U.S. NAVAL
RESEARCH
LABORATORY

HSTS Supports Targeted Surveillance

Paul Syverson and Matt Traudt
Center for High Assurance Computer Systems (CHACS)
U.S. Naval Research Laboratory
Washington DC

FOCI `18: Free and Open
Communications on the Internet
Baltimore MD, U.S.A.
September 12 2017

- HTTP Strict Transport Security (HSTS): widely used (IETF) Internet Standard
  - improves security: forces encrypted connections
  - allows a site to individuate and track users, even if they clear cookies and try to erase their history
- Everybody knew that from the beginning

U.S. NAVAL RESEARCH LABORATORY

Main take-aways

- It's much worse than was recognized/acknowledged: Using HSTS headers
  - sites can track how recently someone visited
  - sites can track despite recent Safari anti-tracking countermeasures
  - 3rd parties (Ad services, CDNs) can track users **across** visited sites
  - can censor the content, services, and destinations users are offered
- There are things we can do to improve the situation while problem is still anecdotal
- "HSTS Supports Targeted Surveillance" is recursively paradoxical

Example: what HSTS is for

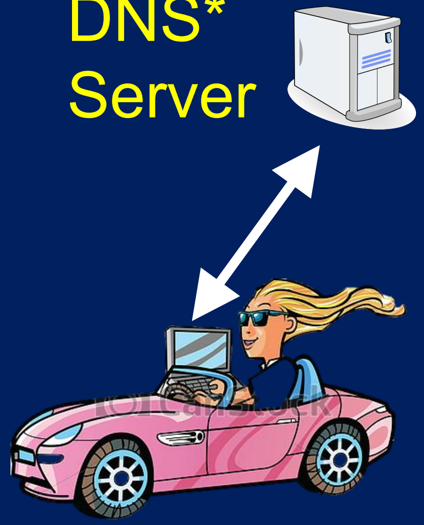
maps.google.com

Alice: lost & late for meeting,
looking up route on Google Maps

Example: What HSTS is for

maps.google.com

DNS*
Server



Q:
maps.google.com
A:
172.217.1.174

Alice: lost & late for meeting,
looking up route on Google Maps

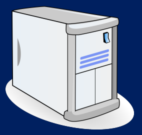
maps.google.com
IP Address:
172.217.1.174

