Authentication

- Verifying an identity
- People authentication
- Host authentication

Authentication vulnerabilities

- eavesdropping
- password database
- replay
- online/ offline guessing
- session maybe hijacked after authentication!

Authenticating people

Computer verifying who you are

- **what you know : password**
- **what you have : physical keys**
- **what you are : fingerprint *etc*.**

Best : at least two of the above

Authentication protocols

- one-way
  - password
  - challenge/response
  - public-key
- two-way (mutual authentication)
  - trusted intermediary (Kerberos)
  - public-key

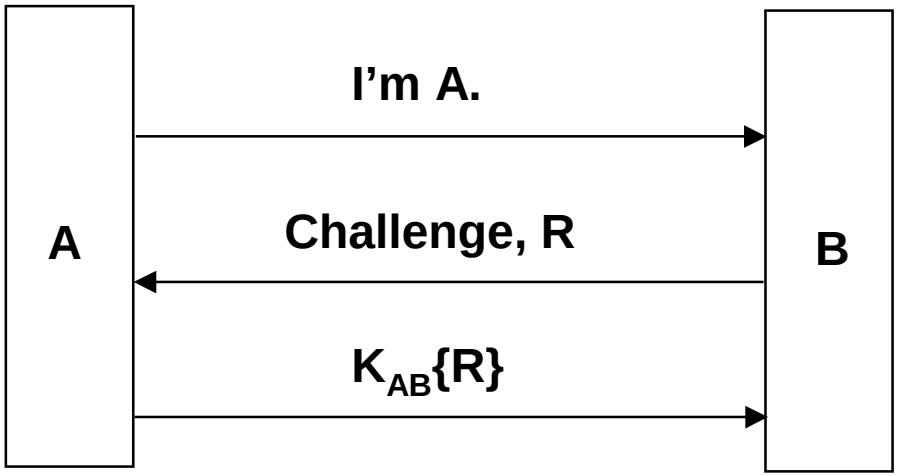
Authentication Systems

- Password-based authentication
  - Off-line vs On-line Password guessing
  - Storing user passwords
- Address-based authentication
  - etc/hosts.equiv, .rhosts (UNIX)
- Trusted Intermediaries
  - KDC (Key Distribution Center)
  - CA (Certification Authorities)
  - Multiple Trusted Intermediaries

Password authentication

- easy and popular
- Assuming
  - No eavesdropping
  - No bad guys
- Replacing clear password with cryptographic challenge/response

Shared secret(I)

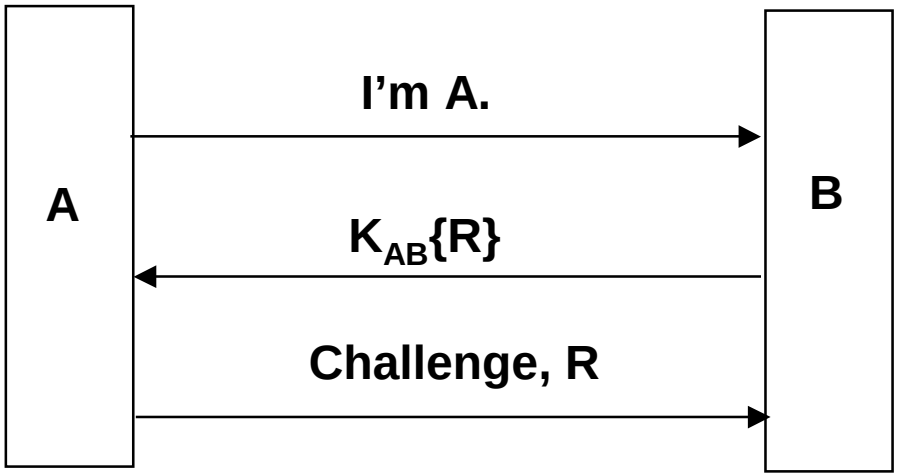


K_{AB} : Shared secret key between A and B.

