Entity authentication and symmetric key establishment

Prof. Bart Preneel
COSIC
Bart.Preneel(at)esatDOTkuleuven.be
http://homes.esat.kuleuven.be/~preneel

March 2013

Outline

- 1. Cryptology: concepts and algorithms
  - symmetric algorithms for confidentiality
  - symmetric algorithms for data authentication
  - public-key cryptology
- **2. Cryptology: protocols**
  - **identification/entity authentication**
  - **key establishment**
- 3. Public-Key Infrastructure principles
- 4. Networking protocols
  - email, web, IPsec, SSL/TLS
- 5. New developments in cryptology
- 6. Cryptography best practices

Definitions (ctd)

Confidentiality
Integrity
Availability

	data	entities
confidentiality	encryption	anonymity
authentication	data authentication	identification

- Authorisation
- Non-repudiation of origin, receipt
- Contract signing
- Notarisation and Timestamping
- E-voting, e-auction,…

Don't use the word authentication without defining it

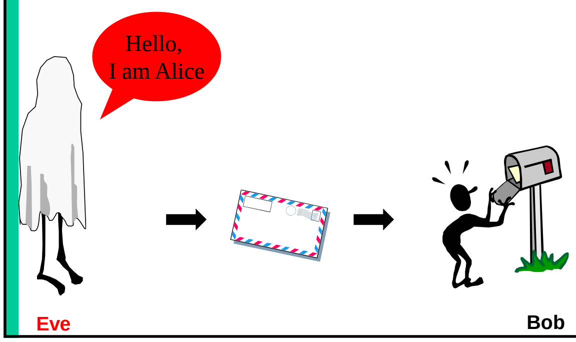
3

Identification

- the problem
- passwords
- challenge response with symmetric key and MAC (symmetric tokens)
- challenge response with public key (signatures, ZK)
- biometry

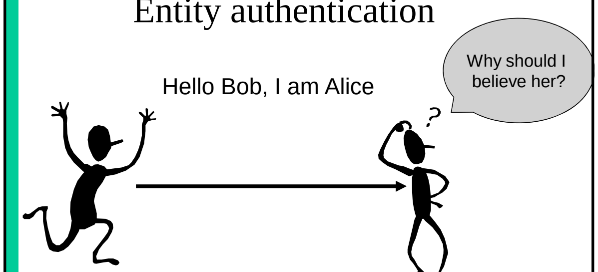
4

Entity authentication



5

Entity authentication



entity authentication: one is corroborated of the identity of another party, and of the fact that this party is **alive (active)** during the protocol

6

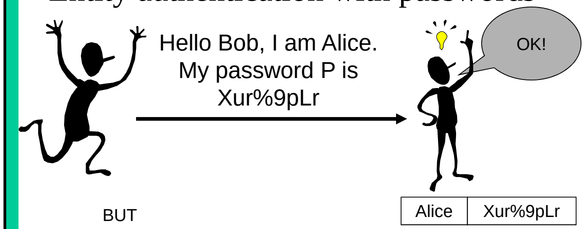
Entity authentication is based on one or more of the following elements:

- what someone knows
  - password, PIN
- what someone has
  - magstripe card, smart card
- what someone is (biometrics)
  - fingerprint, retina, hand shape,...
- how someone does something
  - manual signature, typing pattern
- where someone is
  - dialback, location based services (GSM, Galileo)

ert5^r$#89Oy

7

Entity authentication with passwords



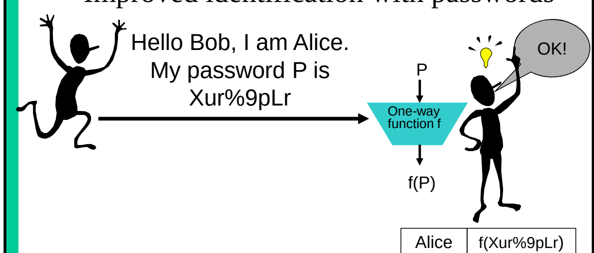
BUT

- Eve can guess the password
- Eve can listen to the channel and learn Alice's password
- Bob needs to know Alice's secret
- Bob needs to store Alice's secret in a secure way