*DNS: Domain Name System

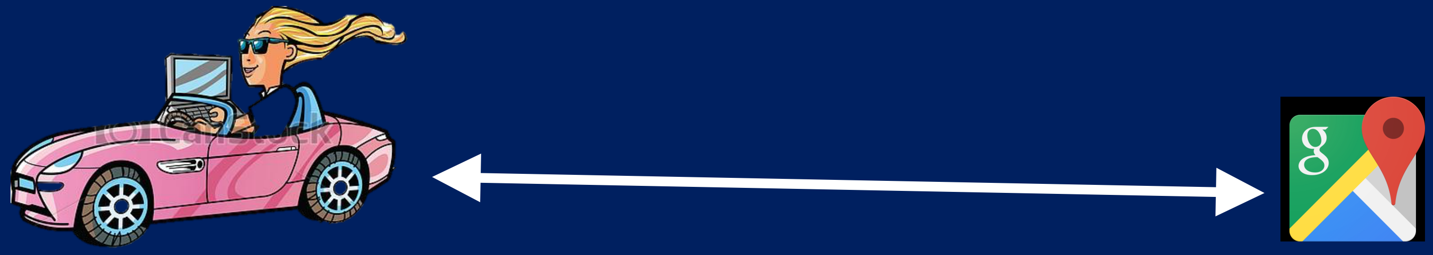
Example: What HSTS is for

maps.google.com

DNS*
Server



Q:
maps.google.com
A:
172.217.1.174

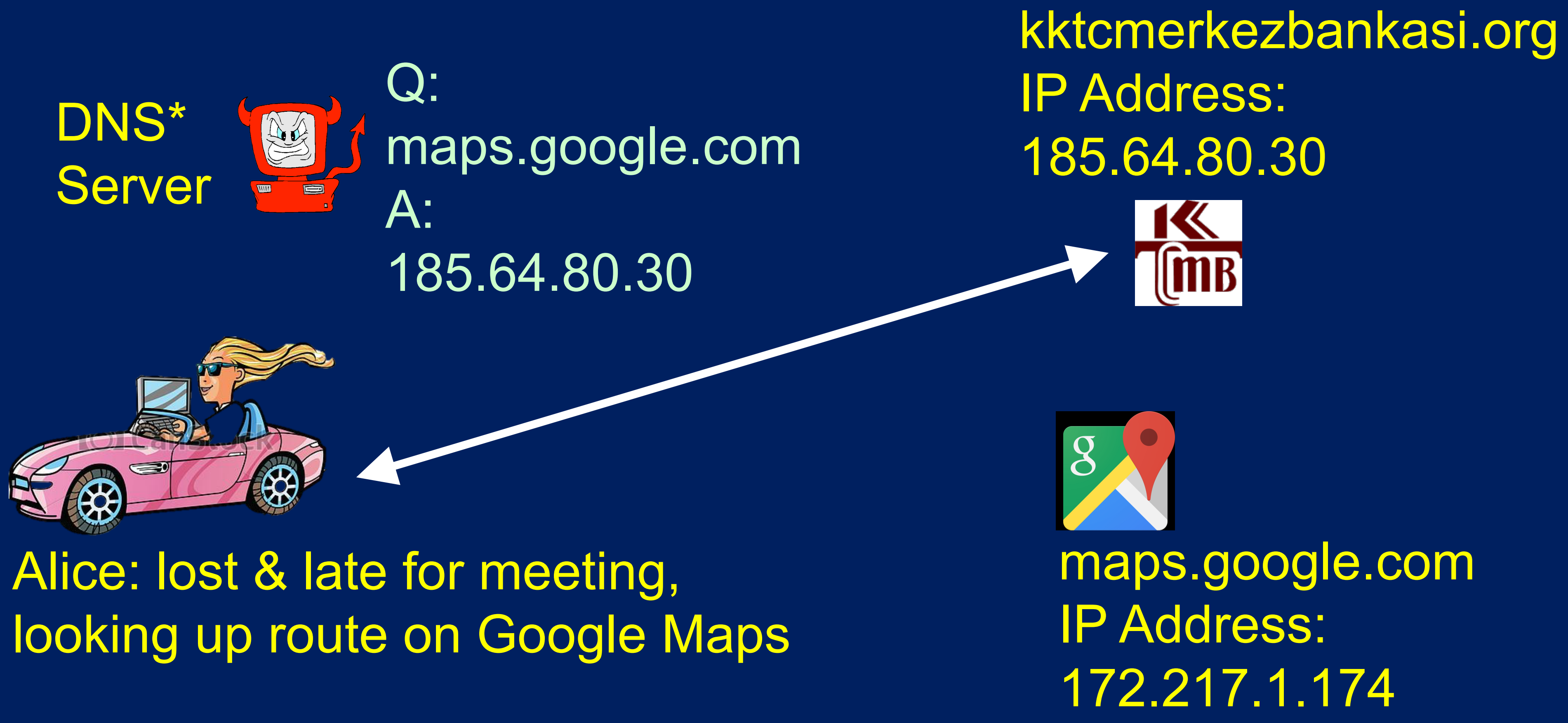


maps.google.com
IP Address:
172.217.1.174

*DNS: Domain Name System

Example: What HSTS is for

Address lookup is not secure



DNS* Server

Q: maps.google.com
A: 185.64.80.30

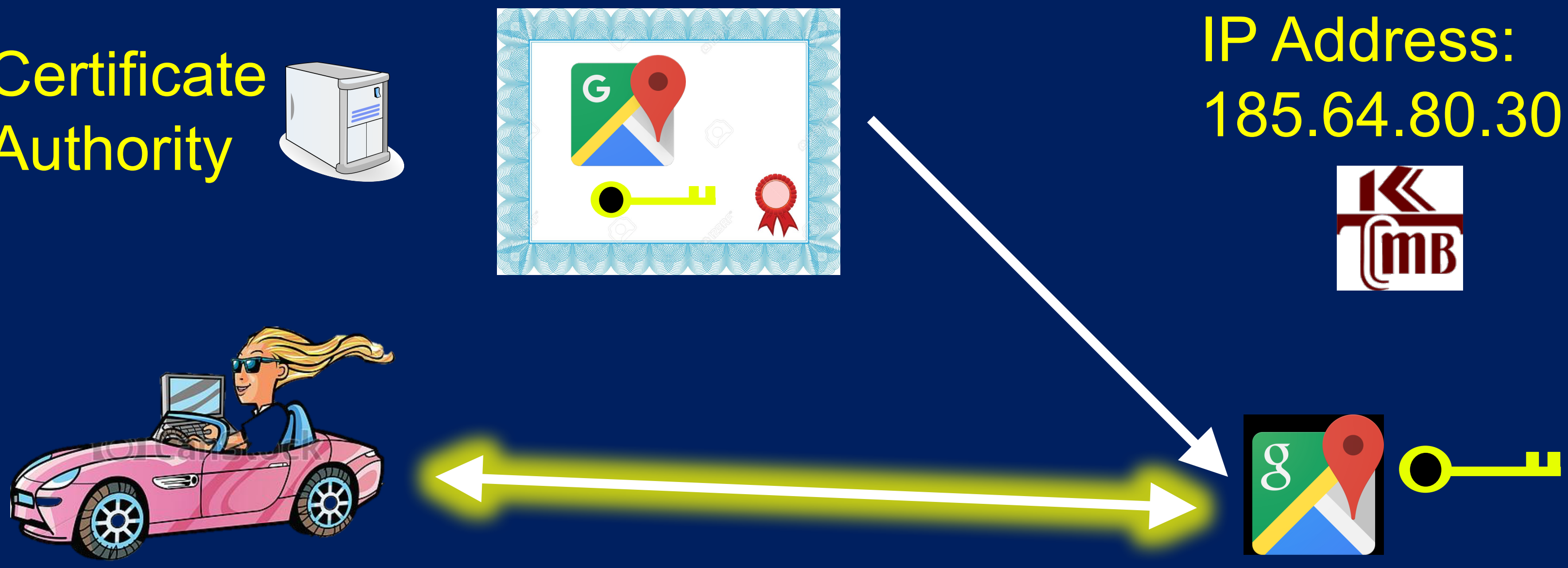
kktcmerkezbankasi.org
IP Address:
185.64.80.30

