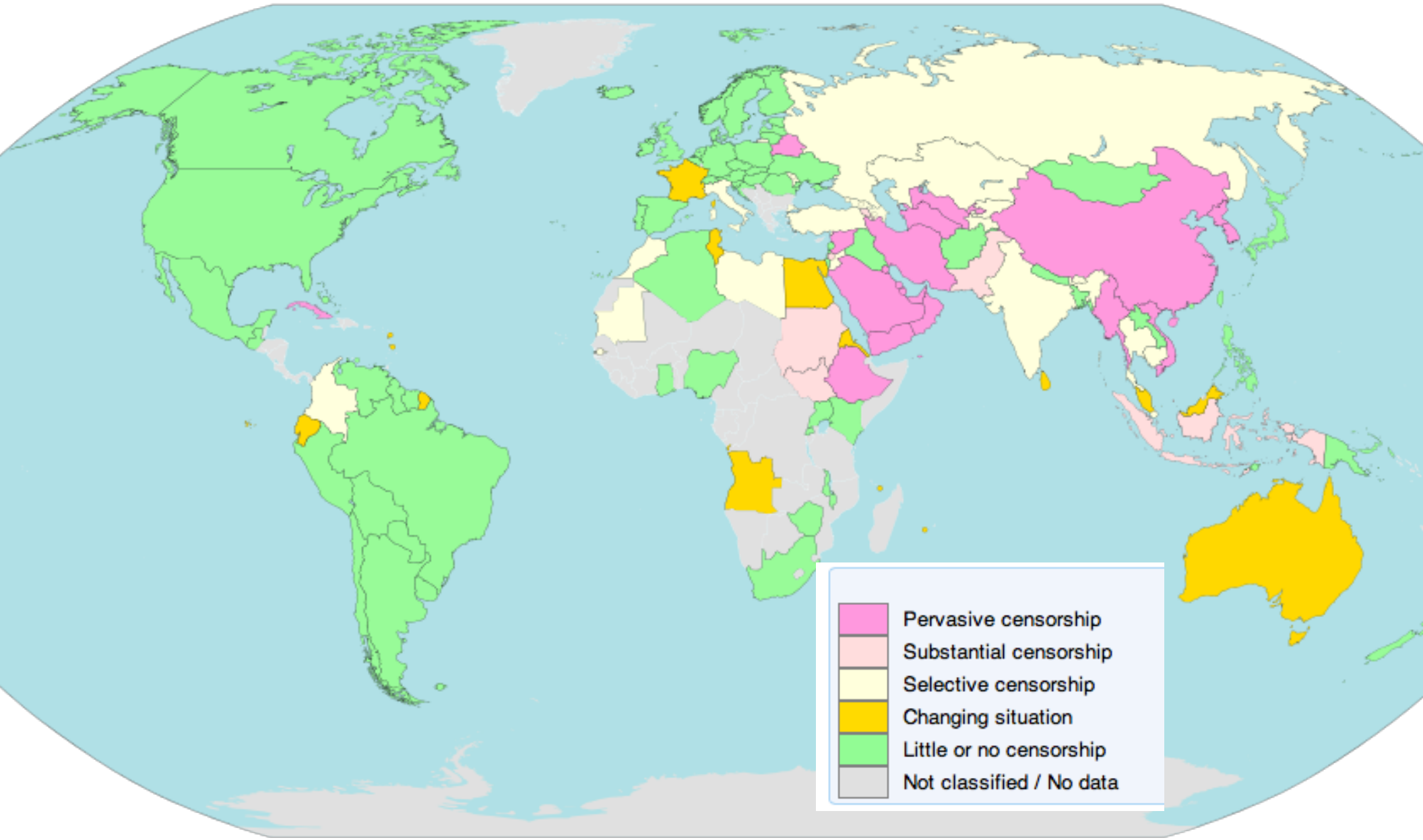
Protocol Misidentification Made Easy with

