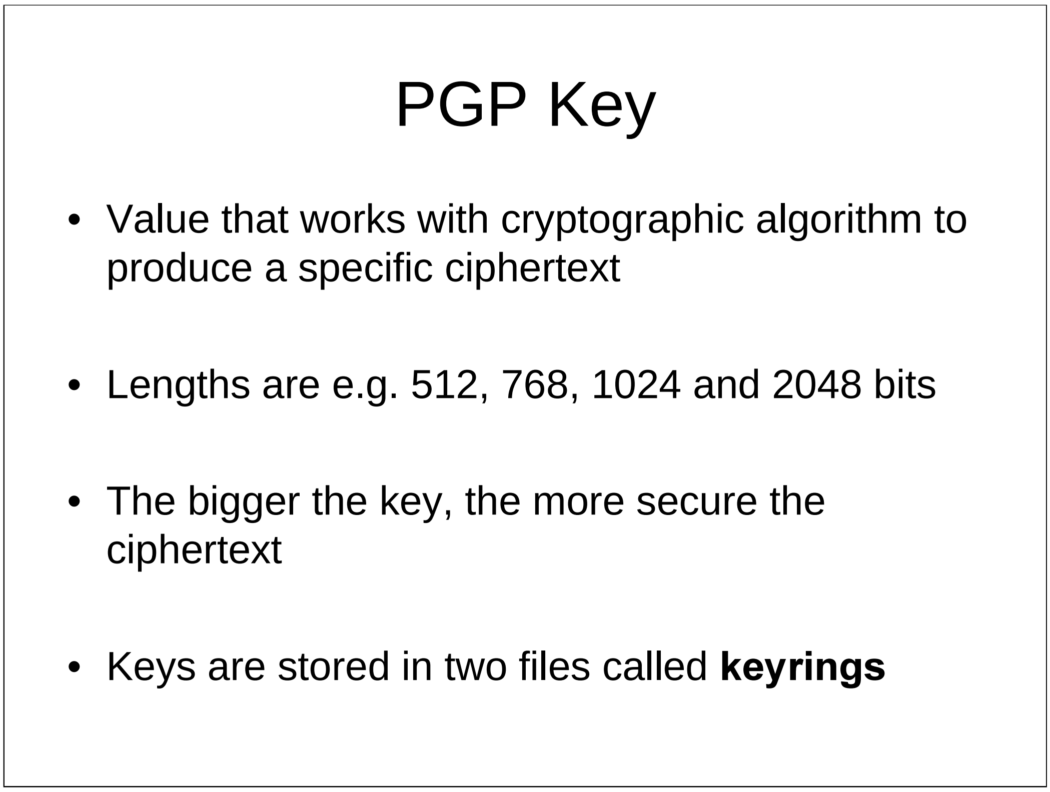
Encryption of plaintext with PGP


plaintext is encrypted with session key
session key is encrypted with public key
ciphertext + encrypted session key



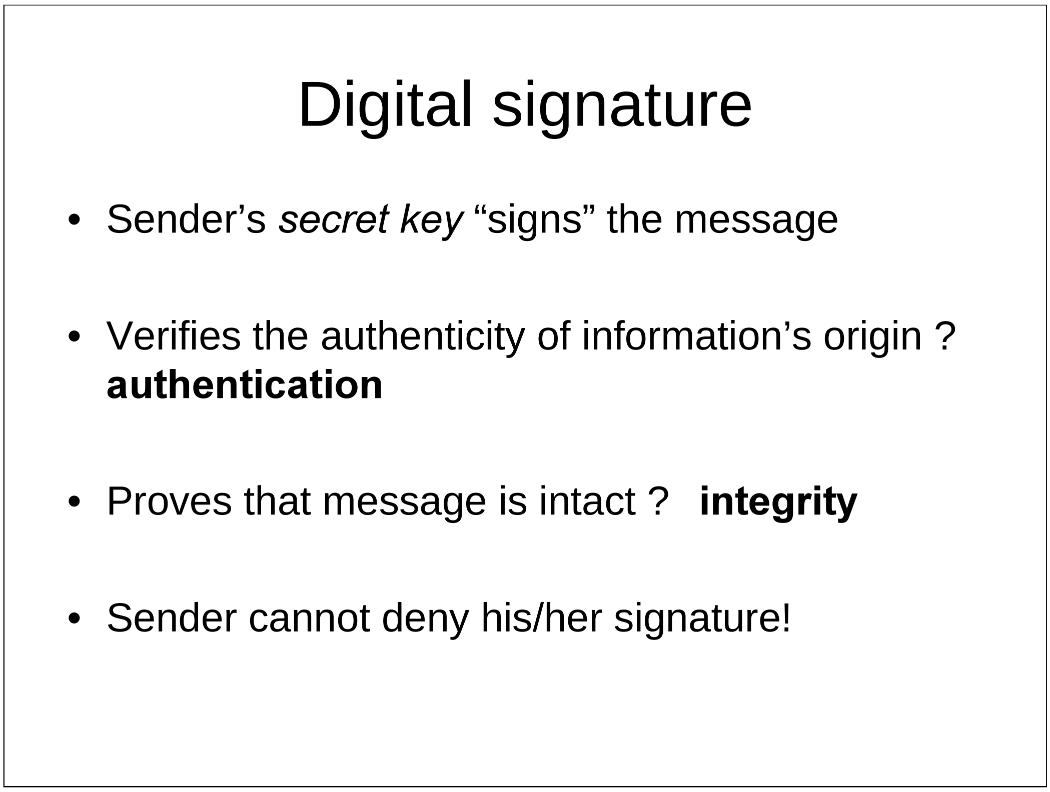
Decryption

- Decryption works in reverse
  - The receivers PGP software uses his/her *secret key* to recover the session key
  - Session key decrypts the conventionally-encrypted ciphertext
- PGP combines the convenience of public key with the speed of conventional encryption
  - Conventional single-key encryption is about 1000 times faster than public key encryption
  - Public key encryption is used only for encrypting the session key



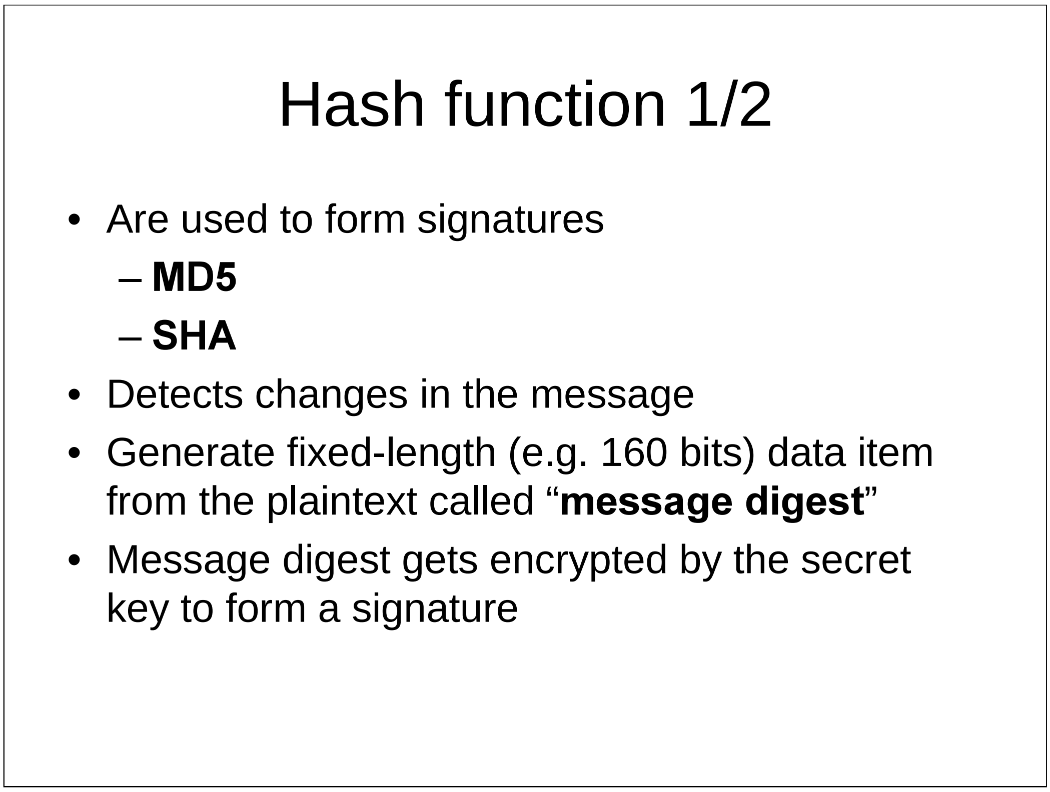
PGP Key

- Value that works with cryptographic algorithm to produce a specific ciphertext
- Lengths are e.g. 512, 768, 1024 and 2048 bits
- The bigger the key, the more secure the ciphertext
- Keys are stored in two files called **keyrings**



Digital signature

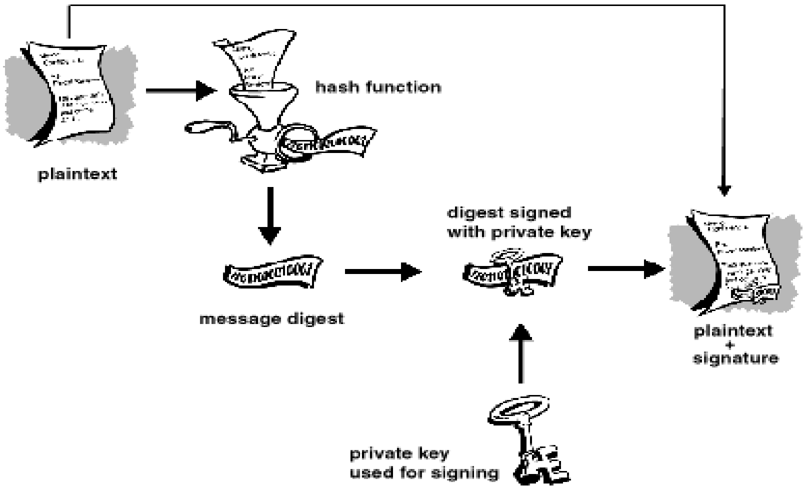
- Sender’s *secret key* “signs” the message
- Verifies the authenticity of information’s origin ? **authentication**
- Proves that message is intact ? **integrity**
- Sender cannot deny his/her signature!



Hash function 1/2

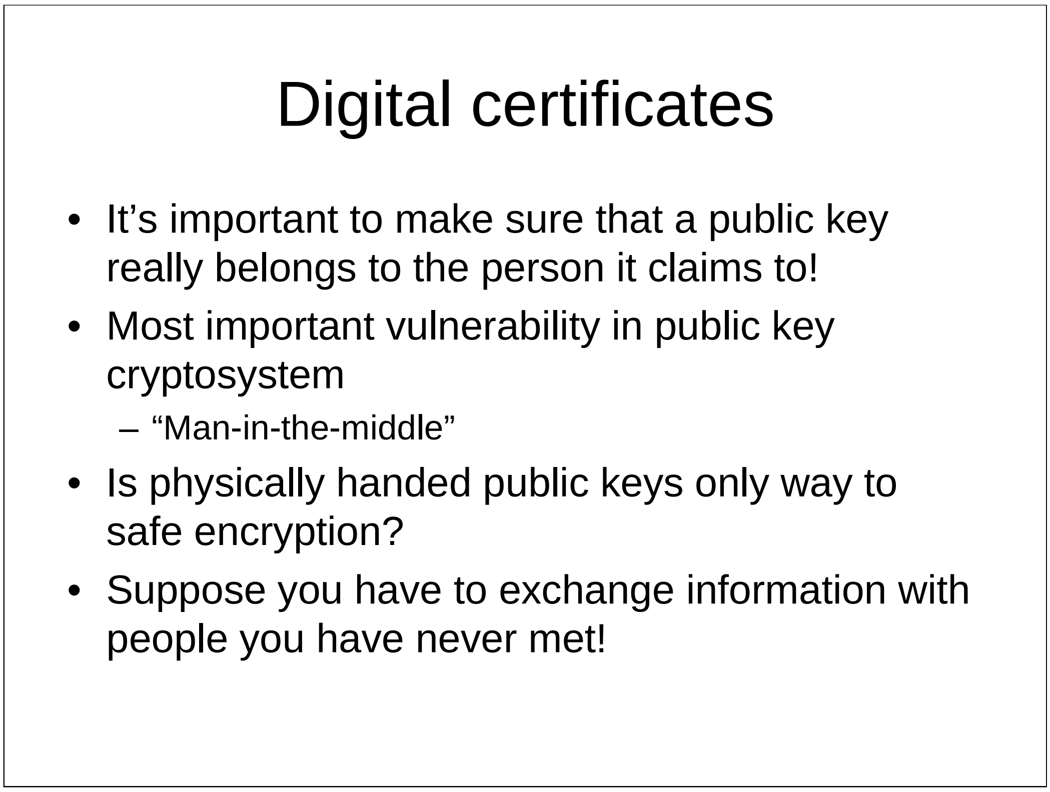
- Are used to form signatures
  - **MD5**
  - **SHA**
- Detects changes in the message
- Generate fixed-length (e.g. 160 bits) data item from the plaintext called “**message digest**”
- Message digest gets encrypted by the secret key to form a signature

Hash function 2/2



The process of using signature

- Hash function generates the digest from the plaintext
- PGP uses the digest and secret key to create the signature
- **This process can be then combined to the encryption before sending the message**
- Receiver’s PGP re-computes the digest from the received plaintext
- Then receivers PGP opens the signature with sender’s public key to get the original digest
- If the digests are equal the sender is verified

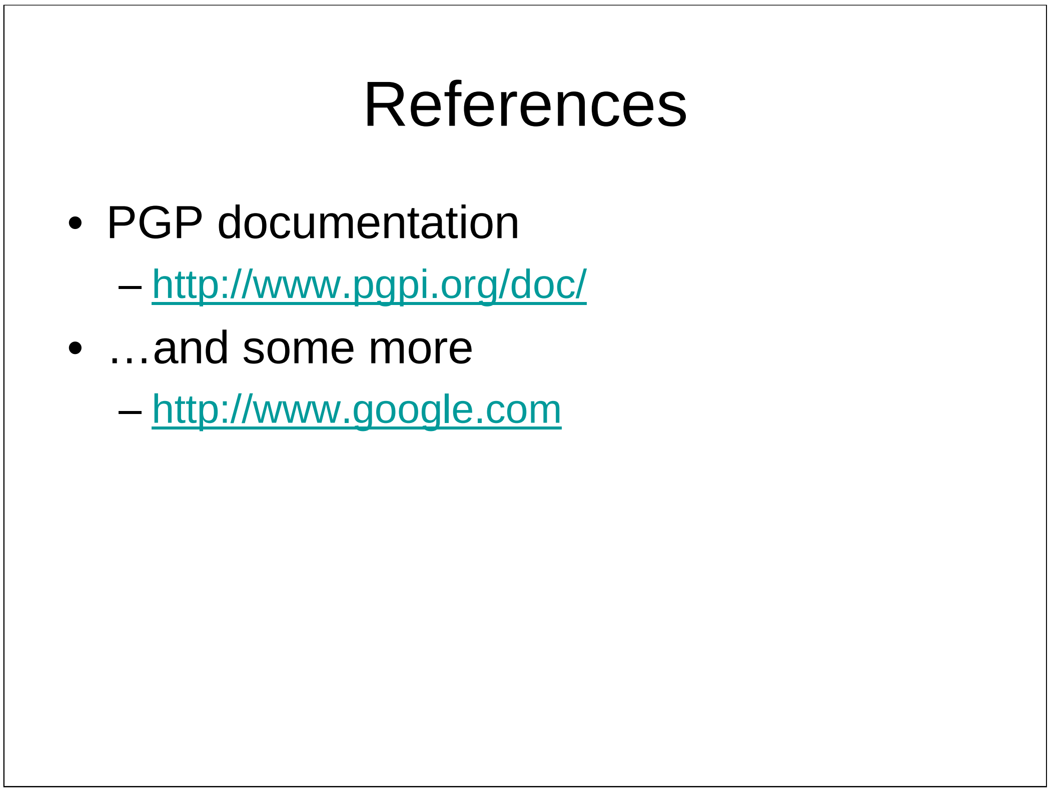


Digital certificates

- It’s important to make sure that a public key really belongs to the person it claims to!
- Most important vulnerability in public key cryptosystem
  - “Man-in-the-middle”
- Is physically handed public keys only way to safe encryption?
- Suppose you have to exchange information with people you have never met!