Alice: lost & late for meeting,
looking up route on Google Maps

maps.google.com
IP Address:
172.217.1.174

*DNS: Domain Name System

Example: What HSTS addresses



Certificate Authority

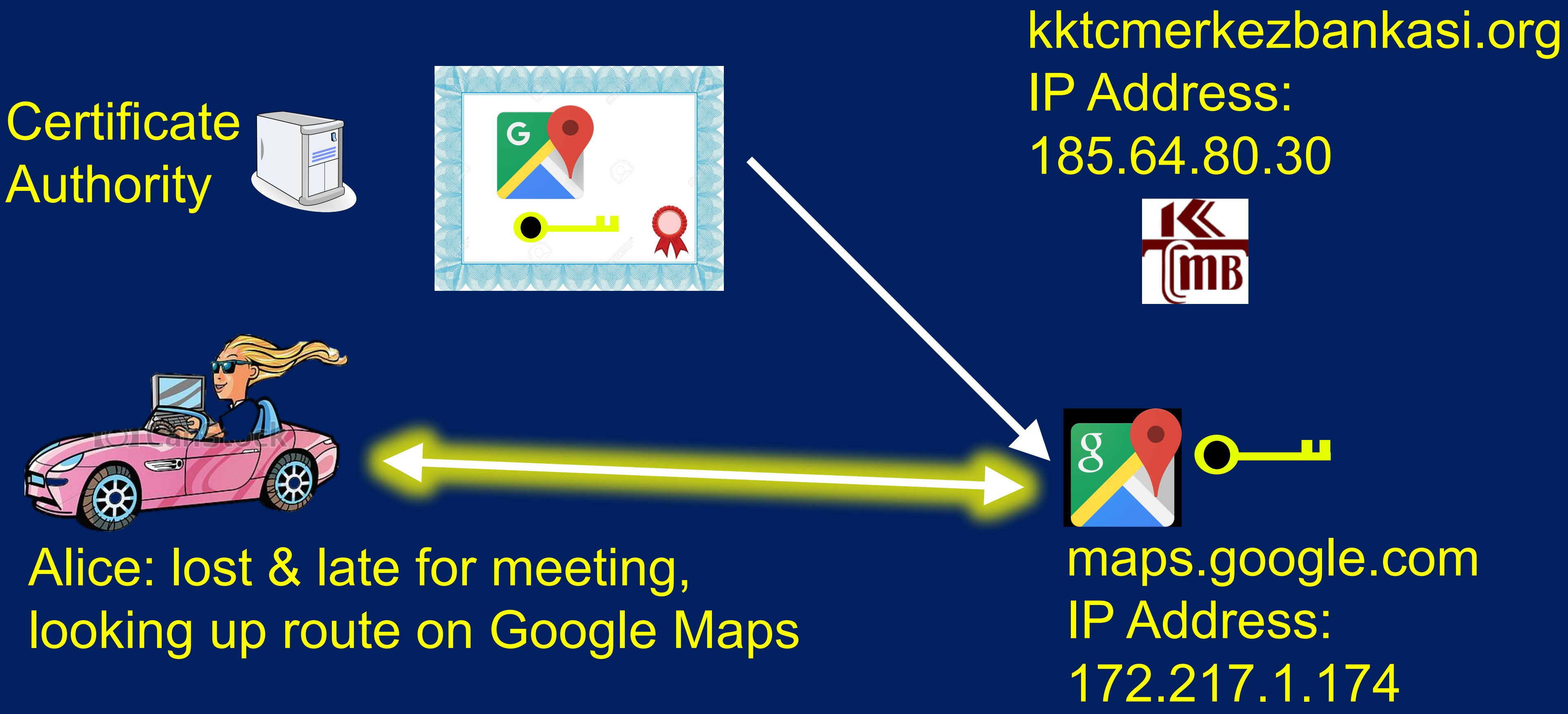
kktcmerkezbankasi.org
IP Address:
185.64.80.30

Alice: lost & late for meeting,
looking up route on Google Maps

maps.google.com
IP Address:
172.217.1.174

Example: What HSTS is for

Alice enters “maps.google.com”



