DNS, DNSSEC

dog network service

slides
bit.ly/cs161-disc

feedback
bit.ly/abhifeedback

hack of the day

hack of the day

- DNS resolver insecurities found on thousands of websites out of 7000 sampled

hack of the day

- DNS resolver insecurities found on thousands of websites out of 7000 sampled
  - small/big businesses, governmental services

hack of the day

- DNS resolver insecurities found on thousands of websites out of 7000 sampled
  - small/big businesses, governmental services
  - "...at least 25 were using static source ports... thousands of more domains using static source ports were discovered."

hack of the day

- DNS resolver insecurities found on thousands of websites out of 7000 sampled
  - small/big businesses, governmental services
  - "...at least 25 were using static source ports... thousands of more domains using static source ports were discovered."
  - none of the 25 used DNSSEC, etc.

hack of the day

- DNS resolver insecurities found on thousands of websites out of 7000 sampled
  - small/big businesses, governmental services
  - "...at least 25 were using static source ports... thousands of more domains using static source ports were discovered."
  - none of the 25 used DNSSEC, etc.
  - cache poisoning used to hijack WordPress

general questions, concerns, etc.

domain name system (DNS)

domain name system (DNS)

- remember: computers use IP addresses to communicate

domain name system (DNS)

- remember: computers use IP addresses to communicate
- but humans use domain names (google.com)

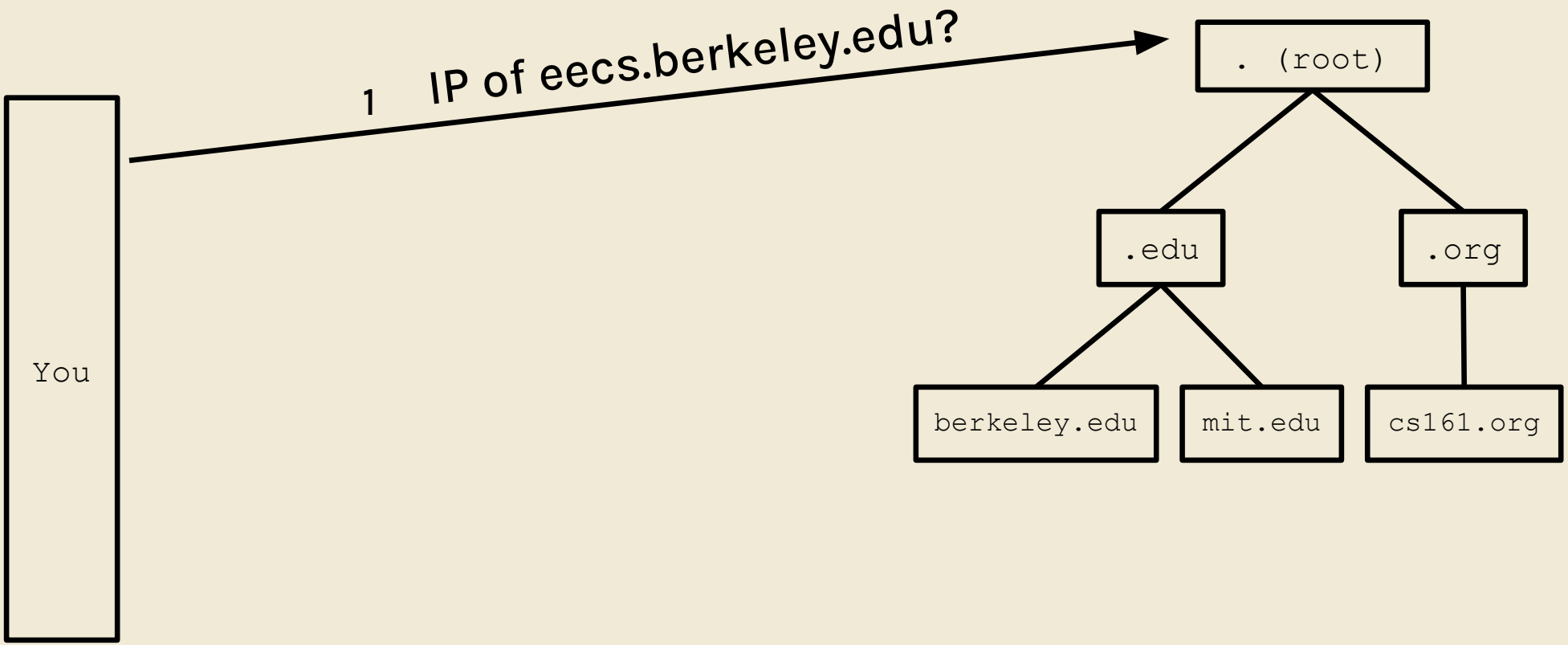
domain name system (DNS)

- remember: computers use IP addresses to communicate
- but humans use domain names (google.com)
- domain name -> IP address?

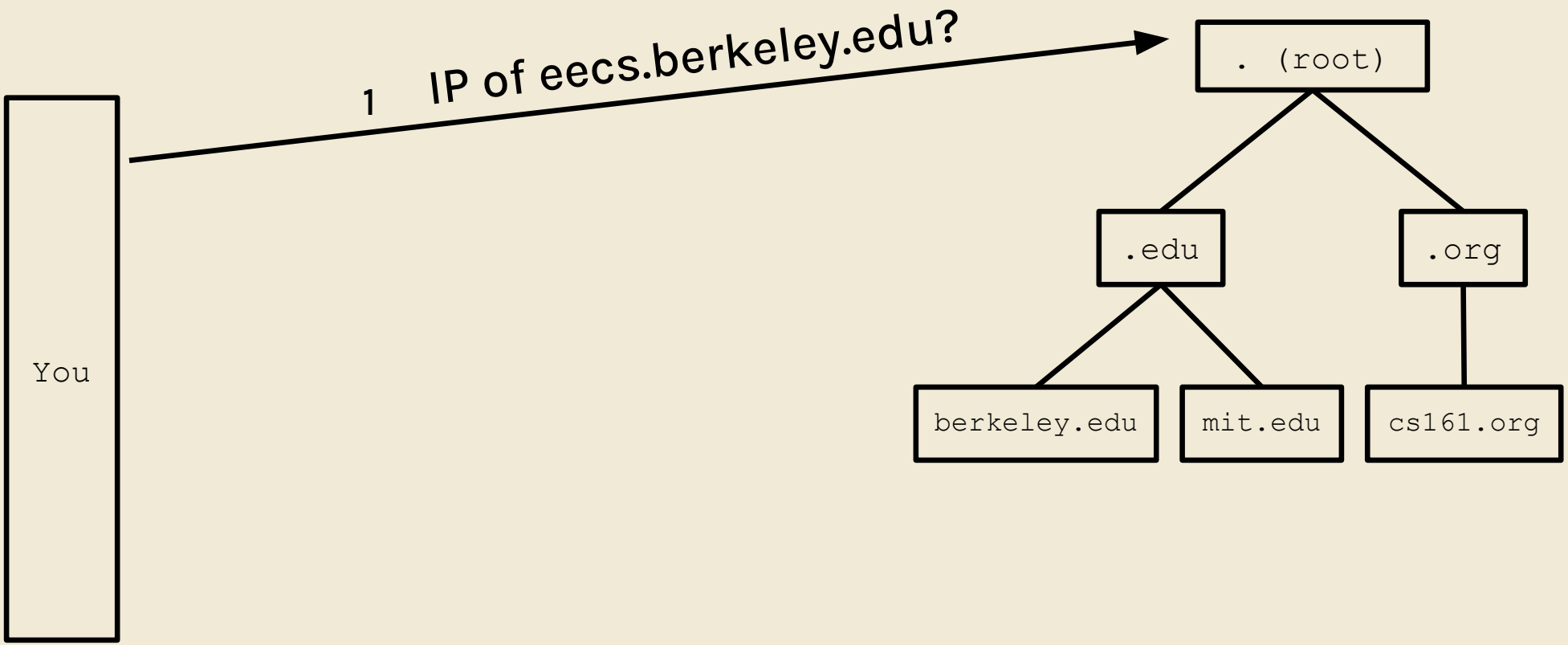
name servers

- a server on the Internet responsible for answering DNS queries
  - “what is the IP of google.com?”
- use a hierarchy of nameservers to find the IP