Possibility of replay: liveliness is missing

8

Improved identification with passwords

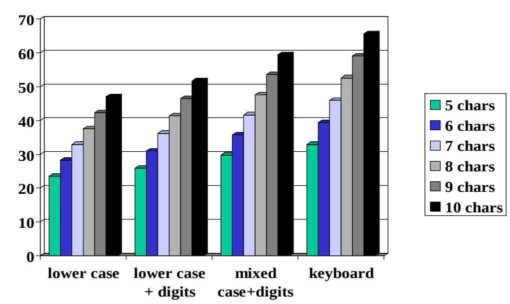


Bob stores f(P) rather than Alice's secret P

- it is difficult to deduce P from f(P)

9

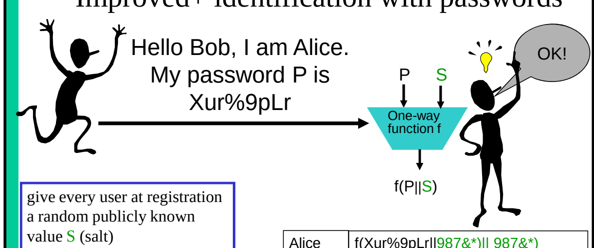
Password entropy: effective key length



Problem: passwords from dictionaries

10

Improved+ identification with passwords

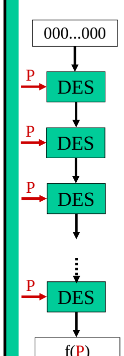


Bob stores f(P,S) || S rather than Alice's secret P

it is harder to attack the passwords of all users simultaneously

11

Example: UNIX



- Function f() = DES applied 25 times to the all zero plaintext with as key the password P (8 7-bit characters)
- Salt: 12-bit modification to DES
- etc/passwd public
- PC: 20-40 million passwords/second
- But time-memory tradeoff…
  - Precomputation per salt 25 . 2^{56}
  - Storage per salt: 2 Terabyte
  - Find one key in time 25.2^{38}

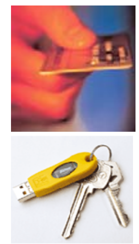
12

Improving password security

- Apply the function f "x" times to the password (iteratively)
  - if x = 100 million, testing a password guess takes a few seconds
  - need to increase x with time (Moore's law)
- Disadvantage: one cannot use the same hashed password file on a faster server and on an embedded device with an 8-bit microprocessor
  - need to use different values of x depending on the computational power of the machine

13

Problem: human memory is limited



- Solution: store key K on magstripe, USB key, hard disk
- Stops guessing attacks

But this does not solve the other problems related to passwords

And now you identify the card, not the user….

Possibility of replay: liveliness is missing

14

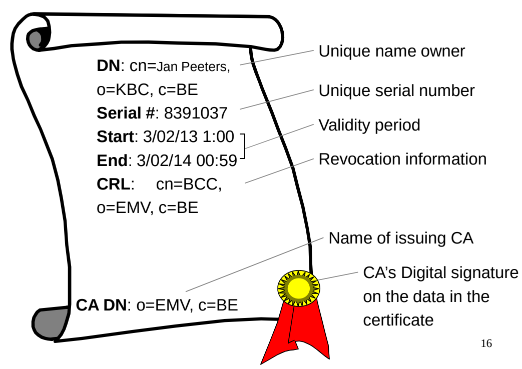
Improvement: Static Data Authentication

- Replace K by a signature of a third party CA (Certification Authority) on Alice's name: $SigSK_{CA}$ (Alice) = special certificate
- Advantage: can be verified using a public string PK_{CA}
- Advantage: can only be generated by CA
- Disadvantage: signature = 40..128 bytes
- Disadvantage: can still be copied/intercepted