Certificate Authority

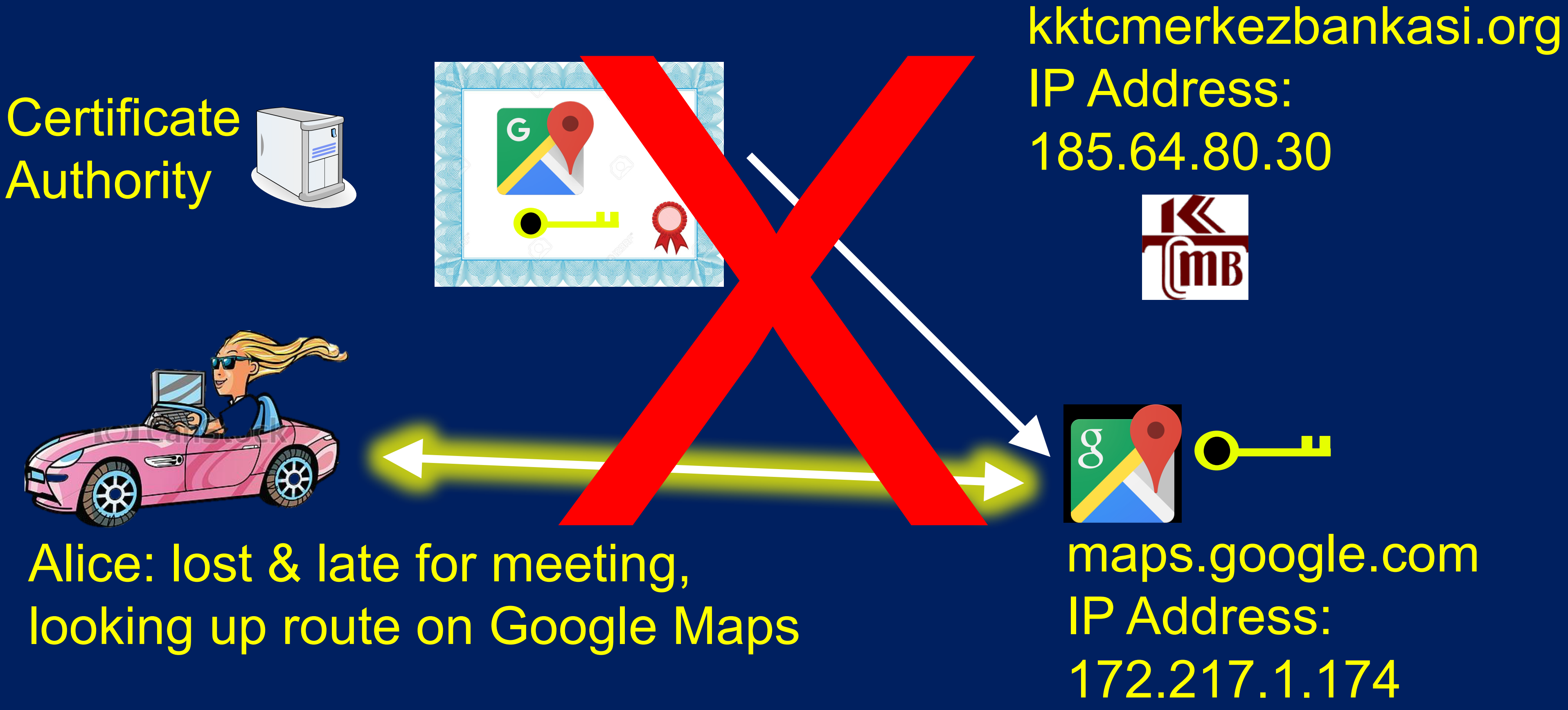
kktcmerkezbankasi.org
IP Address:
185.64.80.30

Alice: lost & late for meeting,
looking up route on Google Maps

maps.google.com
IP Address:
172.217.1.174

Example: What HSTS is for

Alice enters “maps.google.com”



