RW354
Principles of Computer Networking

A.E. Krzesinski and B.A. Bagula
Department of Computer Science
University of Stellenbosch

Last updated: 14 October 2004

The material presented in these slides is used with permission from

- *Larry L. Peterson and Bruce S. Davie. Computer Networks: A Systems Approach (Second Edition). Morgan Kaufmann Publishers. ISBN 1-55860-577-0.*
- *William Stallings. Data and Computer Communications (Sixth Edition). Prentice-Hall Inc. ISBN 0-13-571274-2.*
- *Andrew S. Tannenbaum. Computer Networks (Fourth Edition). Prentice Hall Inc. ISBN 0-13-349945-6.*

Security: overview

- *Cryptography functions*
  - *secret key (e.g. DES)*
  - *public key (e.g. RSA)*
  - *message digest (e.g. MD5).*
- *Security services*
  - *privacy: preventing unauthorized release of information*
  - *authentication: verifying the identity of the remote participant*
  - *integrity: making sure that the message has not been altered.*

Security: overview

- Security
  - Cryptography Algorithms
    - Secret Key (e.g., DES)
    - Public Key (e.g., RSA)
    - Message Digest (e.g., MD5)
  - Security Services
    - Privacy
    - Authentication
    - Message Integrity

Let P denote a plaintext message & K a key

$$D_K(E_K(P)) = P.$$

The encryption algorithms are well known: only the keys are secret.

Security: cryptanalysis

The longer the key, the more work has to be done to decrypt a message by exhaustive search of the key space.

The cryptanalysis problem

- *ciphertext only: the ciphertext is available but no plaintext*
- *known plaintext: some matched ciphertext & plaintext*
- *chosen plaintext: the cryptanalyst can encrypt some pieces of plaintext of her own choosing.*

Security: cryptanalysis

Substitution ciphers replace each letter or group of letters by another letter or group of letters

- *monoalphabetic substitution:* $26! = 4 \times 10^{26}$ *possible keys, 1μsec per trial will take* 10^{13} *years.*

Transposition ciphers re-order the plaintext letters but do not disguise them.

One time pads are unbreakable. The key is a random bit string known by both parties. The ciphertext is the EXCLUSIVE OR of the message & the key. The message length is limited by the length of the key.

Security: transposition cipher

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

Columnar transposition: the cipher is keyed by a word or phrase not containing any repeated letters. The key numbers the columns.

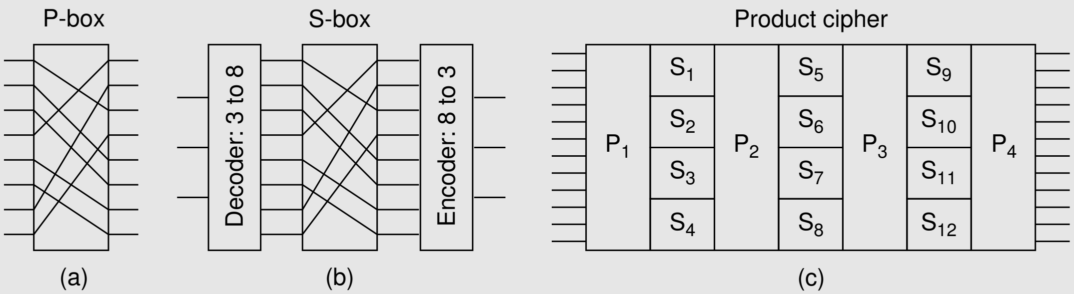
The plain text is written in rows. The ciphertext is read by columns, starting with the lowest numbered column.

Security: secret key algorithms

Modern cryptography makes use of transposition & substitution which are implemented by simple circuits.

A P-box performs transposition (permutation). In the figure below the 8 bits (top to bottom) are denoted `01234567`*: the output is* `36071245`*.*

An S-box performs substitution. In the figure below the 3-bit input selects 1 of 8 lines & sets it to 1: the other lines are 0. Inputting 8 octal numbers `01234567` *yields* `24506713`*.*



P-boxes & S-boxes are cascaded to form a product cipher.