Digital certificates

- Main idea:
  - Certificate is information included with person’s public key that help others to be confirmed that key is **genuine** or **valid**
- Certificate can be e.g. signature of some trusted person
- *Certificate server* or *key server* is a database that allows users to submit and retrieve digital certificates
- *Public Key Infrastructures (PKIs)* are more structured systems that provide additional key management features

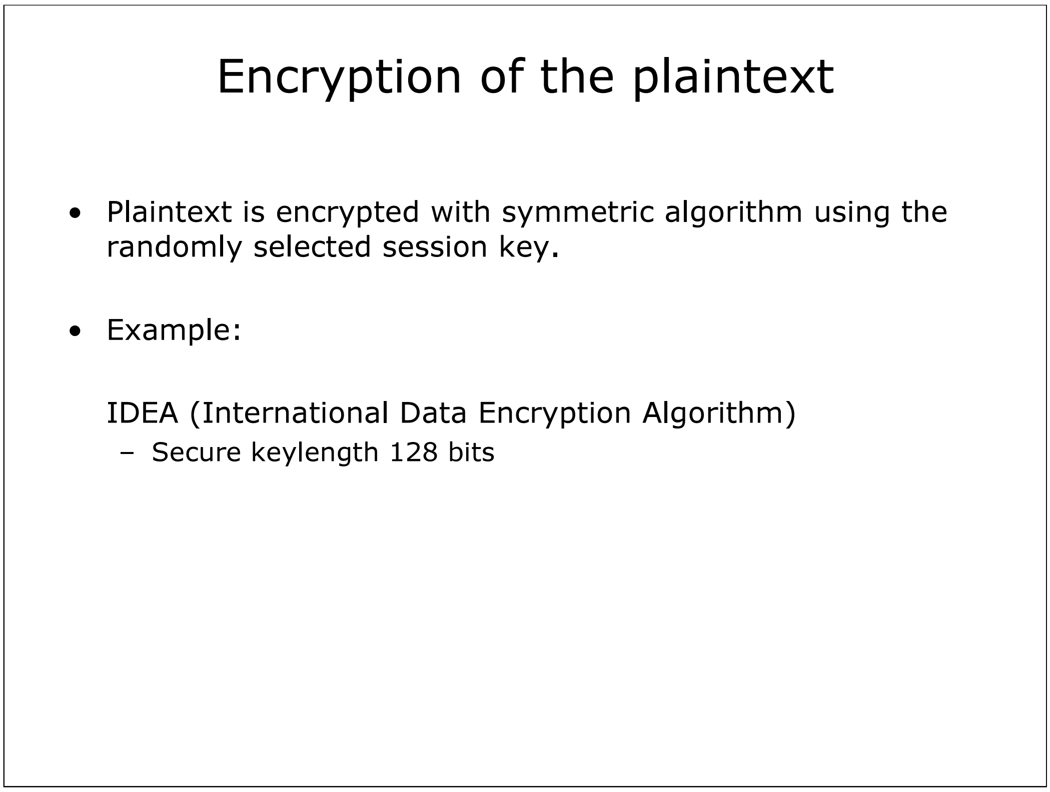


References

- PGP documentation
  - – http://www.pgpi.org/doc/
- …and some more
  - – http://www.google.com

PGP Algorithms

Antti Gröhn
57395T
atgrohn@cc.hut.fi



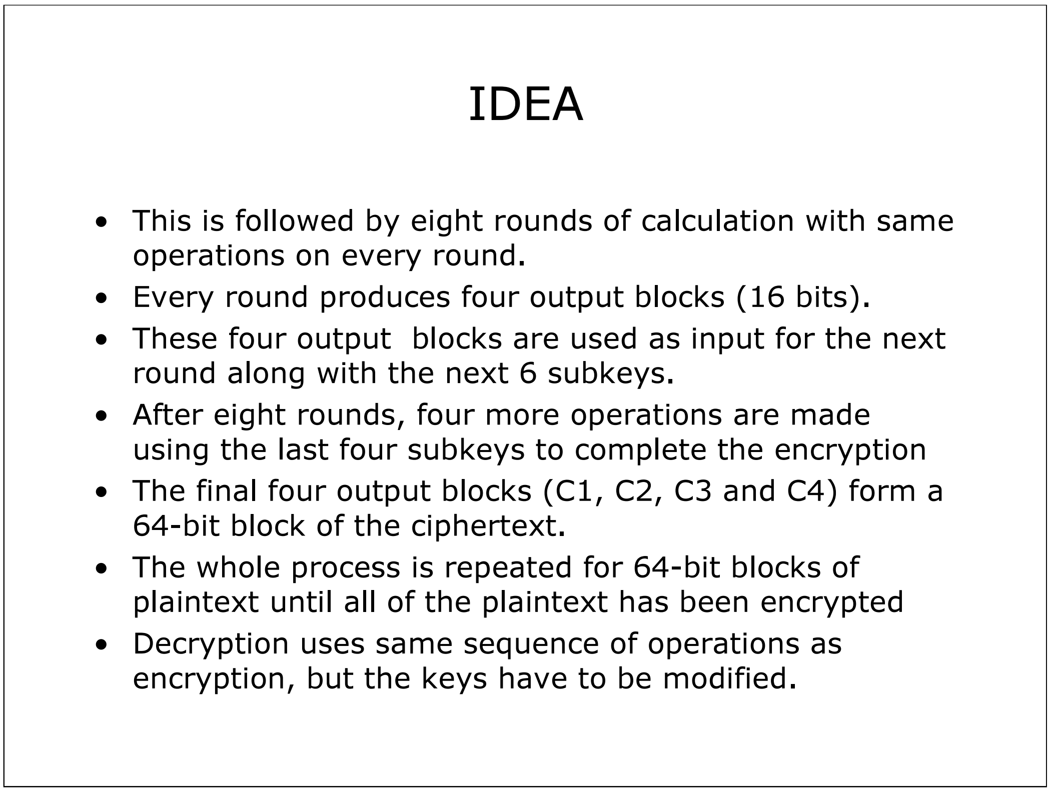
Encryption of the plaintext

- Plaintext is encrypted with symmetric algorithm using the randomly selected session key.
- Example:

  IDEA (International Data Encryption Algorithm)
  - Secure keylength 128 bits

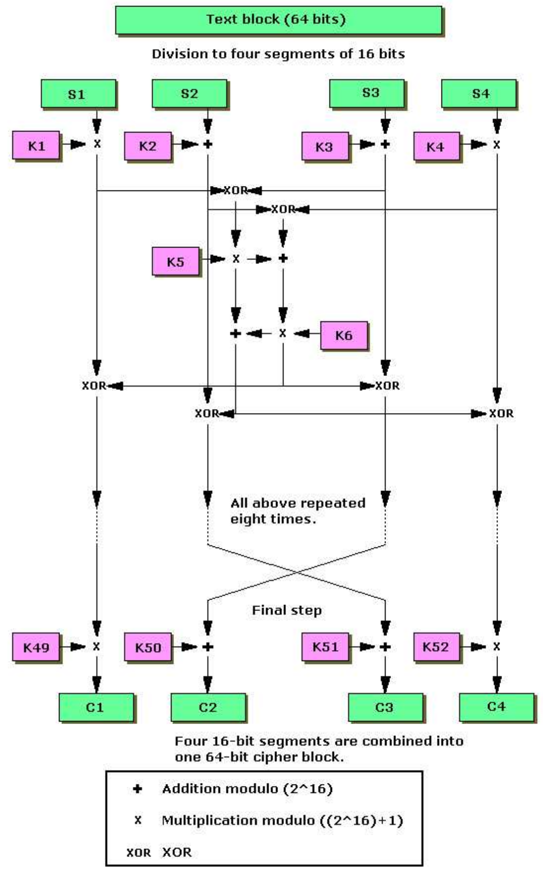
IDEA

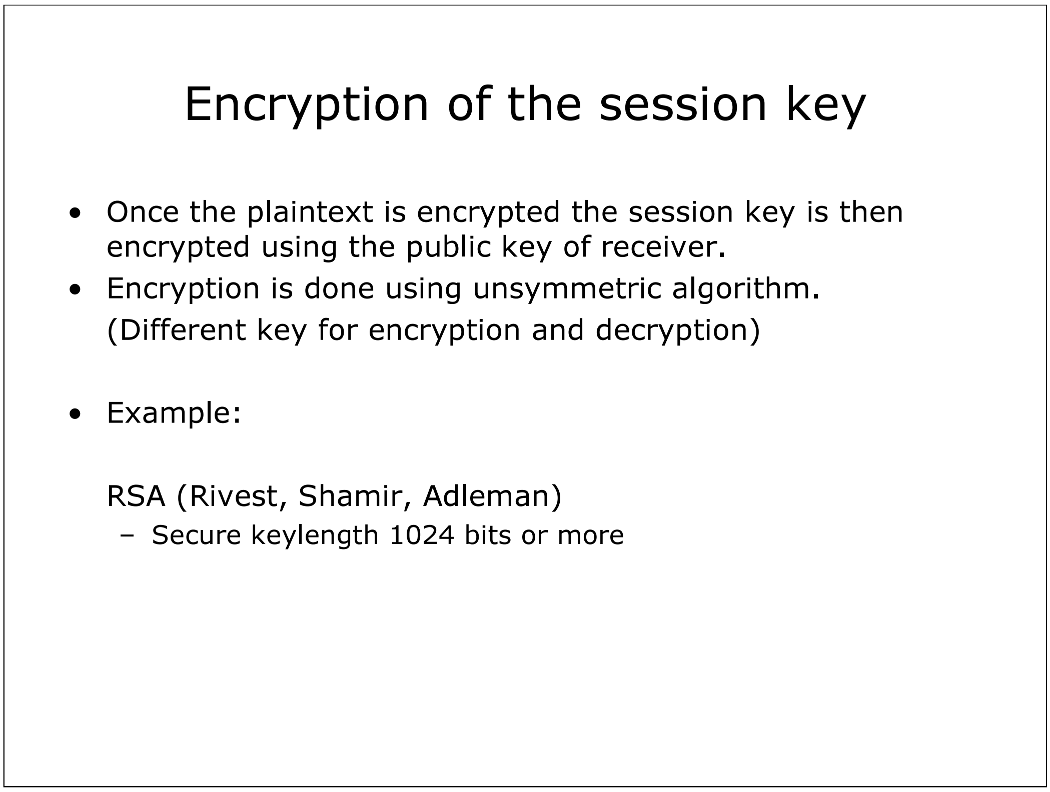
- The 128-bit key is split into 52 subkeys in following way:
  - First the key is split into eight 16-bit keys which are the first eight subkeys.
  - Then the digits of the 128-bit key are shifted 25 bits to the left in order to make a new key for the next eight 16-bit subkeys.
    - Repeated until 52 subkeys have been generated. (K1, K2, K3, K4, ..., K52)
- Plaintext is divided into blocks of 64 bits.
  - Every block is divided into four segments (16 bits). (S1, S2, S3 and S4)



IDEA

- This is followed by eight rounds of calculation with same operations on every round.
- Every round produces four output blocks (16 bits).
- These four output blocks are used as input for the next round along with the next 6 subkeys.
- After eight rounds, four more operations are made using the last four subkeys to complete the encryption
- The final four output blocks (C1, C2, C3 and C4) form a 64-bit block of the ciphertext.
- The whole process is repeated for 64-bit blocks of plaintext until all of the plaintext has been encrypted
- Decryption uses same sequence of operations as encryption, but the keys have to be modified.