Possibility of replay: liveliness is missing

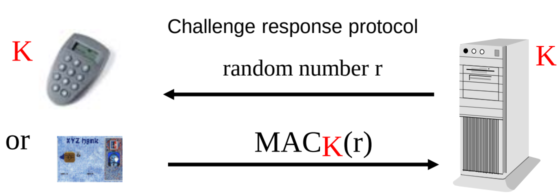
15

"Certificate" for static data authentication



16

Entity authentication with symmetric token

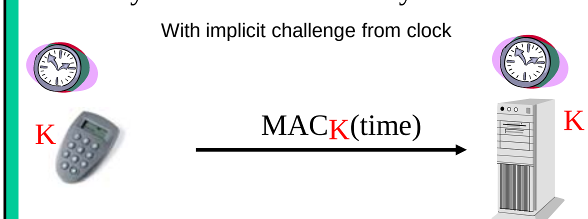


- Eavesdropping no longer effective
- Bob still needs secret key K

Detects whether Alice is alive!

17

Entity authentication with symmetric token

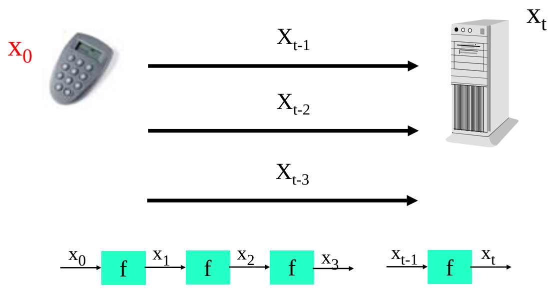


- Eavesdropping no longer effective
- Bob still needs secret key K
- resynchronization mechanism needed

18

Lamport's one-time passwords

iterated one-way function

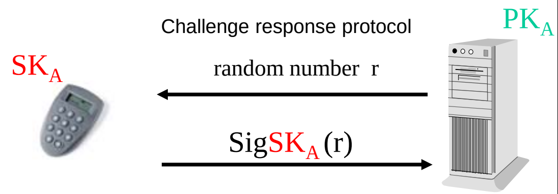


• Disadvantage: only works with one Bob

19

Entity authentication with public key token

Challenge response protocol



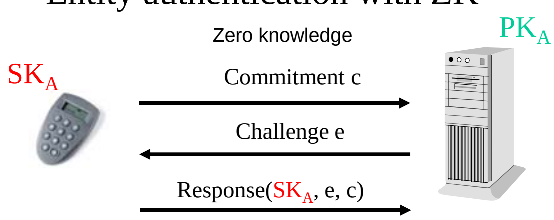
• Eavesdropping no longer effective

• Bob no longer needs a secret – only PK_A

20

Entity authentication with ZK

Zero knowledge



• Mathematical proof that Bob only learns that he is talking to Alice (1 bit of information)

• Bob cannot use this information to convince a third party that he is/was talking to Alice

21

ZK definitions

- **complete:** if Alice knows the secret, she can carry outthe protocol successfully
- **sound:** Eve (who wants to impersonate Alice) can only convinceBob with a very small probability that she is Alice;
- **zero knowledge:** even a dishonest Bob does not learn anything except for 1 bit (he is talking to Alice); he could have produced himself all the other information he obtains during the protocol.