Format-Transforming Encryption

Kevin Dyer, Portland State University (did most of the hard work)
Scott Coull, RedJack
Thomas Ristenpart, University of Wisconsin-Madison
Thomas Shrimpton, Portland State University

OpenNet Initiative (ONI)
Reporters Without Borde
(via wikipedia; updated Ja



Magenta-colored countries are "**internet black holes**":

packet filtering

payload

TCP info	"HTTP: … free+speech …"

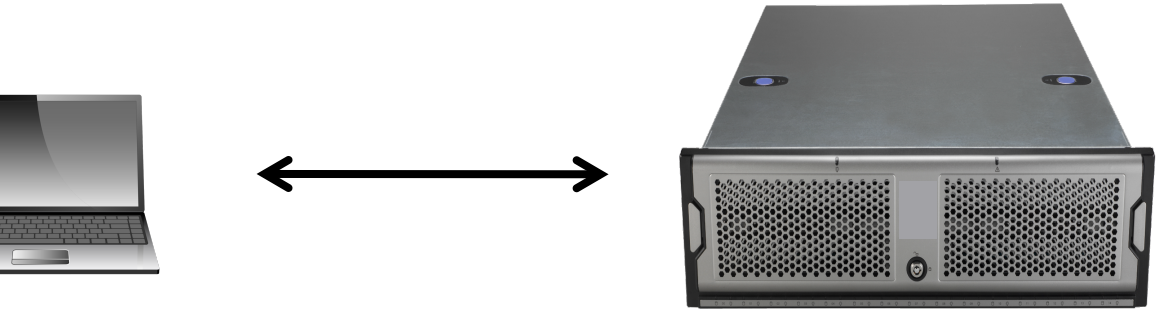
Nation-state level packet filter

A packet can tell you:

- source address
- destination address/port
- application-level protocols
- keywords in payloads
- …

payload

TCP info	"HTTP: … free+speech …"



Use a proxy se
e.g. Tor

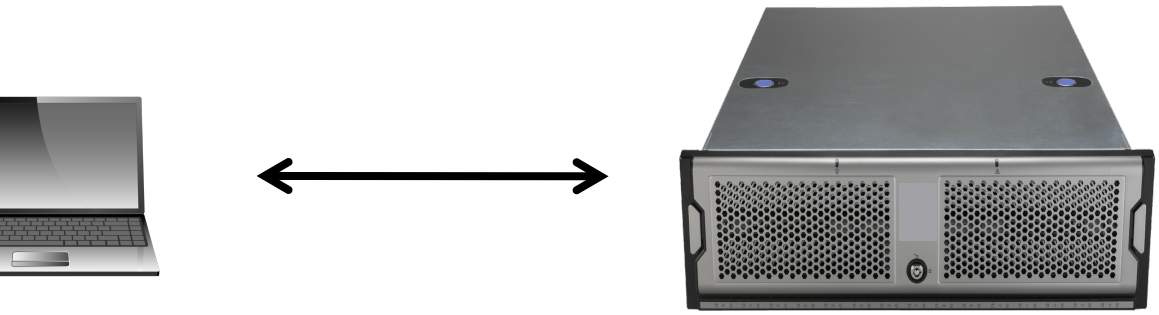
A packet can tell you:

- source address
- destination address/port
- application-level protocols
- keywords in payloads
- …

Deep Packet Inspection (DPI)