Text block (64 bits)
Division to four segments of 16 bits
S1
S2
S3
S4
K1
K2
K3
K4
XOR
K5
K6
All above repeated eight times.
Final step
K49
K50
K51
K52
C1
C2
C3
C4
Four 16-bit segments are combined into one 64-bit cipher block.
+ Addition modulo (2^16)
x Multiplication modulo ((2^16)+1)
XOR XOR

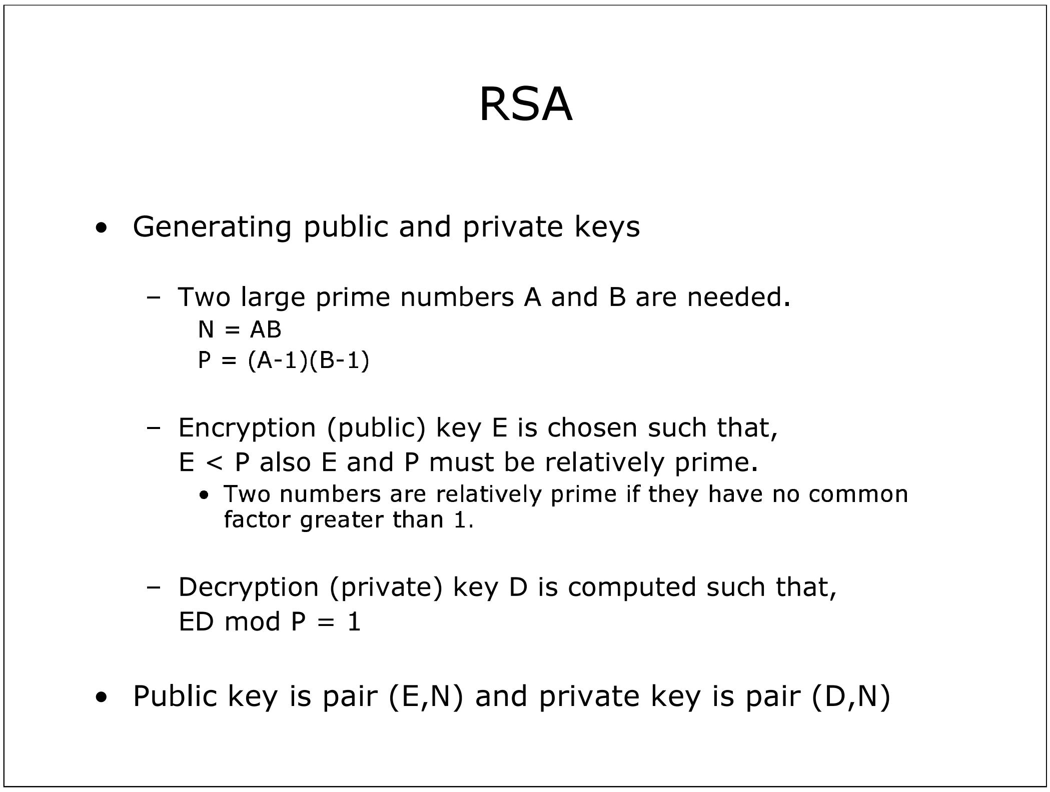


Encryption of the session key

- Once the plaintext is encrypted the session key is then encrypted using the public key of receiver.
- Encryption is done using unsymmetric algorithm.
  (Different key for encryption and decryption)

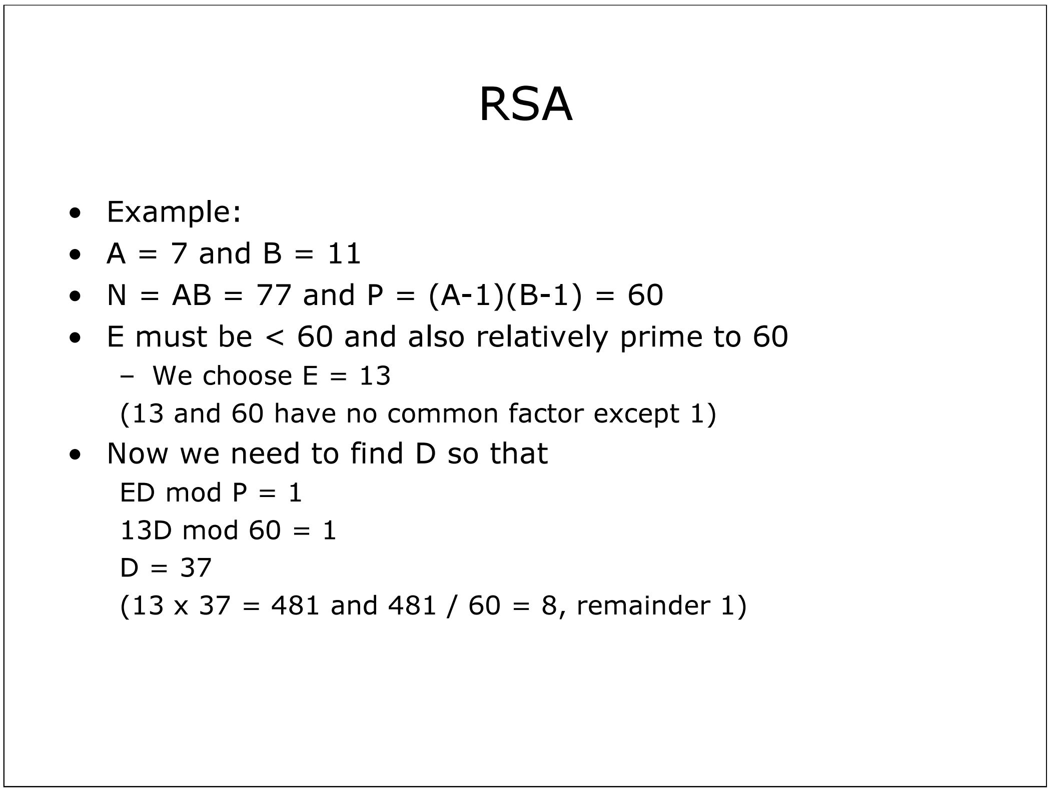
- Example:

  RSA (Rivest, Shamir, Adleman)
  - Secure keylength 1024 bits or more



RSA

- Generating public and private keys
  - Two large prime numbers A and B are needed.
    N = AB
    P = (A-1)(B-1)
  - Encryption (public) key E is chosen such that,
    E < P also E and P must be relatively prime.
    - Two numbers are relatively prime if they have no common factor greater than 1.
  - Decryption (private) key D is computed such that,
    ED mod P = 1
- Public key is pair (E,N) and private key is pair (D,N)



RSA

- Example:
- A = 7 and B = 11
- N = AB = 77 and P = (A-1)(B-1) = 60
- E must be < 60 and also relatively prime to 60
  - We choose E = 13

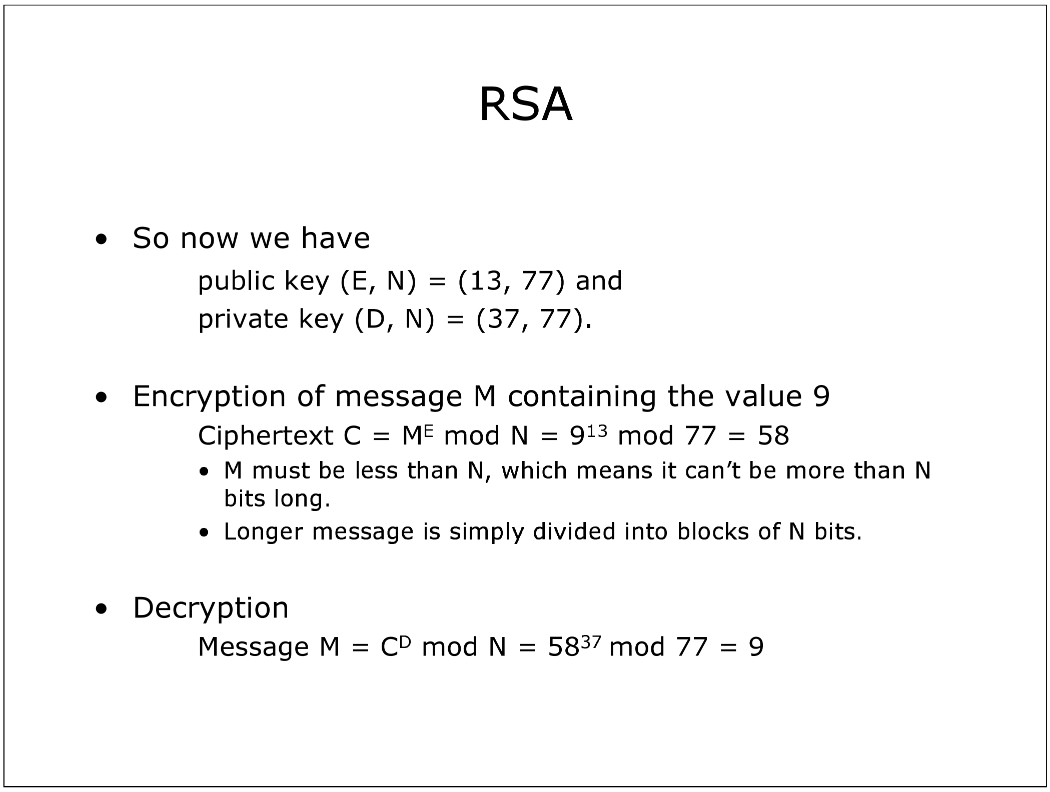
  (13 and 60 have no common factor except 1)
- Now we need to find D so that

  ED mod P = 1

  13D mod 60 = 1

  D = 37

  (13 x 37 = 481 and 481 / 60 = 8, remainder 1)



RSA

- So now we have

  public key (E, N) = (13, 77) and

  private key (D, N) = (37, 77).

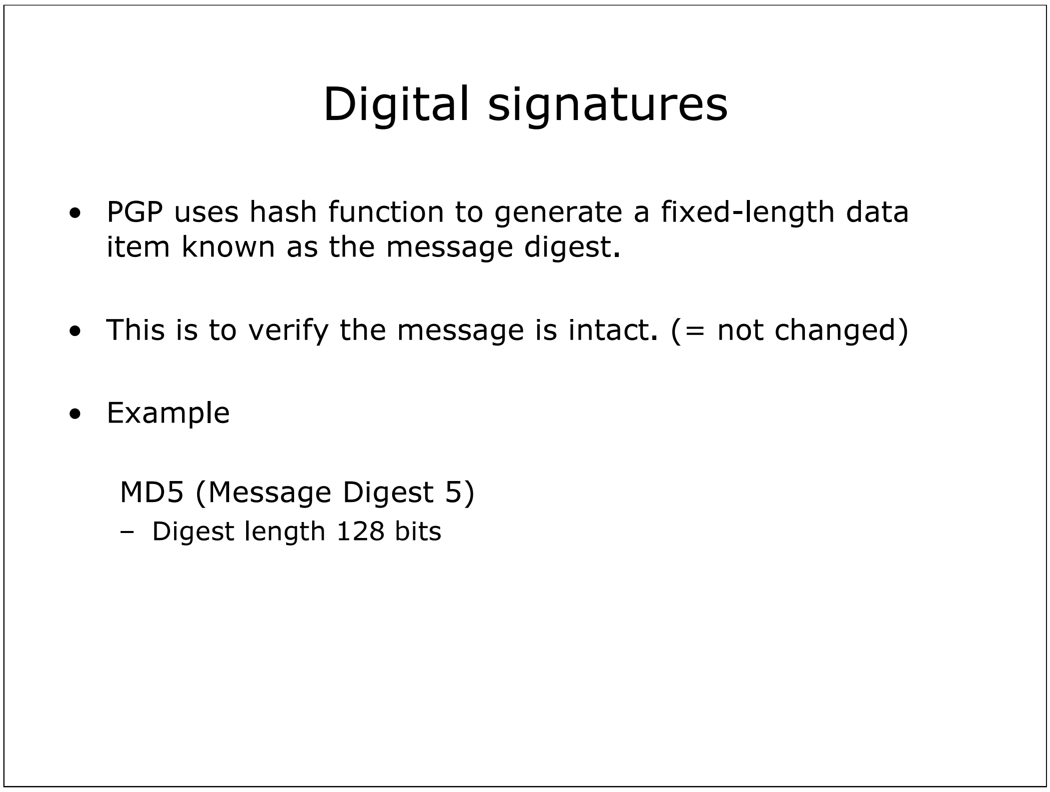
- Encryption of message M containing the value 9

  Ciphertext C = M^E mod N = 9^{13} mod 77 = 58

  - M must be less than N, which means it can’t be more than N bits long.
  - Longer message is simply divided into blocks of N bits.

- Decryption

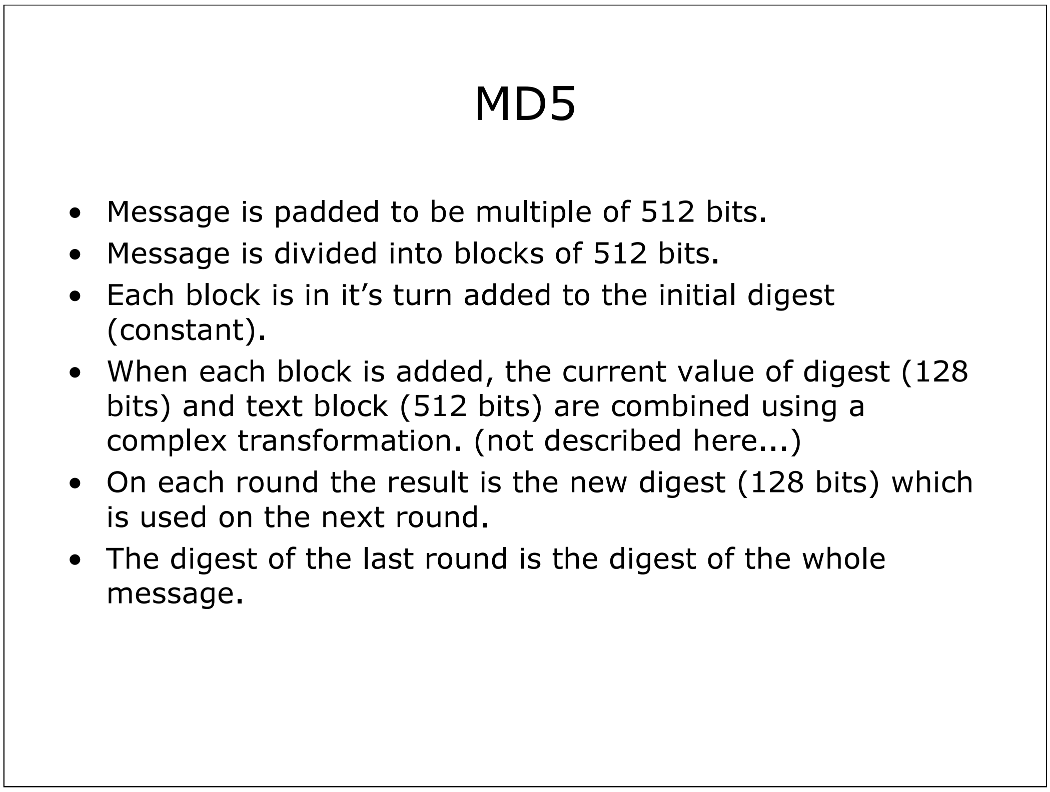
  Message M = C^D mod N = 58^{37} mod 77 = 9



Digital signatures

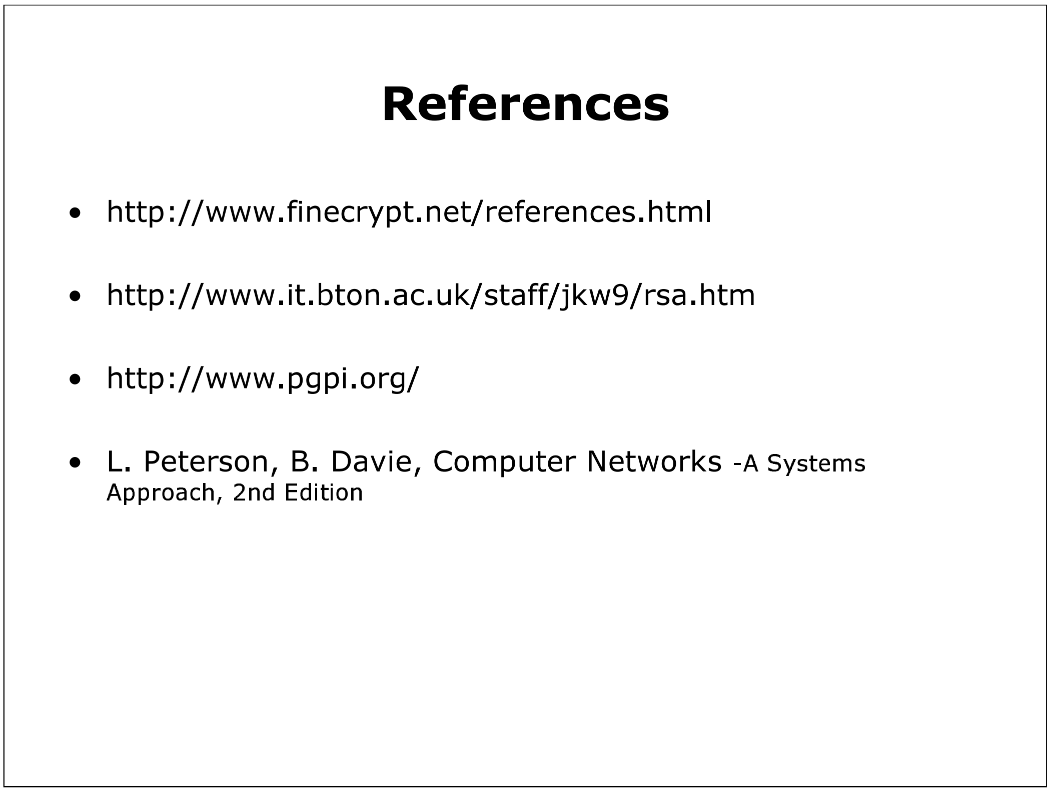
- PGP uses hash function to generate a fixed-length data item known as the message digest.
- This is to verify the message is intact. (= not changed)
- Example

  MD5 (Message Digest 5)
  - Digest length 128 bits



MD5

- Message is padded to be multiple of 512 bits.
- Message is divided into blocks of 512 bits.
- Each block is in it’s turn added to the initial digest (constant).
- When each block is added, the current value of digest (128 bits) and text block (512 bits) are combined using a complex transformation. (not described here...)
- On each round the result is the new digest (128 bits) which is used on the next round.
- The digest of the last round is the digest of the whole message.