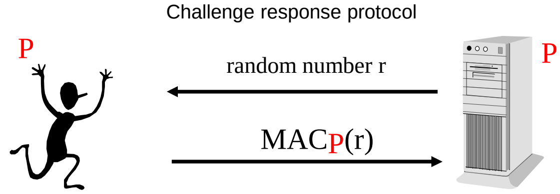
22

Overview Identification Protocols

	Guess	Eavesdrop channel (liveliness)	Impersonation by Bob	Secret info for Bob	Security
Password	-	-	-	-	1
Magstripe (SK)	+	-	-	-	2
Magstripe (PK)	+	-	-	+	3
Dynamic password	+	+	-	-	4
Smart card (SK)	+	+	-	-	4
Smart Card (PK)	+	+	+	+	5

Entity authentication with password

Challenge response protocol



• Eavesdropping no longer effective
• Bob still needs secret key P
• Exhaustive search for P is easy based on a single transcript

24

Entity authentication in practice

- Phishing – mutual authentication
- Forward credentials - biometry
- Interrupt after initial authentication – authenticated key establishment
- Mafia fraud – distance bounding
- Protocol errors – check that local device authentication is linked to entity authentication protocol (example: EMV)

25

Mutual authentication

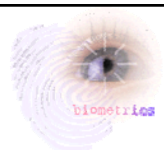
- Phishing is impersonating of the verifier (e.g. the bank)
- Most applications need entity authentication in two directions
- !! This is not complete the same as 2 parallel unilateral protocols for entity authentication

2 stage authentication

- Local: user to device
- Device to rest of the world

26

Biometry

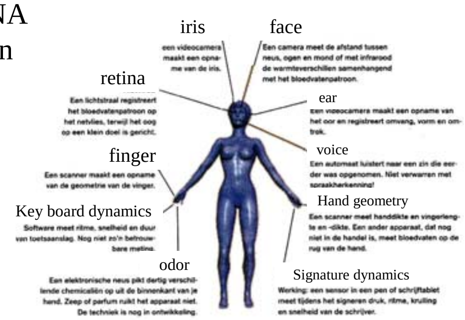


- Based on our unique features
- Identification or verification
  - Is this Alice?
  - Check against watchlist
  - Has this person ever registered in the system?

27

Some unique features

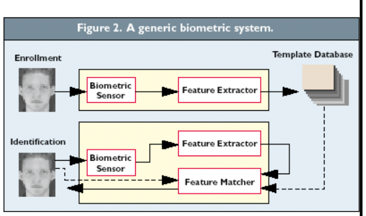
DNA
skin
…



28

Biometric procedures

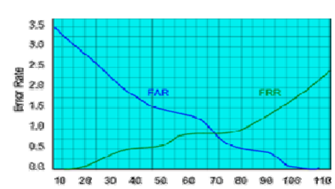
- Registration
- Template extraction
- Measurement
- Processing
- Template matching
- Link with applications



29

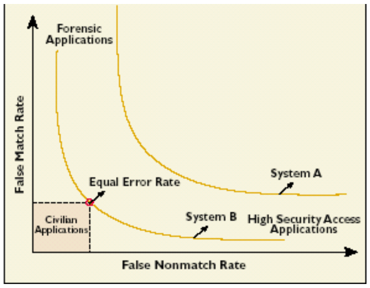
Robustness/performance

- Performance evaluation
  - False Acceptance Ratio or False Match Rate
  - False Rejection Ratio or False Non-Match Rate
- Application dependent



30

Robustness/performance (2)



31

Fingerprint

- Used for PC/laptop access
- Widely available
- Reliable and inexpensive
- Simple interface

minutiae

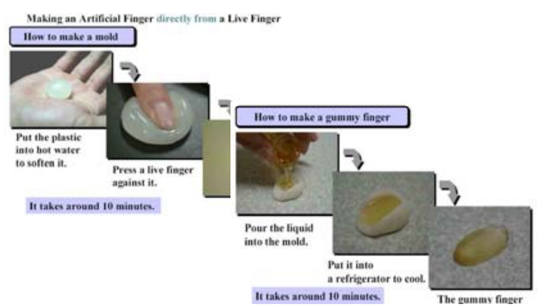
32

Fingerprint (2)

- Small sensor
- Small template (100 bytes)
- Commercially available
  - Optical/thermical/capacitive
  - Liveness detection
- Problems for some ethnic groups and some professions
- Connotation with crime