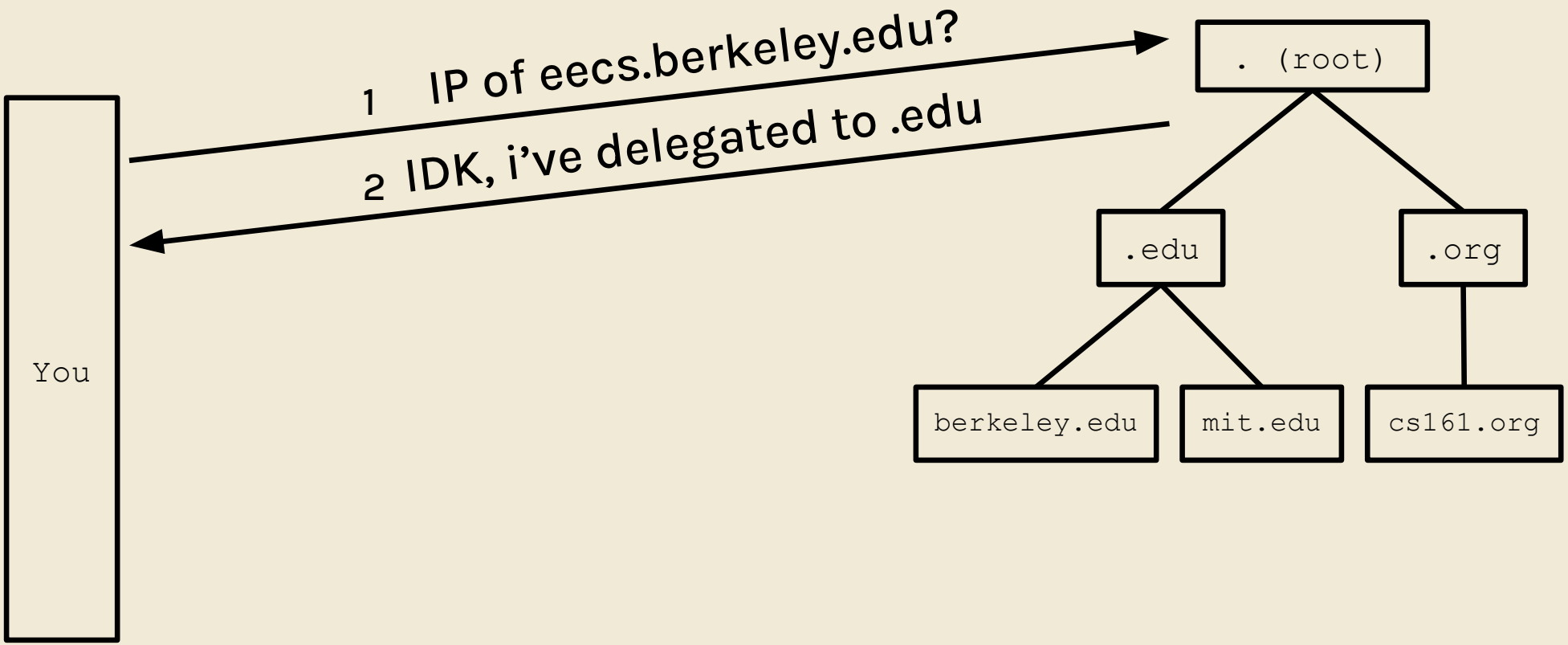
name servers



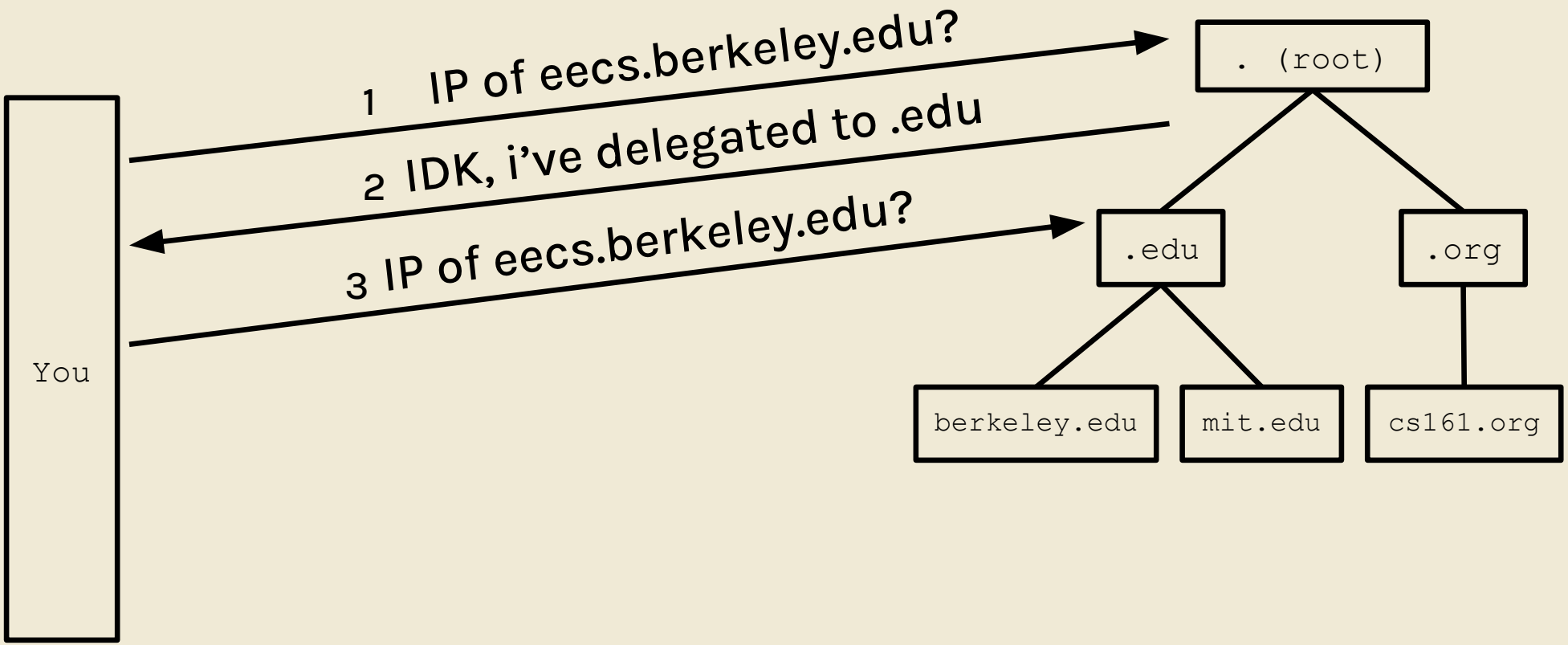
name servers



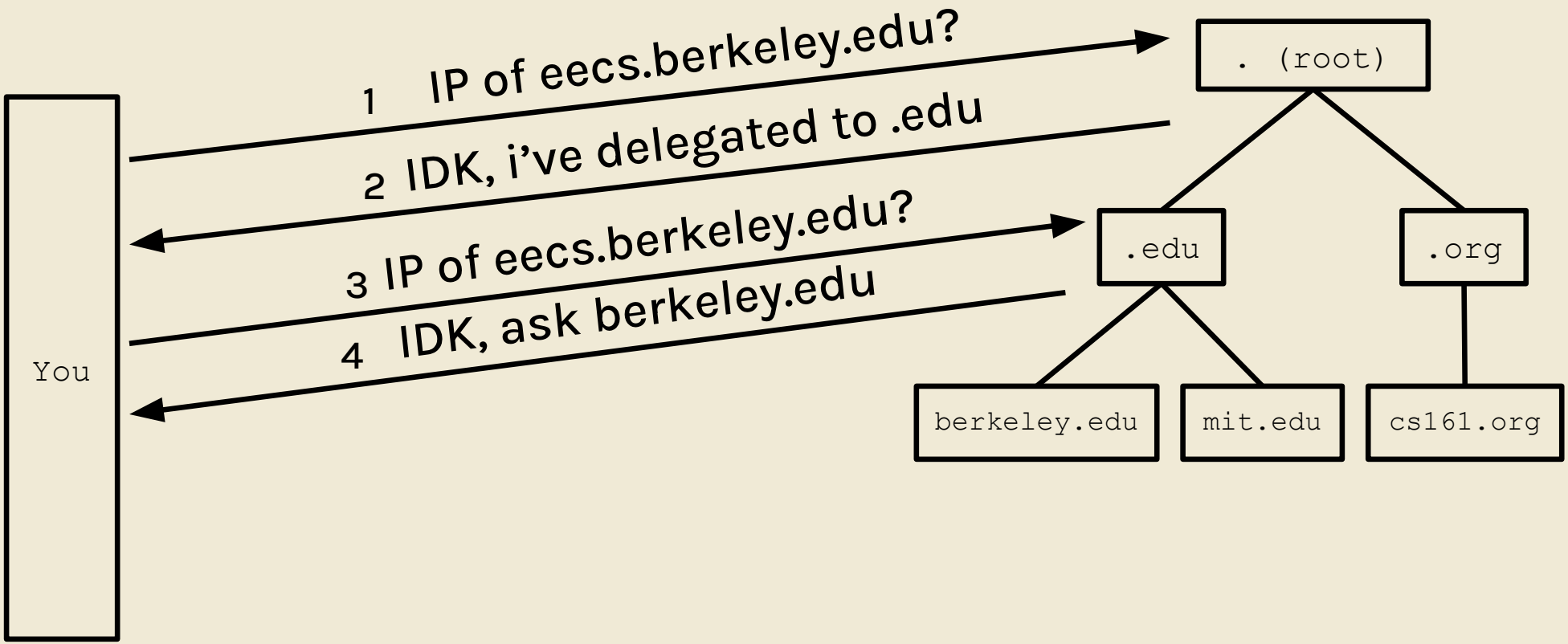
name servers



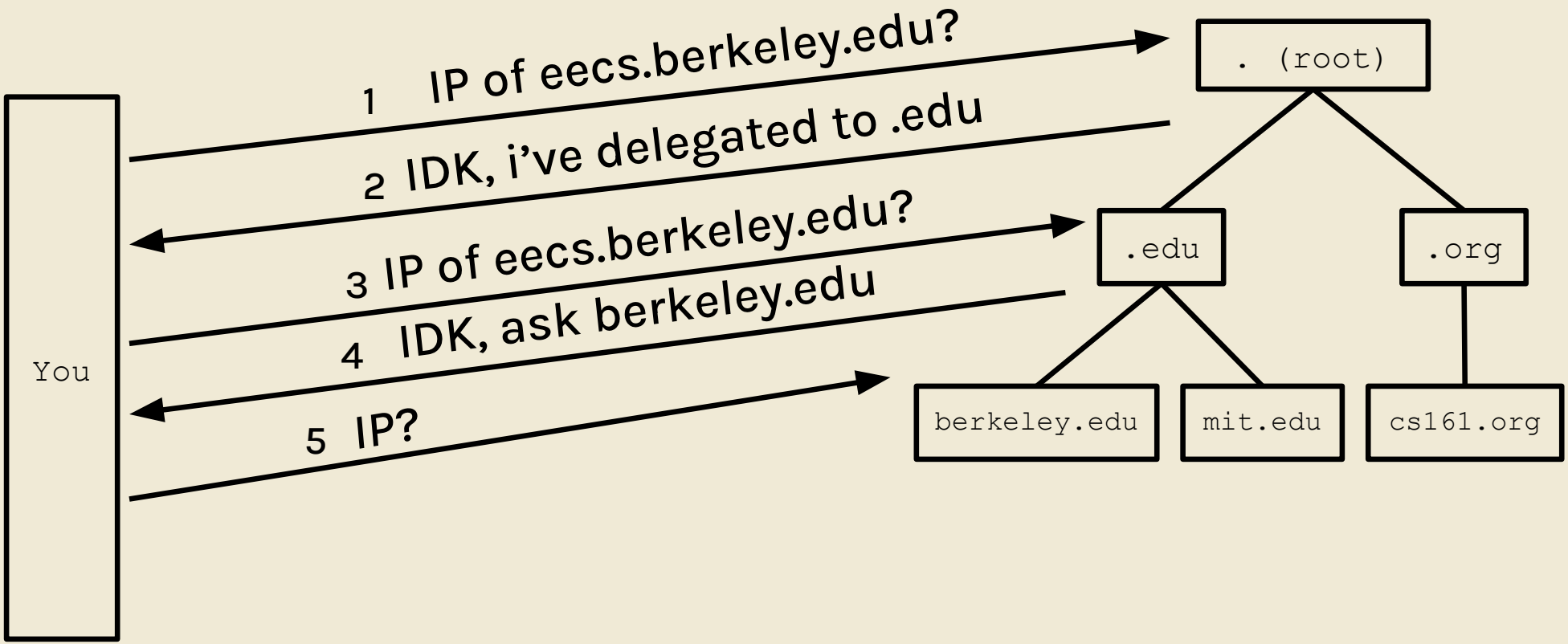
name servers



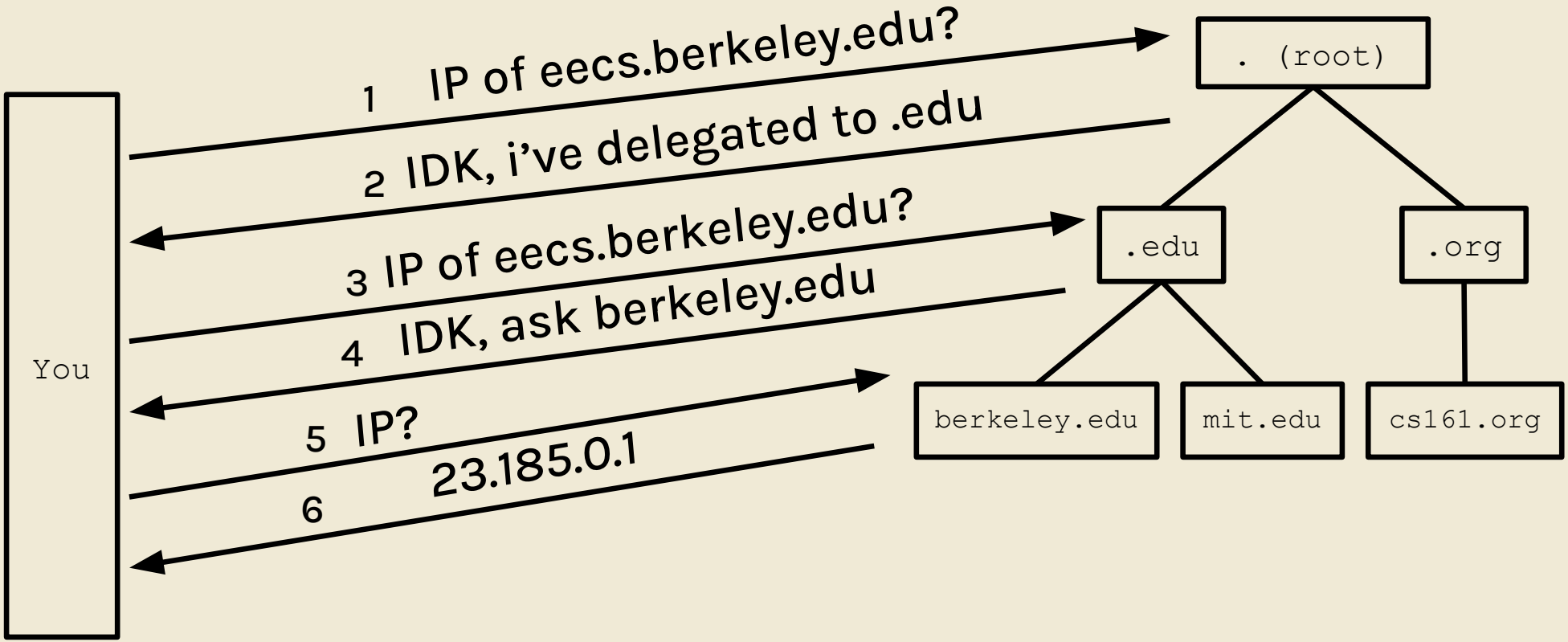
name servers



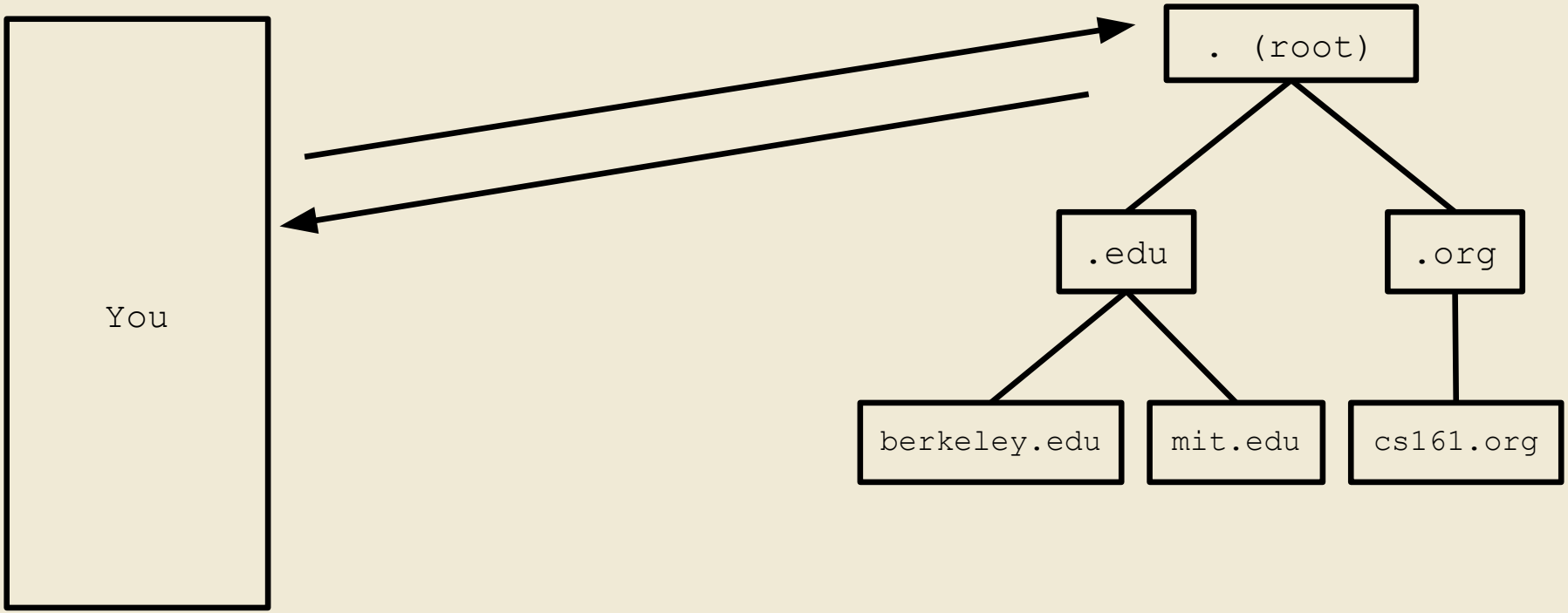
name servers



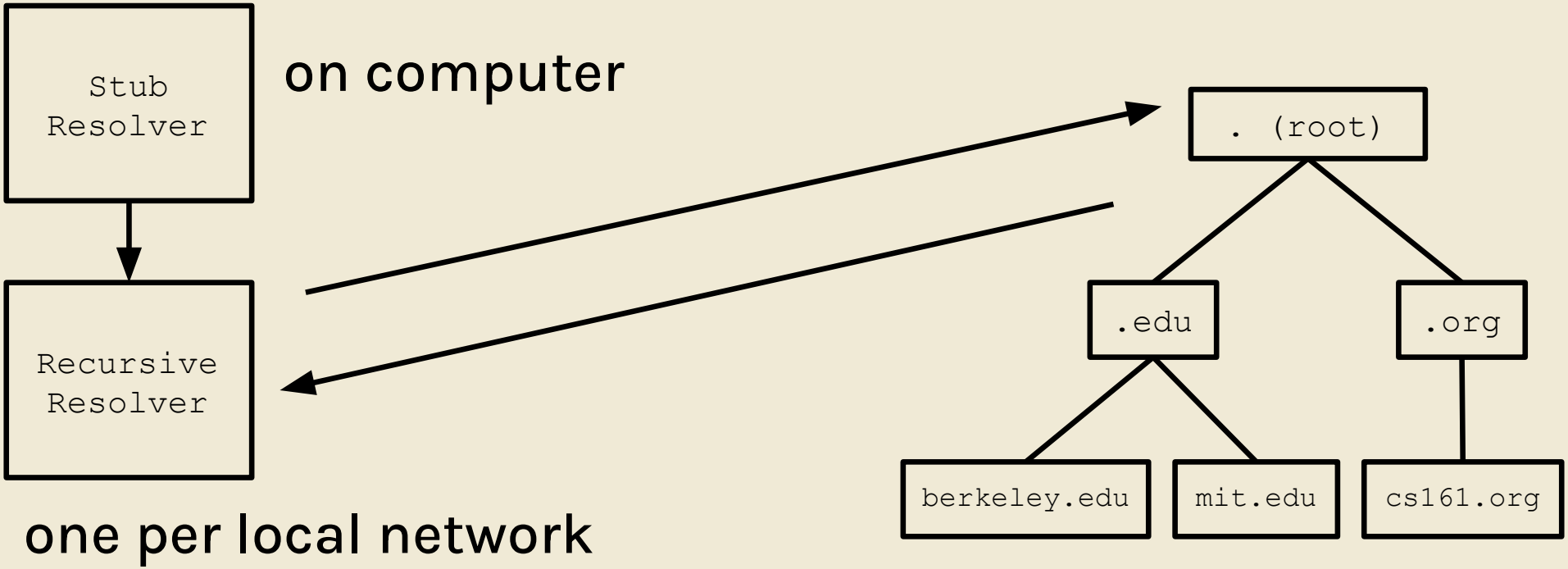
name servers



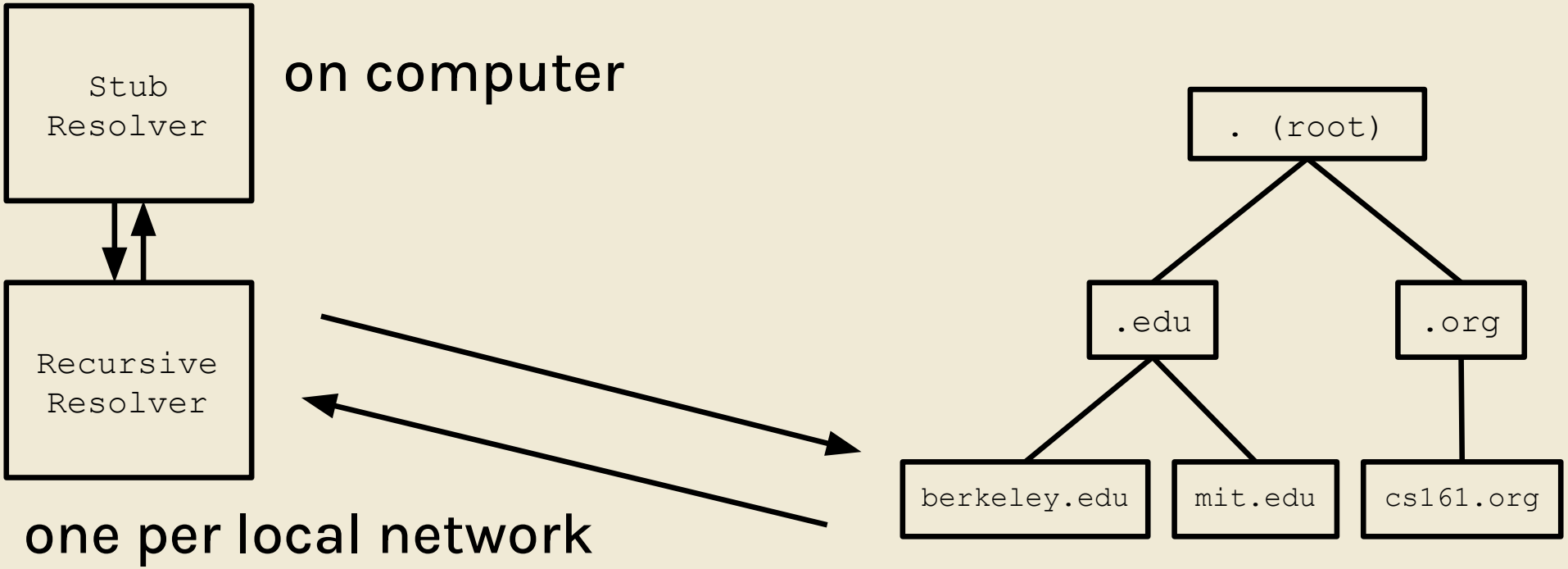
stub/recursive resolvers



stub/recursive resolvers



stub/recursive resolvers

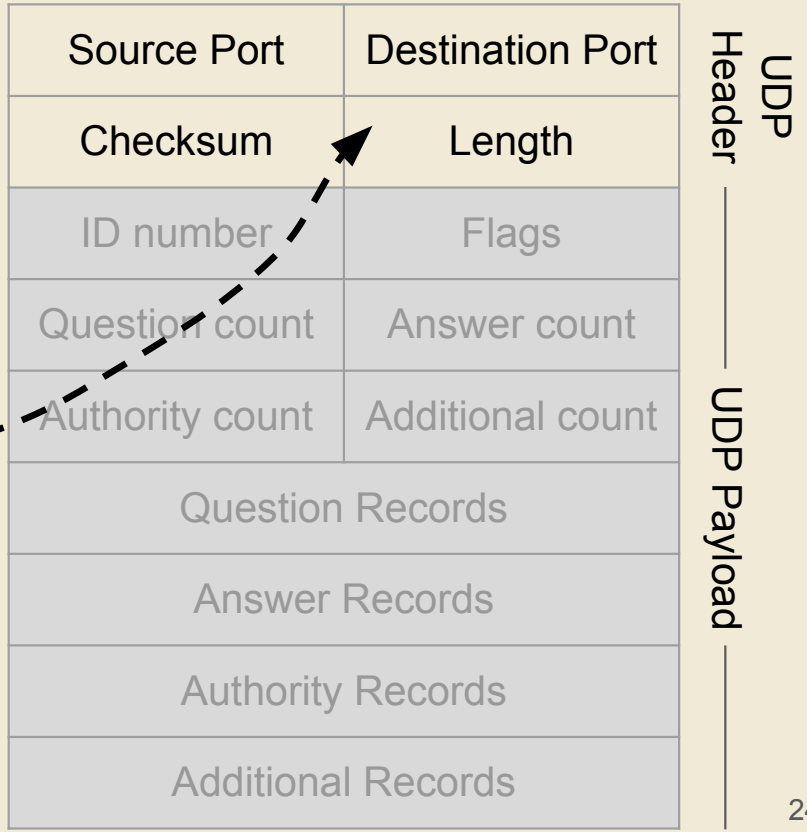


DNS packets

16-bit random

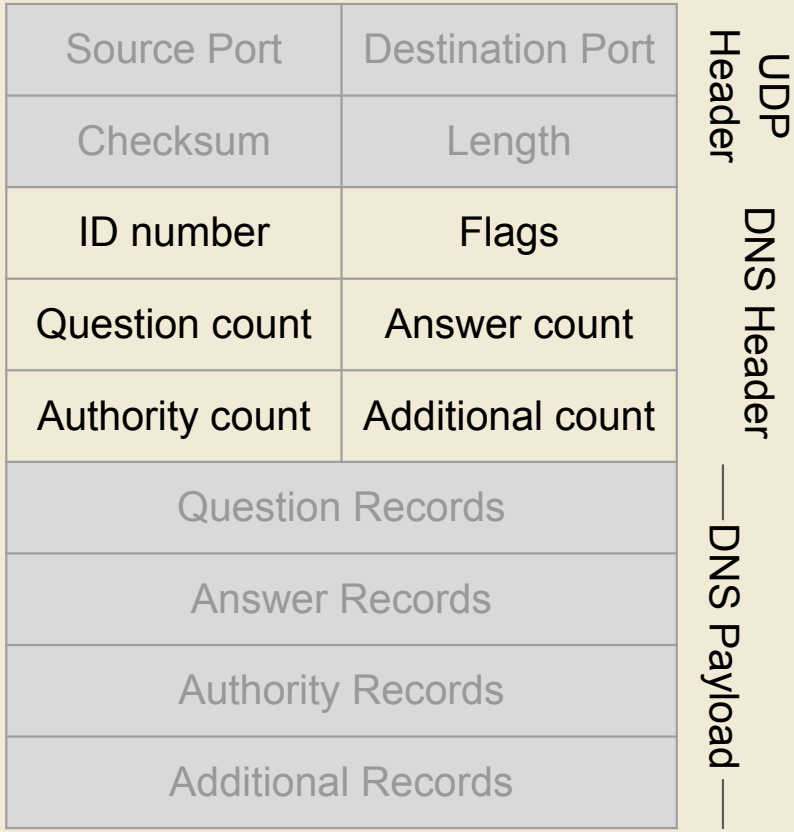
check for corruption

16 bits, usually 53

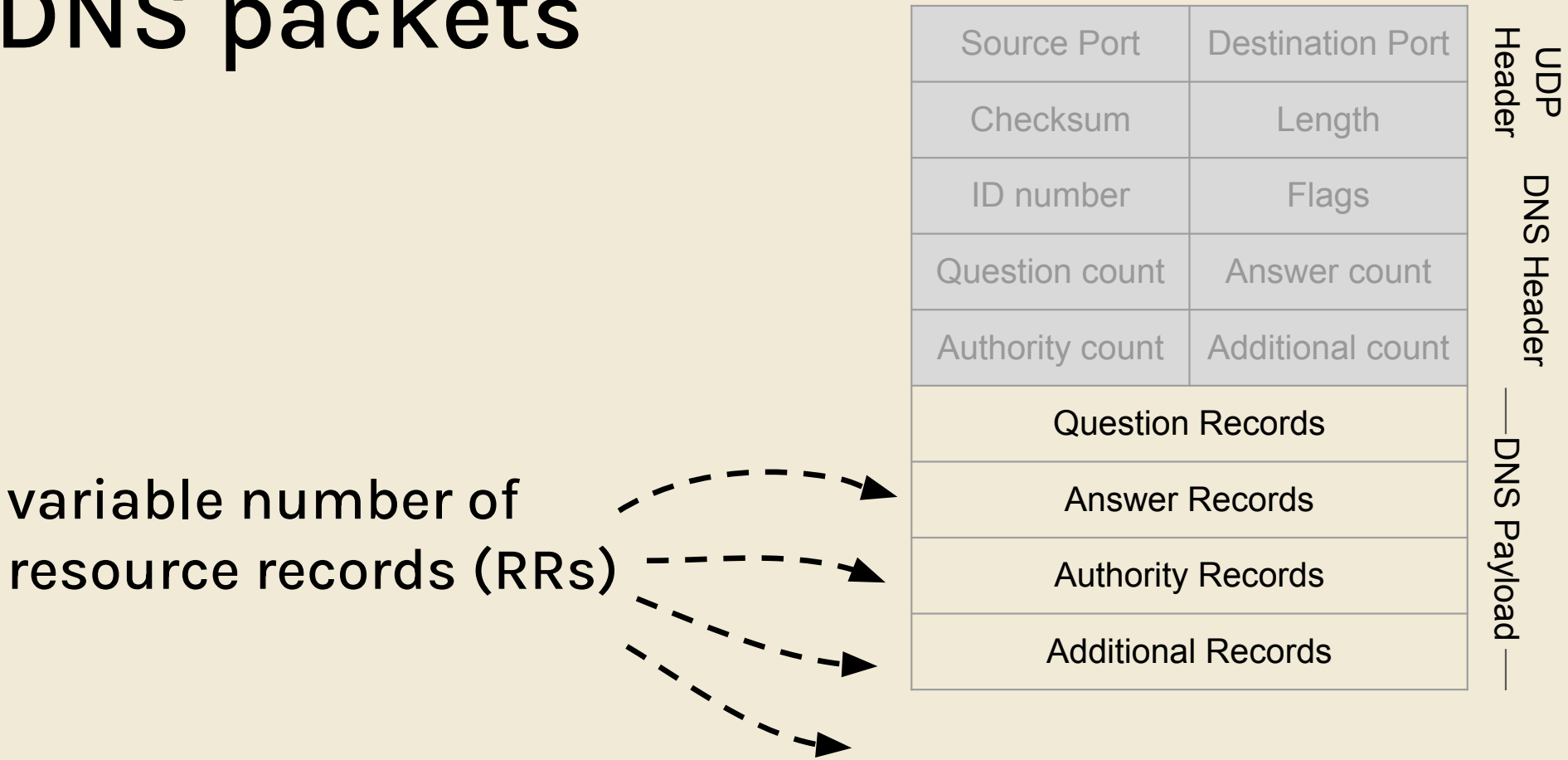


DNS packets

query ID same as response ID



DNS packets


Source Port
Destination Port
Checksum
Length
ID number
Flags
Question count
Answer count
Authority count
Additional count
Question Records
Answer Records
Authority Records
Additional Records
UDP Header
DNS Header
DNS Payload
variable number of resource records (RRs)

DNS lookup walkthrough

```
$ dig +norecurse eecs.berkeley.edu @198.41.0.4

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26114
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 27

;; QUESTION SECTION:
;eecs.berkeley.edu.          IN    A

;; AUTHORITY SECTION:
edu.                172800   IN    NS    a.edu-servers.net.
edu.                172800   IN    NS    b.edu-servers.net.
edu.                172800   IN    NS    c.edu-servers.net.
...

;; ADDITIONAL SECTION:
a.edu-servers.net.  172800   IN    A      192.5.6.30
b.edu-servers.net.  172800   IN    A      192.33.14.30
c.edu-servers.net.  172800   IN    A      192.26.92.30
...
```

pick any one (multiple for redundancy)

this NS record says that a.edu-servers.net is a .edu name server.

DNS lookup walkthrough

```
$ dig +norecurse eecs.berkeley.edu @198.41.0.4

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26114
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 27

;; QUESTION SECTION:
;eecs.berkeley.edu.          IN    A

;; AUTHORITY SECTION:
edu.                172800   IN    NS    a.edu-servers.net.
edu.                172800   IN    NS    b.edu-servers.net.
edu.                172800   IN    NS    c.edu-servers.net.
...

;; ADDITIONAL SECTION:
a.edu-servers.net.  172800   IN    A     192.5.6.30
b.edu-servers.net.  172800   IN    A     192.33.14.30
c.edu-servers.net.  172800   IN    A     192.26.92.30
...
```

next nameserver IP to contact

DNS lookup walkthrough

```
$ dig +norecurse eecs.berkeley.edu @192.5.6.30

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36257
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 5

