Basic Cryptography and Number Theory

Andrei Sabelfeld
andrei AT cs.chalmers.se

Acknowledgement: Peter Gemmell

November 2007

What is cryptography?

- Generalized methods to hide (encrypt) and authenticate information

- Goal: secure communication in presence of malicious adversary

An Important Distinction

Encryption = maintaining information secret/confidential

Authentication = proving and maintaining information integrity

Cryptographic work can be at different levels

-- algorithms/primitives: e.g. encryption algorithms, signature algorithms, hash algorithms

-- protocols between more than 1 party: e.g. threshold sharing

-- systems: e.g. electronic cash systems, smartcard systems

-- “Attacks” - on all the above

Some applications of cryptography

- network, operating systems security
- private internet, telephone communications
- electronic payments
- database security
- software protection
- pay television
- confidential, authentic military communications

Open system design

vs

closed system design

Open design: the algorithm, protocol, or system design may be public information. The only secret will be the private or symmetric key(s)

Closed design: as much information as possible is kept secret

Types of Security

-- unconditional or "information theoretic": the security is provable free of assumptions

-- reducible or "provable": one can prove that the security is as valid as some common unproven assumption

-- ad hoc: the security seems good

Types of algorithms

Symmetric (encryption)

Alice

k

sender

Bob

k

receiver

encryption

M $\longrightarrow$ ciphertext $\longrightarrow$ ciphertext $\longrightarrow$ M

$\mathbf{Enc}_k$

decryption

$\mathbf{Dec}_k$

Types of algorithms

Symmetric (authentication)

Alice

k

sender

authentication

Bob

k

receiver

verification

$M \longrightarrow M, \text{Auth}_k(M) \longrightarrow M, \text{Auth}_k(M) \longrightarrow$ "OK"

Auth_k **Verify_k**

Types of algorithms

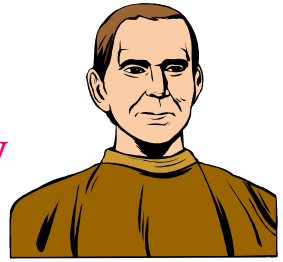
Public Key (assymmetric encryption)

Alice

pubkey

sender

Bob



privkey

receiver

encryption

M ⟶ ciphertext ⟶ ciphertext ⟶ M

$\mathbf{Enc}_{\mathbf{pubkey}}$

decryption

$\mathbf{Dec}_{\mathbf{privkey}}$

Types of algorithms

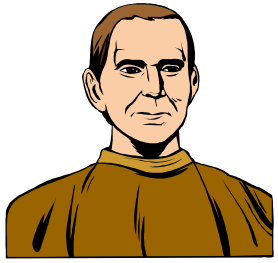
Public Key (assymmetric authentication)

Alice

privkey 

sender

Bob

 pubkey

receiver

authentication

verification

$M \longrightarrow M, \text{Sign}_{privkey}(M) \longrightarrow M, \text{Sign}_{privkey}(M) \longrightarrow$ "OK"

Sign$_{\textbf{privkey}}$

Verify$_{\textbf{pubkey}}$

Digital Signatures

A public key technique to authenticate information in a *non-repudiable* way

may be legally binding

Recipient knows:
a) that the message is that of the supposed sender
b) can prove (a) to a third party

Why Public Key is so important

It lessens the number of keys needed in a general purpose virtual private network (VPN)

Less reliance on a “trusted center” for system availability and secrecy (e.g. electronic cash)

Non-repudiation

Math Background
Modular (clock) Arithmetic

A mod N = remainder of A divided by N

$B \equiv A \bmod N$ if $B \bmod N = A \bmod N$

$A \bmod N + B \bmod N \equiv A + B \bmod N$

$A \bmod N * B \bmod N \equiv A*B \bmod N$

Encryption Algorithms Historical and Toy

Caesar Cipher: {a, b, c … z} $\longleftrightarrow$ {1, 2, 3, … 26}

Enc(X) = Enc(x1 … xN)
= x1 + 3 mod 26 … xN + 3 mod 26 = C1 … CN

E.G. Enc("Security") = Enc(19,5,3,21,18,9,20,25)
= 22,8,6,24,21,12,23,2 = "Vhfxulyb"

Generalizations of Caesar Cipher

(all weak security)

Shift: $Enc_k(x) = x+k \bmod 26$

Affine: $Enc_{k1,k2}(x) = k1 *x + k2 \bmod 26$

Substitution: $Enc_{perm}(x) = perm(x)$

Other Historial Ciphers

-- WWII American use of Navajo

-- WWII German Enigma machine

-- WWII Japanese Purple Machine

D. Kahn. The Codebreakers. Macmillan Co., New York, 1967.

Unconditionally Secure Cipher

One-time pad

key = random* bits = 1100010011100100011…
message = bits = 1110011001100110001…

cipher text = *XOR*
of key, message = 00100010100000010010...

Problems: number of random bits = length of all messages being encrypted (not reusable), random bits must be known to both sender and recipient.