payload

TCP info	**"HTTP: … free+speech …"**



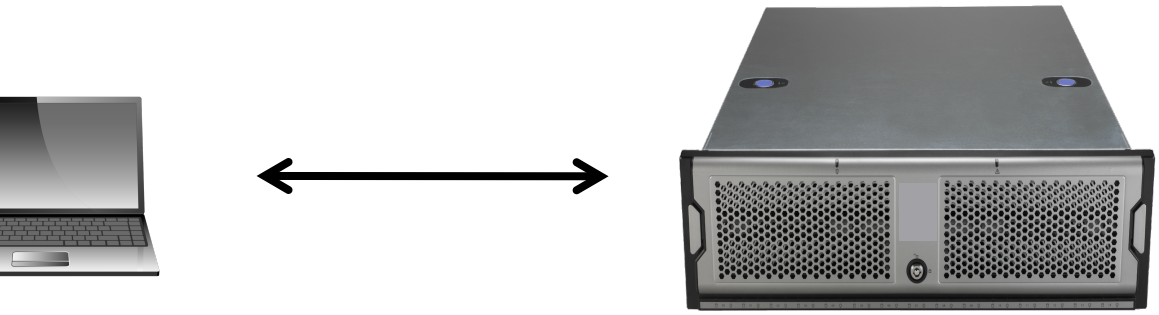
Making payload inform
unhelpful is the new ch

A packet can tell you:

- source address
- destination address/port
- application-level protocols
- keywords in payloads
- …

(TLS, SSH, VPNs, Tor)

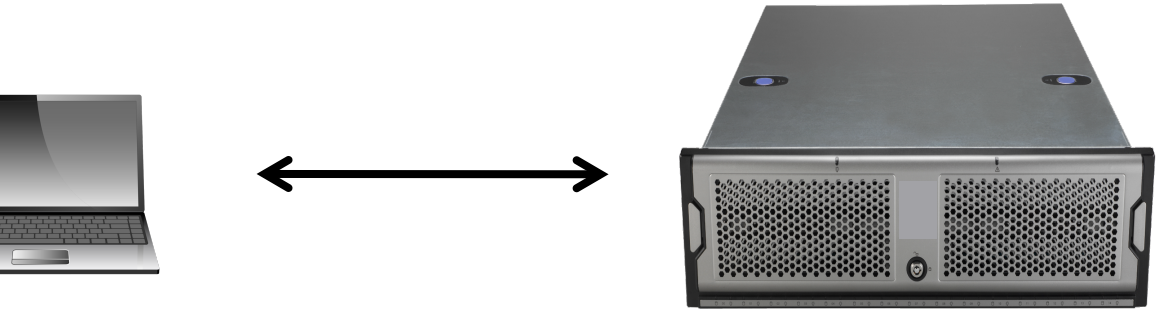
TCP info	"TLS..." ??? ... ???



Hides the protocol insi
the encrypted tunnel...

(TLS, SSH, VPNs, Tor)

TCP info	"TLS..." ??? ... ???



Hides the protocol insid
the encrypted tunnel...

But use of the encryption protocol is still visible.

n Bans Encryption

Soulskill on Tuesday August 30, 2011 @06
for-undecipherable-reasons dept.

Iran reportedly blocking encrypted traffic

The Iranian government is reportedly blocking access to websites tha

by **Jon Brodkin** - Feb 10 2012, 9:44pm IST

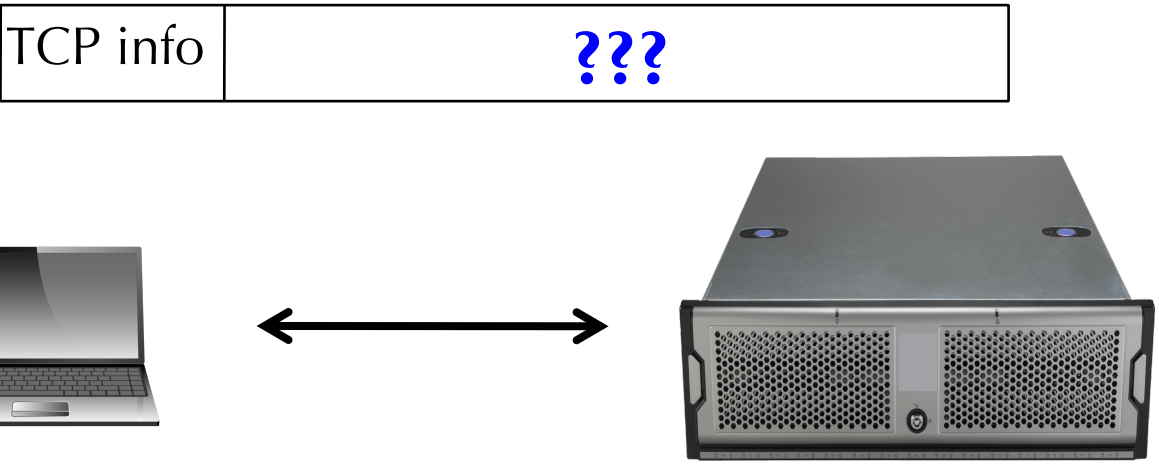
ne Great Firewall of China is Blocking Tor