33

Fingerprint (3): gummy fingers



34

Hand geometry

- Flexible performance tuning
- Mostly 3D geometry
- Example: 1996 Olympics

35

Voice recognition

- Speech processing technology well developed
- Can be used at a distance
- Can use microphone of our gsm
- But tools to spoof exist as well
- Typical applications: complement PIN for mobile or domotica

36

Iris Scan

- No contact and fast
- Conventional CCD camera
- 200 parameters
- Template: 512 bytes
- All etnic groups
- Reveals health status

37

Retina scan

- Stable and unique pattern of blood vessels
- Invasive
- High security

38

Manual signature

- Measure distance, speed, accelerations, pressure
- Familiar
- Easy to use
- Template needs continuous update
- Technology not fully mature

39

Facial recognition

- User friendly
- No cooperation needed
- Reliability limited
- Robustness issues
  - Lighting conditions
  - Glasses/hair/beard/...

40

Comparison

Feature	Uniqueness	Permanent	Performance	Acceptability	Spoofing
Facial	Low	Average	Low	High	Low
Fingerprint	High	High	High??	Average	High??
Hand geometry	Average	Average	Average	Average	Average
Iris	High	High	High	Low	High
Retina	High	Average	High	Low	High
Signature	Low	Low	Low	High	Low
Voice	Low	Low	Low	High	Low

41

Biometry: pros and cons

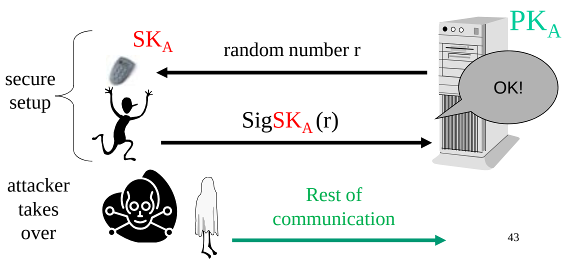
- Real person
- User friendly
- Cannot be forwarded
- Little effort for user
- Secure implementation: derive key in a secure way from the biometric

- Privacy (medical)
- Intrusive?
- Liveliness?
- Cannot be replaced
- Risk for physical attacks
- Hygiene
- Does not work everyone, e.g., people with disabilities
- Reliability
- No cryptographic key

42

Keeping authenticity alive

- Establish who someone is
- Establish that this person is active/liveliness
- But what if the connection is broken after the initial phase?



43

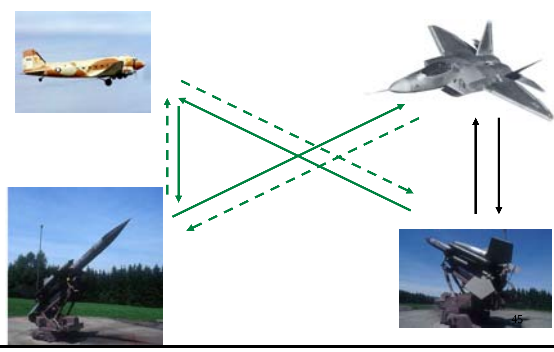
Solution

- Authenticated key agreement

- Run a mutual entity authentication protocol
- Establish a key
- Encrypt and authenticate all information exchanged using this key

44

The mafia fraud – or the grandmaster chess problem



45

Location-based authentication

- Distance bounding: try to prove that you are physically close to the verifier

- Other uses of "location"
  - Dial-back: can be defeated using fake dial tone
  - IP addresses and MAC addresses can be spoofed
  - Mobile/wireless communications: operator knows access point, but how to convince others?
  - Trusted GPS: Galileo?

46

Authentication with device

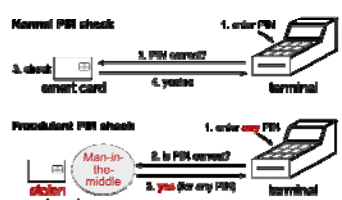
- E.g. smart card, secure login token
- Needs 2 stages
  - Local: user to device
  - Device to rest of the world
- Are these 2 stages connected properly?