Risks

- Not mutual authentication
- Off-line password guessing attack
- Some who reads B's database can later impersonate A.

Shared secret(II)



K_{AB} : Shared secret key between A and B.

Risks

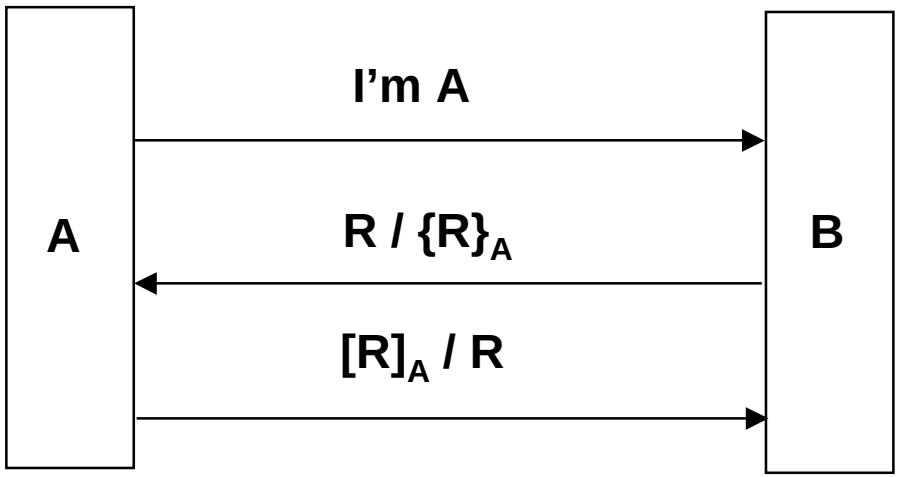
If R is recognizable quantity, password guessing attack is possible

Shared secret(III)

B authenticates A based on synchronized clocks and a shared secret

B authenticates A based on high resolution time and a shared secret

Public Key



B authenticates A based on her public key signature.
B authenticates A if she can decrypt a message encrypted with her public key
$[R]_A$: A signs R with private key.