Philipp Winter and Stefan Lindskog

NEWS

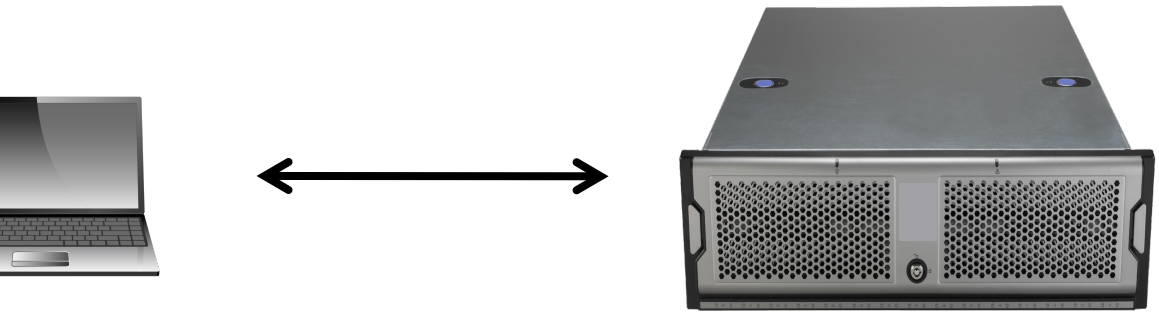
Ethiopian government blocks

(e.g. with a stream cipher)
(e.g. Tor's "obfs" pluggable tr



(e.g. with a stream cipher)
(e.g. Tor's "obfs" pluggable tr

TCP info	???



"I don't recognize this as any legitimate protocol."

What happens if DPI allow only whitelisted protocols

torus [Weinberg et al., 2012],

eMorph [Moghaddam et al. 2012],

Wave [Houmansadr et al., 2013], etc.

e represent nice steps in the right direction, but

Poor performance: 16-256Kbps reported (best case)

Inflexible: not quickly adaptable to changes in DPI rules.

e.g. what if you're using SkypeMorp
and Skype becomes blocked? (Ethic

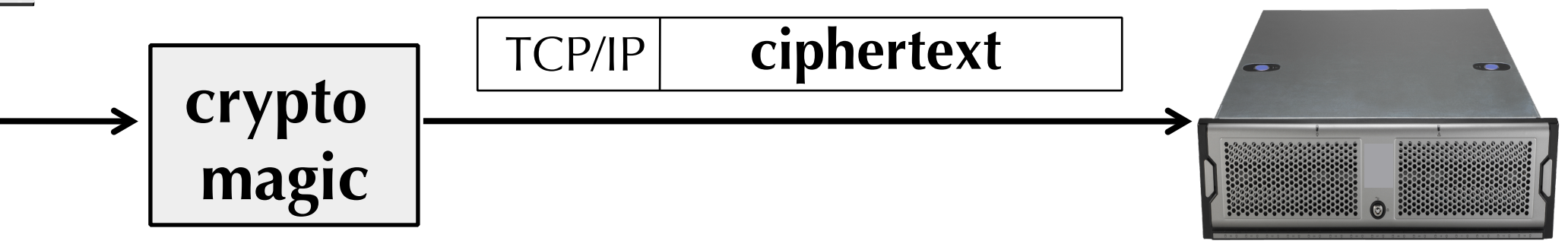
Not empirically validated: do they work against real DPI?