Data Encryption Standard (DES)
(symmetric key)

$Enc_k(M)$ =
wild permutation, XOR's of M, S-boxes, and k

16 "rounds," 64-bit block input and output
not clean and concise (like RSA and one-time pad)

Standard for encryption of unclassified data since 1977

56 bits (40 bits in exported versions) yield valid
concerns about vulnerability to "exhaustive key search"

Extensions to DES

that may give a longer effective key

Triple-DES: ciphertext =

$EncDES_{k1}(EncDES_{k2}(EncDES_{k3}(\text{ plaintext})))$

$EncDES_{k1}(DecDES_{k1 \text{ xor } k2}(EncDES_{k2}(\text{ plaintext})))$

DESX: ciphertext = k1 xor $EncDES_{k1}$(message xor k3)

RC5 by Ron Rivest of RSA
(symmetric encryption algorithm)

Rotations, XOR's, modular additions

With variable key lengths and being more succinct than DES, it is faster and may inspire more confidence

RSA

(public key encryption*)

- Public key: (N,e)
- Private key: (p,q,d): p,q large primes;
  $N = pq$; d : $(m^e)^d = m \bmod N$
- The key d is computed from p,q and e by the extended Euclidean algorithm
- The equality above holds by Euler's theorem

*RSA encryption can be modified easily to work as the RSA signature function

RSA

(public key encryption, continued)

- Express message M as a number between
  1 and N *
- Compute $\text{EncRSA}_{N,e}(M) = M^e \bmod N$
- Compute $\text{DecRSA}_{p,q,d}(M^e) = (M^e)^d = M \bmod N$
- Assumed hard:
  factoring,
  discrete logs modulo N

RSA Security

- How can we be sure that key generation for RSA is secure: given public key (N,e) is it feasible to find private key: (p,q,d)?
- Depends wholly on the problem of factoring large numbers, for which there is no known polynomial algorithm

Hash functions

H

- compress length of message

  1M bits (any number) $\longrightarrow$ 128 or 160 bits

- "collision-free"

  for any x one can not compute $y \neq x$, $\mathbf{H}(x) = \mathbf{H}(y)$

- "random-like" behavior

E.G. SHA-1, MD5, MD2

Authentication
unconditionally secure

Alice

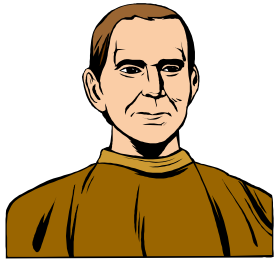
a,b

sender

authentication

Bob

a,b



receiver

verification

M → M, aM+b ⟶ M, aM+b → "OK"

a,b can be used only once

Authentication

Keyed hash function (symmetric)

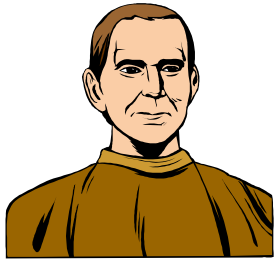
Alice

k

sender

authentication

Bob



k

receiver

verification

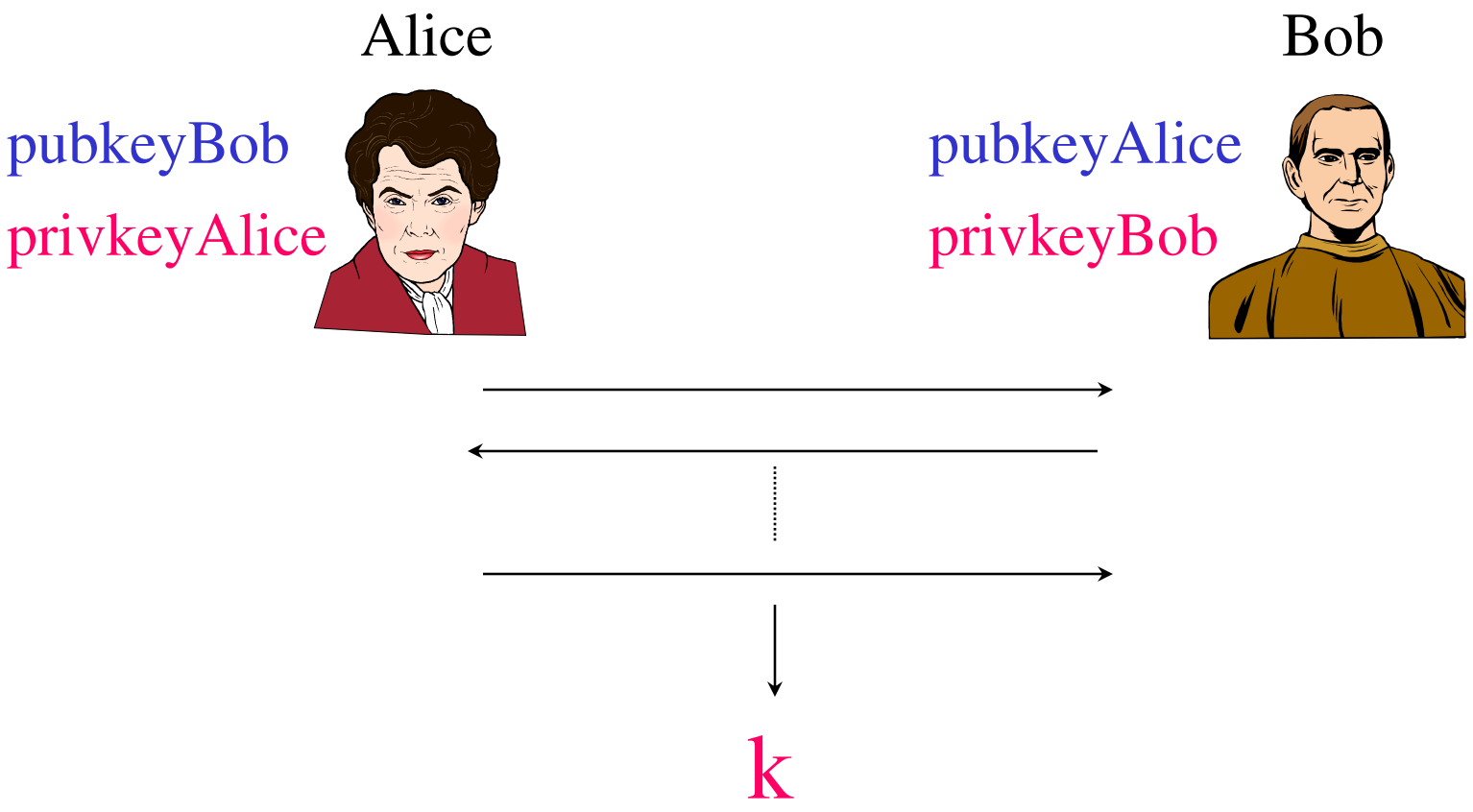
M → M, H(k,M) ⟶ M, H(k,M) →“OK”