Security: secret key (DES) encryption

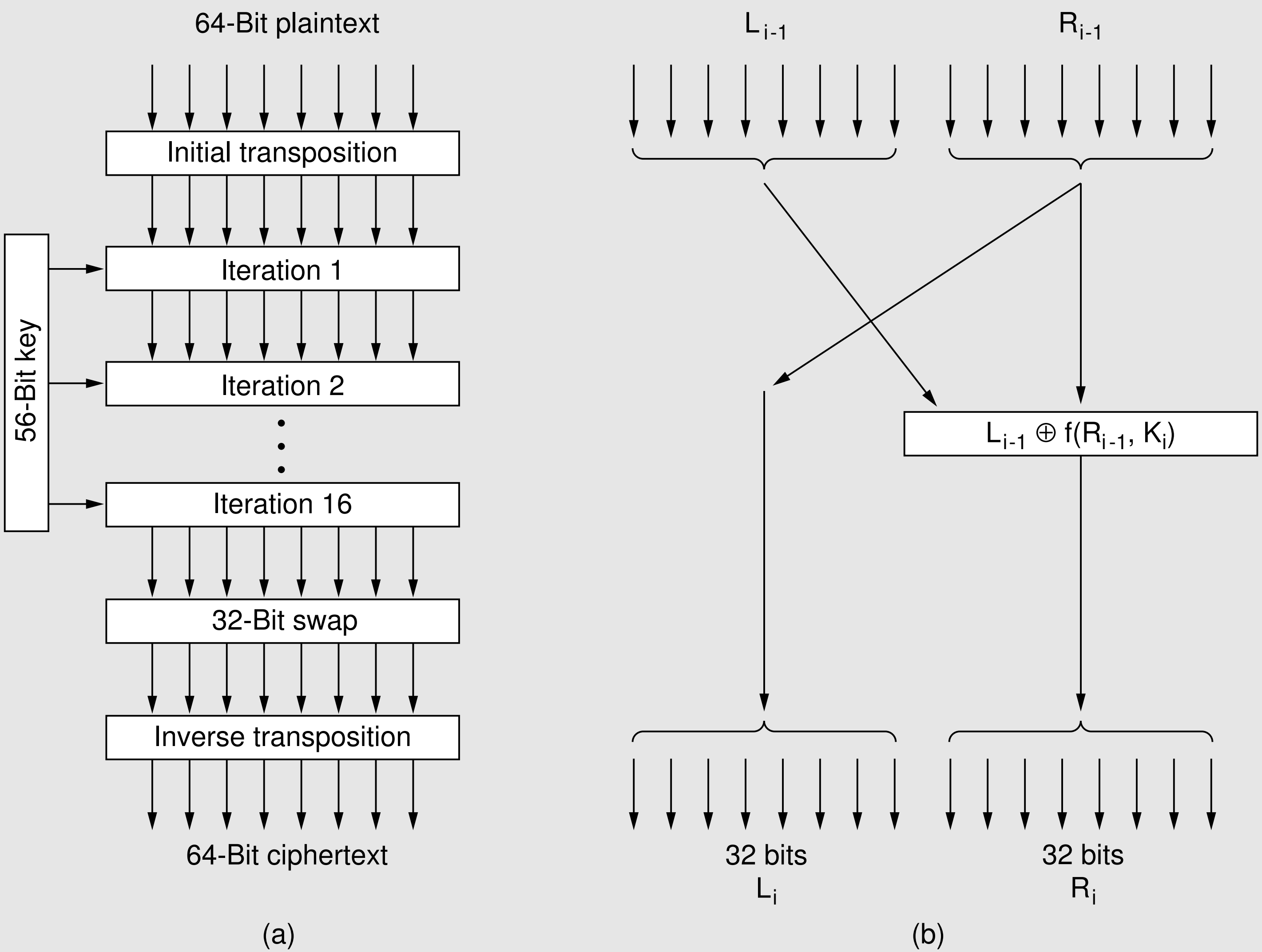
DES encrypts a 64-bit block of plaintext into 64 bits of ciphertext using a 56-bit key

- *symmetric: both participants share a single secret key*
- *64-bit plaintext blocks*
- *64-bit key (56-bits + 8-bit parity)*
- *16 rounds of encryption.*

Each 64-bit plaintext block is mangled in a sequence of parameterised iterations to produce a 64-bit ciphertext block.