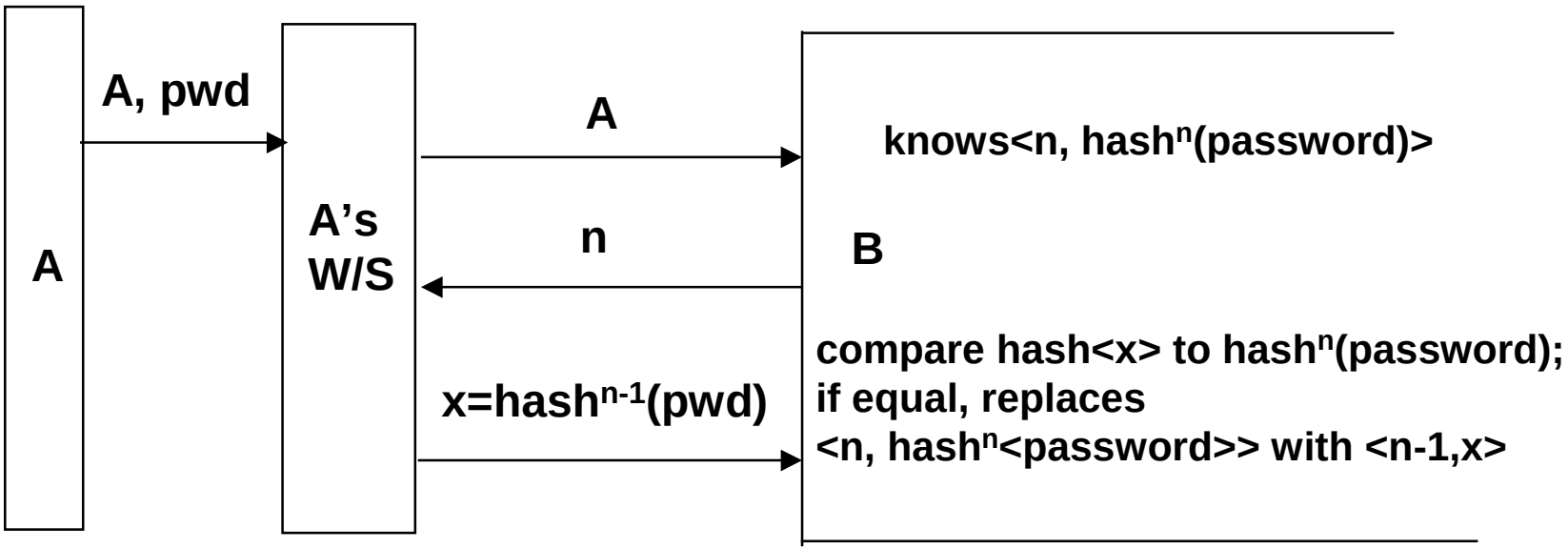
Risk : man-in-the middle attack

Lamport's hash(I)

- **A remembers passwd**
- **B has DB for eash user**
  - **username**
  - **n, an integer which decrements each time B authenticates the user. (최) n=1000**
  - **$hash^n$(passwd) i.e., hash(hash..hash(passwd)…))**
- **Risks**
  - **password access in system DB**
  - **eavesdropping communication line**
  - **revelation of password by careless user**

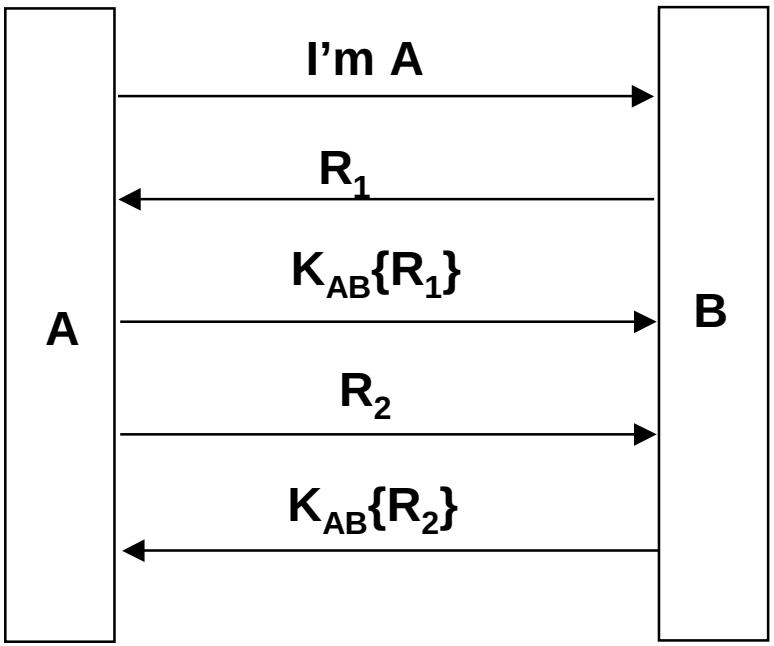
* L. Lamport, "Password Authentication with Insecure Channel",Comm. of the ACM, pp. 770-772, No.11, Vol.24, Nov., 1981

Lamport's hash(II)



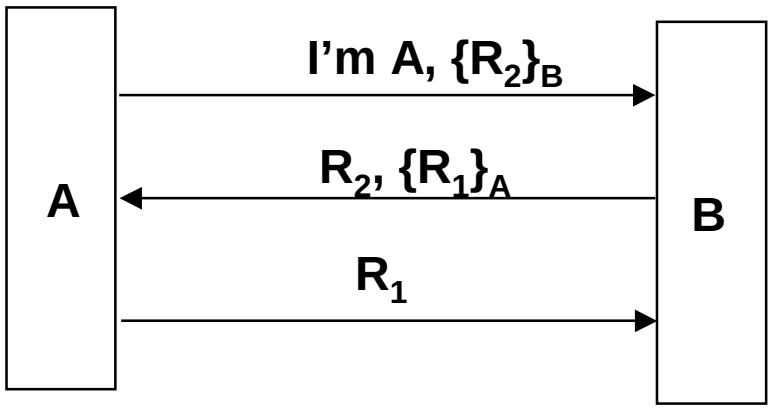
- Solving Encryption and integrity together :
  use password||salt instead of password only -> advance to S/KEY
- No mutual authentication