Efficiency considerations and the need for symmetric session keys

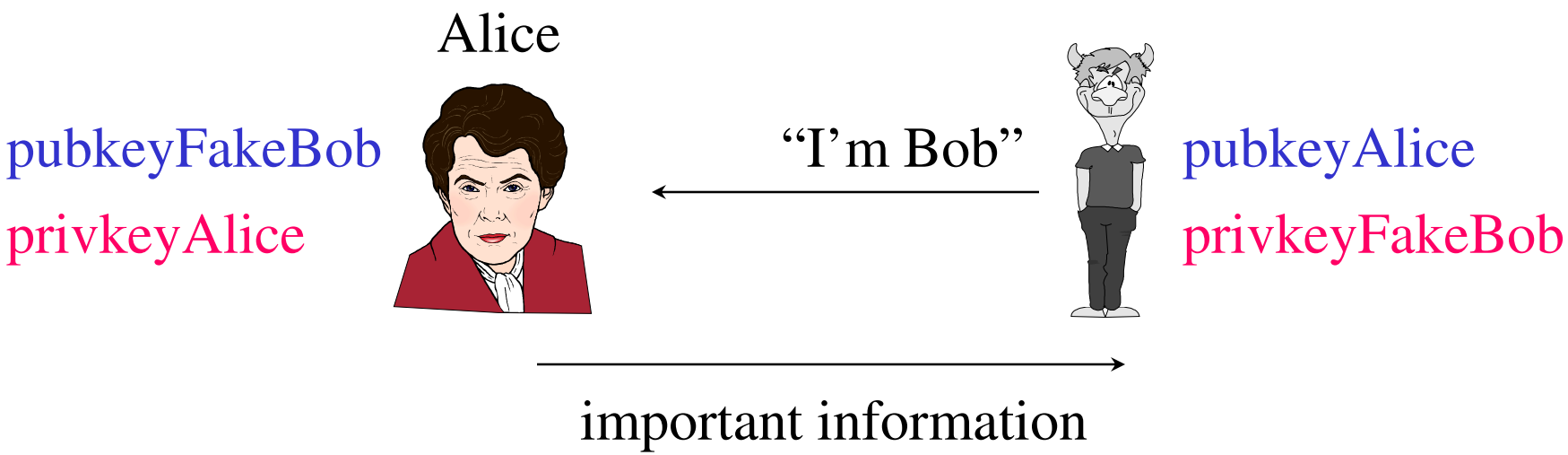
public key algorithms tend to be slow.

E.G. RSA decrypts at the order of 10k bits/second
DES encrypts/decrypts at the order of 1Mbit/second

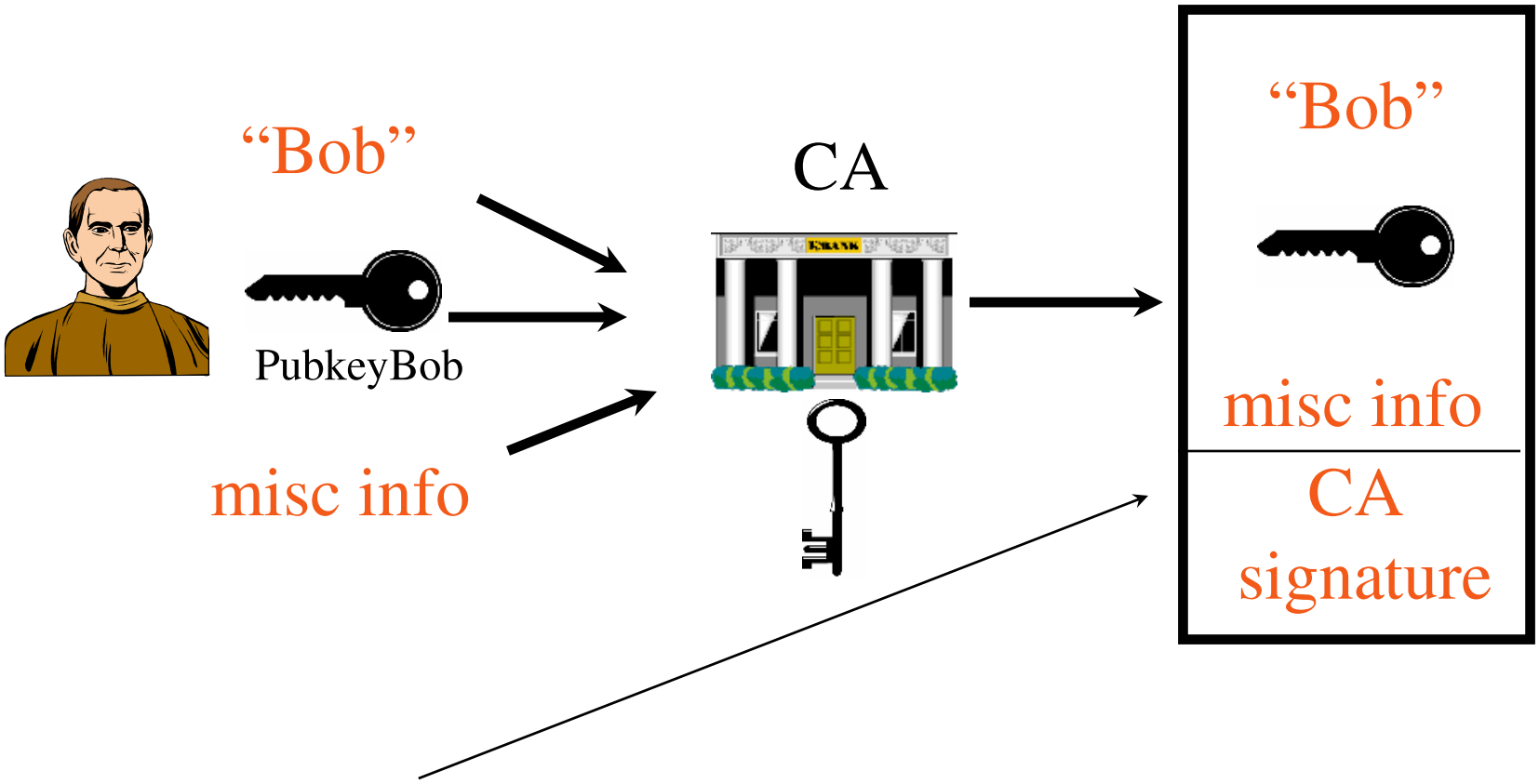
Key Exchange: Establishing a (symmetric) Session Key k