;; QUESTION SECTION:
;eecs.berkeley.edu.            IN    A

;; AUTHORITY SECTION:
berkeley.edu.         172800   IN    NS    adns1.berkeley.edu.
berkeley.edu.         172800   IN    NS    adns2.berkeley.edu.
berkeley.edu.         172800   IN    NS    adns3.berkeley.edu.

;; ADDITIONAL SECTION:
adns1.berkeley.edu.   172800   IN    A     128.32.136.3
adns2.berkeley.edu.   172800   IN    A     128.32.136.14
adns3.berkeley.edu.   172800   IN    A     192.107.102.142
...
```

DNS lookup walkthrough

```
$ dig +norecurse eecs.berkeley.edu @192.5.6.30

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36257
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 5

;; QUESTION SECTION:
;eecs.berkeley.edu.            IN    A

;; AUTHORITY SECTION:
berkeley.edu.         172800   IN    NS    adns1.berkeley.edu.
berkeley.edu.         172800   IN    NS    adns2.berkeley.edu.
berkeley.edu.         172800   IN    NS    adns3.berkeley.edu.

;; ADDITIONAL SECTION:
adns1.berkeley.edu.   172800   IN    A     128.32.136.3
adns2.berkeley.edu.   172800   IN    A     128.32.136.14
adns3.berkeley.edu.   172800   IN    A     192.107.102.142
...
```

next nameserver IP to contact

DNS lookup walkthrough

```
$ dig +norecurse eecs.berkeley.edu @128.32.136.3

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52788
