How Sites Can Manage HTTPS When Users Don’t

Andy Ozment
MIT Lincoln Laboratory

Stuart Schechter
MIT Lincoln Laboratory

Rachna Dhamija
Harvard University

W3C Workshop
on Transparency & Usability of Web Authentication

16 March 2006

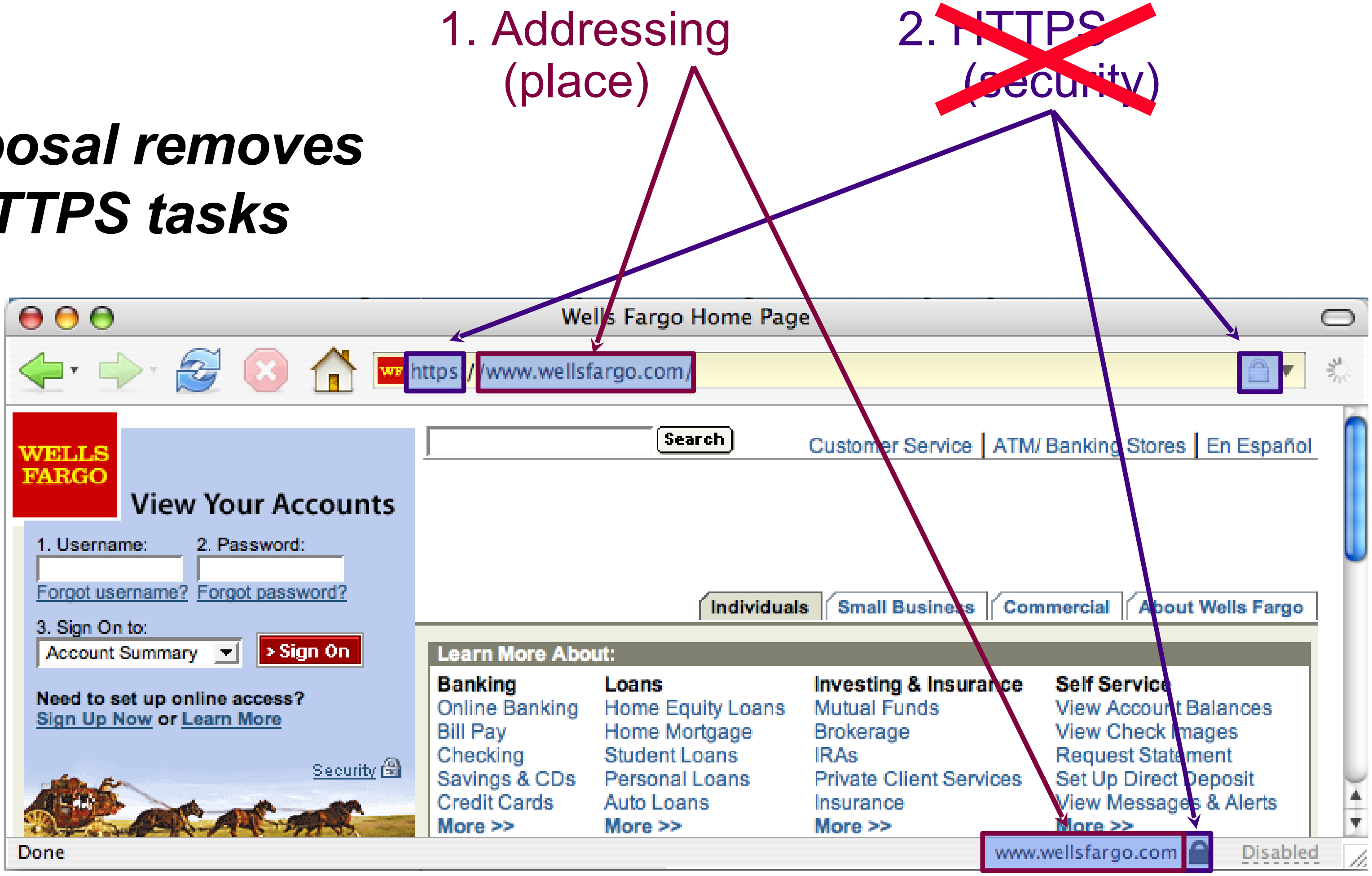
This work is sponsored by the I3P under Air Force Contract FA8721-05-0002. Opinions, interpretations, conclusions and recommendations are those of the author(s) and are not necessarily endorsed by the United States Government.

This work was produced under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College, and supported under Award number 2003-TK-TX-0003 from the U.S. Department of Homeland Security, Science and Technology Directorate. Points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security, the Science and Technology Directorate, the I3P, or Dartmouth College.

Simplifying Users' Tasks & Decisions

Users currently burdened with tasks related to:

Our proposal removes users' HTTPS tasks



1. Addressing (place)

2. ~~HTTPS (security)~~

Goals

- Reduce users’ cognitive overhead
- Free users to focus on location (place)
- Complement ‘anti-phishing’ research