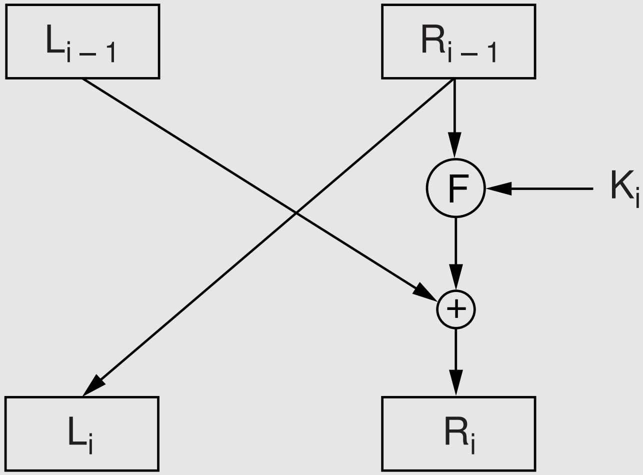
Security: DES

DES has 3 phases: an initial shuffle of the 64-bit block, 16 rounds of encryption & a final shuffle.



Security: DES

Manipulation at reach round i of DES



- *the 64-bit block from round $i-1$ is broken into two 32-bit halves L_{i-1}, R_{i-1}*
- *a 48-bit key K_i is selected in a complicated way from the 56-bit key K_{i-1}*
- *the two 32-bit halves L_i, R_i are computed as $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$.*

Security: DES

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

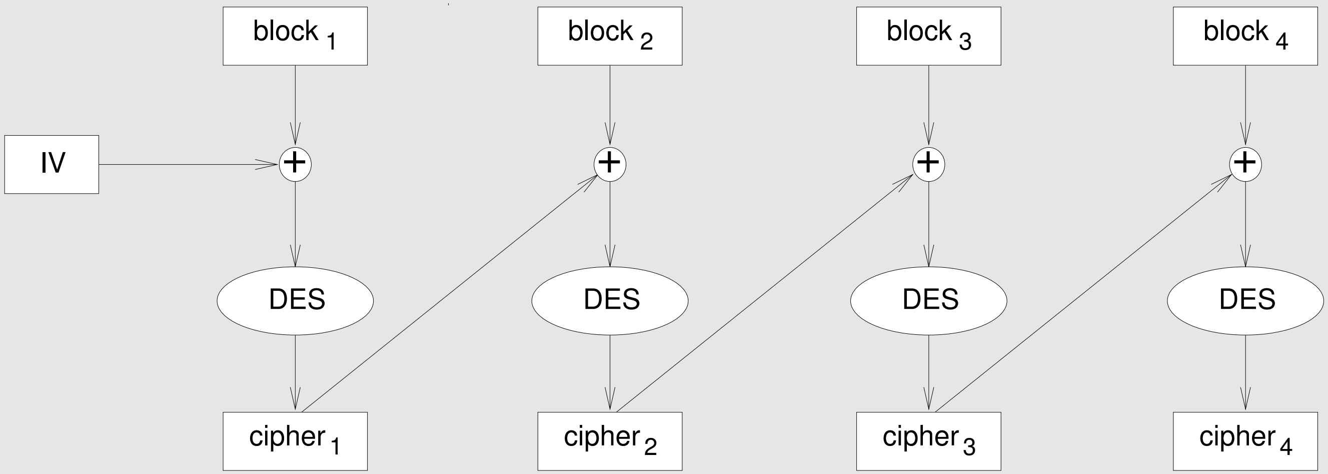
The function F expands R_{i-1} from 32- to 48-bits to combine it with the key 48-bit K_i.

The operation $L_{i-1} \oplus F(R_{i-1}, K_i)$ is performed in eight 6-bit chunks. Each 6-bit result is fed into an S-box (there are 8 of them, one per chunk) which outputs a 4-bit chunk.

The 32 bits are then passed through a P-box.