Impersonation Attack

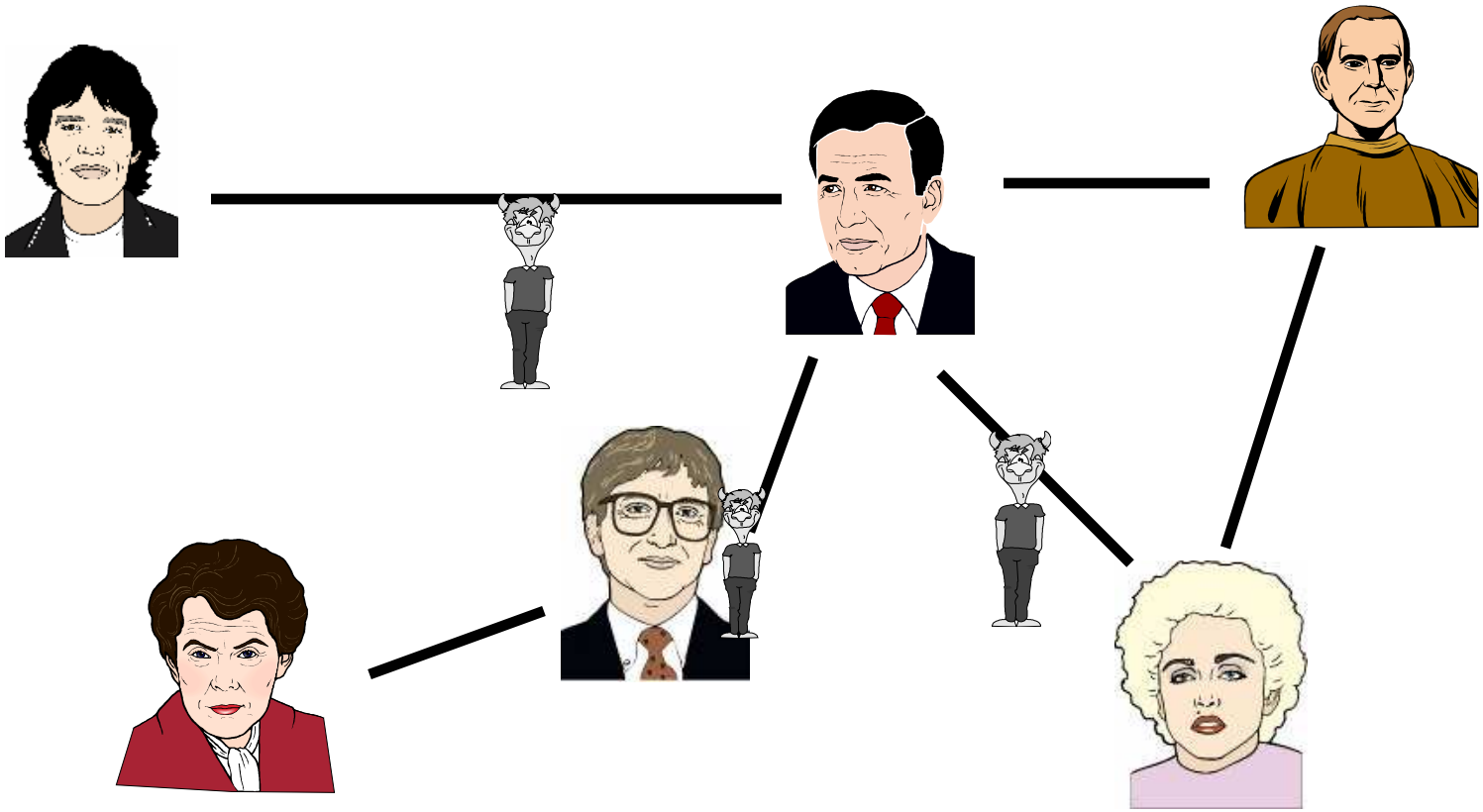


Certification Authority (CA)