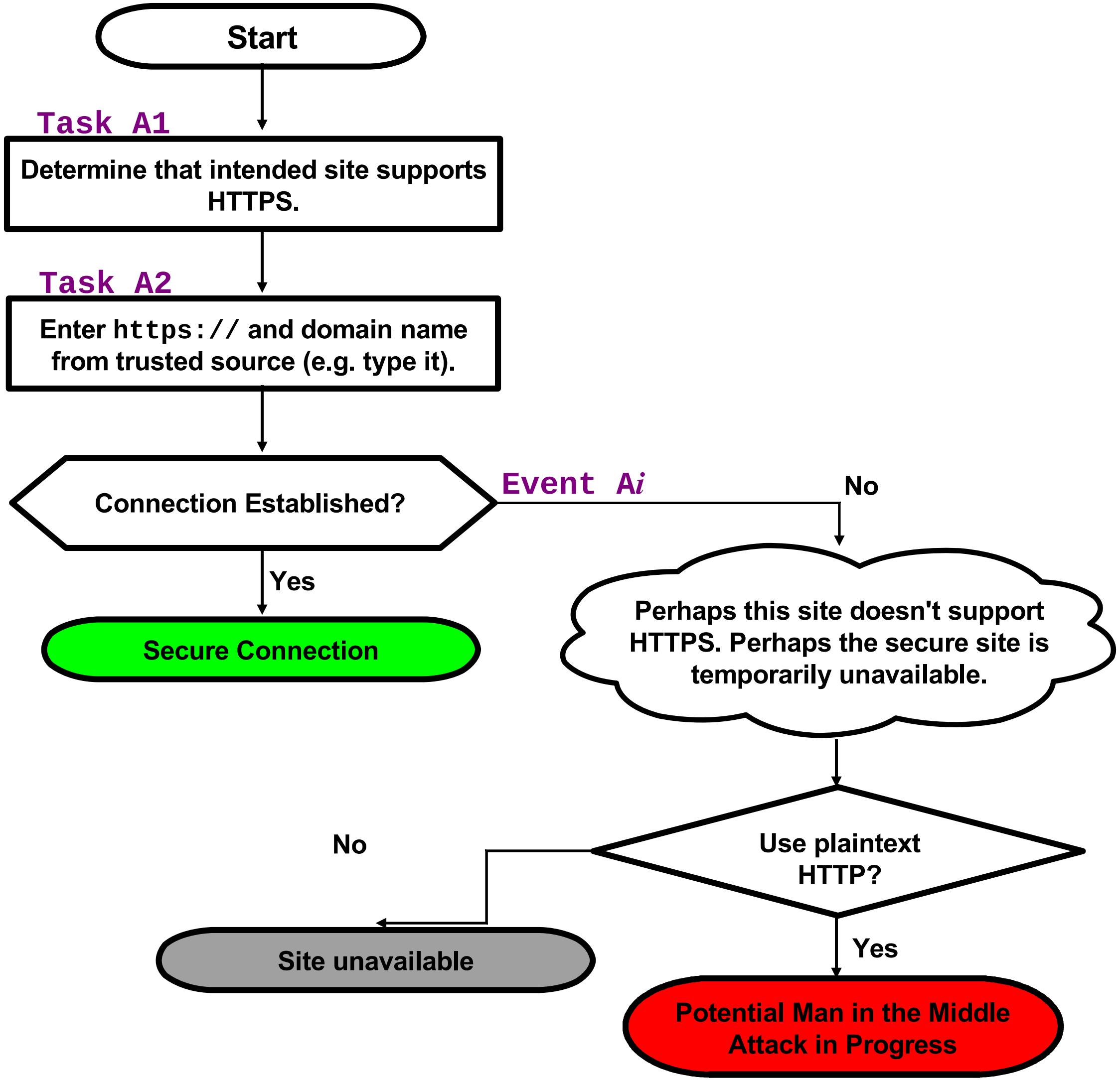
Browsing to a Secure Site:
What’s Expected of Users Today

Two methods by which to ensure a secure connection

- § Request a secure connection to a site
- § Verify security after the connection is established

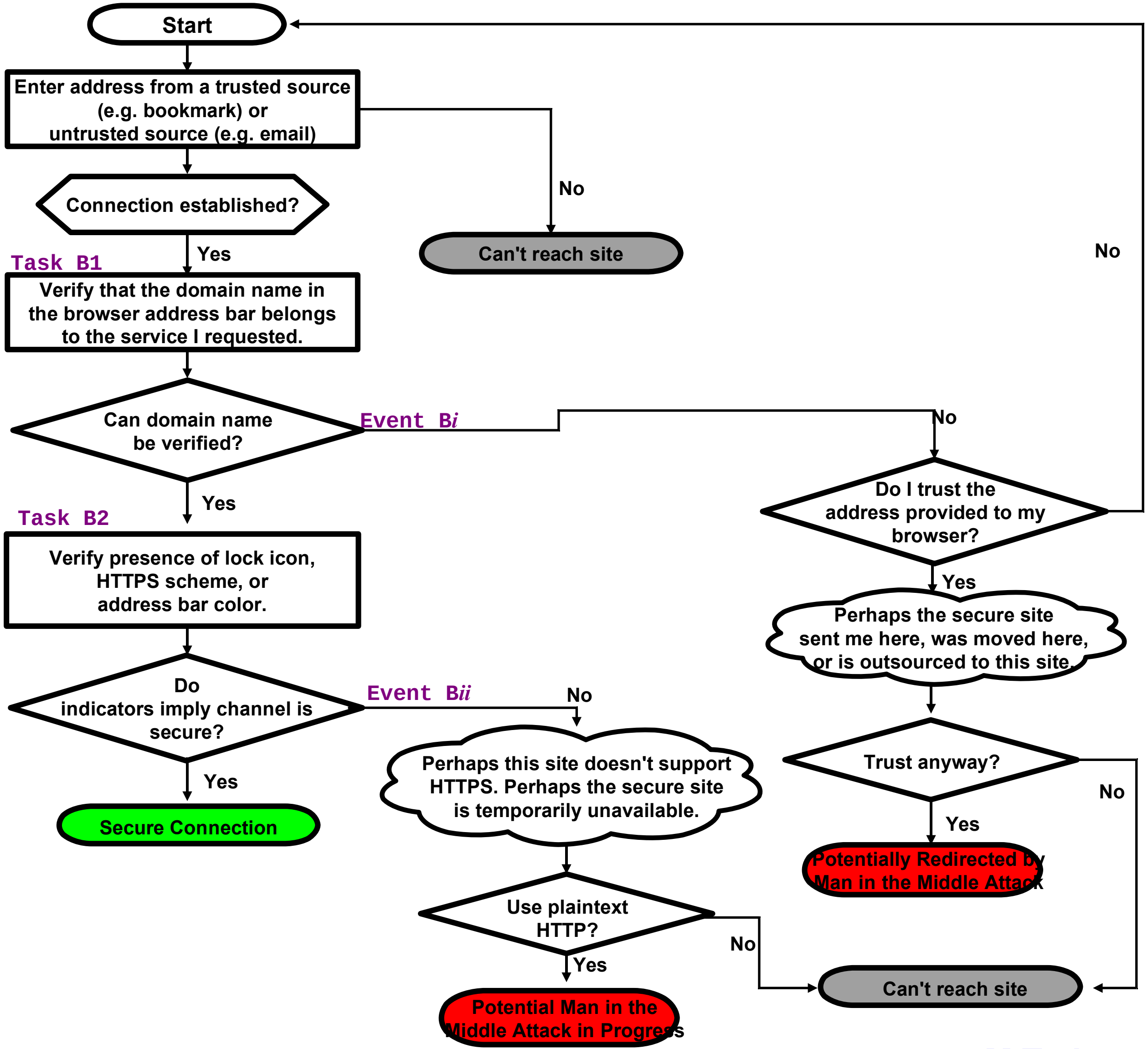
Current process:

Method A: Request


Start
Task A1
Determine that intended site supports HTTPS.
Task A2
Enter https:// and domain name from trusted source (e.g. type it).
Connection Established?
Event Ai
No
Yes
Secure Connection
Perhaps this site doesn't support HTTPS. Perhaps the secure site is temporarily unavailable.
Use plaintext HTTP?
No
Site unavailable
Yes
Potential Man in the Middle Attack in Progress

Current process

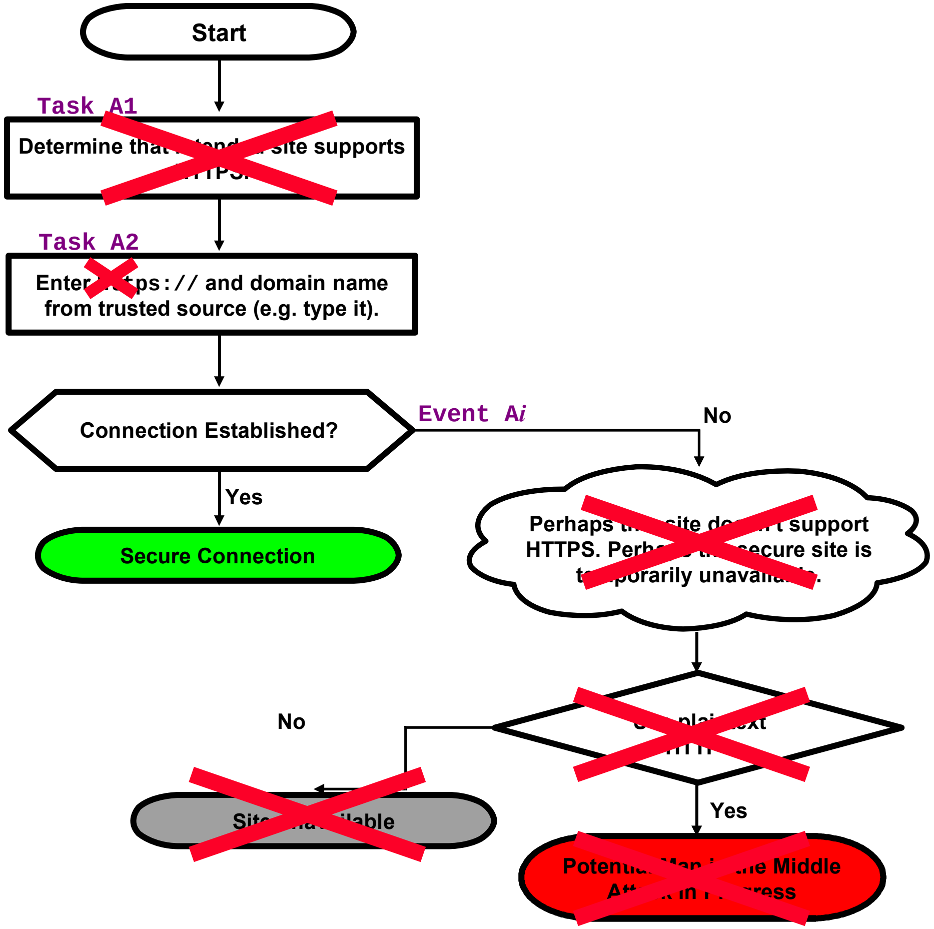
Method B: Verify



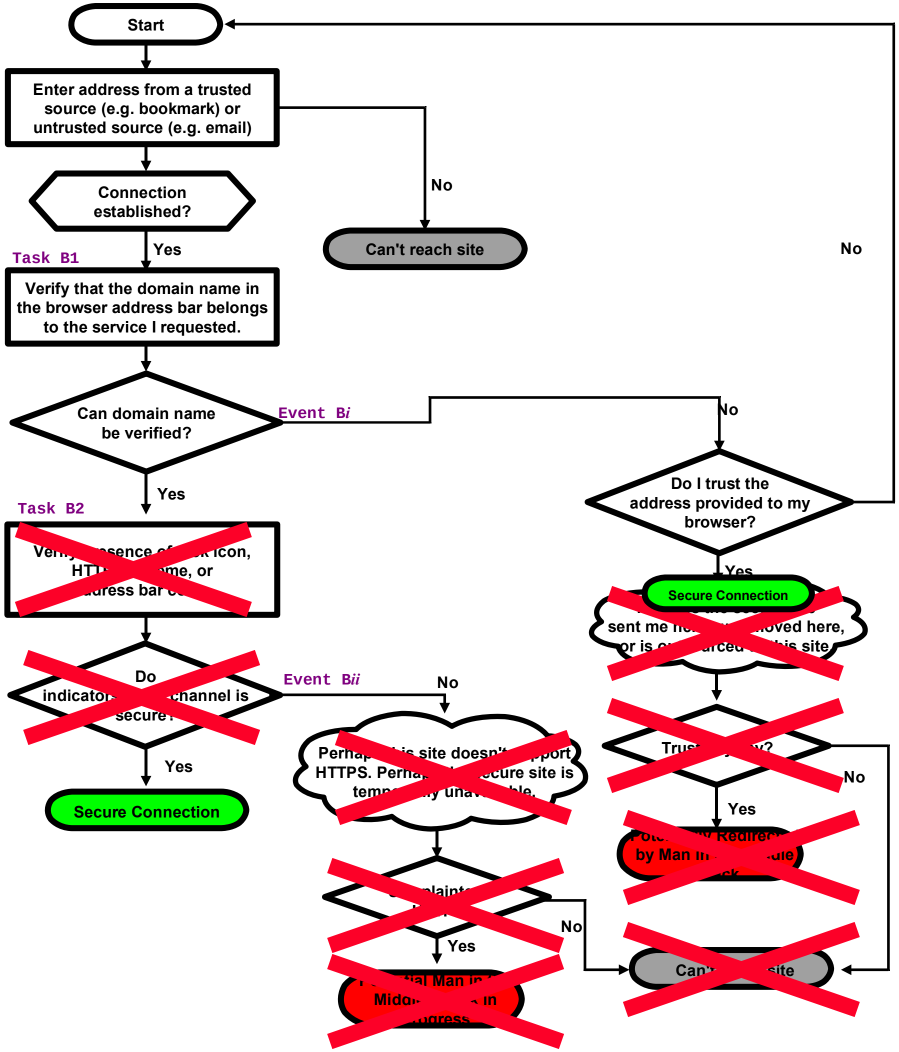
Start
Enter address from a trusted source (e.g. bookmark) or untrusted source (e.g. email)
No
Connection established?
Can't reach site
Yes
Task B1
Verify that the domain name in the browser address bar belongs to the service I requested.
No
Can domain name be verified?
Event Bi
No
Do I trust the address provided to my browser?
Yes
Task B2
Yes
Verify presence of lock icon, HTTPS scheme, or address bar color.
Perhaps the secure site sent me here, was moved here, or is outsourced to this site.
Do indicators imply channel is secure?
Event Bii
No
Trust anyway?
No
Yes
Secure Connection
Perhaps this site doesn't support HTTPS. Perhaps the secure site is temporarily unavailable.
Yes
Potentially Redirected by Man in the Middle Attack
Use plaintext HTTP?
No
Yes
Can't reach site
Potential Man in the Middle Attack in Progress

