1) Authentication
2) Authorization
3) Confidentiality

Sign, Verify

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

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

Encrypt, Decrypt

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

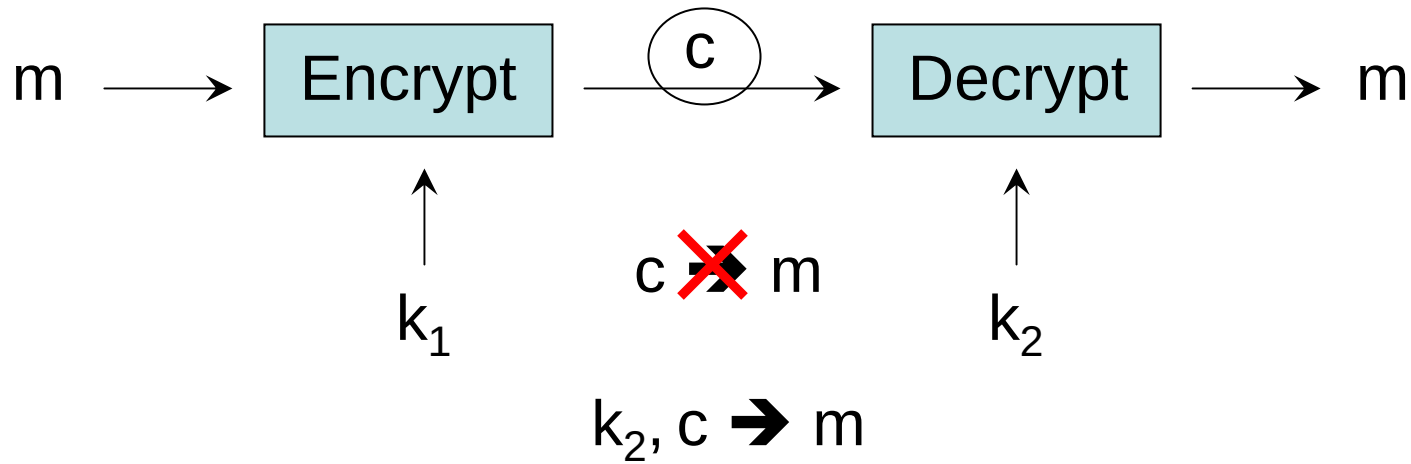
$\mathbf{dec(c, k_2) \rightarrow m}$

Secure Comm. Channel

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

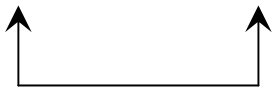
1) freshness
2) appropriateness ←
3) forward secrecy

Confidentiality



Confidentiality + Authentication

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



Authenticate

sign(m, k_{auth})

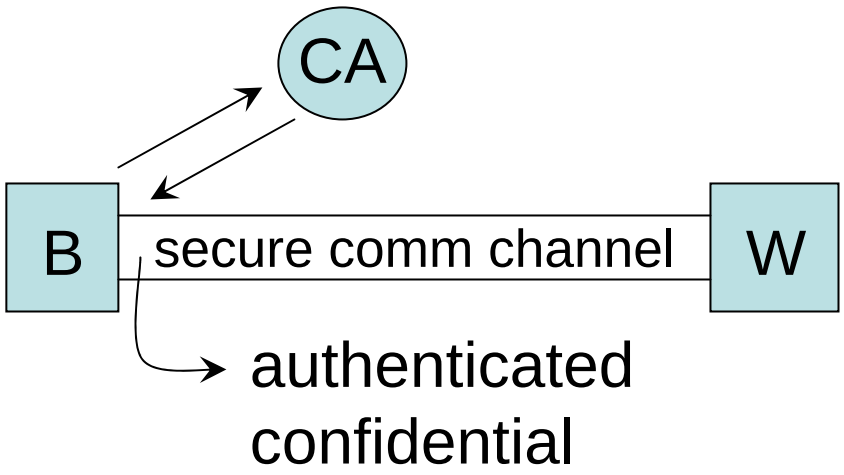
freshness – (e.g. T)