References

- http://www.finecrypt.net/references.html
- http://www.it.bton.ac.uk/staff/jkw9/rsa.htm
- http://www.pgpi.org/
- L. Peterson, B. Davie, Computer Networks -A Systems Approach, 2nd Edition

PGP keys and key rings

- There are four kinds of keys in PGP
  - one-time session keys
  - public keys
  - private keys
  - pass phrase keys
- There are two kinds of key rings
  - public key rings
  - secret key rings

How to protect public keys from tampering

- In the public key cryptosystem you don’t have to protect public keys from exposure
- It is important to protect public keys from tampering

  - this may be the most important vulnerability in a public-key cryptosystem

Potential disaster

- You have a private message to Alice
- You download Alice’s public key certificate
- You encrypt your letter to Alice with a public key and send it to her
- Another user named Charlie has generated a public key of his own with Alice’s user ID attached to it
- Charlie covertly substitutes his bogus key in place of Alice’s real public key
- All looks normal because this bogus key has Alice’s user ID

Potential disaster

- You unwittingly use the bogus key belonging to Charlie instead of Alice’s real public key
- Now Charlie can decipher the message intended for Alice because he has the matching secret key
- He may even re-encrypt the deciphered message with Alice’s real public key and send it to her, so that no one suspects anything
- Furthermore, with the secret key Charlie can falsify Alice’s signatures

How to prevent the disaster

- The only way to prevent the disaster is to prevent anyone from tampering with public keys
  1. If you got Alice’s public key directly from Alice
  2. You could get Alice’s public key from a mutual trusted friend David
     - ➢ David could sign Alice’s public key
     - ➢ David would create this signature with his own secret key
     - ➢ This would create *a signed public key certificate*
     - ➢ This requires you have a known proper copy of David’s public key to check his signature
     - ➢ Perhaps David could also provide Alice with a signed copy of your public key
     - ➢ David is thus serving as *an introducer* between you and Alice

Service of introducing users to each other

- A widely trusted person could even specialize in providing service of introducing users to each other
  - The trusted person could be regarded as a key server
- Any public key certificates bearing the key server’s signature could be trusted as truly belonging to whom they appear to belong to
- All users who wanted to participate need a known copy of just the key server’s public key
- A trusted centralized key server or Certifying Authority is especially appropriate for large impersonal centrally-controlled corporate or government institutions
- There is no Public Key Infrastructure (PKI) in PGP

Service of introducing users to each other

- Some institutional environments use hierarchies of *Certifying Authorities*
- More decentralized environments allowing all users to act as a trusted introducers to their friends, would probably work better than a centralized key server
  - PGP tends to emphasize the decentralized non-institutional approach
  - PGP allows people to better choose who they can trust for key management
- Protecting public keys from tampering is the most difficult problem in the public key applications
  - It is the Achilles’ heel of public key cryptography

Using a public key

- ***A public key should only be used after a certainty is received that the key actually belongs to the person it claims to***
- A reasonable certainty is received if the public key certificate is learned directly from its owner, or if it bears the signature of someone else that you trust, from whom you already have received a good public key
- Also the user ID should have the full name of the key’s owner
- No matter how tempted you are, ***NEVER give in to expediency and trust a public key you downloaded from a bulletin board, unless it is signed by someone you trust***
- The uncertified public key could have been tampered by anyone

Certifying keys

- If you are asked to sign someone else’s public key certificate, make certain that it really belongs to that person
- This is because your signature on her public key certificate is a promise by you that this public key really belongs to her
- Other people who trust you, will accept her public key, because it bears your signature
- Preferably you should sign someone else’s public key certificate only if you got it directly from her
- ***In order to sign a public key, you must be far more certain of that key’s ownership than if you only want to use that key to encrypt a message***

Certifying keys

- To be convinced of a key is valid enough to use it, certifying signatures from trusted introducers should be sufficient
- To sign a key yourself, you should require your own independent firsthand knowledge of who owns that key
- Bear in mind that your signature on public key certificate does not vouch for the integrity of the person, but only vouches for the integrity (the ownership) of the person’s public key
- Other people would accept the key as belonging to him because you signed it (assuming they trusted you), but they would not trust the key’s owner
- ***Trusting a key is not the same as trusting the key’s owner***

PGP and a public key ring

- It would be a good idea to keep your own public key on hand with a collection of certifying signatures attached from a variety of introducers
- You could post your key with its attached collection of certifying signatures on various bulletin boards
- If you sign someone else’s public key, return it to them with your signature so that they can add it to their own collection of credentials for their own public key
- PGP keeps track of which keys on your public key ring are properly certified with signatures from introducers that you trust
- All you have to do is to tell PGP which people you trust as introducers, and certify their keys yourself with your own ultimately trusted key

Secure of your public key ring

- Make sure no one can tamper with your own public key ring
- Checking a new signed public key certificate must ultimately depend on the integrity of the trusted public keys that are already on your own public key ring
- ***Maintain physical control of your public key ring***
  - This is to protect it from tampering, not from disclosure
- Keep a trusted backup copy of your public key ring and your secret key ring on write protected media
- ***Your trusted public key is the most important key to protect from tampering***

How does PGP keep track of which keys are valid

- PGP knows which keys on your public key ring are properly certified
- All you have to do is to tell PGP which people you trust as introducers
- Keys that have been certified by a trusted introducer are deemed valid by PGP
- PGP also allows for the possibility of you having several shades of trust for people to act as introducers
- Your trust for the key's owner to act as an introducer does not just reflect your estimation of their personal integrity
  - ***It should also reflect how competent you think they are at understanding key management and using good judgment in signing keys***
- This trust information is stored on your key ring with their key
- When PGP is calculating the validity of a public key, it examines the trust level of all attached certifying signatures

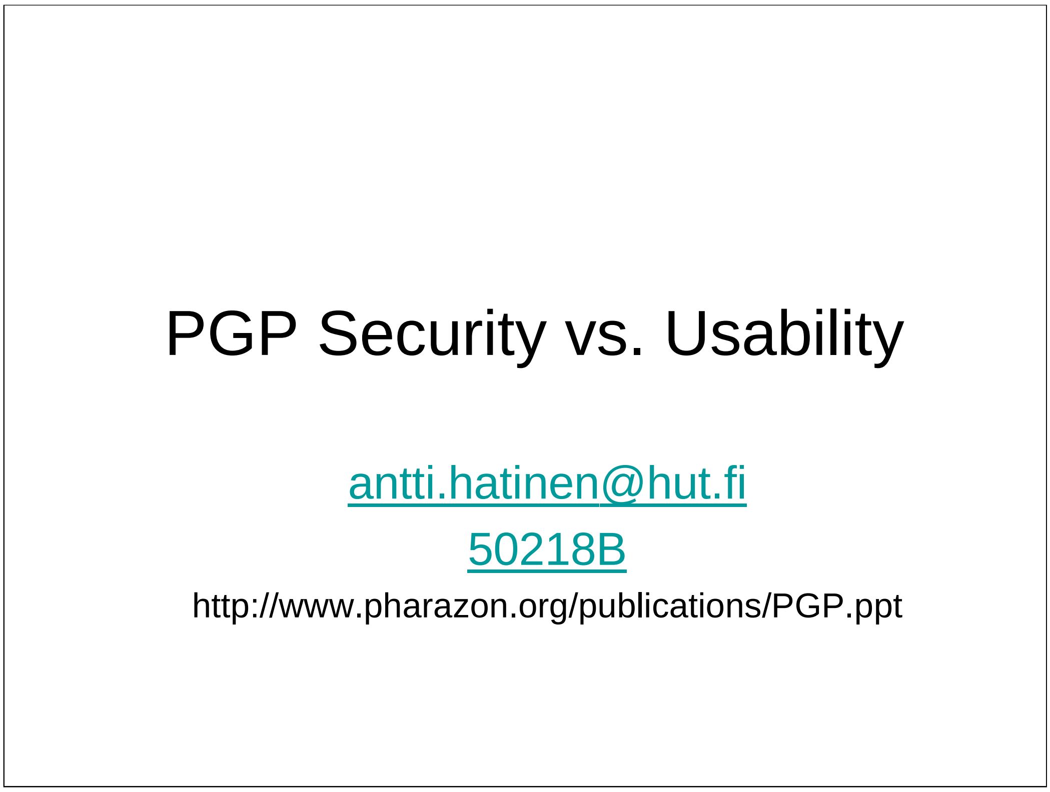
How does PGP keep track of which keys are valid

- As time goes on, you will accumulate keys from other people that you may want to designate as trusted introducers
- Everyone else will each choose their own trusted introducers
- Everyone will gradually accumulate and distribute with their key collection of certifying signatures from other people, with the expectation that anyone receiving it will trust at least one or two of the signatures
- This will cause the emergence of a decentralized fault tolerant web of confidence for all public keys
- ***PGP lets you alone choose who you trust, putting you at the top of your own private certification pyramid***
- PGP is for people who prefer to pack their own parachutes

References

- PGP documentation
  - http://www.pgpi.org/doc/

- S-38.153 Lecture slides 2002
  - http://www.netlab.hut.fi/opetus/s38153/k2002/slides/S38153_12b.pdf
- Cryptographic Reference Center
  - http://pgp.rasip.fer.hr/



PGP Security vs. Usability

antti.hatinen@hut.fi

50218B

http://www.pharazon.org/publications/PGP.ppt

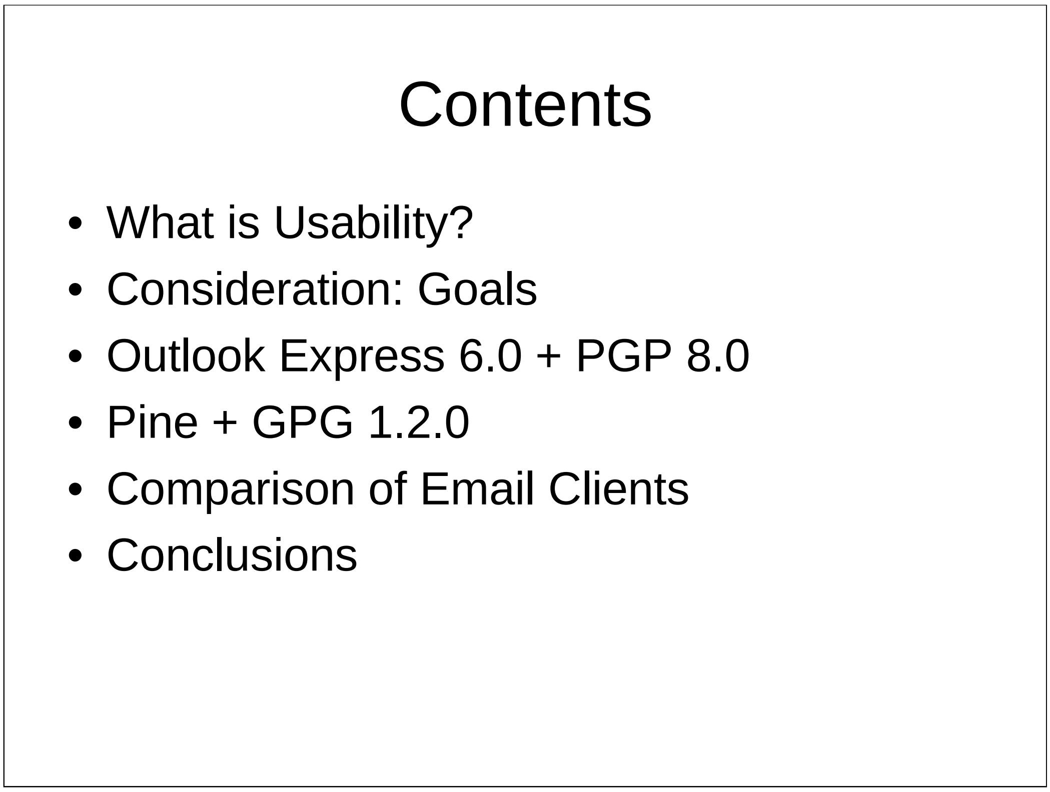
Helsinki University of Technology

Department of Electrical and Communications Engineering

3.2.2003

About the Author

Antti Hätinen studies 5th year Telecommunication at the Helsinki University of Technology and second major at the University of Helsinki, Department of Computer Science. He has also minors in Work Psychology, Mathematics and Marketing & Entrepreneurship at the Helsinki School of Economics. His special interests are usability, software engineering, internet security and entrepreneurship. Other works are available at http://www.pharazon.org/publications/ .