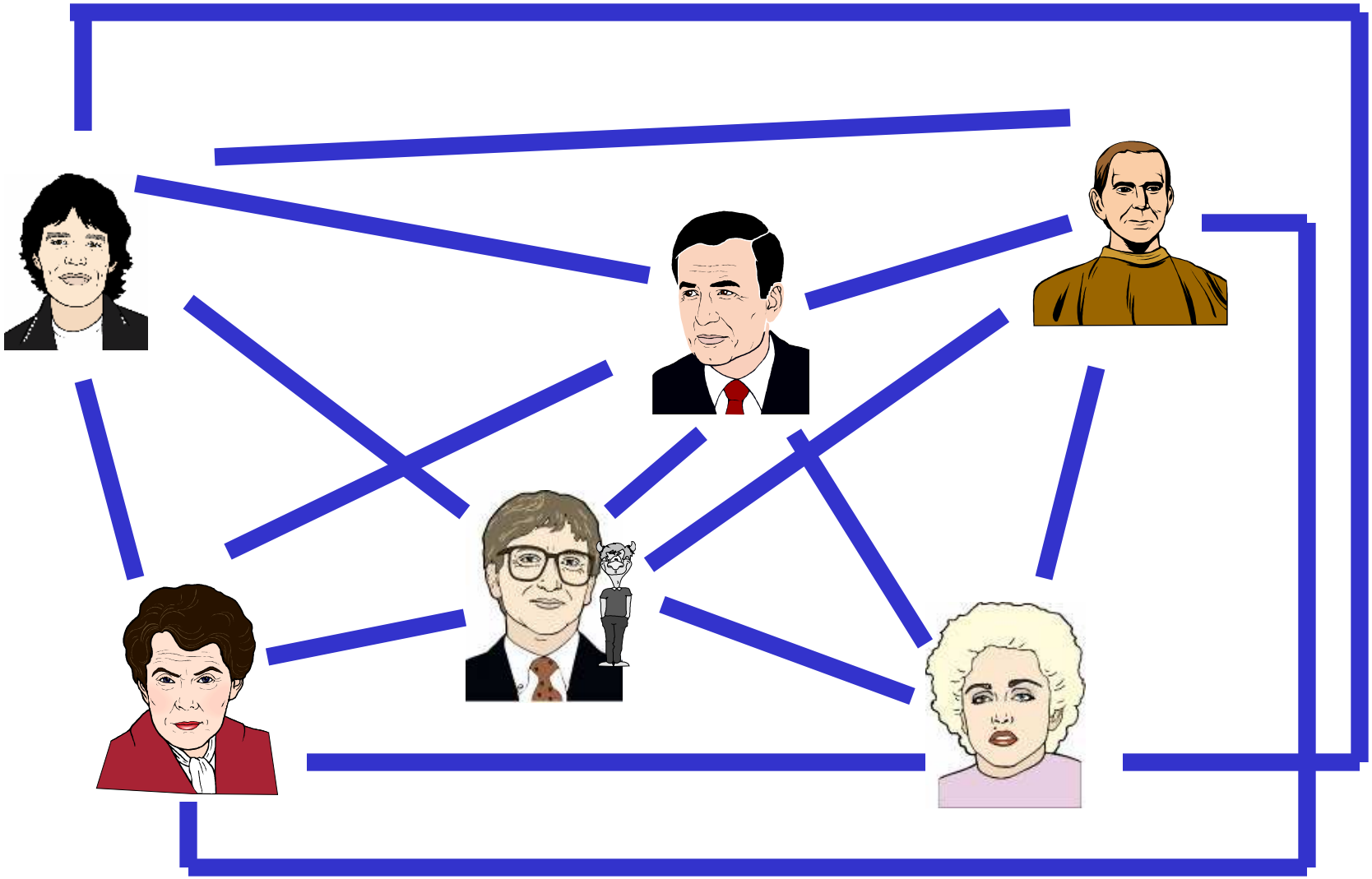


certificate binds a name to a public key

Open Network



Virtual Private Networks (VPN)

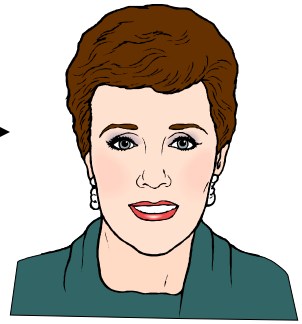


Electronic Payment Systems
"credit card," "debit card" style

Visa, bank, or other

"You have received payment"