Certificate Authority

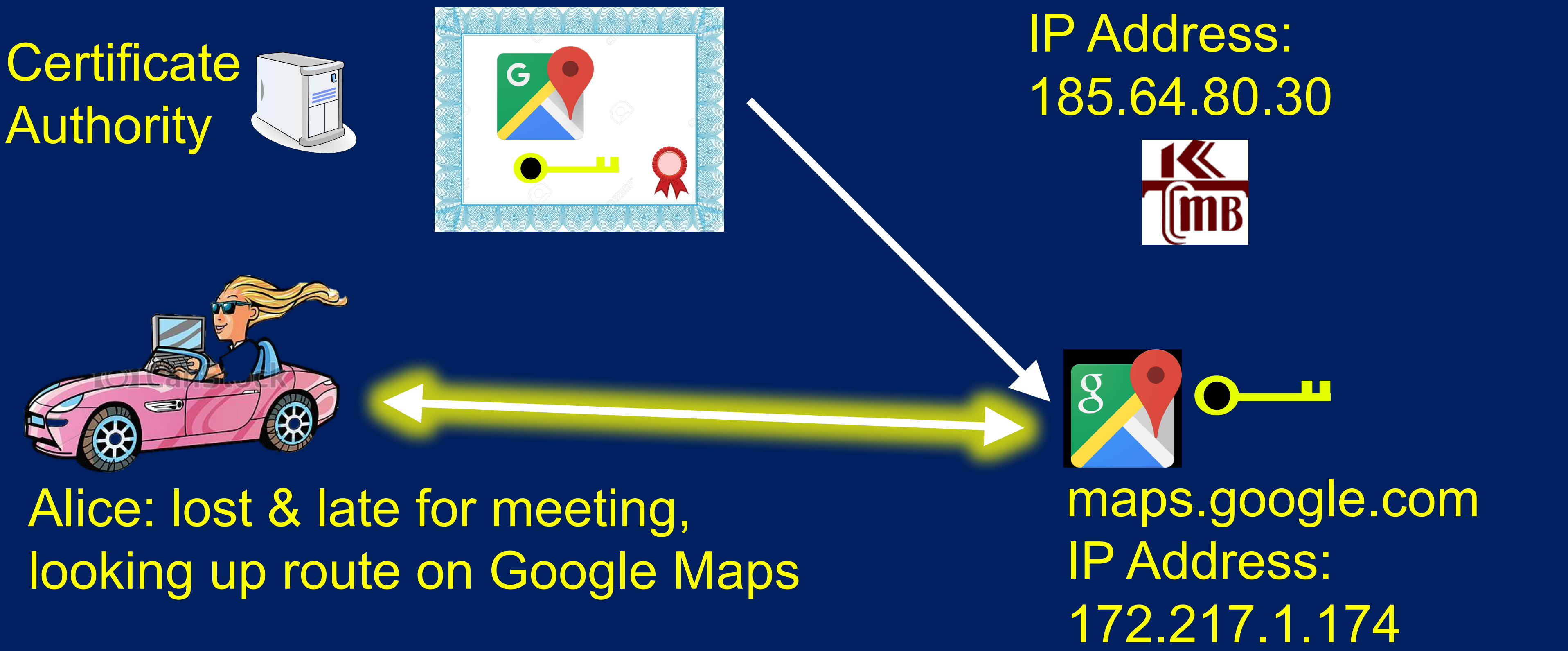
kktcmerkezbankasi.org
IP Address:
185.64.80.30

Alice: lost & late for meeting,
looking up route on Google Maps

maps.google.com
IP Address:
172.217.1.174

Example: What HSTS is for

Alice enters maps.google.com,
HSTS forces her browser to only connect via
HTTPS://maps.google.com



HSTS basics

Alice enters maps.google.com,
HSTS forces her browser to only connect via HTTPS://maps.google.com

How?

HSTS basics

Alice enters maps.google.com,
HSTS forces her browser to only connect via HTTPS://maps.google.com

How?

- maps.google.com sends header

```
Strict-Transport-Security: max-age=31536000
```

- Alice's browser remembers this
  - will only connect to maps.google.com via TLS for one year
    - whether typed, selected, or redirected
  - will not allow user to click through warning

HSTS basic tracking

- Send invisible pixels and HSTS headers for them

<img src=“https://01.foo.com/FQd23.jpg”, width=“1”, height=“1”>

01.foo.com/FQd23.jpg, send HSTS header

02.foo.com/FQd23.jpg, don’t send HSTS header

03.foo.com/FQd23.jpg, send HSTS header

04.foo.com/FQd23.jpg, send HSTS header

05.foo.com/FQd23.jpg, don’t send HSTS header

06.foo.com/FQd23.jpg, send HSTS header

etc.

HSTS vector
1
0
1
1
0
1
Etc.

HSTS basic tracking

- When client returns, attempt HTTP connection to all resources, and see which force HTTPS

	HSTS vector
01.foo.com/FQd23.jpg, send HSTS header	1
02.foo.com/FQd23.jpg, don’t send HSTS header	0
03.foo.com/FQd23.jpg, send HSTS header	1
04.foo.com/FQd23.jpg, send HSTS header	1
05.foo.com/FQd23.jpg, don’t send HSTS header	0
06.foo.com/FQd23.jpg, send HSTS header	1
Etc.	Etc.