ur goal: to cause real DPI systems
o reliably misclassify our traffic

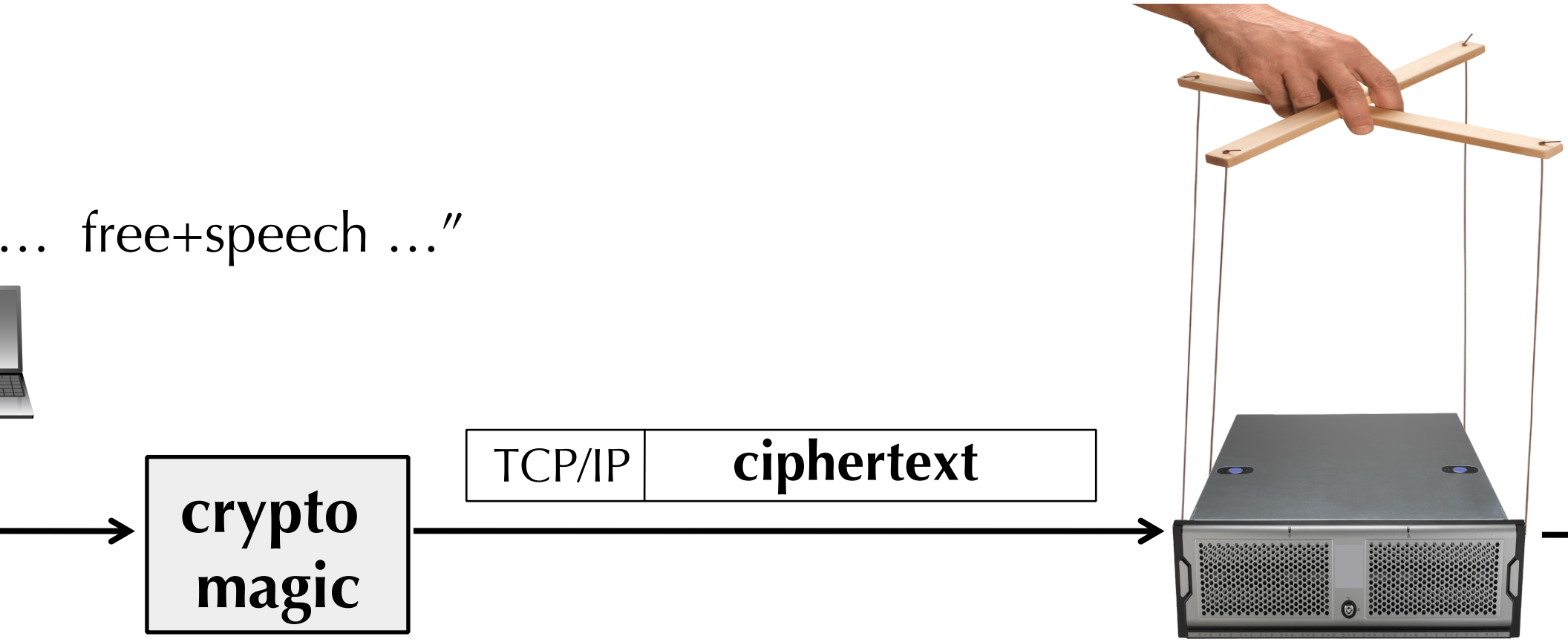
for example: HTTP misclassified as F

…. free+speech …"

"This is an
FTP mes
Let it pa



ur goal: to cause real DPI systems
o reliably misclassify our traffic
s whatever protocol we want.