Internet payer

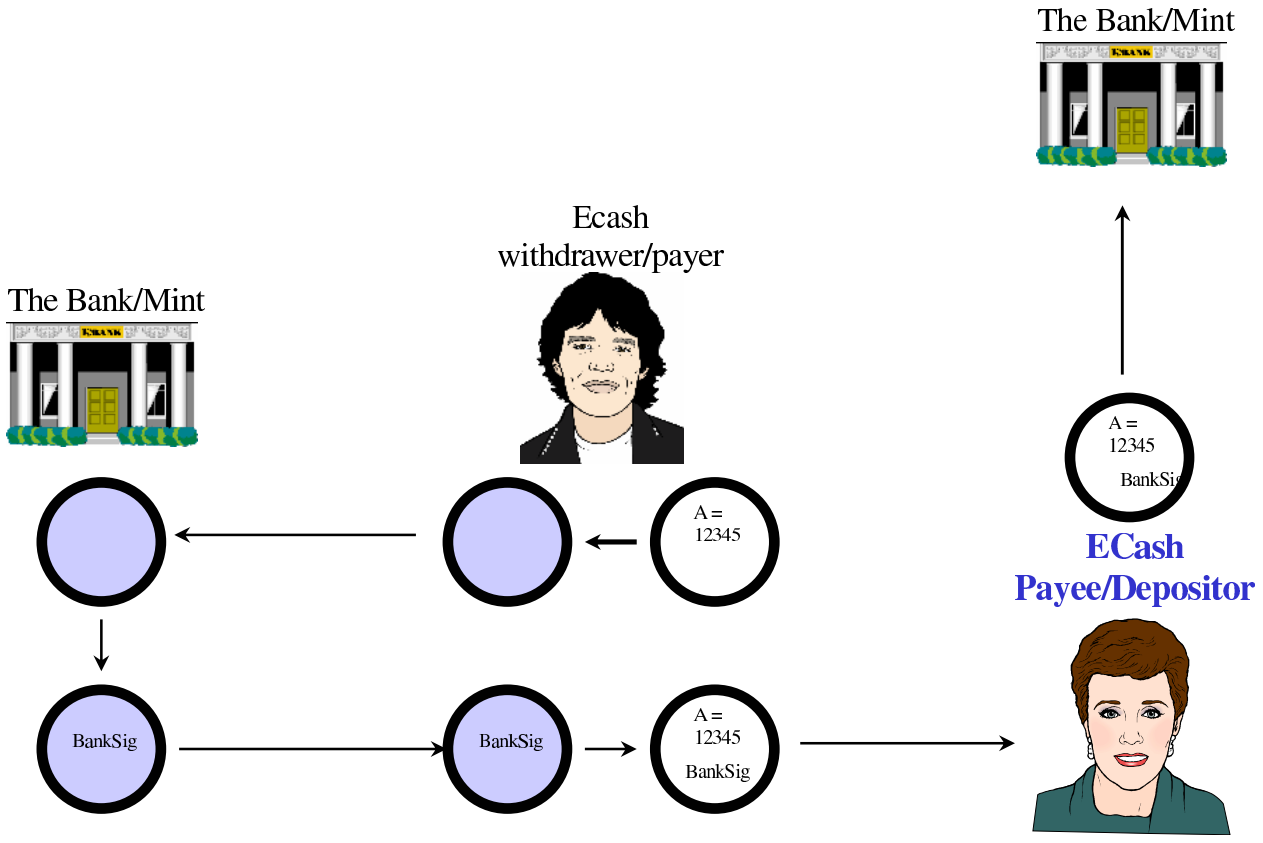


Internet payee

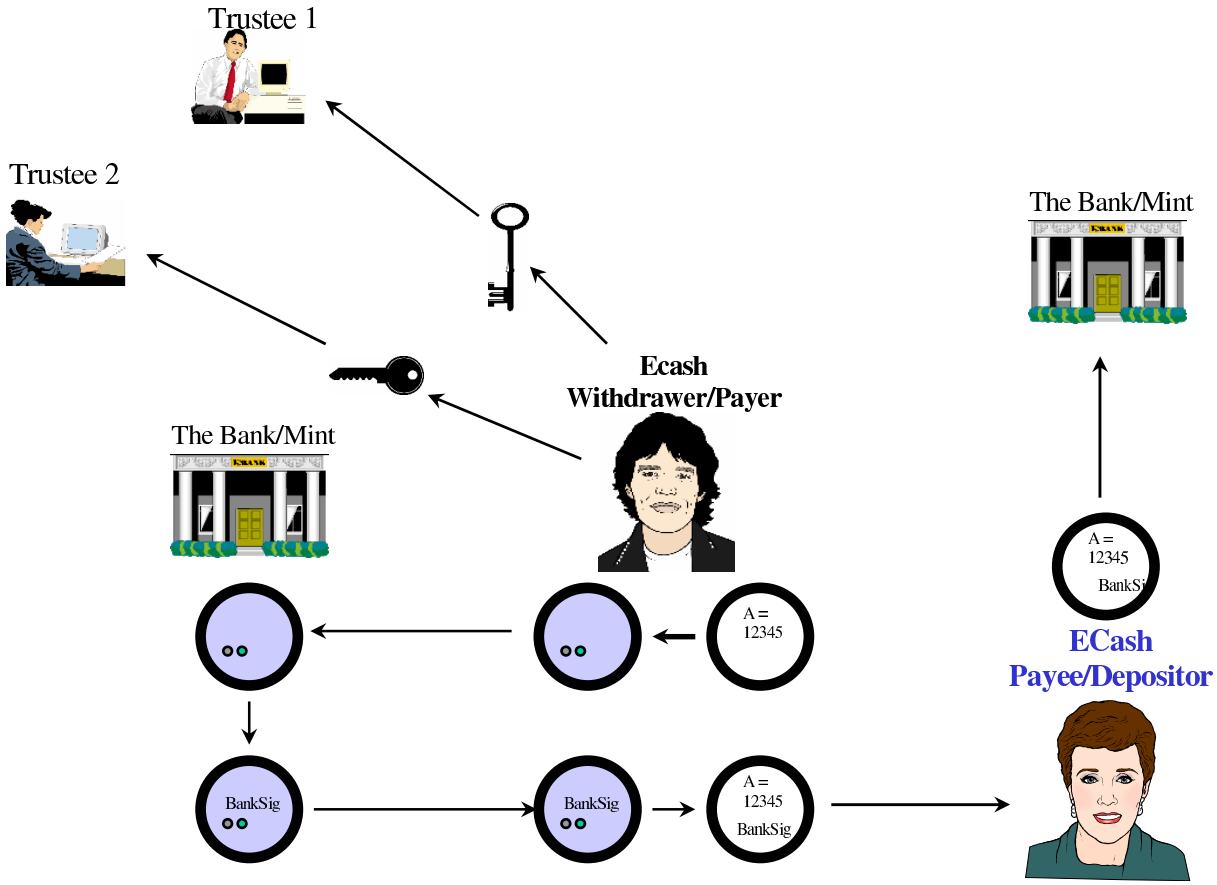
Sample Payment Protocols

- iKP of IBM
- SEPP: IBM, Netscape,GTE,CyberCash, and MasterCard
- VISA's design
- First Virtual
- Secure Courier, STT

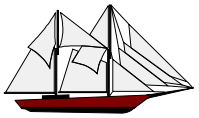
Electronic Payment Systems
Chaum-style untraceable Cash