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;eecs.berkeley.edu.         IN   A

;; ANSWER SECTION:
eecs.berkeley.edu.  86400   IN   A   23.185.0.1
```

DNS lookup walkthrough

```
$ dig +norecurse eecs.berkeley.edu @128.32.136.3

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52788
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;eecs.berkeley.edu.          IN    A

;; ANSWER SECTION:
eecs.berkeley.edu.   86400    IN    A    23.185.0.1
```

one A type record in answer section: `eecs.berkeley.edu`'s IP is `23.185.0.1`.

attack: cache poisoning

```
$ dig +norecurse eecs.berkeley.edu @128.32.136.3

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52788
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;eecs.berkeley.edu.            IN    A

;; ANSWER SECTION:
eecs.berkeley.edu.    86400    IN    A    23.185.0.1

;; ADDITIONAL SECTION:
www.google.com.       172800   IN    A    6.6.6.6
```

We made a query to a malicious **berkeley.edu** name server…

…and it returned a malicious record for **www.google.com**!

attack: cache poisoning

```
$ dig +norecurse eecs.berkeley.edu @128.32.136.3

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52788
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;eecs.berkeley.edu.            IN    A

;; ANSWER SECTION:
eecs.berkeley.edu.    86400    IN    A     23.185.0.1

;; ADDITIONAL SECTION:
www.google.com.       172800   IN    A     6.6.6.6
```

We made a query to a malicious **berkeley.edu** name server…

…and it returned a malicious record for **www.google.com**!

now google.com has IP 6.6.6.6 in our cache! :(

defense: bailiwick checking

- **bailiwick checking**: the resolver only accepts records if they are in the name server's zone

defense: bailiwick checking

- **bailiwick checking**: the resolver only accepts records if they are in the name server's zone
  - ex: berkeley.edu NS can provide a record for eecs.berkeley.edu, but not mit.edu

defense: bailiwick checking

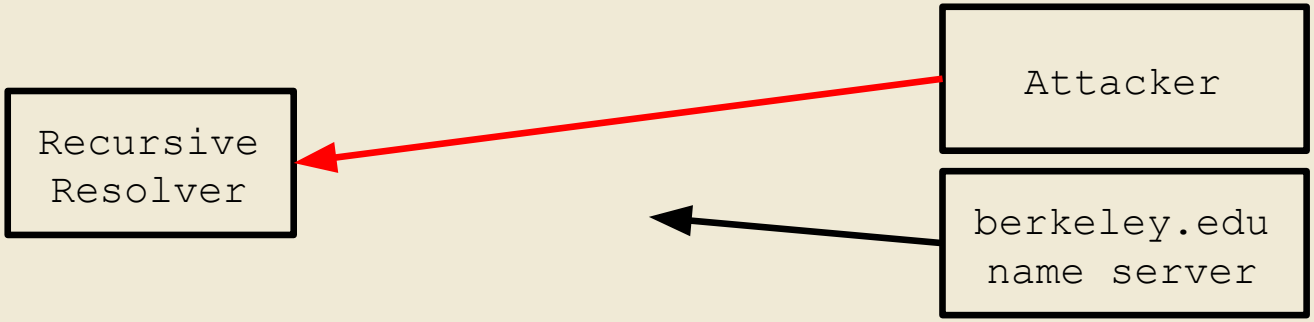
- **bailiwick checking**: the resolver only accepts records if they are in the name server's zone
  - ex: berkeley.edu NS can provide a record for eecs.berkeley.edu, but not mit.edu
  - ex: root name server can provide a record for any domain (everything is in bailiwick for the root)

attack: MITM, on-path

- <u>MITM:</u> can change DNS answer, poison cache

attack: MITM, on-path

- <u>MITM:</u> can change DNS answer, poison cache
- <u>on-path:</u> can spoof DNS response, no fields to guess



attack: MITM, on-path

- <u>MITM:</u> can change DNS answer, poison cache
- <u>on-path:</u> can spoof DNS response, no fields to guess
- <u>off-path:</u> need to guess 16-bit ID number