duce a new cryptographic tool, Format Transforming Encry

acterize how real DPI systems make classification decision

ement an FTE-powered proxy system

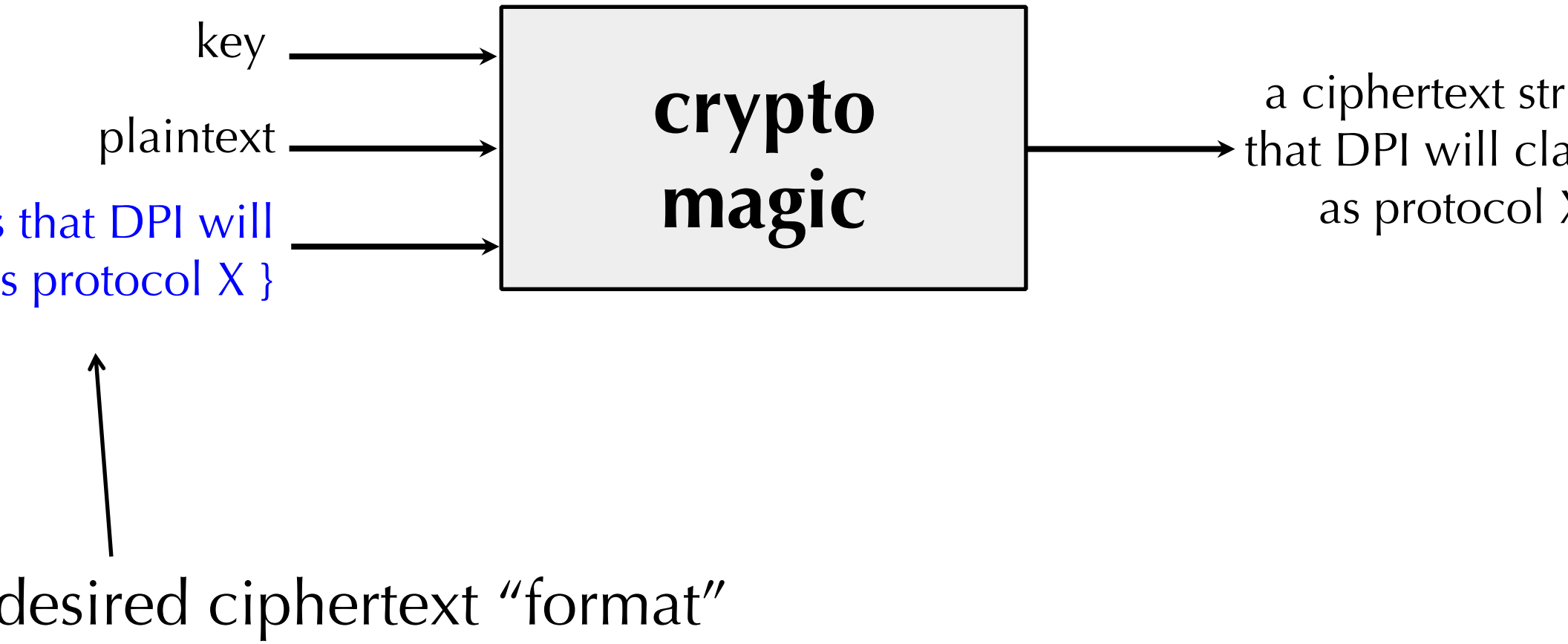
rically evaluate FTE against real DPI in the lab

ort on some live “field tests”


key
plaintext
crypto
magic
a ciphertext str
that DPI will cla
as protocol X

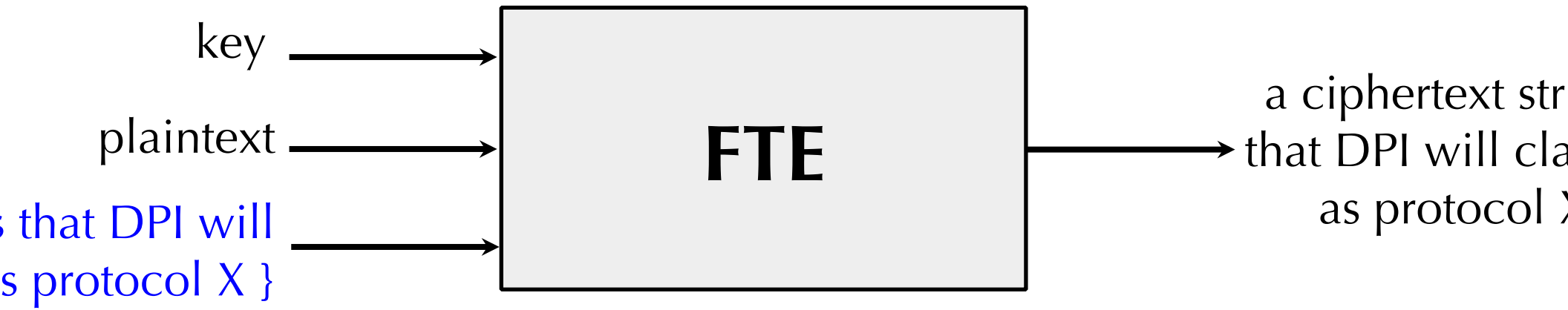
ok inspiration from Format Preserving Encryp