47

Warning about EMV

http://www.cl.cam.ac.uk/research/security/banking/nopin/oakland10chipbroken.pdf

- **EMV PIN verification "wedge" vulnerability** S.J. Murdoch, S. Drimer, R. Anderson, M. Bond, IEEE Security & Privacy 2010



48

Guidelines

NIST Special Publication 800-63 Version 1.0.2 (2006): Electronic Authentication Guideline: identifies four levels of assurance

http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf

See http://csrc.nist.gov/publications/PubsSPs.html

for about 120 Special Publications (800 Series) from NIST on computer security and cryptography

49

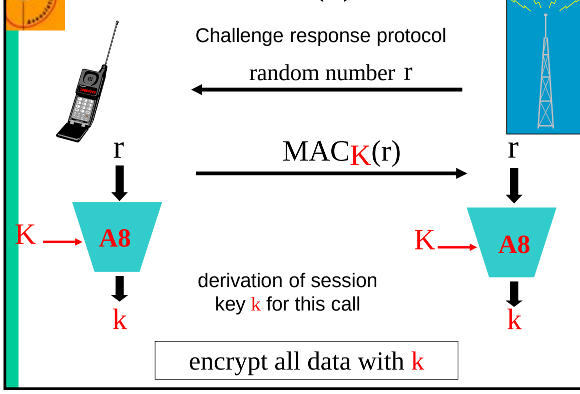
Key establishment

- The problem
- How to establish secret keys using secret keys?
- How to establish secret keys using public keys?
  - Diffie-Hellman and STS
- How to distribute public keys? (PKI)

Key establishment: the problem

- Cryptology makes it easier to secure information, by replacing the security of information by the security of keys
- The main problem is how to establish these keys
  - 95% of the difficulty
  - integrate with application
  - if possible transparent to end users

GSM (1)



GSM (2)

- SIM card with long term secret key K (128 bits)
- secret algorithms
  - A3: MAC algorithm
  - A8: key derivation algorithm
  - A5.1/A5.2: encryption algorithm
- anonimity: IMSI (International Mobile Subscriber Identity) replaced by TIMSI (temporary IMSI)
  - the next TIMSI is sent (encrypted) during the call set-up

Point-to point symmetric key distribution

- Before: Alice and Bob share long term secret K_{AB}

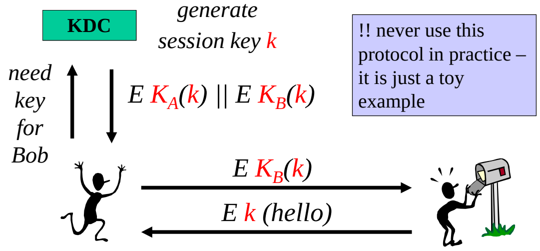
generate session key k → $EK_{AB}(k \,||\, time \,||\, Bob)$ → *decrypt extract k*

← $Ek\,(time \,||\, Alice \,||\, hello)$

- After: Alice and Bob share a short term key *k*
  - which they can use to protect a specific interaction
  - which can be thrown away at the end of the session
- Alice and Bob have also authenticated each other

Symmetric key distribution with 3rd party

- Before (KDC=Key Distribution Center)
  - Alice shares a long term secret with KDC: K_A
  - Bob shares long term secret with KDC: K_B

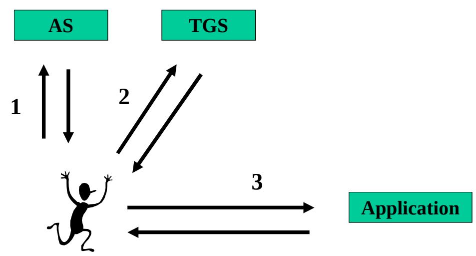


Symmetric key distribution with 3rd party(2)