HSTS basic tracking

- When client returns, attempt HTTP connection to all resources, and see which force HTTPS
- With 30 pixels, 2^30=over a billion possible classifications

HSTS basic tracking

- When client returns, attempt HTTP connection to all resources, and see which force HTTPS
- With 30 pixels, 2^30=over a billion possible classifications
- Some call HSTS state vector “supercookie”
  - Survives clearing cookies and some other ways of clearing data/history

HSTS basic tracking

- When client returns, attempt HTTP connection to all resources, and see which force HTTPS
- With 30 pixels, 2^30=over a billion possible classifications
- Some call HSTS state vector “supercookie”
  - Survives clearing cookies and some other ways of clearing data/history

- Mar 18: Apple reported basic tracking of Safari users in the wild.
- Announced countermeasures
  - Ignore HSTS headers for invisible pixels and similar (domains for which they block cookies).
  - Ignore HSTS except for loaded hostname and TLD+1

  (E.g., for a.a.a.a.foo.com, only respect HSTS headers for a.a.a.a.foo.com name and foo.com, ***not*** for a.a.foo.com)

HSTS redirect tracking

- Mar 18: Apple reported basic tracking of Safari users in the wild.
- Announced countermeasures
  - Ignore HSTS headers for invisible pixels and similar (domains for which they block cookies).
  - Ignore HSTS except for loaded hostname and TLD+1

  (E.g., for a.a.a.a.foo.com, only respect HSTS headers for a.a.a.a.foo.com name and foo.com, ***not*** for a.a.foo.com)

Our main attacks redirect via a chain of **loaded** hostnames

HSTS redirect tracking

- Mar 18: Apple reported basic tracking of Safari users in the wild.
- Announced countermeasures
  - Ignore HSTS headers for invisible pixels and similar (domains for which they block cookies).
  - Ignore HSTS except for loaded hostname and TLD+1

  (E.g., for a.a.a.a.foo.com, only respect HSTS headers for a.a.a.a.foo.com name and foo.com, ***not*** for a.a.foo.com)

Our main attacks redirect via a chain of **loaded** hostnames

- Will this impact performance or raise user suspicion?

HSTS redirect tracking

- See video Redirect Chain Chrome and Safari.webm at https://github.com/pastly/satis-hsts-tracking

Attacks using HSTS redirect tracking

- Entropist fallacy: It’s not **just** about the number of specific clients individuated

Some of the other things attackers can do

- Can send HSTS headers with different values of `max-age=` to treat users who visited at various times differently
- Can offer up different content/services to users who visited different parts of web page, or parts in different order

- A content-delivery-network (CDN), Ad network, analytics network used at multiple sites can track users **across** sites
- Can select content to (not) offer on arbitrary serviced sites (again regardless of clearing cookies)

Attacks using HSTS redirect tracking

- See video HSTS Chrome clickjacking kitten.webm at https://github.com/pastly/satis-hsts-tracking

Attacks using HSTS redirect tracking

- See video HSTS Chrome clickjacking kitten.webm at https://github.com/pastly/satis-hsts-tracking

- More attacks and analysis (e.g. CSS-based cross-domain tracking) in paper with links to code and video
- Also discussions of HSTS-preload and HTTPS Everywhere

Recommendations, Dilemmas, Questions

- Browsers should make it clear how to check (and how to remove?) dynamic HSTS state
  - Chrome only browser we checked with GUI for this, but not as easy to find or use as clearing cookies
  - Firefox state is only stored to file when browser closes
  - Safari stores HSTS state in binary file

Recommendations, Dilemmas, Questions

- Browsers should make it clear how to check (and how to remove?) dynamic HSTS state

Eliminating HSTS header support avoids tracking/censorship, but makes MitM more broadly effective