Contents

- What is Usability?
- Consideration: Goals
- Outlook Express 6.0 + PGP 8.0
- Pine + GPG 1.2.0
- Comparison of Email Clients
- Conclusions

Contents of this presentation

•Usability Theory

•Usability Definitions

•Alice’s Goals

•Corporate Goals

•Threat Model

•Walkthroughs

•Outlook Express 6.0 & PGP 8.0

•Pine 4.50 & GPG 1.2.0

•Measurements

•Email Client Comparison

•Sending Mails

•Reading Mails

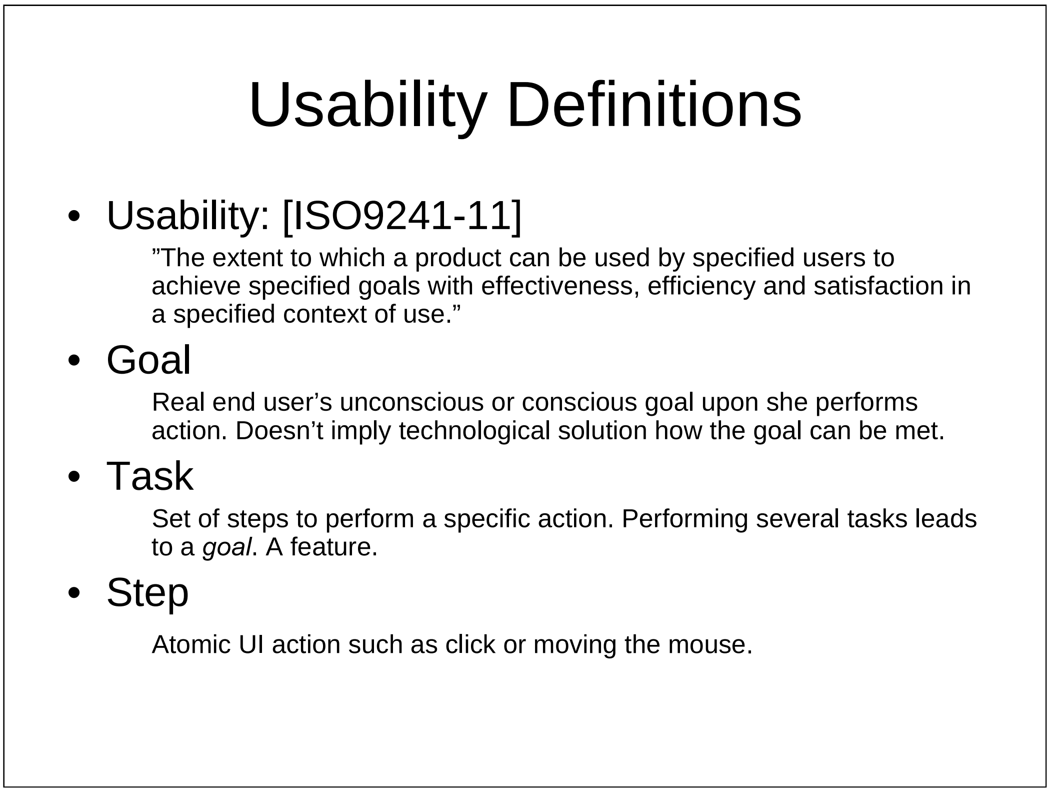
•Conclusions

•Cost of Bad Usability

•Design Implications

•Summary

•References



Usability Definitions

- Usability: [ISO9241-11]

  ”The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.”
- Goal

  Real end user’s unconscious or conscious goal upon she performs action. Doesn’t imply technological solution how the goal can be met.
- Task

  Set of steps to perform a specific action. Performing several tasks leads to a *goal*. A feature.
- Step

  Atomic UI action such as click or moving the mouse.

Usability Theory

In this chapter we first define the used usability terms. Next we introduce two example personas using PGP-encryption and their motives. Finally there’s introduction of the threat models and how they affect people’s goals.

Usability Terms Defined

Alone in Helsinki area there exists at least five different schools of usability who all have more or less different methods and definitions of usability. I personally have adopted the view of Sari Laakso from the Department of Computer Science at the University of Helsinki. In her view, the essential concept is the end user’s goal, the target he performs action to archieve consciously or unconsciously. The concept of goal has different interperations in the literature. Here we call Alan Cooper’s definition Basic Goals [Cooper95]. In his view, people have goals such as not to look stupid, don’t do big mistakes, do enough work and have fun (or at least do not get bored). These goals can be seen to relate to the basic human need hierarchy by for example Maslow [Maslow54].

Sari Laakso has an other definition of design usable goals. Hereafter we make distinction between basic goals defined before, and goals, that aren’t as abstract as Cooper’s basic goals, don’t imply the technical solution the goal can be arcieved and have status data of the situation [Hätinen02b]. For example a goal could be that it’s 18 o’clock and a student wants to write a physics labratory report on her measurements, so that she could pass the course [Hätinen02a]. The student has two options to select the technological device to archieve the goal, she can take a pen and paper and write the report by hand, or she can alternatively use a computer. The design implication of the concept of goal is that the user is likely to choose the apparatus that helps the user to archieve her goal with less physical and cognitive burden. Usually the goal is acompanied with a persona, that has name, picture and description of the user.

Now we can discuss the definition of usability itself. In Finnish the word ”käytettävyys” is even more ambigous than in English. In the field of production management the other widely used definition is availability of machinery, or the uptime of for example a paper machine how many percent of the time it’s in production versus in maintenance. ISO however defines usability quite well as follows:

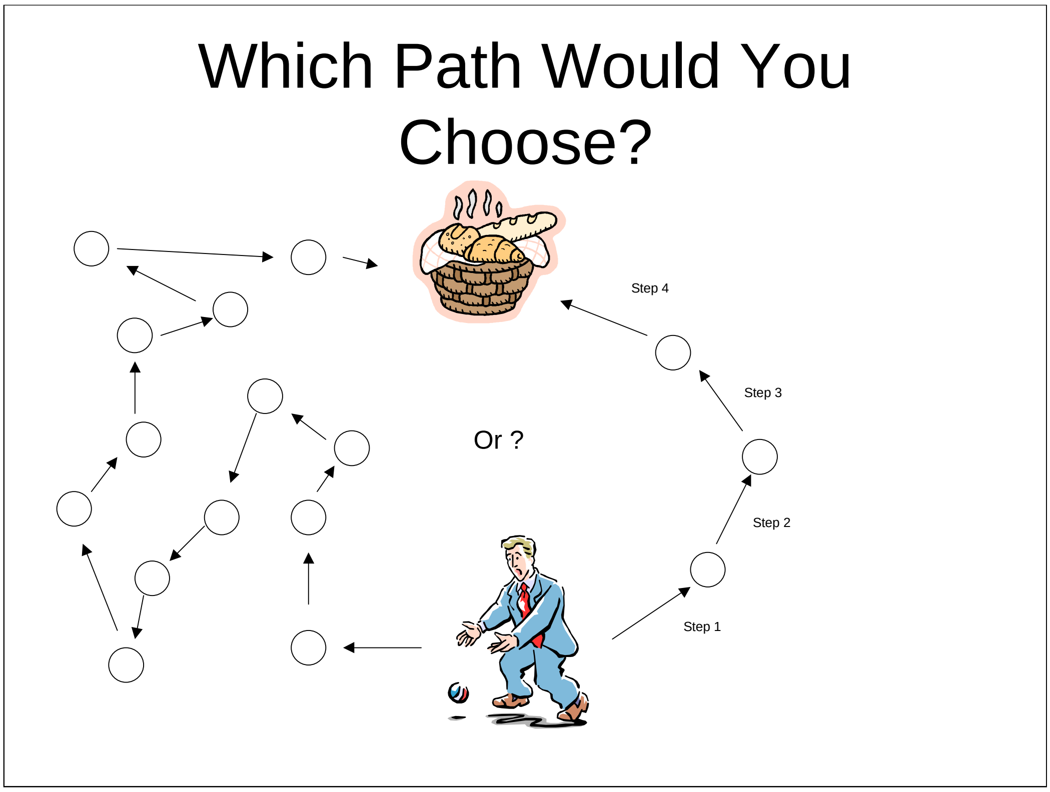
> Usability ”The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.” [ISO9241-11]

The effectiveness means the accuracy and completeness the users achieve specified goals. Efficiency measures the required resources expended to achieve the goals. Satisfaction measures the comfort and acceptability of use.

In the ISO terms, the goal-oriented approach tries to mainly to improve the usability in therms of efficiency. However, also the effectiveness contributes to efficiency, since if the user doesn’t completely archieve her goal, she might be forced to try again and thus the amount of resources increase.

The efficiency of archieving a goal can be divided into tasks and further into steps [Hätinen02b]. A task represents typically a feature in a program. It’s basically a set of steps, that are atomical user interface actions such as moving a mouse or clicking a button. Performing several steps lead to performing a task, and performing several tasks lead to archieving a goal. We later define step more closely.

Summary of terms: usability, basic goal, goal, persona, task, step, effectiveness, efficiency, satisfaction.



Which Path Would You Choose?

The point in the usability model is, however, that if you want to eat the cake, you propably choose the easiest way to get it.

Goals - Alice

Alice

Alice is a 32 year old secretary working for a work rental firm. She is positioned at a research and development department of a large telecom manufacturer. Her job includes arranging trips for the whole 113 people R&D unit with Beatrice, a regular secretary of the unit. She works closely with the head project manager. While she doesn’t understand much of the actual technology, she likes the relaxed atmosphere in the unit and the whole company and likes her work.

Basic Goals [Cooper95]

- Don’t look stupid
- Don’t make big mistakes
- Do enough work (to not get fired)
- Have fun (or at least don’t get bored)

What do people really want?

Alice is a secretary at a large telecom manufacturer.

Alan Cooper introduces few personal goals Alice has about her work.

Later I’ll show how PGP threaths most of these goals and doesn’t help any.

Alice’s Threat-model

-PGP may make Alice look stypid if she can’t use PGP

-PGP slows her down and she can’t do enough work

-Clicking bores her to death

Threats

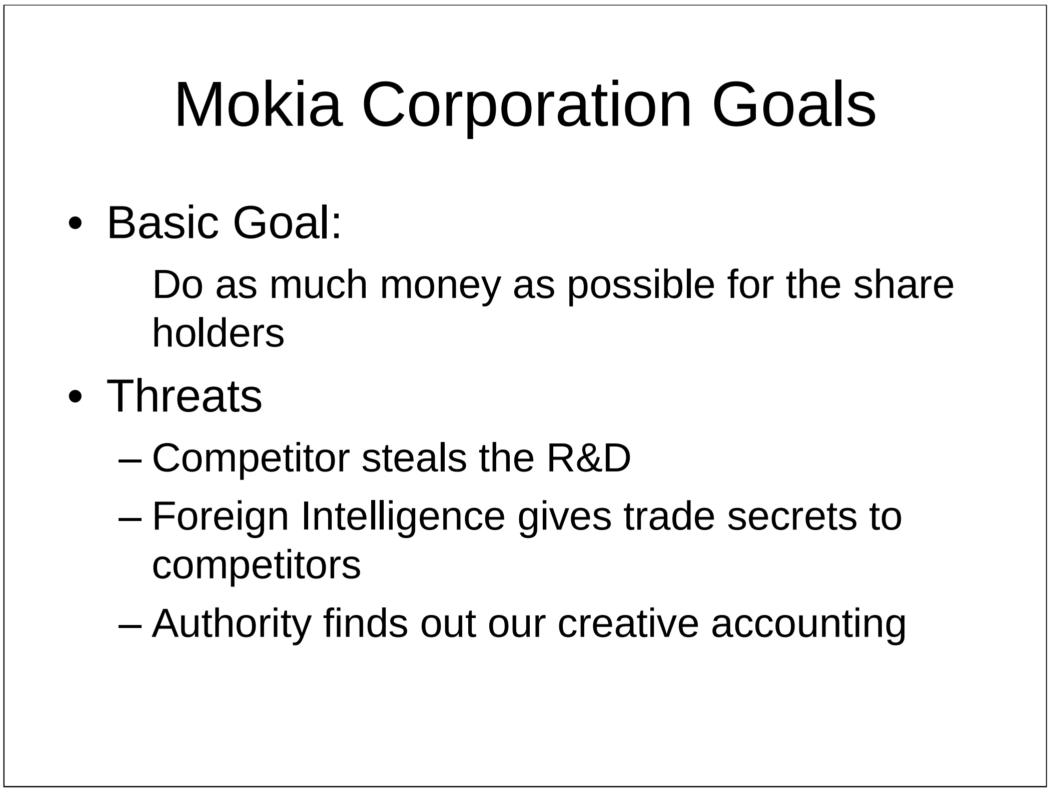
-Somebody reads your email (I couldn’t care less)

-Charles reads Alices email (well, perhaps i don’t send any personal mail)

-Beatrice reads her email exchange with her newest boy friend (never!)

-Police reads your email (Alice isn’t doing anything wrong)

-KGB reads Alices email (unlikely, doesn’t care)



Mokia Corporation Goals

- Basic Goal:

  Do as much money as possible for the share holders

- Threats
  - Competitor steals the R&D
  - Foreign Intelligence gives trade secrets to competitors
  - Authority finds out our creative accounting

Corporate Basic Goals (Alan Cooper)

-make money for shareholders

-do not give anything valuable for free

Threats

-Somebody reads your email (Well who reads the mail? Competitors? Foreign Intelligence?)

-Charles reads Alices email (Boss should be able to supervise his subjects)

-Beatrice reads her email exchange with her newest boy friend (No private mails during work hours)

-Police reads your email (Well we are cooperating with the authorities)

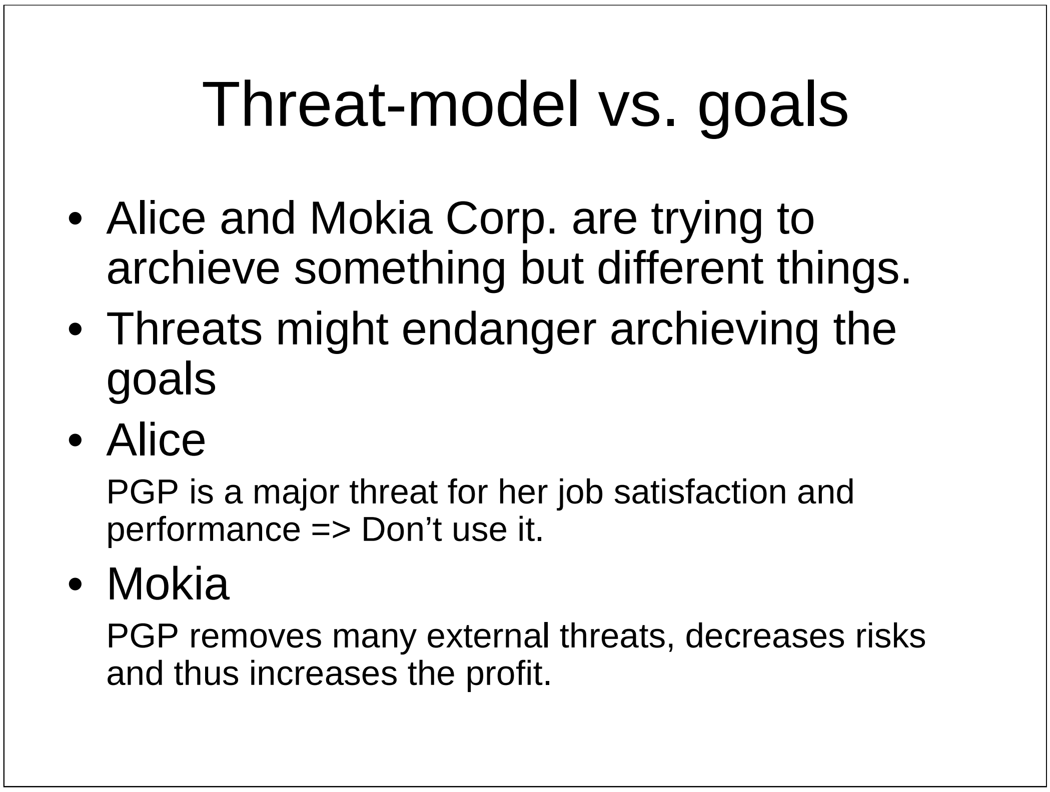
-Competing Fredriksson corporation employee reads the mail (Huge security risk)

-KGB reads Alices email (Leaks the techonology to competitors and new market entrants from Russia)

PGP:

- Licences increase costs

+ Removes many threats



Threat-model vs. goals

- Alice and Mokia Corp. are trying to archieve something but different things.
- Threats might endanger archieving the goals
- Alice

  PGP is a major threat for her job satisfaction and performance => Don’t use it.
- Mokia

  PGP removes many external threats, decreases risks and thus increases the profit.