- After: Alice and Bob share a short term key k
- Need to trust third party!
- Single point of failure in system

Kerberos/Single Sign On (SSO)

- Alice uses her password only once per day



Kerberos/Single Sign On (2)

- Step 1: Alice gets a "day key" K_A from AS (Authentication Server)
  - based on a Alice's password (long term secret)
  - K_A is stored on Alice's machine and deleted in the evening
- Step 2: Alice uses K_A to get application keys k_i from TGS (Ticket Granting Server)
- Step 3: Alice can talk securely to applications (printer, file server) using application keys k_i

A public-key distribution protocol: Diffie-Hellman

- Before: Alice and Bob have never met and share no secrets; they know a public system parameter α

Alice		Bob
generate x, *compute* α^x	α^x →	*generate* y, *compute* α^y
	← α^y	
compute $k=(\alpha^y)^x$		*compute* $k=(\alpha^x)^y$

- After: Alice and Bob share a short term key k
  - Eve cannot compute k : in several mathematical structures it is hard to derive x from α^x (this is known as the discrete logarithm problem)

Diffie-Hellman (continued)

Alice		Bob
generate x, *compute* α^x	α^x →	*generate* y, *compute* α^y
	← α^y	
compute $k=(\alpha^y)^x$		*compute* $k=(\alpha^x)^y$

- BUT: How does Alice know that she shares this secret key k with Bob?
- Answer: Alice has no idea at all about who the other person is! The same holds for Bob.

Meet-in-the middle attack

- Eve shares a key *k1* with Alice and a key *k2* with Bob
- Requires *active* attack

Alice → Eve: α^{x1}
Eve → Alice: α^{y1}
Eve → Bob: α^{x2}
Bob → Eve: α^{y2}

$k1 = (\alpha^{y1})^{x1} = (\alpha^{x1})^{y1}$ $k2 = (\alpha^{y2})^{x2} = (\alpha^{x2})^{y2}$

Entity authentication with password: EKE

[Bellovin,Merritt '92]
All operations mod p

P		P
$x \in_R [1,p-1]$	$A \,\|\, E_P(\alpha^x)$ →	$y \in_R [1,p-1]$
	← $A \,\|\, E_P(\alpha^y \,\|\, r_B)$	r_B 128-bit string
r_A 128-bit string	$E_k(r_A \,\|\, r_B)$ →	$k = (\alpha^x)^y$
$k = (\alpha^y)^x$	← $E_k(r_A)$

- Adds entity authentication to Diffie Hellman
- Attacker cannot perform off-line exhaustive search for the password P
- Attacker can still try on-line attacks; need to restrict number of uses of the account
- Literature: PAKE: Password Authenticated Key Establishment

62

Station to Station protocol (STS)

- The problem can be fixed by adding digital signatures
- This protocol plays a very important role on the Internet (under different names)

choose x	α^x →	
	← α^y	choose y
$k=(\alpha^y)^x$		$k=(\alpha^x)^y$
	$SigA(\alpha^x,\alpha^y)$ →	
√SigB	← $SigB(\alpha^y,\alpha^x)$	√SigA

IKE - Main Mode with Digital Signatures

Initiator		Responder
	proposed attributes →	
	← selected attributes	
	g^x, N_i →	
	← g^y, N_r	
K derived from master = prf($N_i \,\|\, N_r, g^{xy}$) SIG_i = Signature on H(master, $g^x \,\|\, g^y \,\|\, \ldots \,\|\, ID_i$)	E(K, ID_i, [Cert(i)], SIG_i) →	SIG_r = Signature on H(master, $g^y \,\|\, g^x \,\|\, \ldots \,\|\, ID_r$)
	← E(K, ID_r, [Cert(r)], SIG_r)	

H is equal to prf or the hash function tied to the signature algorithm
(all inputs are concatenated)

Key transport using RSA

generate k $E_{PKB}(k)$	$E_{PKB}(k)$ →	*decrypt using SKB to obtain k*

- How does Bob know that ***k*** is a fresh key?
- How does Bob know that this key ***k*** is coming from Alice?
- How does Alice know that Bob has received the key ***k*** and that Bob is present (entity authentication)?