Kaminsky attack

- the attacker includes
  - <img src="http://fake1.google.com">
  - <img src="http://fake2.google.com">
  - <img src="http://fake3.google.com">
  - <img src="http://fake4.google.com">
- client makes request for domain name for each one
- off-path attacker can guess ID every time & poison

defenses (off-path)

- source port randomization (have to guess 32 bits)
- random domain query capitalization
- glue validation (don’t cache glue records)
  - not implemented by all DNS software

worksheet

(on 161 website)

DNSSEC

duh, no security sucks. err, cryptography?

securing DNS

securing DNS

- DNS over TLS?

securing DNS

- DNS over TLS?
  - slow

securing DNS

- DNS over TLS?
  - slow
  - doesn’t defend against malicious nameservers

securing DNS

- DNS over TLS?
  - slow
  - doesn’t defend against malicious nameservers
- we want integrity—no cache poisoning, tampering

securing DNS

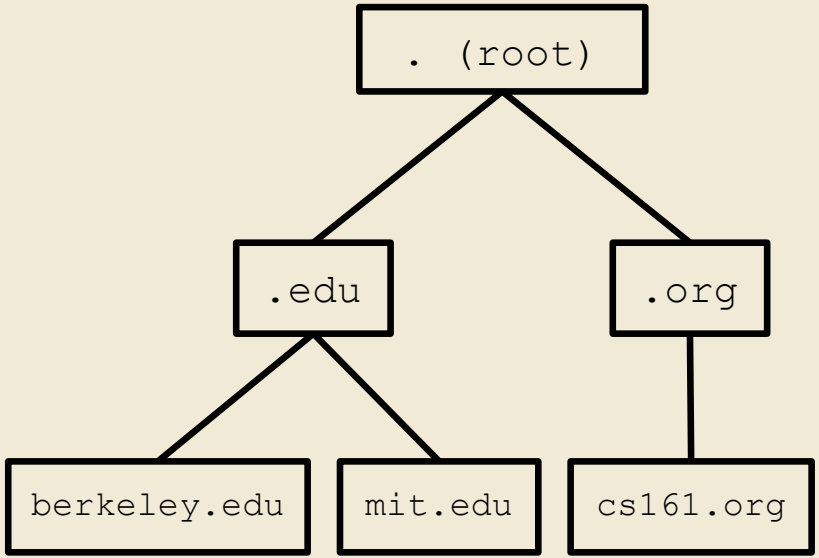
- DNS over TLS?
  - slow
  - doesn’t defend against malicious nameservers
- we want integrity—no cache poisoning, tampering
- we don’t need confidentiality
  - why?

securing DNS

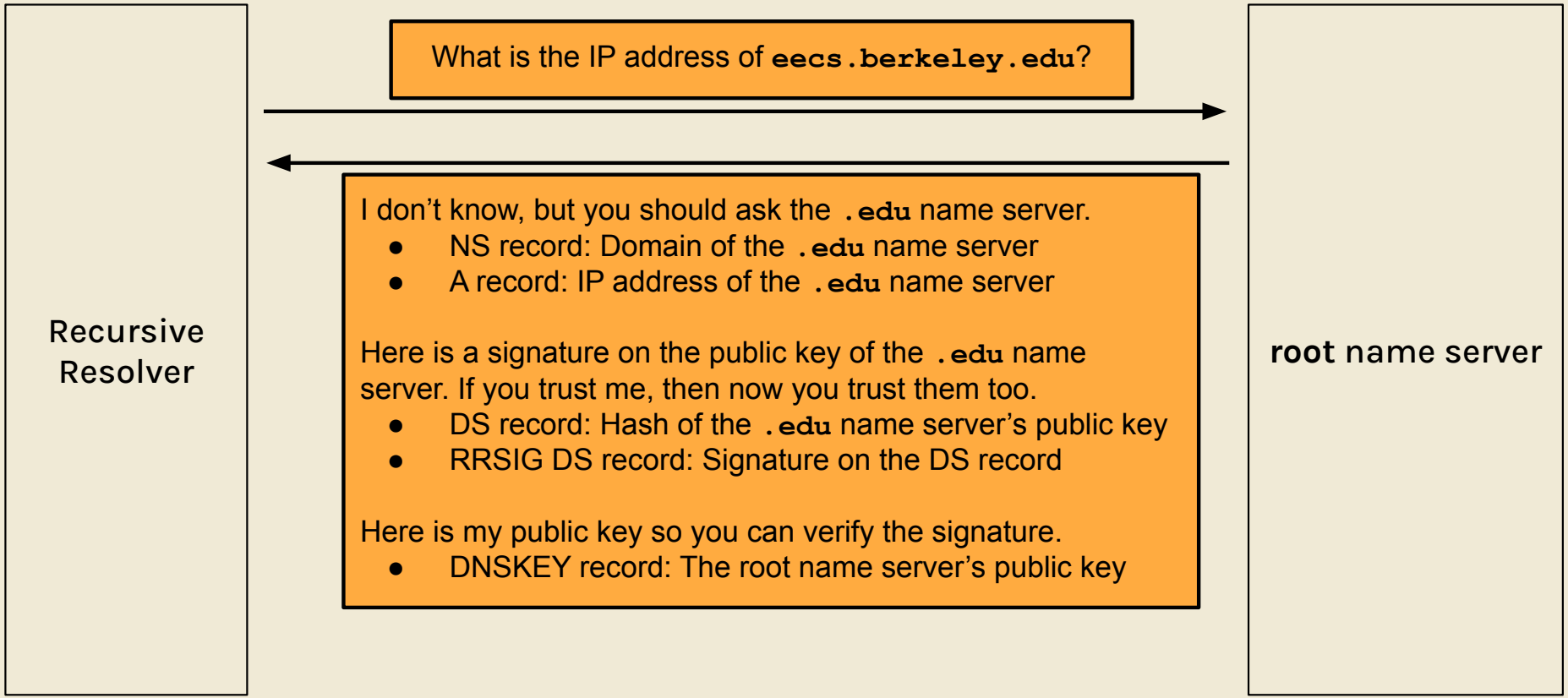
- DNS over TLS?
  - slow
  - doesn’t defend against malicious nameservers
- we want integrity—no cache poisoning, tampering
- we don’t need confidentiality
  - why?
  - anyone can make a DNS lookup, attackers can see IPs you connect to anyway

DNSSEC (DNS Security Extensions)

- delegate trust—parent signs child's public key



DNSSEC lookup



DNSSEC lookup

Recursive Resolver

What is the IP address of **`eecs.berkeley.edu`**?

I don't know, but you should ask the **`berkeley.edu`** name server.

- NS record: Domain of the **`berkeley.edu`** name server
- A record: IP address of the **`berkeley.edu`** name server

Here is a signature on the public key of the **`berkeley.edu`** name server. If you trust me, then now you trust them too.