When usability meets security, the important addition to the model is the concept of threat model. The security itself isn’t anyone’s goal. Security can be thought to be measures to prevent someone forcefully taking something that is valuable for you. Ie. in a modern society the government provides the police service to raise the level to unrightly take someone’s valuable property. The crook’s need might be for example get a new fast car (social status) and he archieves this by stealing one instead of working for years for it in a regular work. Of course this action has some risks involved, such as the police catches the crook and puts him in prison. The crook compares the seeming benefit and risk involved.

The owner’s need might the same as the crooks, but the measures to archieve the need are different. However, the crook, is a threat to the owner’s need of having the car if the crook steals it. The owner has valuable possesions the crook threat to steal.

Threat can be defined as a conflict of two instances different goals. One must take measures to highten the level of security to protect own valuable assets from the other party’s goals.

Generalizing, people have goals some other instances might threaten of archieving. Normally bad usability might be threat for goals like job satisfaction or performance. For valuable possessions the threats might include espionage or burglary. The design target of goal-oriented design is to minimize the steps required to archieve a goal. Adding threats into the picture usually adds extra steps that are taken to prevent threats from realising (as can be seen later).

Minimum number of steps to archieve a goal $S_{min.}$ Number of steps to archieve a goal with threats S_{threat}

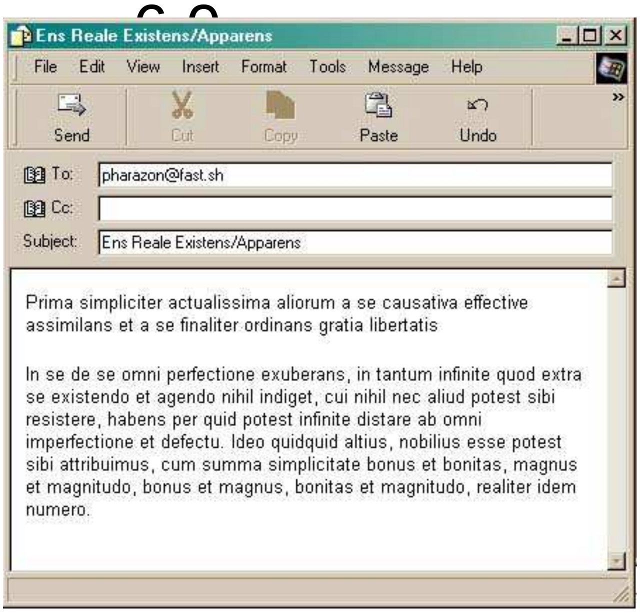
Hypothesis: S_{min} <= S_{threat}

The design target should, however, still be to minimize the required steps even if additional threats are included.

PGP 8.0 + Outlook Express 6.0

Task 1 / 4

Write the mail to David asking him out tomorrow.



To: pharazon@fast.sh

Cc:

Subject: Ens Reale Existens/Apparens

Prima simpliciter actualissima aliorum a se causativa effective assimilans et a se finaliter ordinans gratia libertatis

In se de se omni perfectione exuberans, in tantum infinite quod extra se existendo et agendo nihil indiget, cui nihil nec aliud potest sibi resistere, habens per quid potest infinite distare ab omni imperfectione et defectu. Ideo quidquid altius, nobilius esse potest sibi attribuimus, cum summa simplicitate bonus et bonitas, magnus et magnitudo, bonus et magnus, bonitas et magnitudo, realiter idem numero.

Next we study two email clients and show with screenshots how sending encrypted mail works on them.

First Outlook Express 6.0 and PGP 8.0 , then Pine 4.50 and GPG 1.2.0.

Outlook Express 6.0 and PGP 8.0

Goal

Alice wants to ask her boyfriend Fred out tonight.

Threats

Beatrice, the coworking secretary at the unit reads her personal mail and tells everybody.

Task

Write crypted email

Steps to do

- Write the mail to David asking him out tomorrow.
- Find David’s public key.
- Encrypt the mail to keep it from the boss and coworkers.
- Send the mail.

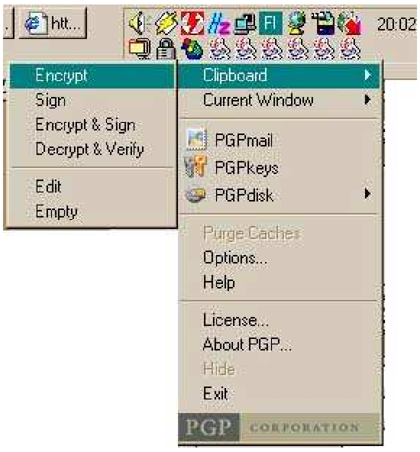
Note

Straight encryption requires registered version of PGP8.0, so we use clipboard.

PGP 8.0 + Outlook Express 6.0

Task 2 / 4

Encrypt mail



Task

Encrypt email using clipboard

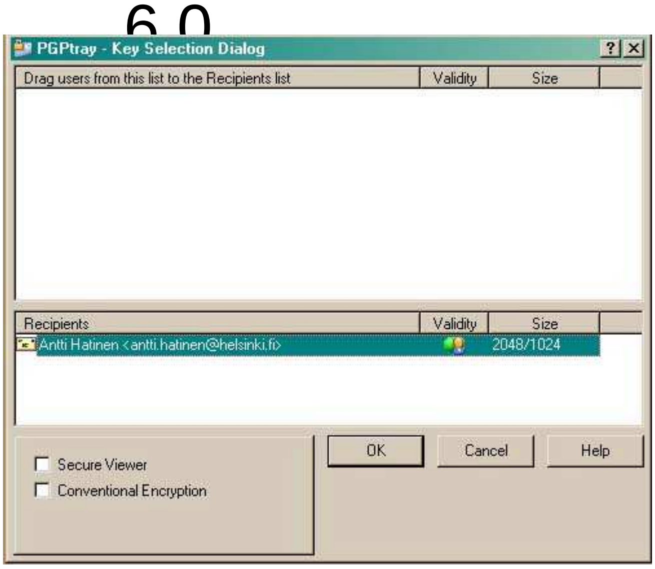
Steps

- Select the text to encrypt
- Copy (ctrl-c)
- pgp-key from icons (right click) -> clipboard -> encrypt

PGP 8.0 + Outlook Express 6.0

Task 3 / 4

Select recipients



Recipients	Validity	Size
Antti Hatinen <antti.hatinen@helsinki.fi>	●	2048/1024

□ Secure Viewer

□ Conventional Encryption

Task

Encrypt email using clipboard

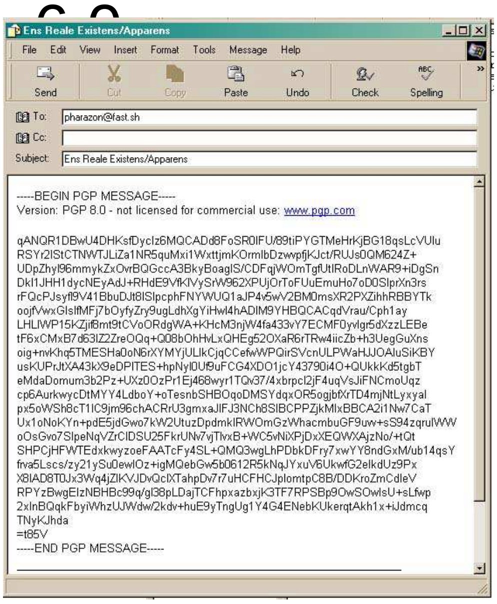
Steps

- Select the key
- Drag recipients to upper box
- Push OK

PGP 8.0 + Outlook Express 6.0

Task 4 / 4

Send mail


Ens Reale Existens/Apparens
File Edit View Insert Format Tools Message Help
Send
Cut
Copy
Paste
Undo
Check
Spelling

To: pharazon@fast.sh

Cc:

Subject: Ens Reale Existens/Apparens

```
-----BEGIN PGP MESSAGE-----
Version: PGP 8.0 - not licensed for commercial use: www.pgp.com

qANQR1DBwU4DHKsfDycIz6MQCADd8FoSR0IFU/89tiPYGTMeHrKjBG18qsLcVUlu
RSYr2IStCTNWTJLiZa1NR5quMxi1WxttjmKOrmIbDzwvpfjKJct/RUJs0QM624Z+
UDpZhyl96mmykZxOvrBQGccA3BkyBoagIS/CDFqjWOmTgfUtIRoDLnWAR9+iDgSn
Dkl1JHH1dycNEyAdJ+RHdE9VfKIVySrW962XPUjOrToFUuEmuHo7oD0SIprXn3rs
rFQcPJsyfl9V41BbuDJt8ISIpcphFNYWUQ1aJP4v5wV2BM0msXR2PXZihhRBBYTk
oojfVwxGIslfMFj7bOyfyZry9ugLdhXgYiHwl4hADIM9YHBQCACqdVrau/Cph1ay
LHLIWP15KZjif8mt9tCVoORdgWA+KHcM3njW4fa433vY7ECMF0yvlgr5dXzzLEBe
tF6xCMxB7d63IZ2ZreOQq+Q08bOhHvLxQHEg52OXaR6rTRw4iicZb+h3UegGuXns
oig+nvKhq5TMESHa0oN6rXYMYjULIkCjqCCefwWPQirSVcnULPWaHJJOAluSiKBY
usKUPrJtXA43kX9eDPITES+hpNyl0Uf9uFCG4XDO1jcY43790i4O+QUkkKd5tgbT
eMdaDomum3b2Pz+UXz0OzPr1Ej468wyr1TQv37/4xbrpcl2jF4uqVsJiFNCmoUqz
cp6AurkwycDtMYY4LdboY+oTesnbSHBOqoDMSYdqxOR5ogjbfXrTD4mjNtLyxyal
px5oWSh8cT1IC9jm96chACRrU3gmxaJIFJ3NCh8SIBCPPZjkMIxBBCA2i1Nw7CaT
Ux1oNoKYn+pdE5jdGwo7kW2UtuzDpdmkIRWOmGzWhacmbuGF9uw+sS94zqrulWW
oOsGvo7SIpeNqVZrCIDSU25FkrUNv7vjTlvxB+WC5vNiXPjDxXEQWXAjzNo/+tQt
SHPCjHFWTEdxkwyzoeFAATcFy4SL+QMQ3wgLhPDbkDFry7xwYY8ndGxM/ub14qsY
frva5Lscs/zy21ySu0ewlOz+igMQebGw5b0612R5kNqJYxuV6UkwfG2eIkdUz9Px
X8IAD8T0Jx3Wq4jZIKVJDvQcIXTahpDv7r7uHCFHCJplomtpC8B/DDKroZmCdleV
RPYzBwgEIzNBHBc99q/gI38pLDajTCFhpxazbxjK3TF7RPSBp9OwSOwIsU+sLfwp
2xInBQqkFbyiWhzUJWdw/2kdv+huE9yTngUg1Y4G4ENebKUkerqtAkh1x+iJdmcq
TNyKJhda
=t85V
-----END PGP MESSAGE-----
```

Task

Send the mail

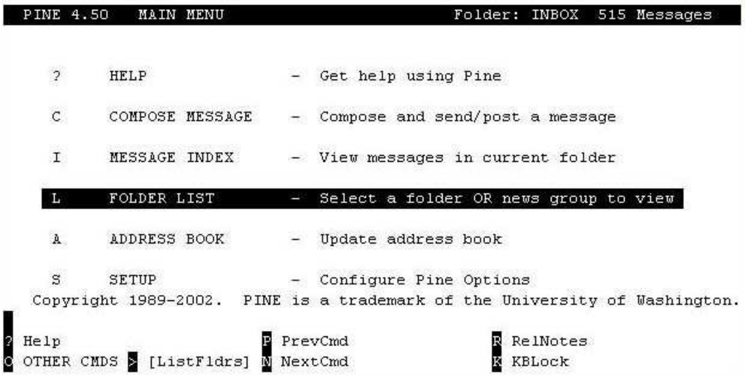
Steps

- Select the text
- Paste (ctrl-v) the crypted message over the original
- Push Send-button

GPG 1.2.0+PinePGP

1 / 3

Pine



```
  PINE 4.50    MAIN MENU                       Folder: INBOX  515 Messages


     ?     HELP               -  Get help using Pine

     C     COMPOSE MESSAGE    -  Compose and send/post a message

     I     MESSAGE INDEX      -  View messages in current folder

     L     FOLDER LIST        -  Select a folder OR news group to view

     A     ADDRESS BOOK       -  Update address book

     S     SETUP              -  Configure Pine Options
   Copyright 1989-2002.  PINE is a trademark of the University of Washington.

? Help                       P PrevCmd               R RelNotes
O OTHER CMDS > [ListFldrs]   N NextCmd               K KBLock
```

Gnu Privacy Guard 1.2.0 - open source PGP clone for unix and several other platforms.