[Bellare et al.



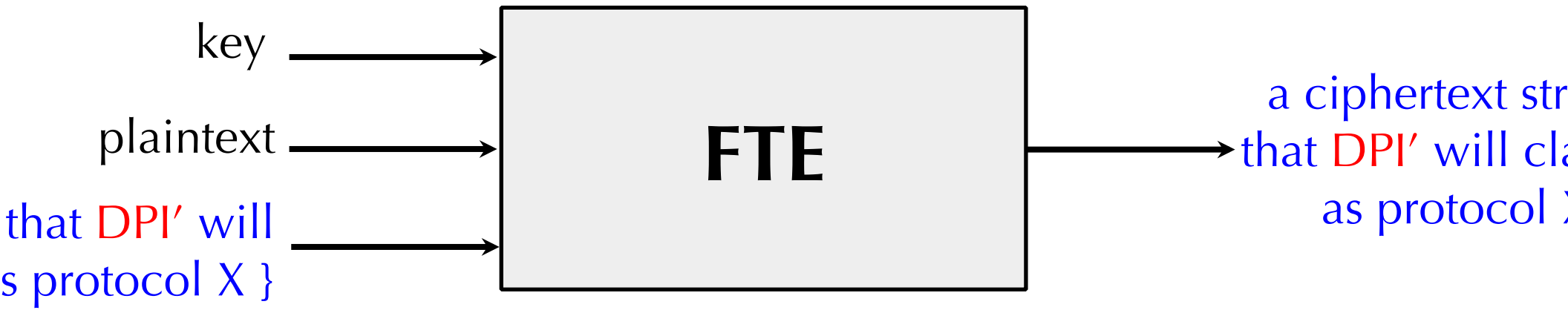
desired ciphertext “format”

Format-*Transforming* Encryption



Like traditional encryption, with the extra operational requirement that ciphertexts fall within the format.

hertext flexibility is built into the FTE syn



Conceptually, adapting to new DPI rules
requires changing only the format

do real DPI devices determine
at protocol a message belongs?

ystem	Classification Tool	P
appid		f
7-filter		f
YAF		f
bro		f
Probe		~300
DPI-X		~$