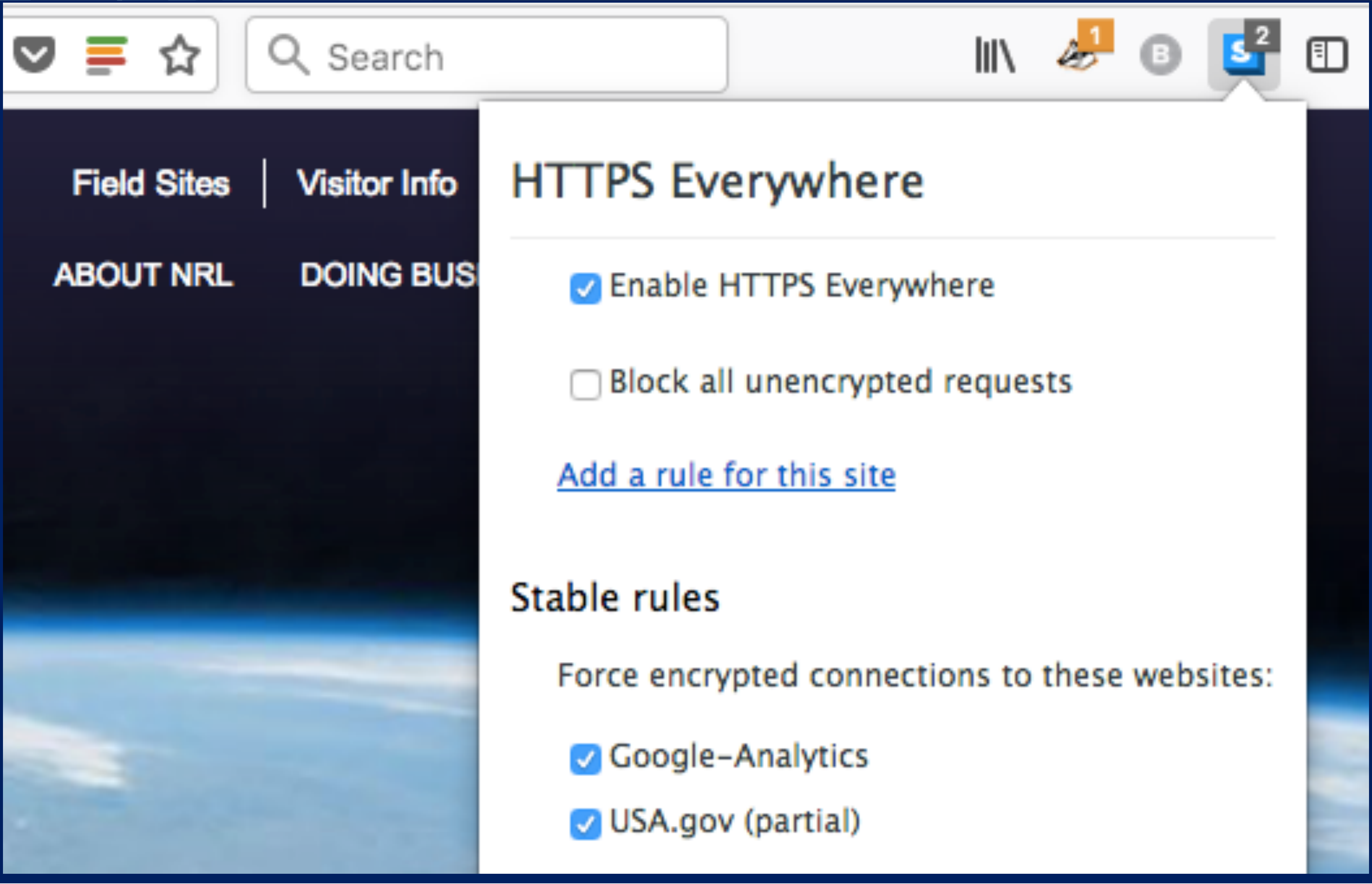
Recommendations, Dilemmas, Questions

- Browsers should make it clear how to check (and how to remove?) dynamic HSTS state

Eliminating HSTS header support avoids tracking/censorship, but makes MitM more broadly effective

- Browsers should make it possible to toggle on/off accepting HSTS headers?
- Browsers should permit toggling all connections TLS-only?

Recommendations, Dilemmas, Questions



HTTPS Everywhere

☑ Enable HTTPS Everywhere

□ Block all unencrypted requests

Add a rule for this site

Stable rules

Force encrypted connections to these websites:

☑ Google-Analytics

☑ USA.gov (partial)

Comments? Questions?

- Browsers should make it clear how to check (and how to remove?) dynamic HSTS state

Eliminating HSTS header support avoids tracking/censorship, but makes MitM more broadly effective

- Browsers should make it possible to toggle on/off accepting HSTS headers?
- Browsers should permit toggling all connections TLS-only?