add timestamp to m

appropriateness

add context

Example: Web

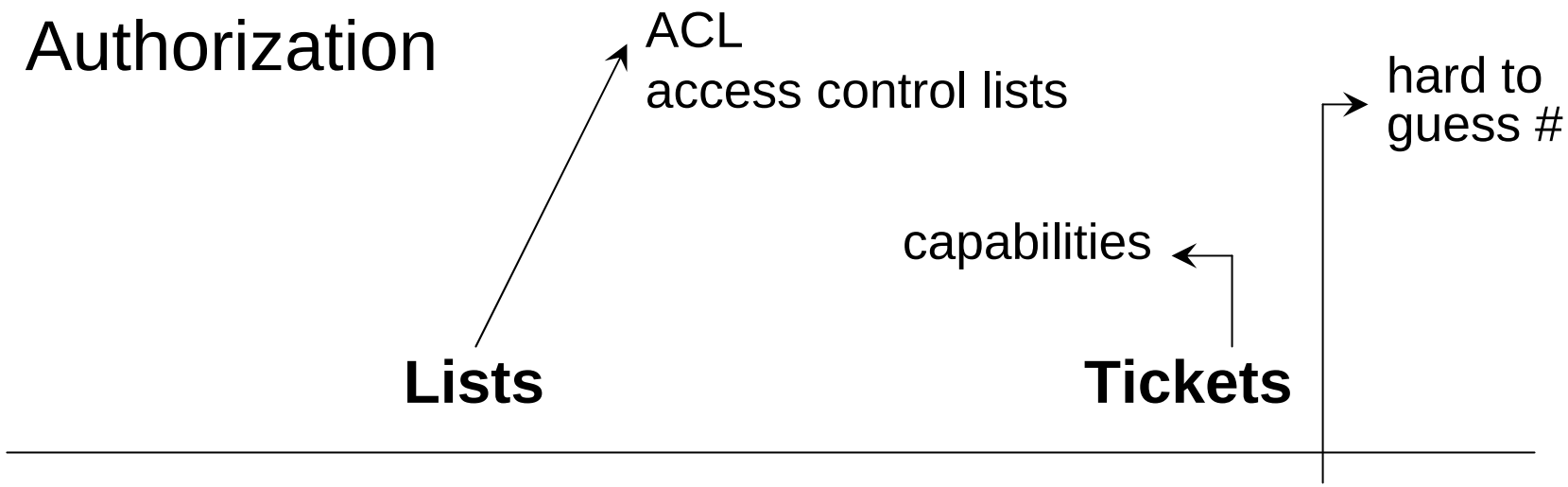


Q. How does W know that B is authorized to access W?

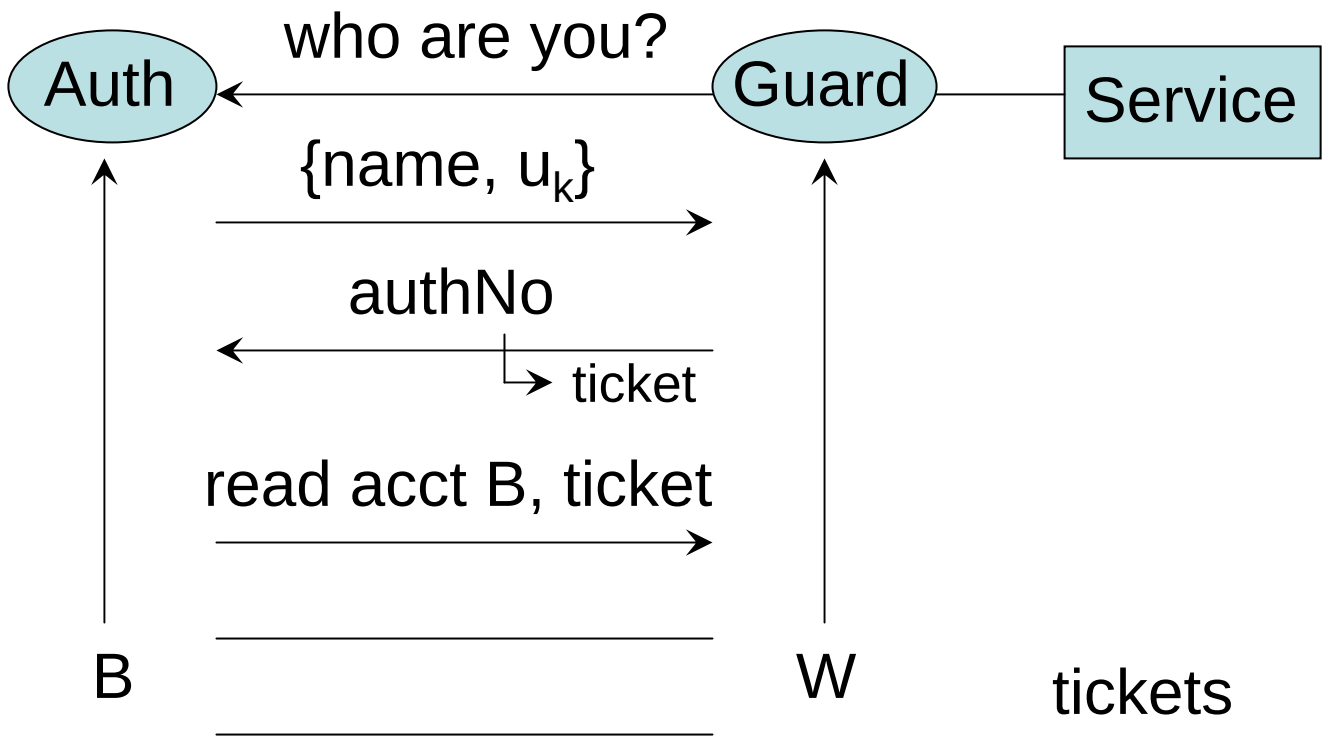
(3) Authorization Functions

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

Authorization



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



Cookie

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

tickets

t	resource
authNo	Acct B