Task

Write email

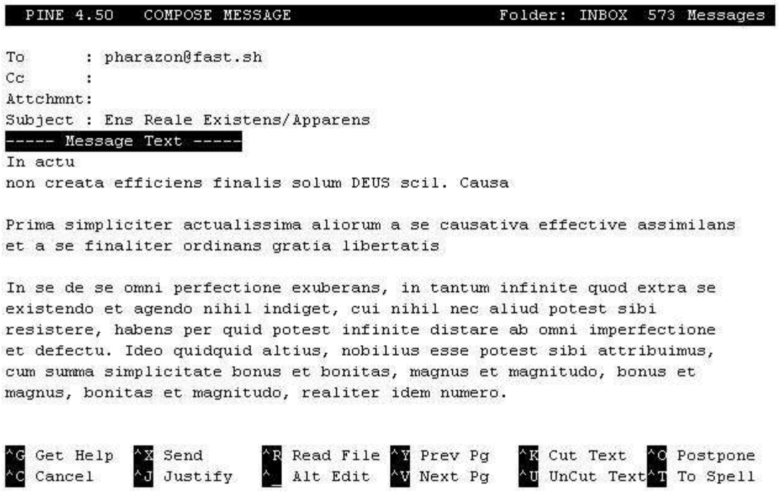
Steps

- Type pine on console
- Select Compose by pushing c

GPG 1.2.0+PinePGP

2 / 3

GPG



```
  PINE 4.50   COMPOSE MESSAGE                    Folder: INBOX  573 Messages

To      : pharazon@fast.sh
Cc      :
Attchmnt:
Subject : Ens Reale Existens/Apparens
----- Message Text -----
In actu
non creata efficiens finalis solum DEUS scil. Causa

Prima simpliciter actualissima aliorum a se causativa effective assimilans
et a se finaliter ordinans gratia libertatis

In se de se omni perfectione exuberans, in tantum infinite quod extra se
existendo et agendo nihil indiget, cui nihil nec aliud potest sibi
resistere, habens per quid potest infinite distare ab omni imperfectione
et defectu. Ideo quidquid altius, nobilius esse potest sibi attribuimus,
cum summa simplicitate bonus et bonitas, magnus et magnitudo, bonus et
magnus, bonitas et magnitudo, realiter idem numero.


^G Get Help  ^X Send      ^R Read File ^Y Prev Pg   ^K Cut Text  ^O Postpone
^C Cancel    ^J Justify   ^_ Alt Edit  ^V Next Pg   ^U UnCut Text^T To Spell
```

Task

Send email

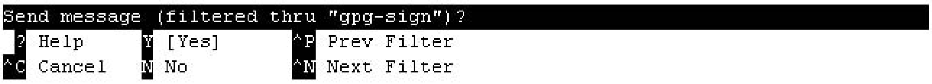
Steps:

- Write the mail
- Push ctrl-x to send

GPG 1.2.0+PinePGP

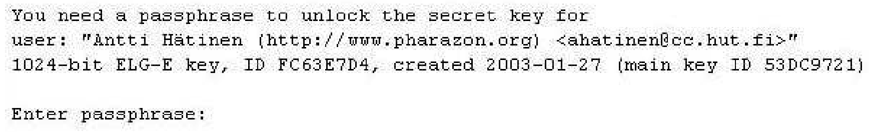
3 / 3
GPG

Filter:



```
Send message (filtered thru "gpg-sign")?
 ? Help       Y [Yes]       ^P Prev Filter
^C Cancel     N No          ^N Next Filter
```

Passphrase:



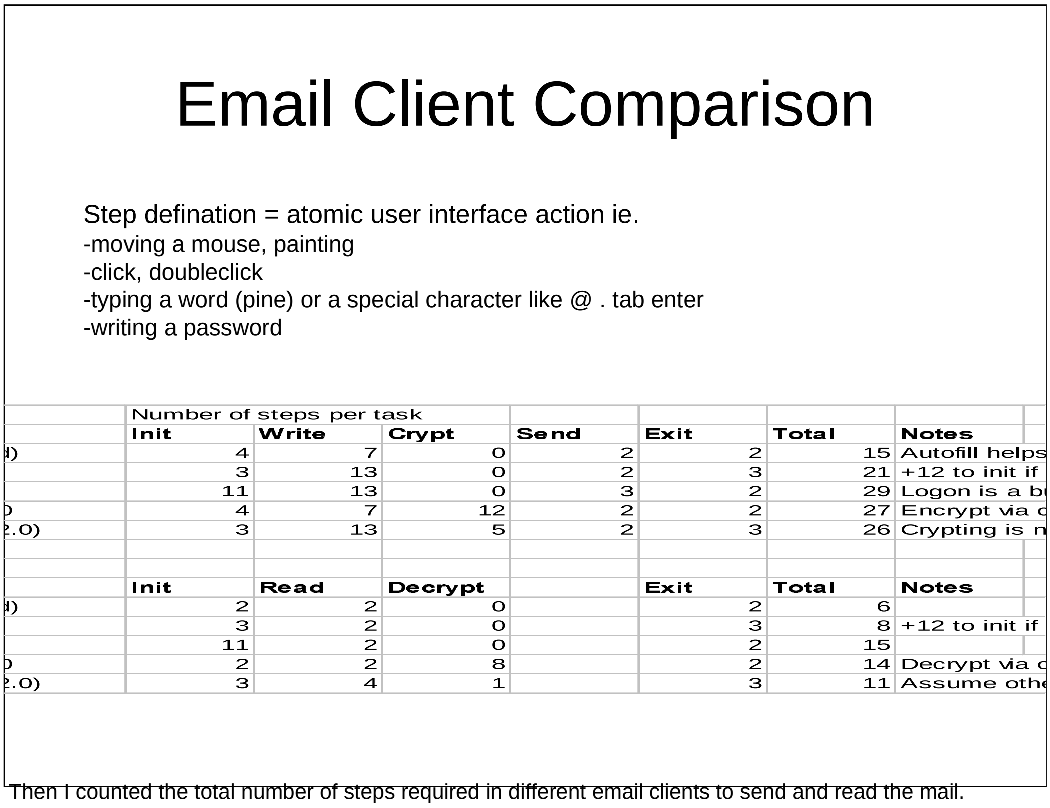
```
You need a passphrase to unlock the secret key for
user: "Antti Hätinen (http://www.pharazon.org) <ahatinen@cc.hut.fi>"
1024-bit ELG-E key, ID FC63E7D4, created 2003-01-27 (main key ID 53DC9721)

Enter passphrase:
```

Task

Encrypt the mail

Steps

- Select proper filter with ctrl-n (gpg-encrypt)
- Type passphrase



Email Client Comparison

Step defination = atomic user interface action ie.
-moving a mouse, painting
-click, doubleclick
-typing a word (pine) or a special character like @ . tab enter
-writing a password

	Number of steps per task						
	Init	Write	Crypt	Send	Exit	Total	Notes
d)	4	7	0	2	2	15	Autofill helps
	3	13	0	2	3	21	+12 to init if
	11	13	0	3	2	29	Logon is a bu
ɔ	4	7	12	2	2	27	Encrypt via c
2.0)	3	13	5	2	3	26	Crypting is n
	Init	Read	Decrypt		Exit	Total	Notes
d)	2	2	0		2	6	
	3	2	0		3	8	+12 to init if
	11	2	0		2	15	
ɔ	2	2	8		2	14	Decrypt via c
2.0)	3	4	1		3	11	Assume othe

Then I counted the total number of steps required in different email clients to send and read the mail.

More specific definition of *step* used in this measurement:

Step = atomic user interface action

Step counting principles (ie. counted as one step)

- -moving a mouse, painting
- -click, doubleclick
- -typing a word (ahatinen), a special character like @ or . , tab or enter
- -writing a password

Assume

The message sent in all measurements had subject = test, payload = test .

keys are pre-stored.

Only encrypt, don’t sign.

Tasks

Init: Starting the email client

Write: Writing the mail (receiver, subject, payload)

Crypt: Encypting the email

Sending: Send the mail

Exit: Changing to other program

Outlook Express 6.0 (uncrypted)

Notes: autofill in address-field helps a lot versus pine when writing ahatinen@cc.hut.fi = 7 steps, but with autofill only 1.

Pine 4.50 (uncrypted)

Notes: start from console assuming text UI. If used with F-secure SSH 5.52 via kosh.hut.fi add extra 11 steps for init and 1 for exit.

Webmail (uncrypted)

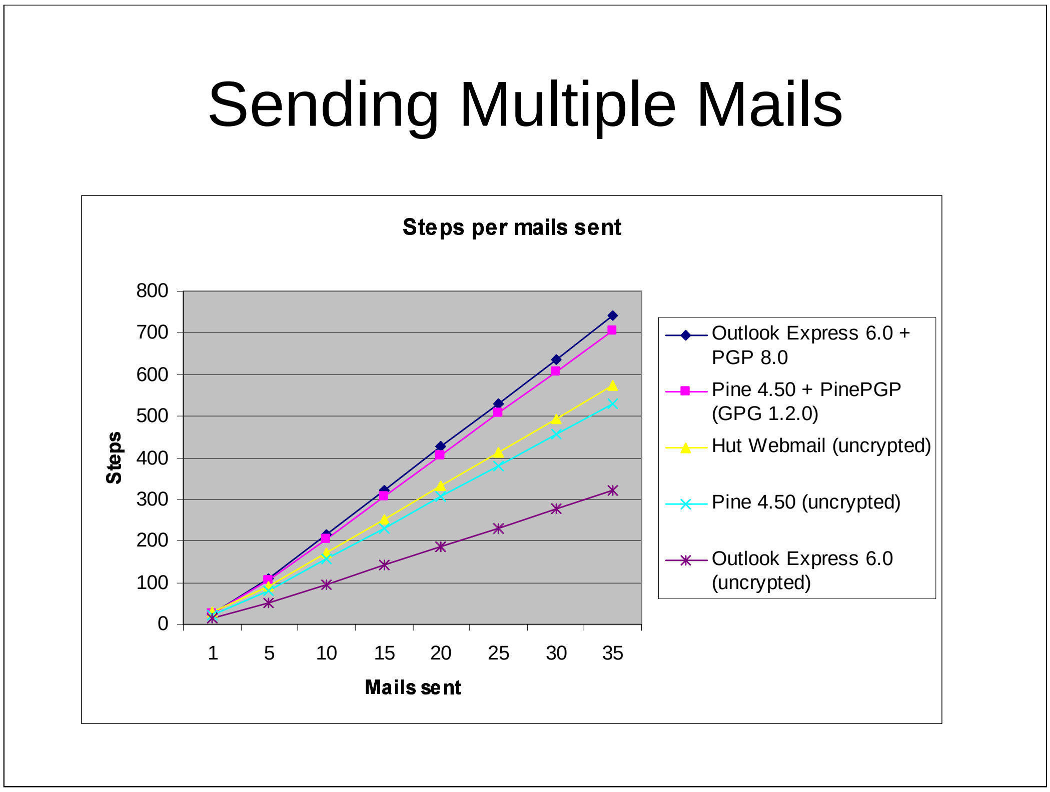
Notes: requiring to login is a initial burden. No autofill.

Outlook Express 6.0 + PGP 8.0 (crypted)

Notes: didn’t ask for passphrase and receiver was the only key stored in PGP.

Pine 4.50 + PinePGP (crypted)

Notes: Pine encryption is actually very straightforward. Later I found out alternative init+write –path of 10+5 steps == 1 shorter than 3+13.



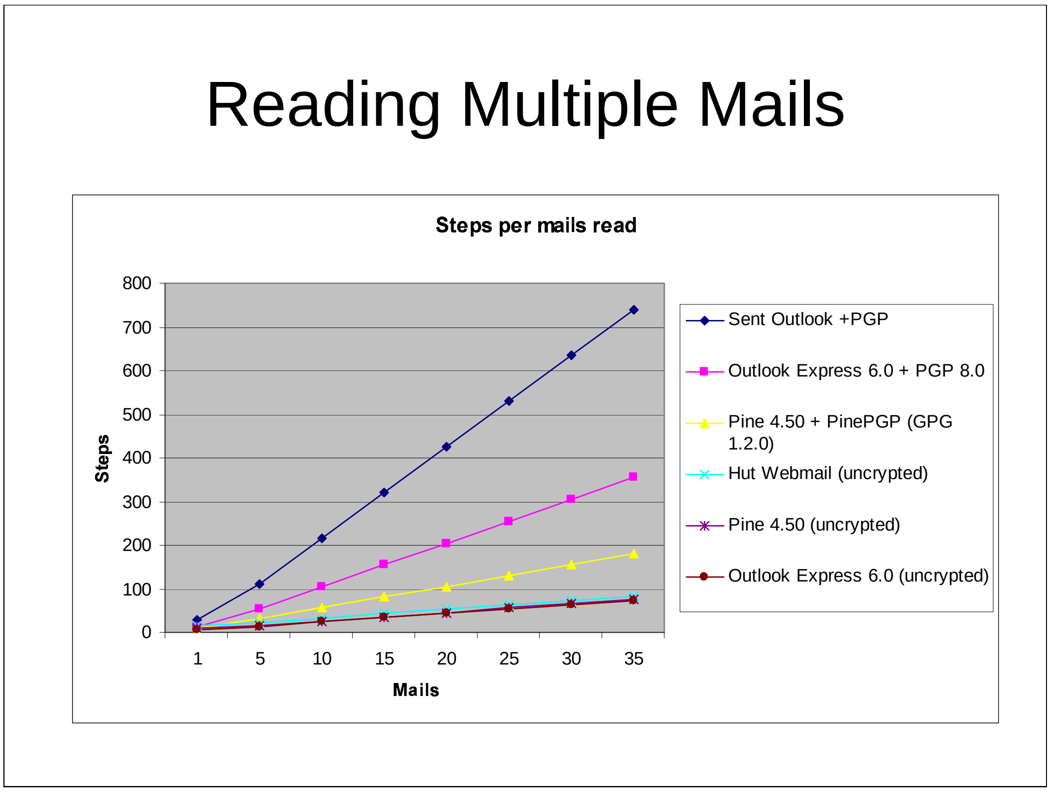
Sending Multiple Mails

Then how about sending multiple emails with different clients?

The difference between uncrypted Outlook and PGP crypted while sending 35 mails is 420 extra steps.

On pine 175 extra steps.