Our Proposal Removes
Tasks & Decisions

Method A: Request


Start
Task A1
Determine that intended site supports HTTPS.
Task A2
Enter https:// and domain name from trusted source (e.g. type it).
Connection Established?
Event Ai
No
Yes
Secure Connection
Perhaps this site doesn't support HTTPS. Perhaps the secure site is temporarily unavailable.
Complaint text
HTTP
No
Yes
Site unavailable
Potential Man in the Middle Attack in Progress

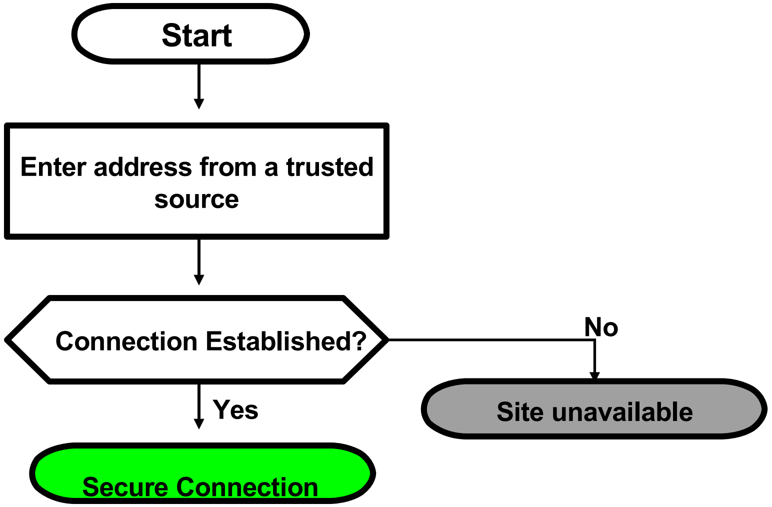
Method B: Verify


Start
Enter address from a trusted source (e.g. bookmark) or untrusted source (e.g. email)
Connection established?
No
Can't reach site
Yes
Task B1
Verify that the domain name in the browser address bar belongs to the service I requested.
Can domain name be verified?
Event Bi
No
No
Do I trust the address provided to my browser?
Yes
Secure Connection
sent me here, moved here, or is outsourced this site.
Yes
Task B2
Verify presence of lock icon, HTTPS scheme, or address bar color
Do indicators channel is secure?
Yes
Secure Connection
Event Bii
No
Perhaps this site doesn't support HTTPS. Perhaps the secure site is temporarily unavailable.
Complaint
Yes
No
Potential Man in the Middle Attack in Progress
Trust y?
No
Yes
Potentially Redirected by Man in the Middle Attack
Can't reach site

Using our SSR Proposal:

Users' Tasks & Decisions

Method 1: Trusted Source



Start

Enter address from a trusted source

Connection Established?

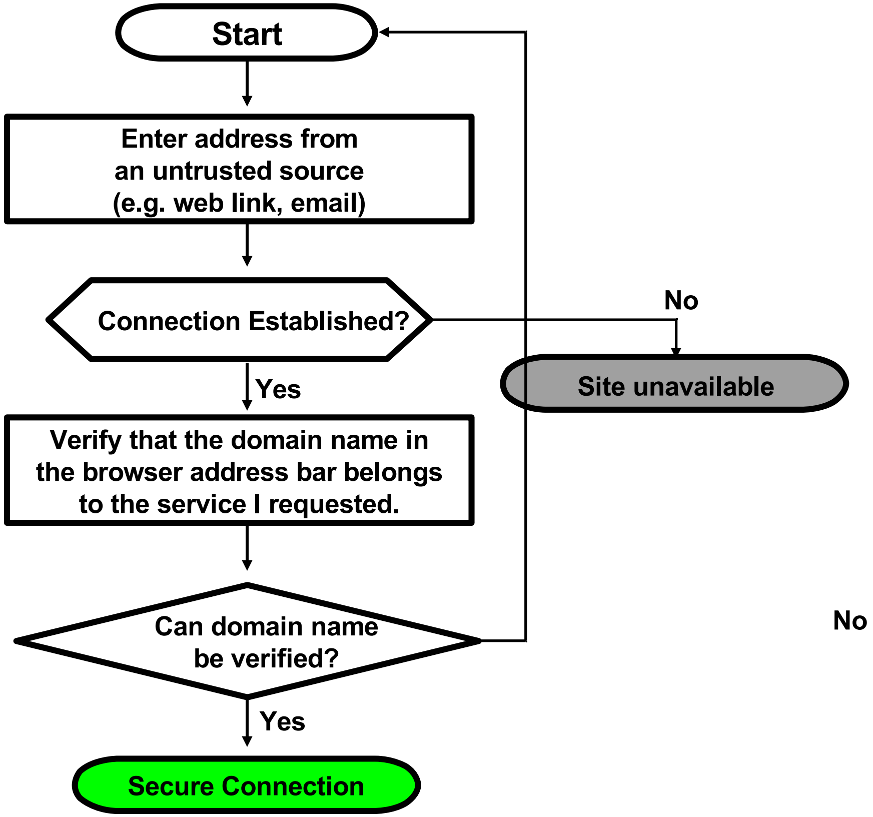
No

Site unavailable

Yes

Secure Connection

Method 2: Untrusted Source



Start

Enter address from an untrusted source (e.g. web link, email)

Connection Established?

No

Site unavailable

Yes

Verify that the domain name in the browser address bar belongs to the service I requested.

Can domain name be verified?

No

Yes

Secure Connection

Why Don’t We Have the Simpler Model?