Electronic Payment Systems
Trustee-traceable Cash



Key Escrow, e.g. *Clipper*



Trustee 1

escrow key1

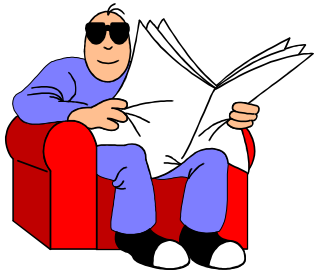
ECash User

escrow key2

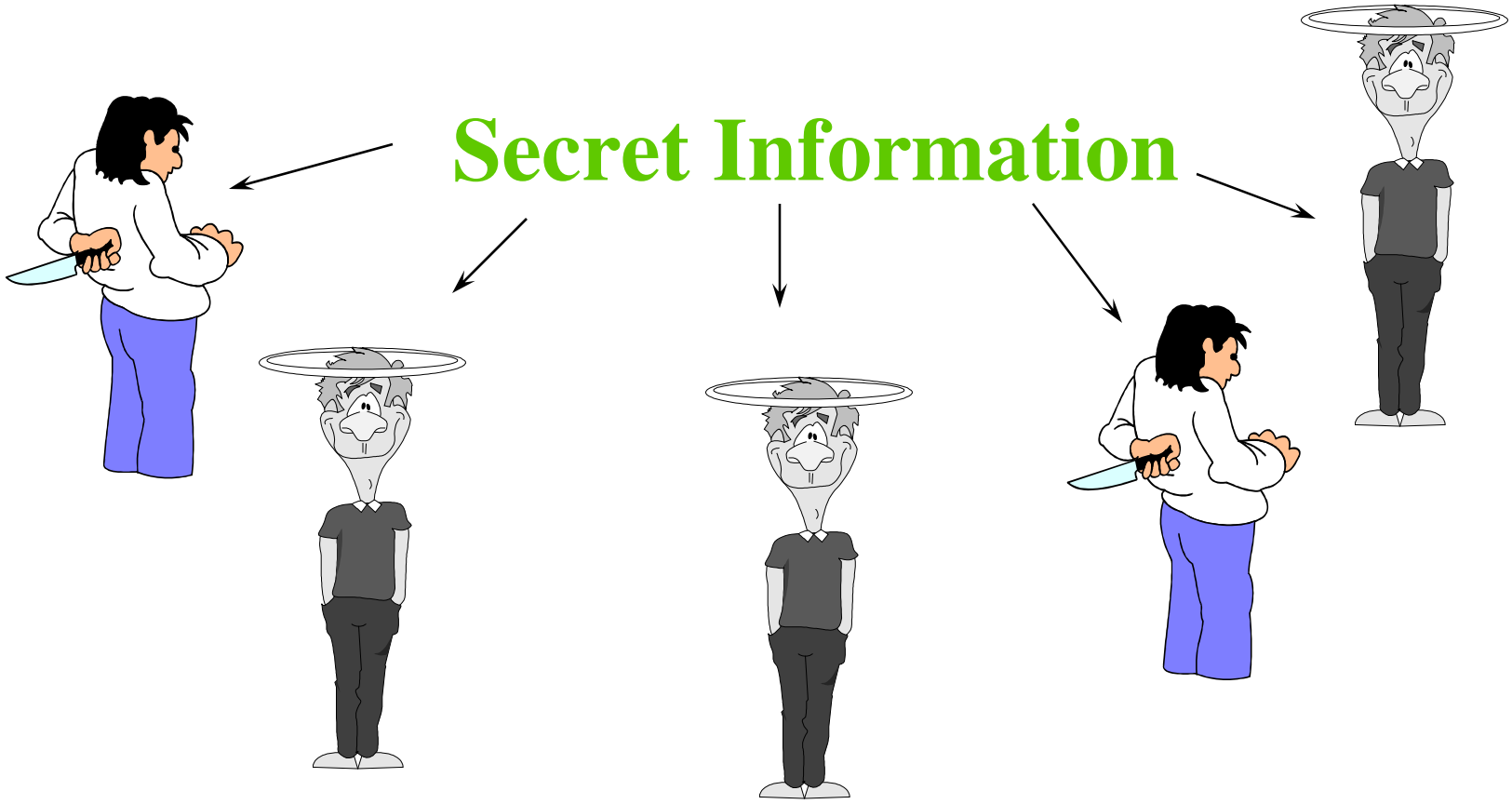
Trustee 2

escrow keyL

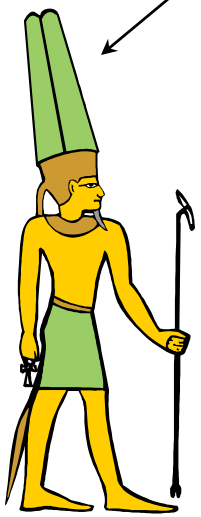
Trustee L



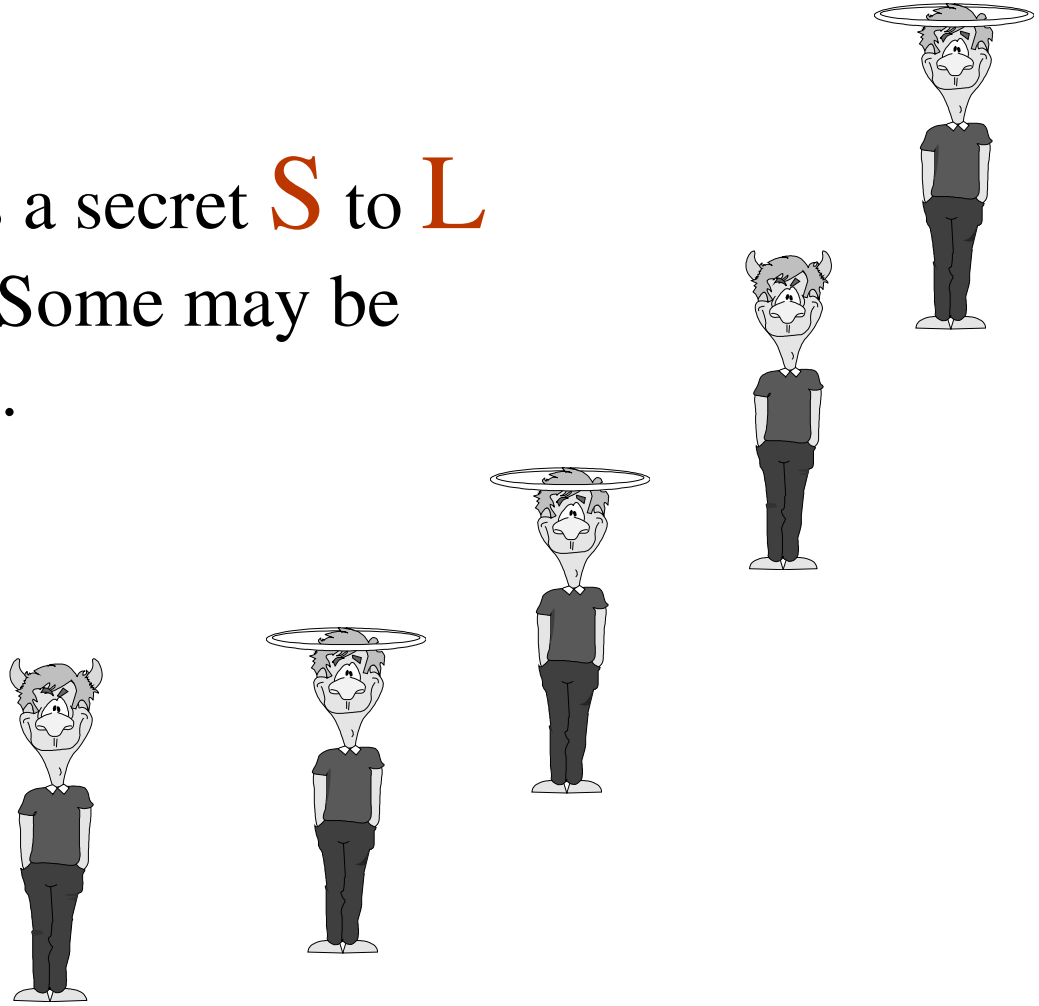
The Threshold Paradigm is one of *Distributed Trust*



Threshold Sharing



Dealer shares a secret S to L shareholders. Some may be untrustworthy.

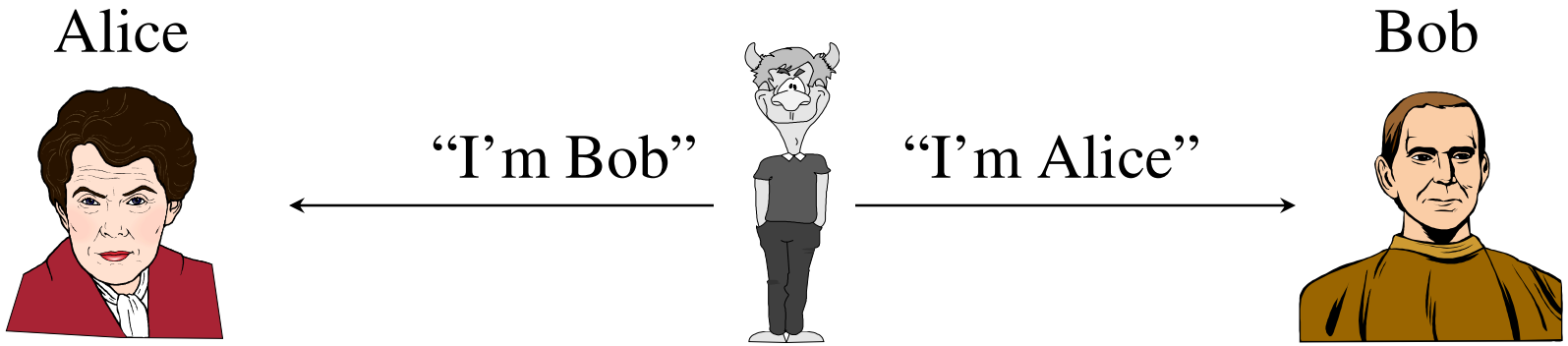


Threshold Sharing protects:

- *Secrecy*: no k-1 shareholders can learn the secret

- *Integrity*: no L-k shareholders can destroy the secret

Man-in-the-Middle Attack



Importance of Mathematical Approach

- Security is complex: mathematical analysis allows for designing and reasoning in terms of basic building blocks
- Mathematical analysis provides rigorous basis for security of cryptosystems
- Number theory and complexity theory are main foundations of cryptography

Crypto Information Resources

- RSA FAQ (http://www.rsa.com/rsalabs/newfaq/)
- Handbook of Applied Cryptography Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone (http://www.dms.auburn.edu/hac/)
- B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, New York, 2nd Edition, 1996
- D.R. Stinson. Cryptography - Theory and Practice. CRC Press, Boca Raton, 1995 http://bibd.unl.edu/~stinson/CTAP.html
- D. Kahn. The Codebreakers. Macmillan Co., New York, 1967

Security-related Courses and Master Program at Chalmers

- Computer Security course (Erland Jonsson)
- Cryptography course (Björn Sydow)

- Network security (Tomas Olovsson)
- Language-based security (Andrei Sabelfeld)

- Master program on secure and dependable computing