6.033 Computer System Engineering
Spring 2009

1) Authentication
2) Authorization
3) Confidentiality

Sign, Verify

$\mathbf{sign(m,\ k_1)\ =\ sig}$

$\mathbf{verify(m,\ sig,\ k_2)}$

Encrypt, Decrypt

$\mathbf{enc(m,\ k_1)\ \rightarrow\ c}$

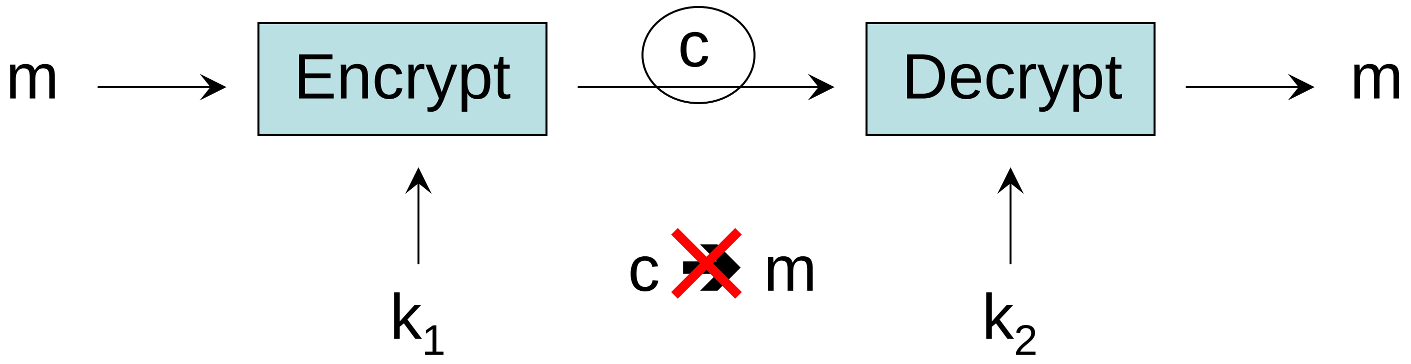
$\mathbf{dec(c,\ k_2)\ \rightarrow\ m}$

Secure Comm. Channel

- use pub key to exchange a shared key
- use shared key to enc. comm

1) freshness
2) appropriateness ←
3) forward secrecy

Confidentiality



$c \not\rightarrow m$

$k_2, c \rightarrow m$

Confidentiality + Authentication

sign(encrypt(m, k_{conf}), k_{auth})

Authenticate

sign(m, k_{auth})

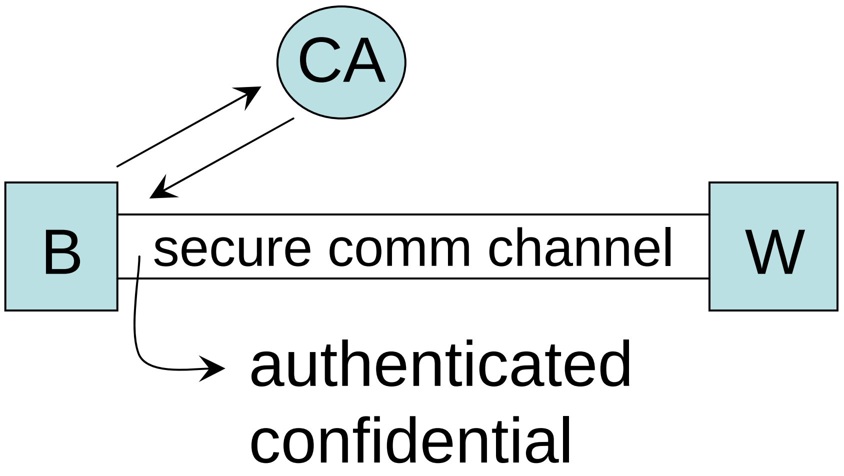
freshness – (e.g. T)