Browsers have needed user input to activate HTTPS

- Browsers *must* default to HTTP
- Security-activation agreement problem
  - No secure means of discovering whether a site offers HTTPS

Our Proposal:
Sites Publish their Security Requirements

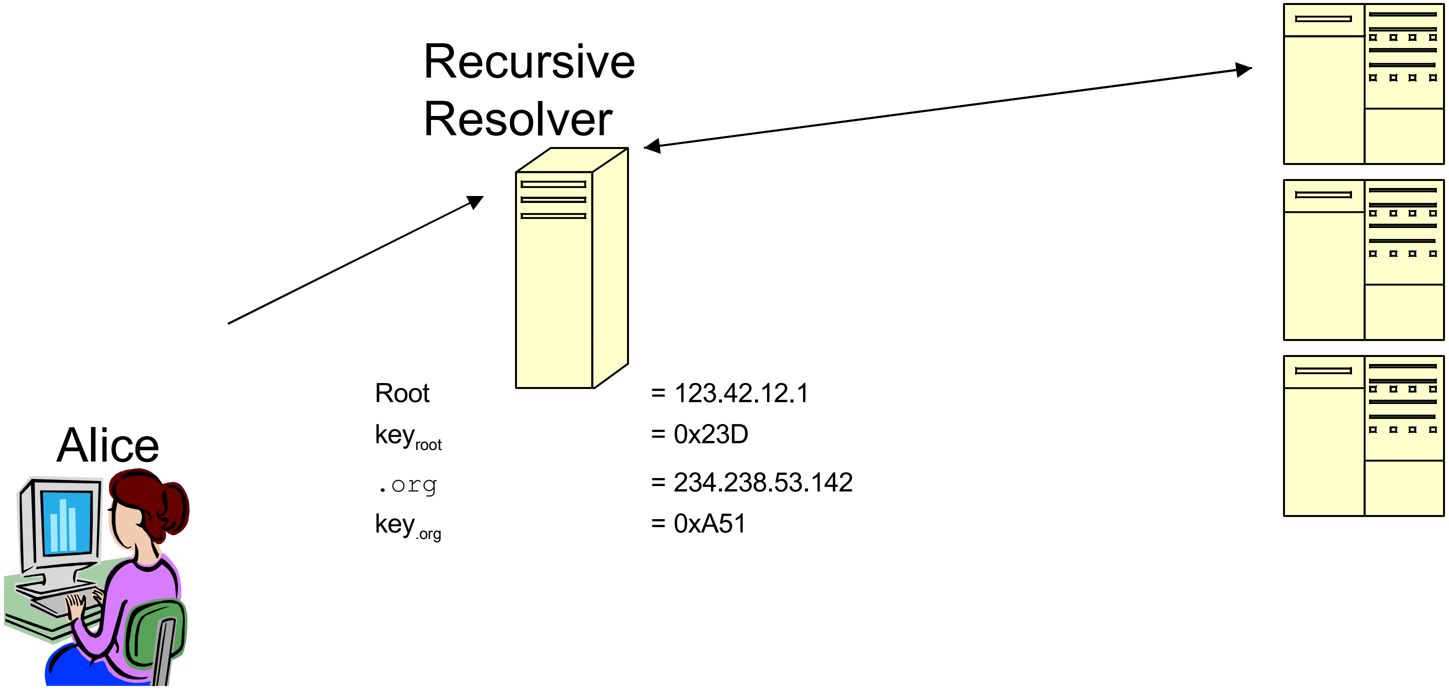
The Service Security Requirements (SSR) record

Example requirement:
All web connections must use HTTPS, minimum SSLv3

- Securely published and universally accessible
- SSR is a record stored in the DNS
- DNSSEC provides the security for the record itself

DNS + DNSSEC + SSR:

Query root zone DNS server



Recursive Resolver

Root	= 123.42.12.1
key_{root}	= 0x23D
`.org`	= 234.238.53.142
$\text{key}_{\text{.org}}$	= 0xA51

. **(root)** zone DNS server

.org zone DNS server

`w3.org` zone DNS server

Alice

Browser

key_{root} = 0x23D

Q: What is the address of:

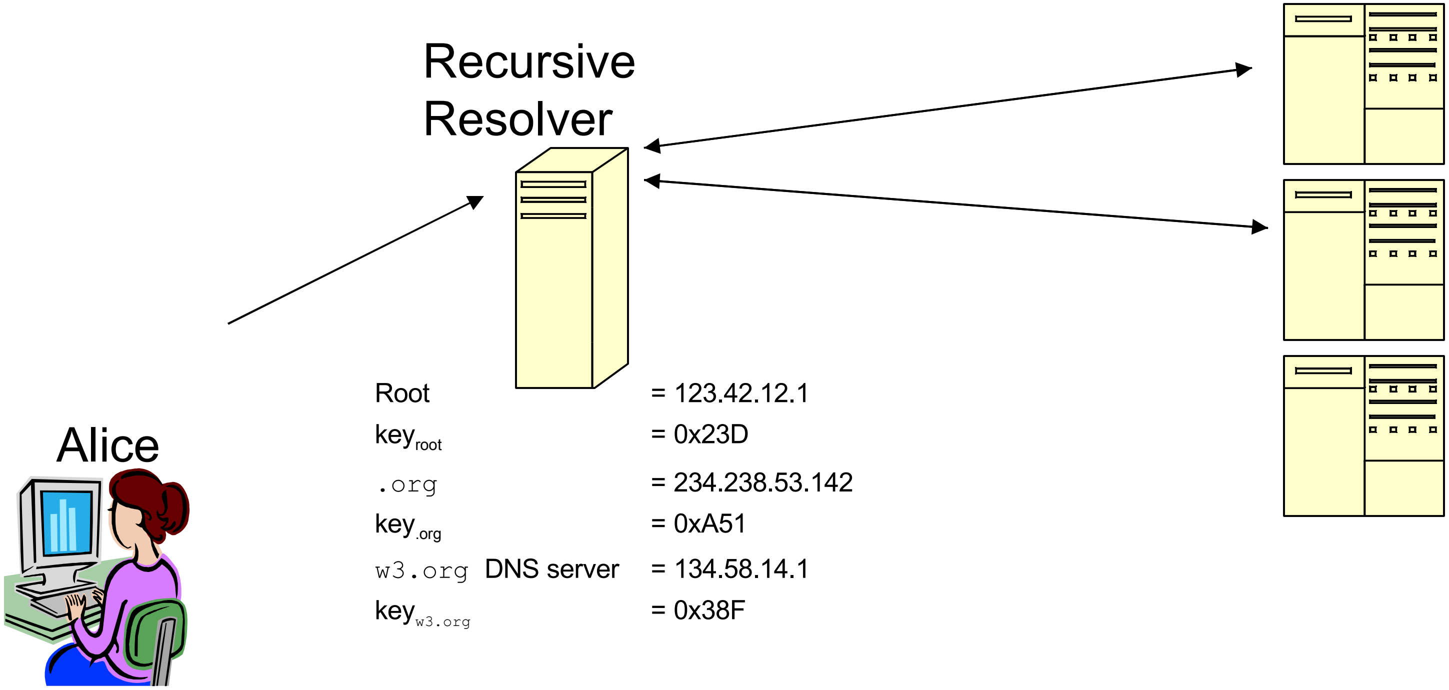
`www.w3.org`

A: I don’t know `www.w3.org`, but…

`.org`	= 134.58.14.3
$\text{key}_{\text{.org}}$	= 0xA51

Signed with key_{root}

DNS + DNSSEC + SSR:
Query .org zone DNS server



Root	= 123.42.12.1
key_{root}	= 0x23D
.org	= 234.238.53.142
$\text{key}_{\text{.org}}$	= 0xA51
w3.org DNS server	= 134.58.14.1
$\text{key}_{\text{w3.org}}$	= 0x38F

Browser

key_{root} = 0x23D

Q: What is the address of:

`www.w3.org`

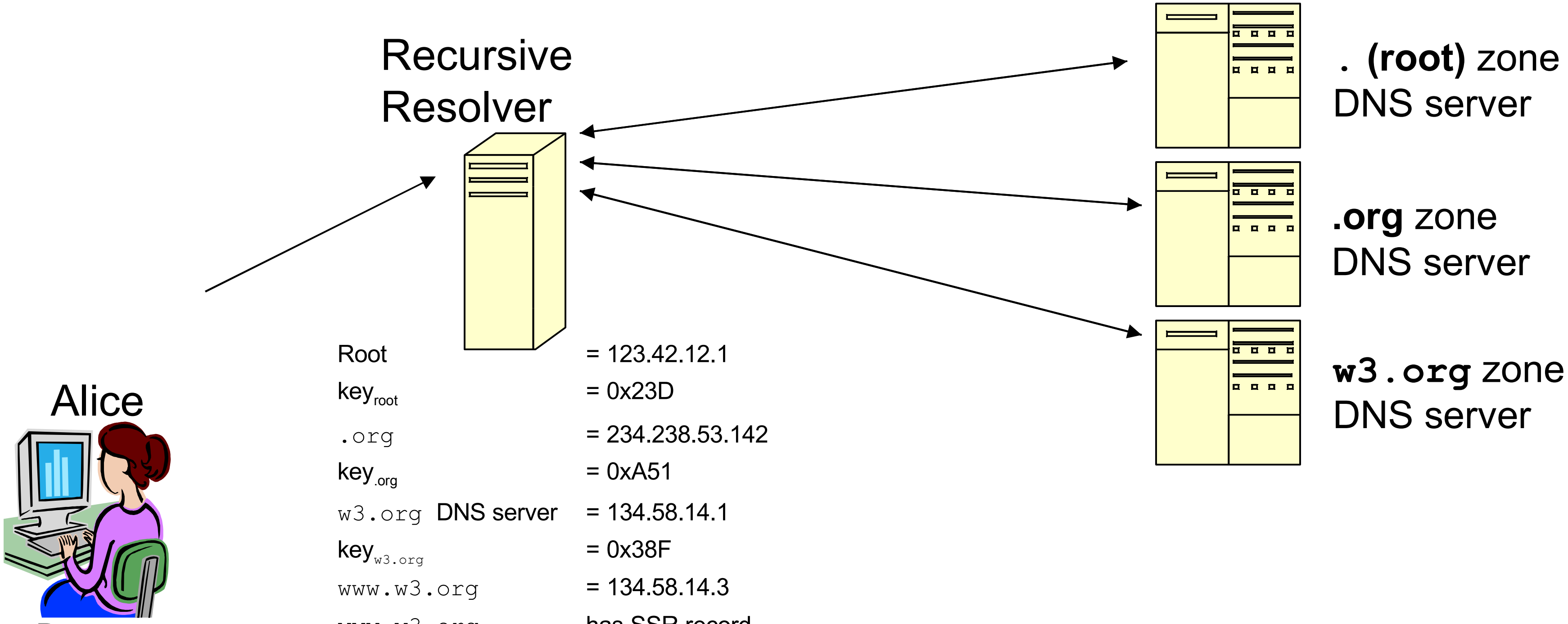
A: The `w3.org` DNS server is:

w3.org	= 134.58.14.3
$\text{key}_{\text{w3.org}}$	= 0xA51

Signed with $\text{key}_{\text{.org}}$

DNS + DNSSEC + SSR:

Query `w3.org` zone DNS server



Recursive Resolver

. **(root)** zone DNS server

.org zone DNS server

`w3.org` zone DNS server

Alice

Browser

key$_{root}$ = 0x23D

Root	= 123.42.12.1
key$_{root}$	= 0x23D
`.org`	= 234.238.53.142
key$_{.org}$	= 0xA51
`w3.org` DNS server	= 134.58.14.1
key$_{w3.org}$	= 0x38F
`www.w3.org`	= 134.58.14.3
`www.w3.org`	has SSR record

Q: What is the address of:

`www.w3.org`

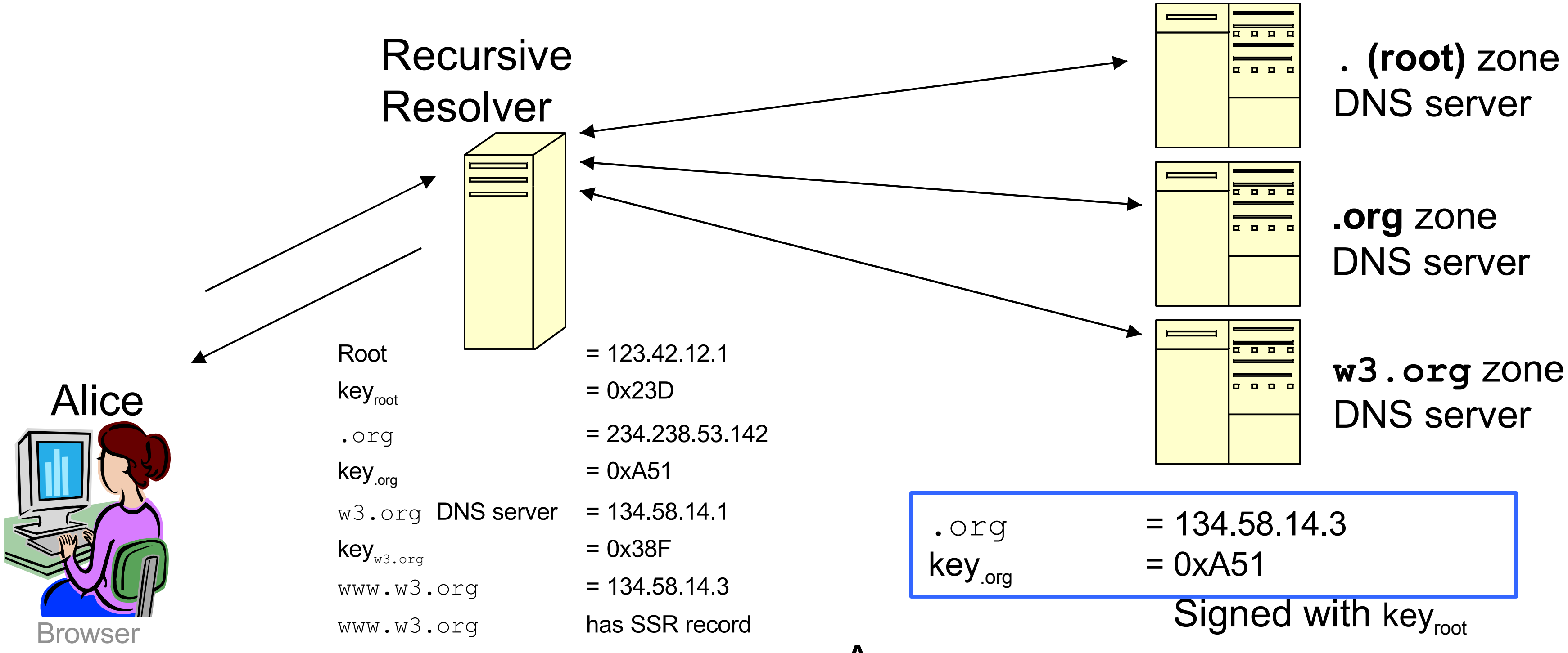
A:

`www.w3.org` = 134.58.14.3
`www.w3.org` has an SSR record

Signed with key$_{w3.org}$

DNS + DNSSEC + SSR:

Return responses to client



Root	= 123.42.12.1
key_{root}	= 0x23D
`.org`	= 234.238.53.142
$\text{key}_{\text{.org}}$	= 0xA51
`w3.org` DNS server	= 134.58.14.1
$\text{key}_{\texttt{w3.org}}$	= 0x38F
`www.w3.org`	= 134.58.14.3
`www.w3.org`	has SSR record

key_{root} = 0x23D

A:

`.org`	= 134.58.14.3
$\text{key}_{\text{.org}}$	= 0xA51

Signed with key_{root}

`w3.org`	= 134.58.14.3
$\text{key}_{\texttt{w3.org}}$	= 0xA51

Signed with $\text{key}_{\texttt{.org}}$

`www.w3.org`	= 134.58.14.3
`www.w3.org`	has an SSR record

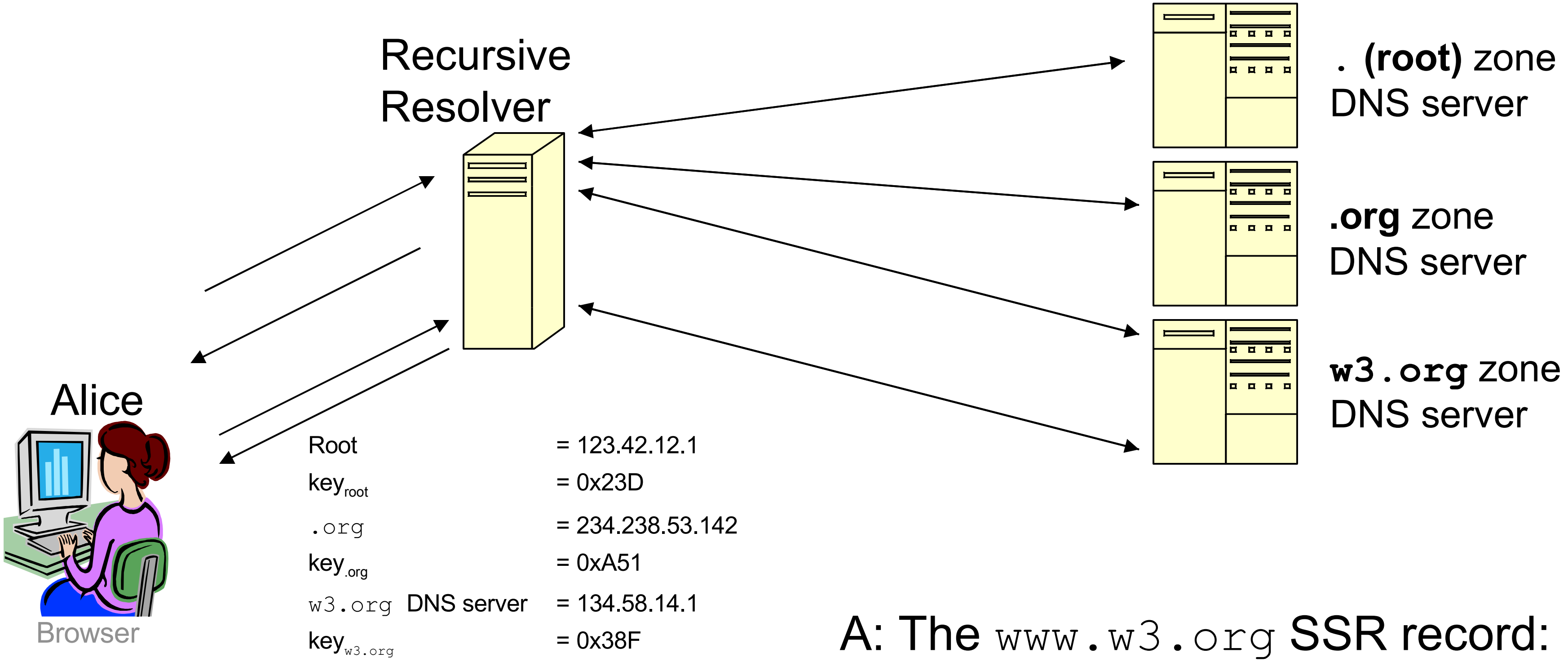
Signed with $\text{key}_{\texttt{w3.org}}$