Security: DES

Cipher block chaining (CBC) is used for larger messages



Security: DES

DES is one of the strongest encryption algorithms ever devised

- *DES is based on an 128-bit key IBM cipher*
- *there is no published proof that DES is secure*
- *the most serious concern is the key size*
- *the only known way to break DES is to search all* $2^{56} = 7.6 \times 10^{16}$ *keys: one computer doing one DES encryption per* μs *would take* 10^3 *years*
- *searching the key space is highly parallelizable*
- *many applications now use triple-DES.*

Security: public key (RSA)

Each participant has a secret key which no-one else knows & a public key which is published

- *participant A encrypts data for participant B using B's public key*
- *B uses its private key to decrypt the data.*

Note that A cannot decrypt the message that it has sent to B.

Security: RSA

Generate a public and a private key

- *choose two 256-bit prime numbers* p, q
- $n = p \times q$
- *chose the encryption key* e *such that* e *and* $(p-1) \times (q-1)$ *are relatively prime*
    - *two numbers are relatively prime if they have no common factor greater than one*
- *compute the decryption key* d *such that*

$$e \times d = 1 \; mod \; ((p-1) \times (q-1))$$

- *the public key is* $\langle e, n \rangle$
- *the private key is* $\langle d, n \rangle$
- *discard (do not disclose) the original primes* p, q*.*

Security: RSA example

- *choose* $p = 7$, $q = 17$
- $n = p \times q = 119$
- *the encryption key* e *and* $(p-1) \times (q-1) = 96$ *are relatively prime:* $e = 5$
- $5d = 1 \ mod \ 96$*: choose* $d = 77$ *because* $5 \times 77 = 385$, $385 \ mod \ 96 = 1$

Suppose we wish to encrypt $m = 19$*.*

The public key $\langle e, n \rangle$ *is used to encrypt the message*

$$c = m^e mod \ n = 19^5 mod \ 119 = 66.$$

The private key $\langle d, n \rangle$ *is used to decrypt the message*

$$m = c^d mod \ n = 66^{77} mod \ 119 = 19.$$

Security: RSA example

A sends a message m to B using B's public key $\langle e, n \rangle$ to encrypt the message m to a ciphertext $c = m^e mod\ n$.

If e is a large number then m^e will overflow.

A simple method (not the best) to compute m^e is

```
C = 1;
for (i=1; i<=e; i++)
  C = (C * i) % n;
```

Thus $16^6 mod\ 116$ is computed as follows:

```
( 1 * 19) mod 119 = 19
(19 * 19) mod 119 =  4
( 4 * 19) mod 119 = 76
(76 * 19) mod 119 = 16
(16 * 19) mod 119 = 66
```

Security: RSA example

Plaintext (M)			Ciphertext (C)		After decryption	
Symbolic	Numeric	M^3	M^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Sender's computation

Receiver's computation

In this example $p = 3$ and $q = 11$. Compute the private keys $\langle e, n\rangle$ and $\langle d, n\rangle$.

Security: message digest

- *Cryptographic checksum: just as a regular checksum protects the receiver from accidental changes to the message, a cryptographic checksum protects the receiver from malicious changes to the message.*
- *One-way function: given a cryptographic checksum for a message, it is virtually impossible to figure out what message produced that checksum; it is not computationally feasible to find two messages that hash to the same cryptographic checksum.*
- *Relevance: if you are given a checksum for a message & you are able to compute exactly the same checksum for that message, then it is highly likely this message produced the checksum you were given.*

Security: performance of encryption algorithms

Software implementations

- *DES and MD5 are several orders of magnitude faster than RSA*
- *typically: DES achieves 36Mbps, MD5 gets 85Mbps and RSA gets 1Kbps.*