Mutual authentication(I)



- Mutual authentication based on shared secret, K_{AB}
- Risk of simplified 3-pass version (Protocol 9-9)
  - Man-in-the-middle attack (reflection attack)
  - password guessing

Mutual authentication(II)



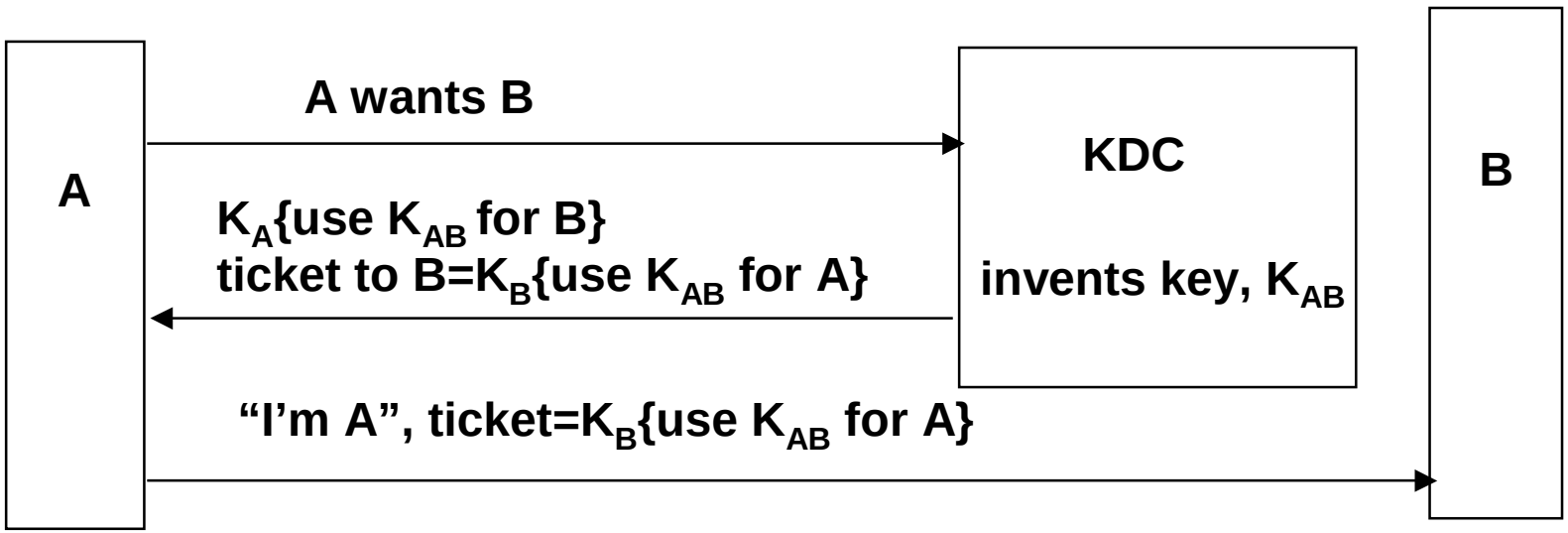
Mutual authentication with public keys assuming that A and B know each other's public keys.