Note the big difference between Outlook vs. Pine&Webmail

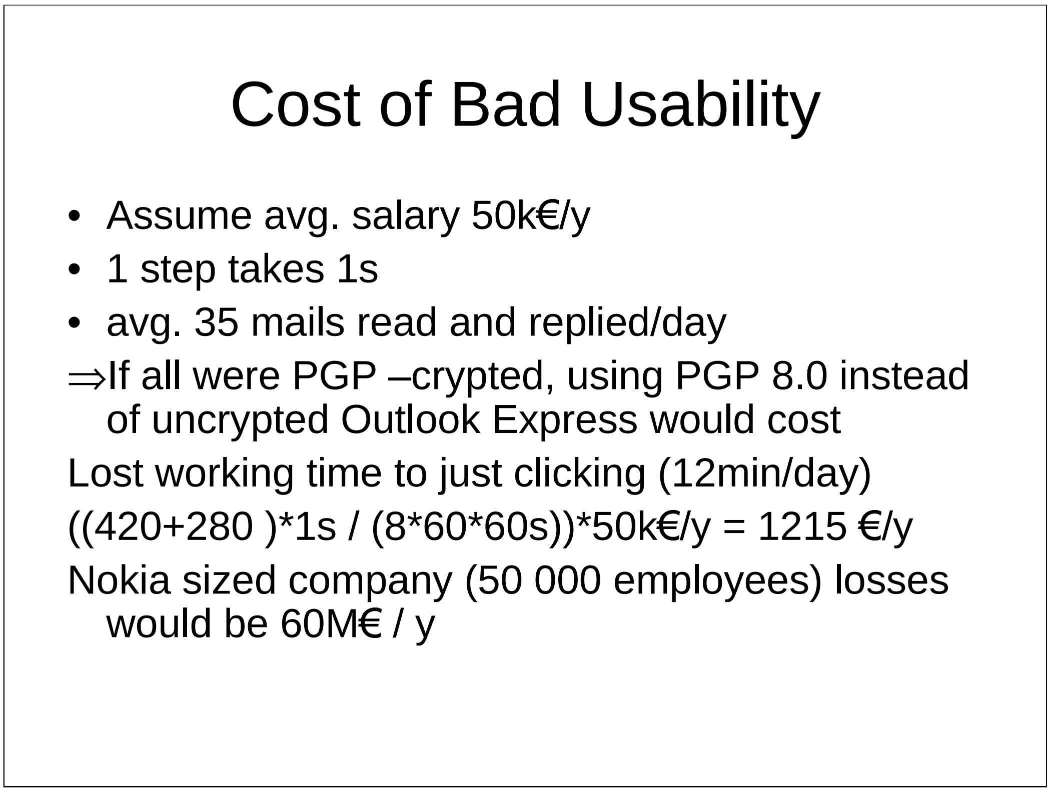


Reading Multiple Mails

Difference 280 mails between crypted and uncrypted Outlook.
On pine 105 (~100) extra steps.

Highest line is from previous picture

Note all uncrypted clients are about the same



Cost of Bad Usability

- Assume avg. salary 50k€/y
- 1 step takes 1s
- avg. 35 mails read and replied/day

⇒If all were PGP –crypted, using PGP 8.0 instead of uncrypted Outlook Express would cost

Lost working time to just clicking (12min/day)

((420+280)*1s / (8*60*60s))*50k€/y = 1215 €/y

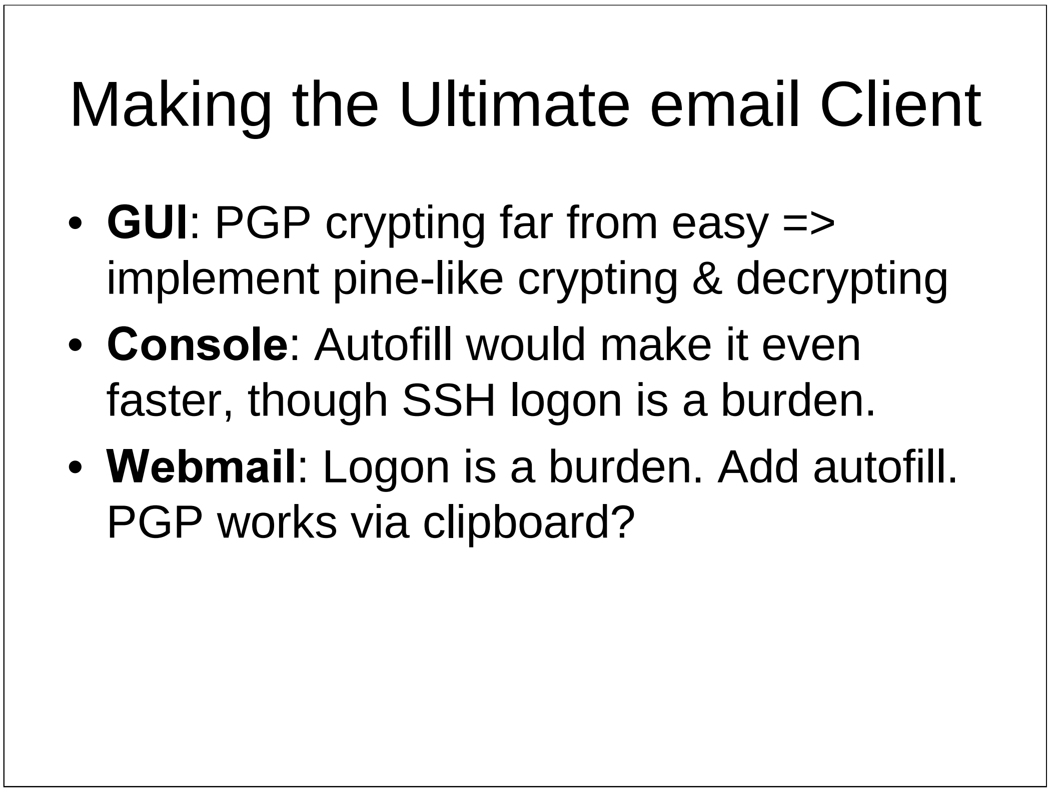
Nokia sized company (50 000 employees) losses would be 60M€ / y

Cost of Bad Usability

The number of extra steps done to archieve the same goal using different tools isn’t unrelevant. Goal-oriented usability design can be concidered as a tayloristic approach since it is very interested also in studying and optimizing the atomic user interface actions, ie. how the user turns the monkey wrench most efficiently.

If we make an assumption that one step takes 1s to complete (Webmail might take more due to network delays, and in GUI some steps might be faster), the cost of a user for a company is 50 000€ / year (includes the side costs) and an avarage of 35 mails read and replied per day, using encrypted mail the user spends every day 12 minutes just clicking and moving the mouse. The lost time is comparison between using Outlook Express with and without PGP. On Pine and GPG the difference isn’t so dramatical, but still exists. How would you like if your every day would start with extra 12 mins of cpushing the button?

The only cost isn’t that all users are bored to death, but the same time is wasted from performing some other more useful activity. The cost of wasting 12 minutes per day is 1215 €/y for the company. For a company of 50 000 employees the cost of wasted time of using PGP 8 would be 60M€/y alone. Cryptographic software is rarely free, so the licence and training costs have to be added to the sum. These kind of figures offer an opportunity for development of pgp-type software into more user friendly versions with lesser steps.



Making the Ultimate email Client

- **GUI**: PGP crypting far from easy => implement pine-like crypting & decrypting
- **Console**: Autofill would make it even faster, though SSH logon is a burden.
- **Webmail**: Logon is a burden. Add autofill. PGP works via clipboard?

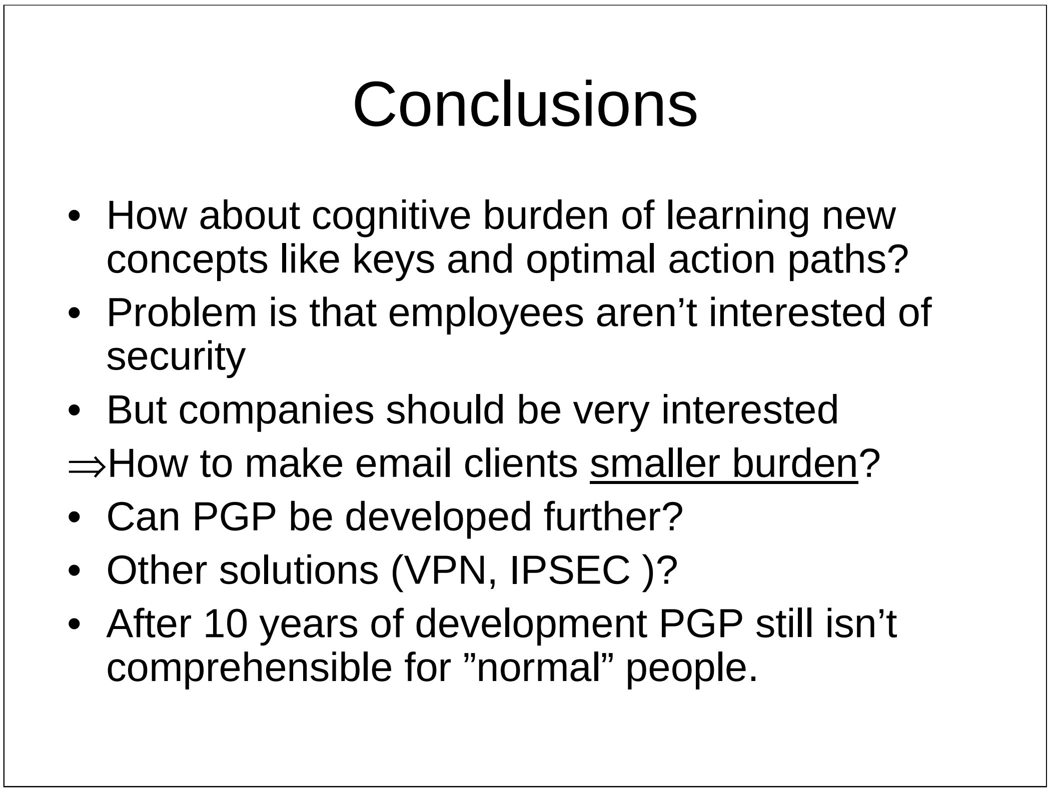
Design Implications for different platforms.

Outlook was the best email client in the comparison with least number of steps while sending email uncrypted. However, it’s encryption and decryption properties with PGP8.0 were far from good and in this area there is much to do to lower the number of steps required to even the same level as pine + gpg has.

While pine is a quite ancient compared to newest GUI clients, it does it’s job well. The only minus today is that the client has to be used over SSH-connections, which make the starting of the client very cumbersome. Also, sending mail would benefit from Outlook-like autofill, even though there is some sort of autocompletion for local addresses. Encryption and decryption is the area where pine and gpg shines. GUI PGP has a long way to archieve as low number of steps as PinePGP offers. However, one could still automate crypting more, for example by automatically decrypting all crypted mails.

Webmail is nowadays perhaps the most popular way of reading mail while not using the home computer. However, the logon process is a burden and many good features present in Outlook and Pine, such as autofill and easy encryption, are missing. PGP works only through the clipboard, which makes the encyption and decryption hard even if the keys would be managed with ease (which might not be the case in most of the situations).

Conclusion is that there are a lot of small improvements every client could benefit from.



Conclusions

- How about cognitive burden of learning new concepts like keys and optimal action paths?
- Problem is that employees aren’t interested of security
- But companies should be very interested

⇒How to make email clients <u>smaller burden</u>?

- Can PGP be developed further?
- Other solutions (VPN, IPSEC)?
- After 10 years of development PGP still isn’t comprehensible for ”normal” people.

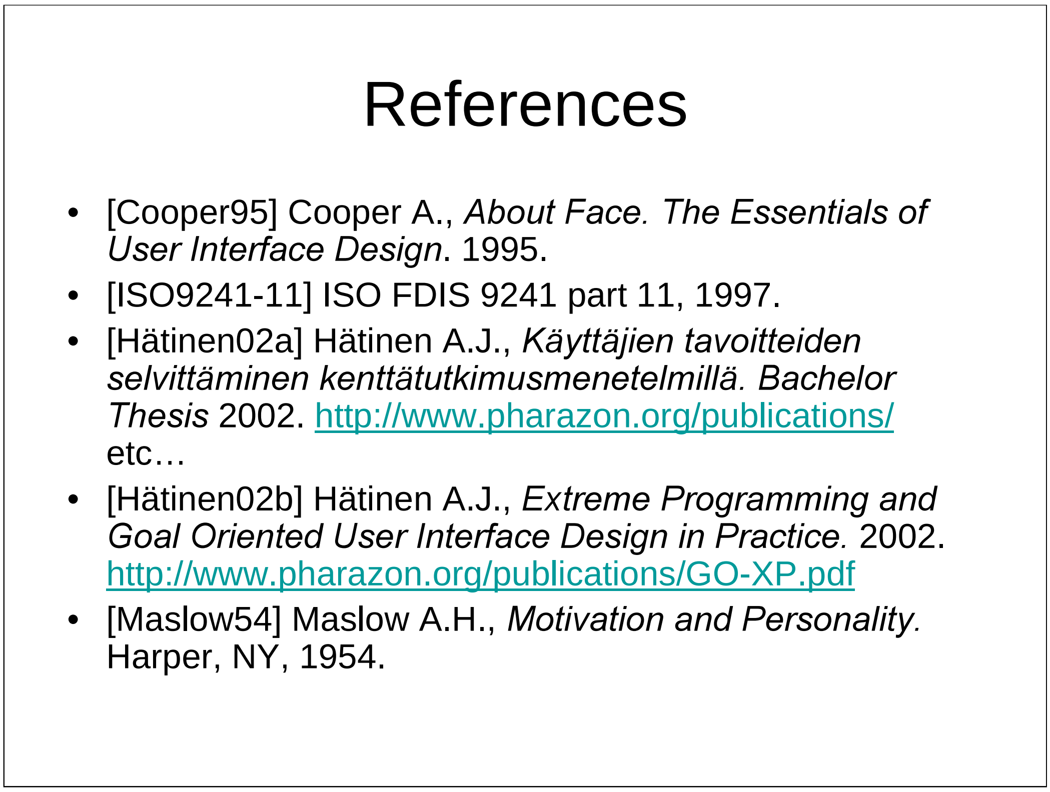
Critique of the study

While counting the steps of some tasks might help to measure the efficiency of different email clients, many other important things have been exqluded from the consideration. For example the key management increases significantly the burden to use all pgp-software in comparison to uncrypted mail. Also the cognitive load required to learn new concepts and usage patterns isn’t measured, but it’s likely to even further hinder the usage of PGP. Personally I made numerous mistakes before I managed to get the screenshots and the measurements, so the efficiency might also cause significiant number of extra steps.

The main problem of PGP is that the company threat model doesn’t concern it’s employees. While using uncrypted email there normally aren’t any personal goals at risk. However, for the firms the threats are real. How can the company line the employee threat models with it’s own threats?

The implication might be that the management should link somehow the firms threaths to employees personal threaths for example by educating the employees of the risks that face the company and thus their jobs. Scaring people that they might lose their jobs, might be a good start to increase overall security.

Another solution could be to improve the technology by tuning the usability of email clients or using totally new solutions like VPN or IPSec, so that the usage would become transparent for the end user. However, it’s clear that even after 10 years of development PGP isn’t comprehensible for normal people.



References

- [Cooper95] Cooper A., *About Face. The Essentials of User Interface Design*. 1995.
- [ISO9241-11] ISO FDIS 9241 part 11, 1997.
- [Hätinen02a] Hätinen A.J., *Käyttäjien tavoitteiden selvittäminen kenttätutkimusmenetelmillä. Bachelor Thesis* 2002. http://www.pharazon.org/publications/ etc…
- [Hätinen02b] Hätinen A.J., *Extreme Programming and Goal Oriented User Interface Design in Practice.* 2002. http://www.pharazon.org/publications/GO-XP.pdf
- [Maslow54] Maslow A.H., *Motivation and Personality.* Harper, NY, 1954.