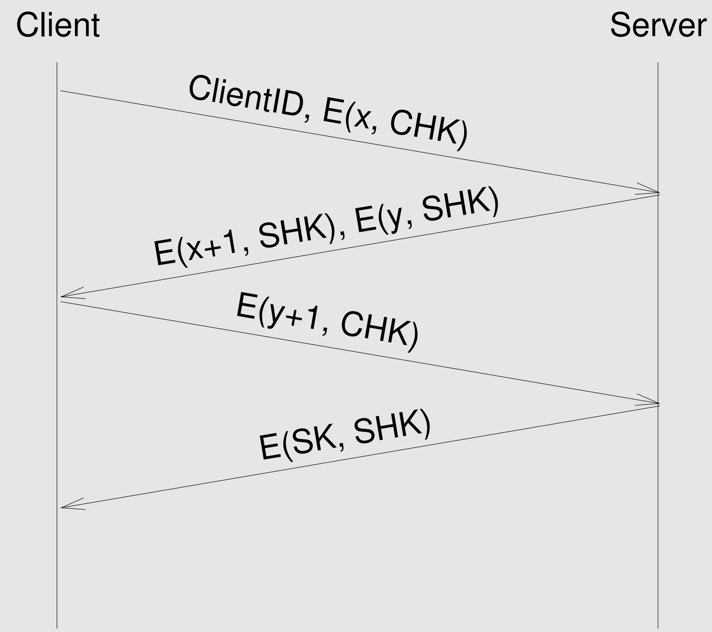
Hardware implementations

- *DES and MD5 achieve several 100's of Kbps*
- *RSA gets 64Kbps.*

RSA is typically used to encrypt small amounts of data such as keys & passwords. These RSA protected secrets are used with DES & MD5 to encrypt larger amounts of data.

Security: authentication protocols

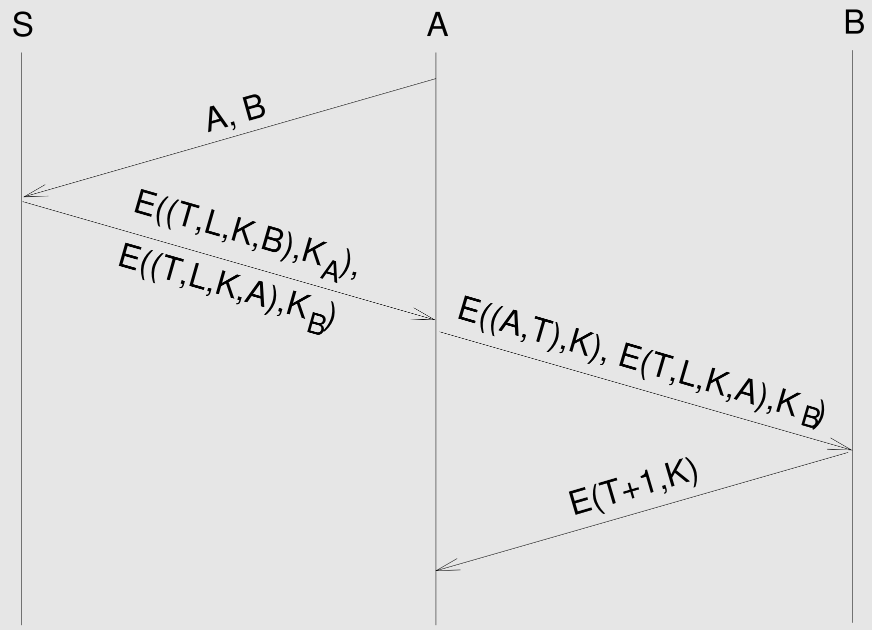
Three-way handshake



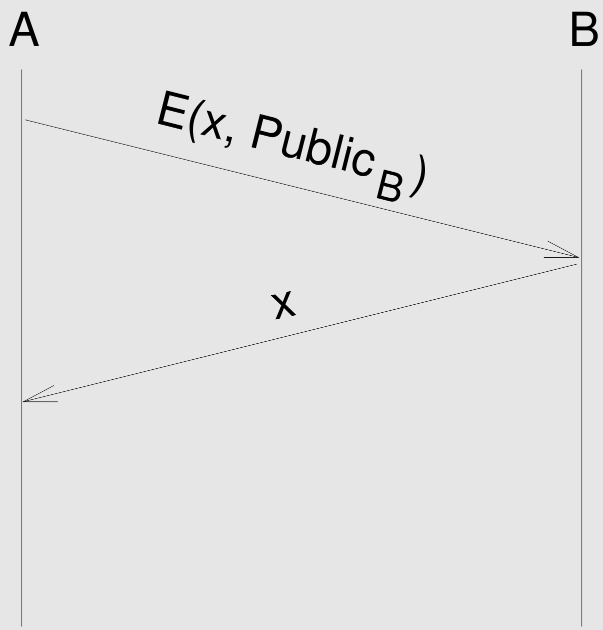
- *CHK: client handshake key*
- *SHK: server handshake key – the key that the server thinks will correspond to the* `ClientId`*.*