add timestamp to m

appropriateness

add context

Example: Web



Q. How does W know that B is authorized to access W?

(3) Authorization Functions

1) Rendezvous (setup)
2) Verification (mediate)
3) Revoke

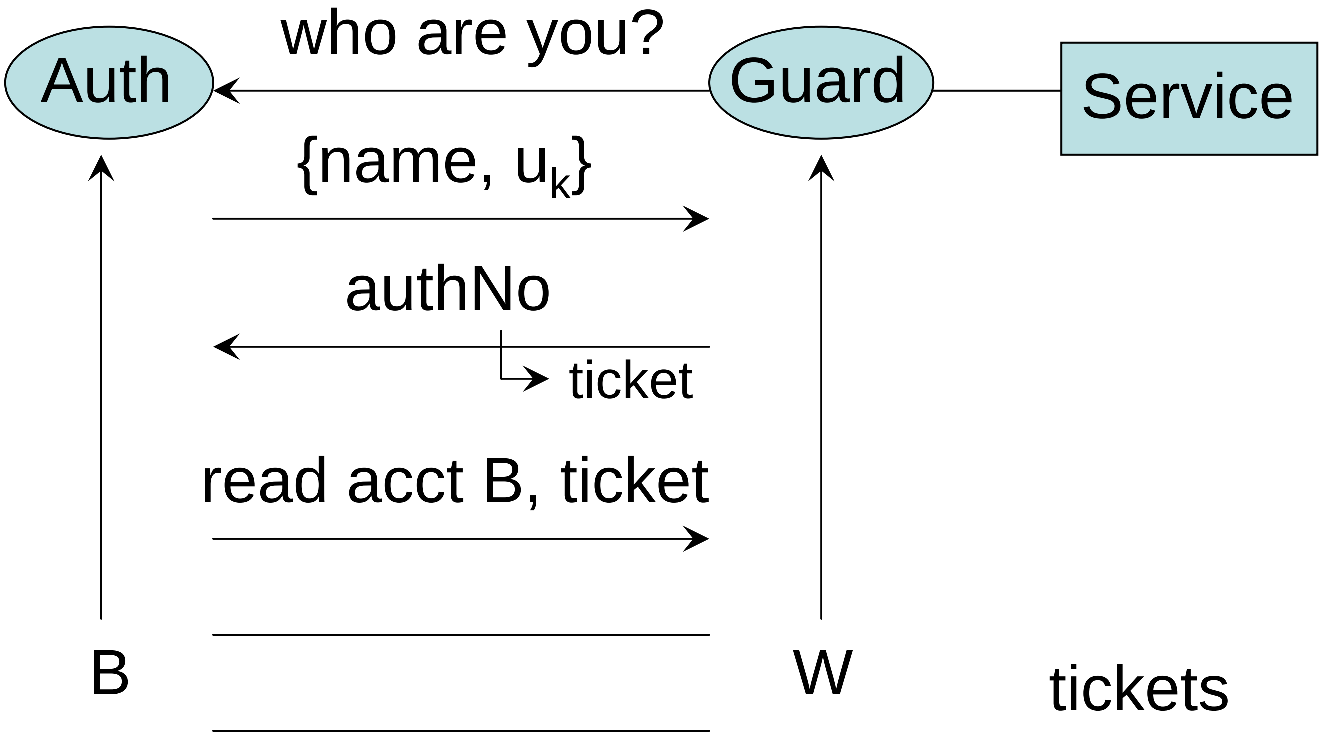
Authorization

ACL
access control lists

hard to
guess #

capabilities

	Lists	**Tickets**
Setup	add to list	generate ticket
Mediate	search list, check credentials	table lookup
Revoke	remove from list	invalidate ticket



Auth

who are you?

Guard

Service

{name, u_k}

authNo

ticket

read acct B, ticket

B

W

Cookie

```
user,
timeout
hash(
  user,
  timeout,
  random #)
```

tickets

t	resource
authNo	Acct B