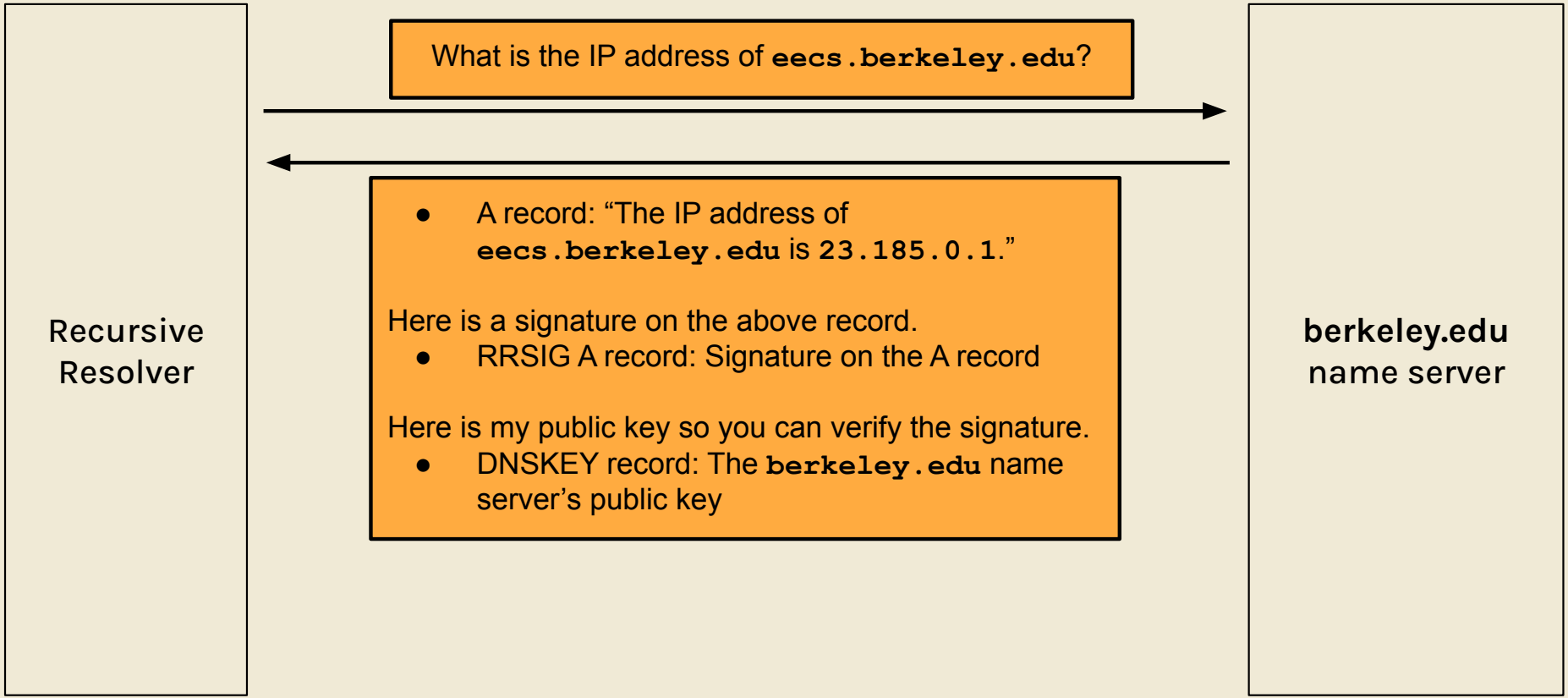
- DS record: Hash of the **`berkeley.edu`** name server's public key
- RRSIG DS record: Signature on the DS record

Here is my public key so you can verify the signature.

- DNSKEY record: The **`.edu`** name server's public key

.edu name server

DNSSEC lookup



nonexistent domains

- what do we do if no records exist for query?

nonexistent domains

- what do we do if no records exist for query?
- do we sign nothing and send it over?

nonexistent domains

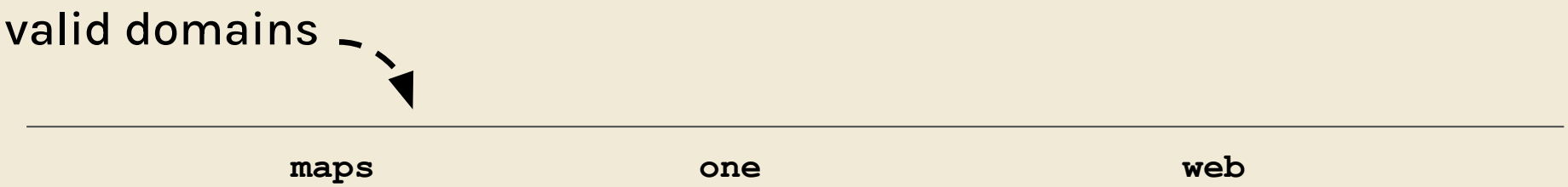
- what do we do if no records exist for query?
- do we sign nothing and send it over?
  - option 1: don’t authenticate NXDOMAIN
    - bad (anyone can spoof)

nonexistent domains

- what do we do if no records exist for query?
- do we sign nothing and send it over?
  - option 1: don’t authenticate NXDOMAIN
    - bad (anyone can spoof)
  - option 2: sign a NXDOMAIN response
    - signing in real time is slow

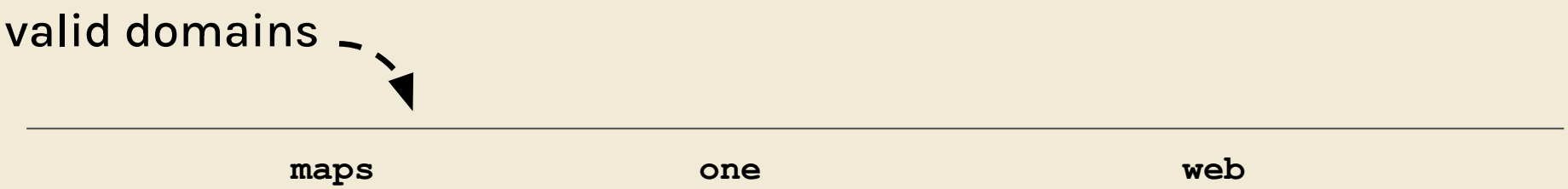
NSEC

- query for nonexistent.google.com



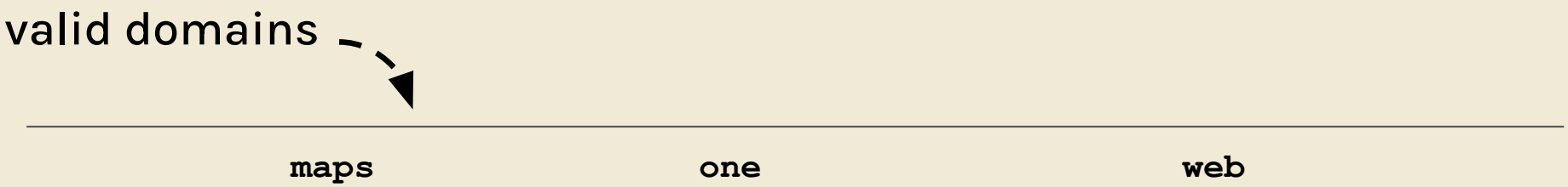
NSEC

- query for nonexistent.google.com
- response: “no domains exist between maps.google.com and one.google.com”
  - can pre-sign all pairs of adjacent domains



NSEC problem

- domain enumeration: can get every pair of domains by asking for arbitrary domains



NSEC3

NSEC3

- hash domain names and store pairs of adjacent hashes (instead of adjacent domain names)

NSEC3

- hash domain names and store pairs of adjacent hashes (instead of adjacent domain names)

NSEC3

- hash domain names and store pairs of adjacent hashes (instead of adjacent domain names)
- query for nonexistent.google.com hashes to d48678...

NSEC3

- hash domain names and store pairs of adjacent hashes (instead of adjacent domain names)
- query for nonexistent.google.com hashes to d48678...
  - NSEC3 response: “There exist no domains which hash to values between c612f3... and d810de...

hack of the day

- DNS resolver insecurities found on thousands of websites out of 7000 sampled
  - small/big businesses, governmental services
  - "...at least 25 were using static source ports... thousands of more domains using static source ports were discovered."
  - none of the 25 used DNSSEC, etc.
  - cache poisoning used to hijack WordPress

feedback

bit.ly/abhifeedback

slides: **bit.ly/cs161-disc**