Security: authentication protocols

Trusted third party (Kerberos)



Public key authentication



Security: message integrity protocols

Digital signature using RSA

- *a special case of message integrity where the signature can only have been generated by one participant*
  - *the sender encrypts his signature with his private key*
  - *the receiver verifies the signature using the sender's public key.*

Security: message integrity protocols

Keyed MD5

- *the sender*

  $$m + MD5(m + k) + E(k, private)$$

- *the receiver*
  - *recovers the random key k using the sender's public key*
  - *applies MD5 to $(m + k)$*
  - *compares the result with the checksum sent with the message.*

Security: message integrity protocols

MD5 with RSA signature

- *the sender*

  $$m + E(MD5(m), private)$$

- *the receiver*
  - *decrypts the signature with the sender's public key*
  - *compares the result with the MD5 checksum sent with the message.*

Security: public key distribution

A wishes to convey his public key to B.

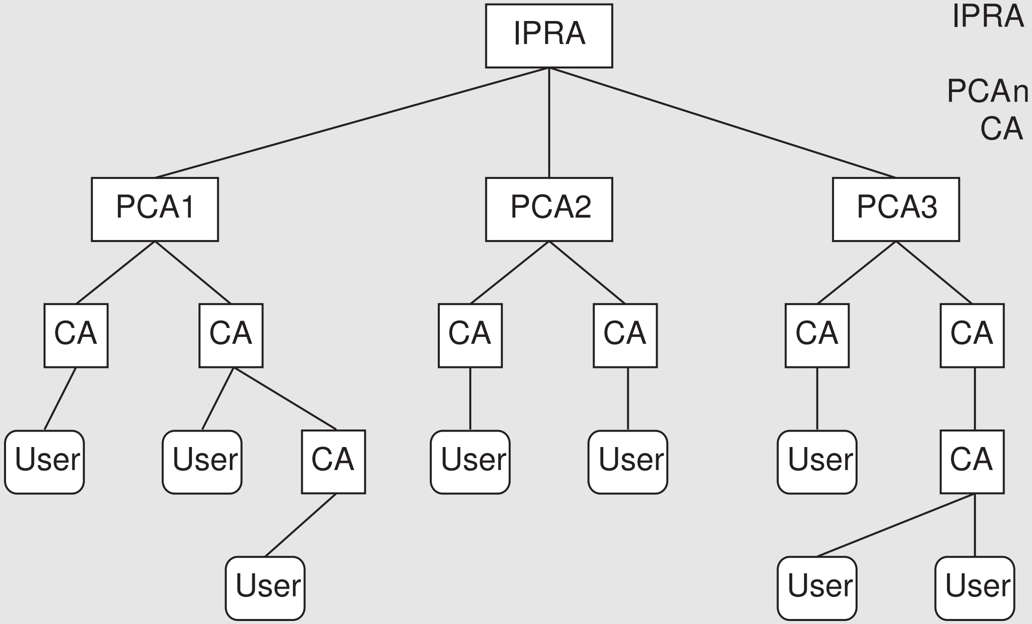
Question: how does B know that the key comes from A?

Answer: use digital certificates which prove that the data were generated by the owner of a certain key & that the data were not modified since it was signed.

Digital certificates are issued by a Certification Authority (CA).

Security: public key distribution

Chains of trust are arranged in a tree-structured hierarchy



IPRA = Internet Policy Registration Authority (root)
PCAn = policy certification authority
CA = certification authority

X.509 defines the standard for digital certificates

- *the name of the entity being certified*
- *the public key of the entity*
- *the name of the Certification Authority*
- *a digital signature.*

Security: certificate revocation

If someone has discovered your private key then that person can impersonate you.

A CA can issue a Certificate Revocation List *(CRL) – a digitally signed list of revoked certificates.*

When a participant receives a digital certificate for B he will check the CRL that B's certificate has not been revoked.