Mediated Authentication(I)

KDC operation (in principle)

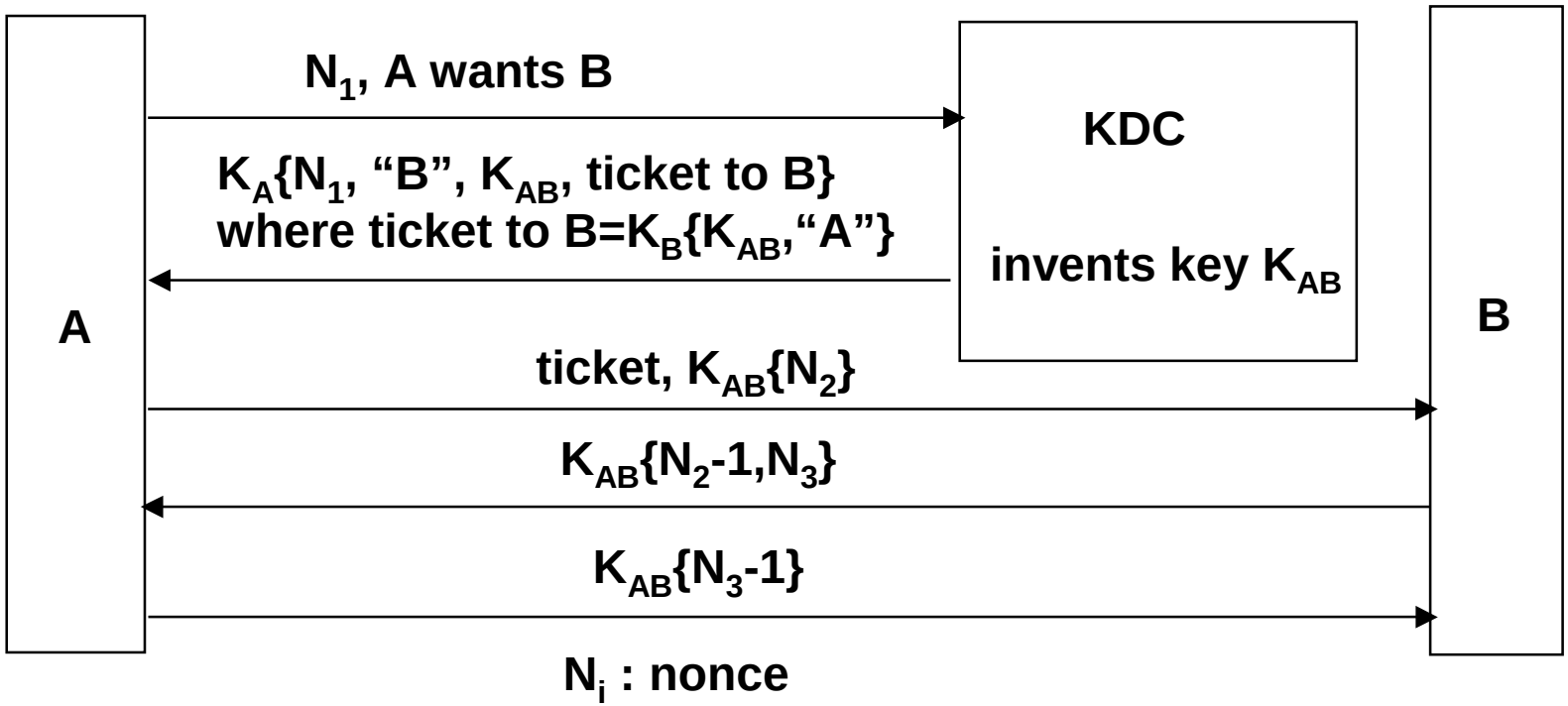
*** anyone can impersonate A**

Mediated Authentication(II)



KDC operation (in practice)

Needham-Schroeder



R.G.Needham and M.D. Schroeder, "Using encryption for authentication in large networks of computers", Comm. of the ACM, pp.993-999, Vol.21, No.12,Dec. 1978