Q: What is the address of:

`www.w3.org`

DNS + DNSSEC + SSR:

Query `w3.org` zone DNS server for SSR



Root	= 123.42.12.1
key_{root}	= 0x23D
`.org`	= 234.238.53.142
$\text{key}_{\text{.org}}$	= 0xA51
`w3.org` DNS server	= 134.58.14.1
$\text{key}_{\texttt{w3.org}}$	= 0x38F
`www.w3.org`	= 134.58.14.3
`www.w3.org`	has SSR record

Q: What are the security requirements for:

`www.w3.org`

A: The `www.w3.org` SSR record:

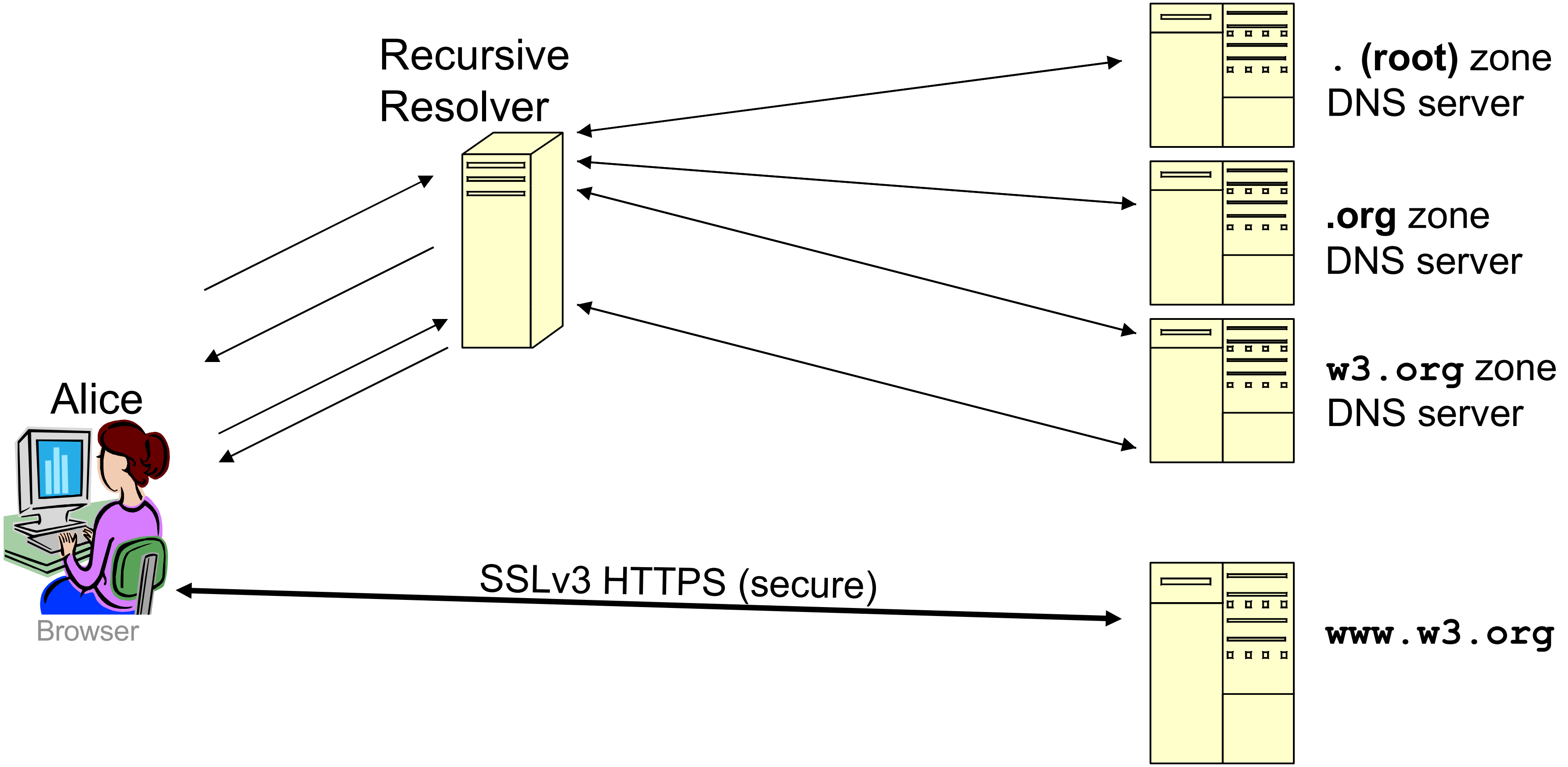
HTTPS required:
SSLv3 or TLS v1

HTTP forbidden

Signed with $\text{key}_{\texttt{w3.org}}$

DNS + DNSSEC + SSR:

Initiate HTTPS connection



The SSR Record’s Capabilities

Enables sites to raise security level of users’ configurations (*exactly what Chuck Wade requested*)

- **Require protocols & protocol options**
  - Cipher, keylength, etc.
  - E.g. HTTPS using SSLv3 and AES-256
- **Forbid protocols & protocol options**
  - E.g. no HTTP, no SHA1
- **Securely redirect**
  - E.g. `etrade.com` → `secure.us.etrade.com`
- **Restrict subdomains**
  - E.g. acceptable subdomains are `login.w3.org, www.w3.org`

Value of Simpler Model

Requirements presented yesterday by Ian Fette (CMU)

- **Concepts at users' level of understanding**
  - ✓ **Removes HTTPS tasks (which are not at user level)**
- **Minimal interaction with user**
  - ✓ **Removes interaction**
- **Should be hard to make mistakes**
  - ✓ **SSR records prevent users' mistakes managing HTTPS**
- **Works wherever the user is**
  - ✓ **Yes, if we can seduce those browser folks (wink, wink)**
- **Consider disabilities**
  - ✓ **Disabled users no longer need to see locks**

SSR Design Questions

Open questions:

- Secure redirection: use S-NAPTR or build into SSR?
- How should we define a service?
  - Yes! It can be used for services other than the web.
- What other abilities could/should SSR provide?
- How to best meet the needs of browser developers?

Questions & Discussion

Questions?

Comments?

Ideas?