Key transport using RSA (2)

generate k $E_{PKB}(k)$	$E_{PKB}(k \,\|\, t_A)$ →	*decrypt using SKB to obtain k*

- Freshness is solved with a timestamp t_A

Key transport using RSA (3)

generate k

$Sig_{SKA}(E_{PKB}(k \,||\, t_A))$ →

decrypt using SKB and verify using PKA

- Alice authenticates by signing the message
- There are still attacks (signature stripping…)

Key transport using RSA (4): X.509

generate k

$Sig_{SKA}(B \,||\, t_A \,||\, E_{PKB}(A \,||\, k)) \,||\, t_A \,||\, E_{PKB}(A \,||\, k)$ →

decrypt using SKB and verify using PKA

Mutual: B can return a similar message including part of the first message

Problem (compared to D-H/STS): lack of **forward secrecy**

If the long term key *SKB* of Bob leaks, all past session keys can be recovered!

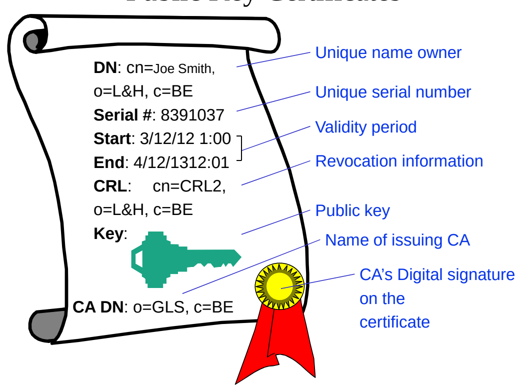
Distribution of public keys

- How do you know whose public key you have?
- Where do you get public keys?
- How do you trust public keys?
- What should you do if your private key is compromised?

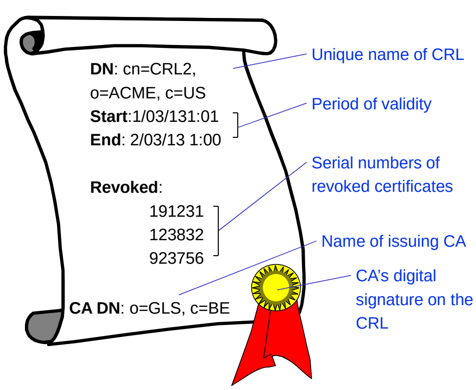
reduce protection of public key of many users to knowledge of a single public key of a Certification Authority (CA)

digital certificates &
Public Key Infrastructure (PKI)

Public Key Certificates



Certificate Revocation List



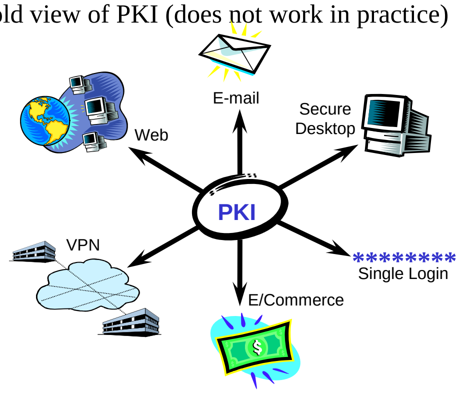
Essential PKI Components

- Certification Authority
- Revocation system
- Certificate repository ("directory")

- Key backup and recovery system
- Support for non-repudiation
- Automatic key update
- Management of key histories
- Cross-certification
- PKI-ready application software

72

PKI-ready application software:
old view of PKI (does not work in practice)



Example of a key hierarchy