Nonce

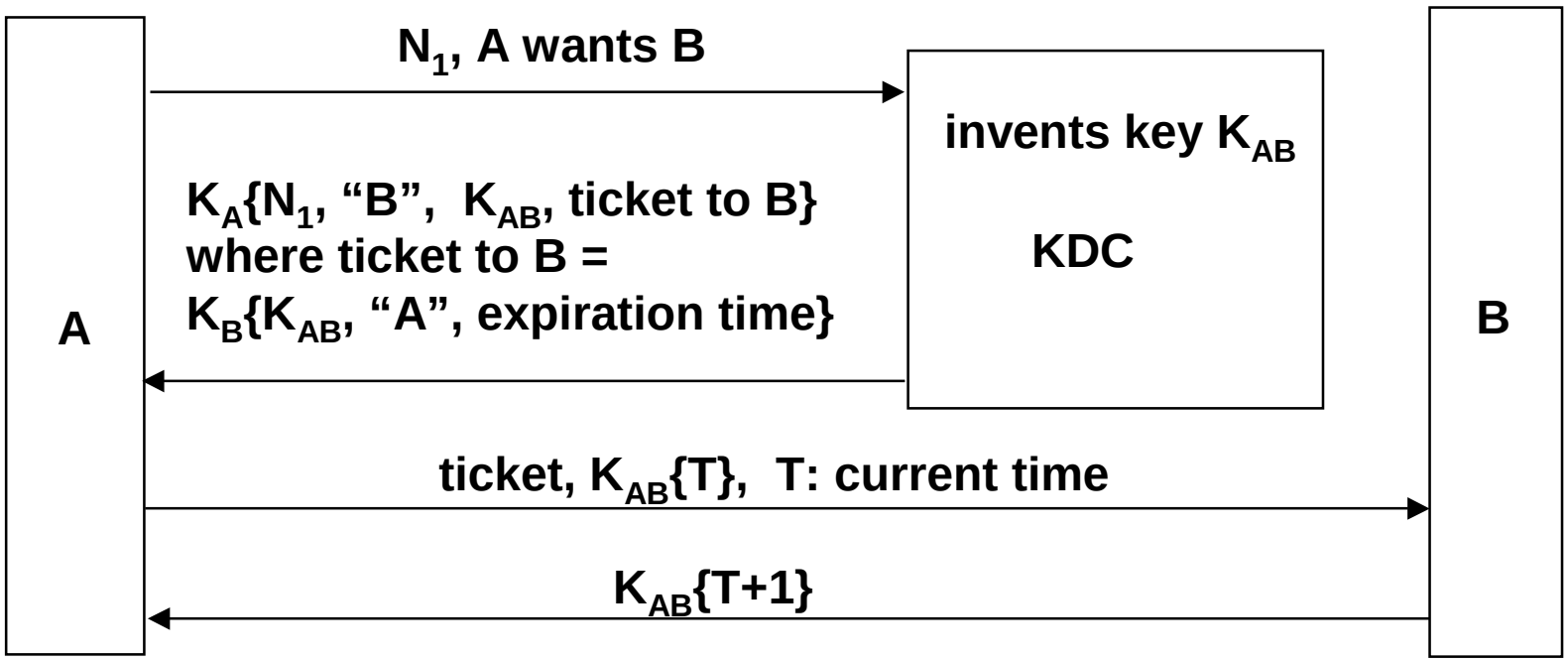
a number use only once

- **timestamp**
  - **synchronized clocks**
  - **guessable**
  - **set clock back**
- **sequence number**
  - **guessable**
  - **requires state**
- **large random number**

Others

- Extension of Needham-Schroeder : p.247
- Otway-Rees : p.248
- Bellovin-Meritt : p.250, p.252
- Kerberos : p.249

Kerberos


A
N_1, A wants B
invents key K_{AB}
KDC
B
$K_A\{N_1$, "B", K_{AB}, ticket to B$\}$
where ticket to B =
$K_B\{K_{AB}$, "A", expiration time$\}$
ticket, $K_{AB}\{T\}$, T: current time
$K_{AB}\{T+1\}$

Performance of protocol

- No. of cryptographic operations using a private key
- No. of cryptographic operations using a public key
- No. of bytes encrypted or decrypted using a secret key
- No. of bytes to be cryptographically hashed
- No. of message transmitted