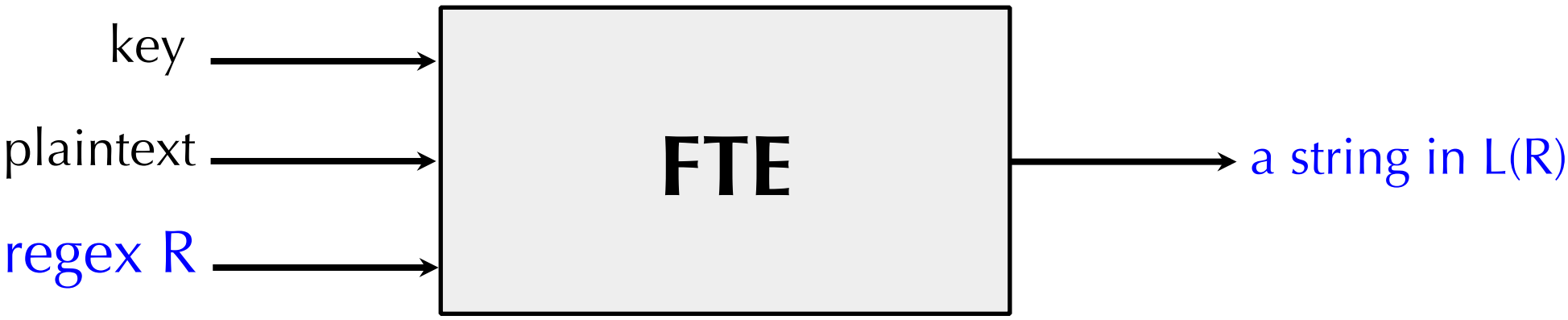
se grade DPI, well-known company

do real DPI devices determine
hat protocol a message belongs?

ystem	Classification Tool	P
appid	**Regular expressions**	f
7-filter	**Regular expressions**	f
YAF	**Regular expressions** (sometimes hierarchical)	f
bro	Simple **regular expression** triage, then additional parsing and heuristics	f
nProbe	Parsing and heuristics (many of them "**regular**")	~300
DPI-X	**???**	~$

Regular languages/expression
figure heavily in state of the a

Regular-expression-based FTE



ce the regex?

DPI is open source (appid, l7-filter, YAF), extract them!

hem manually, using RFCs and (when possible) DPI sourc

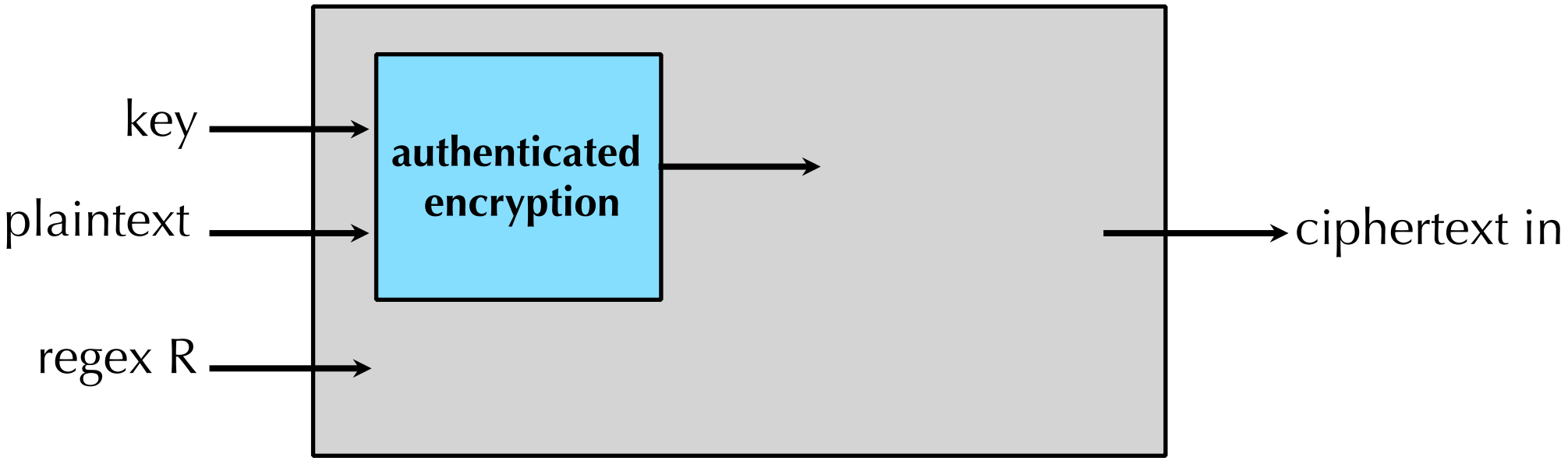
them from traffic that was allowed by the DPI.

...ing regex-based FTE

How should we realize regex-based FTE?

We want: Cryptographic protection for the plain
Ciphertexts in L(R)

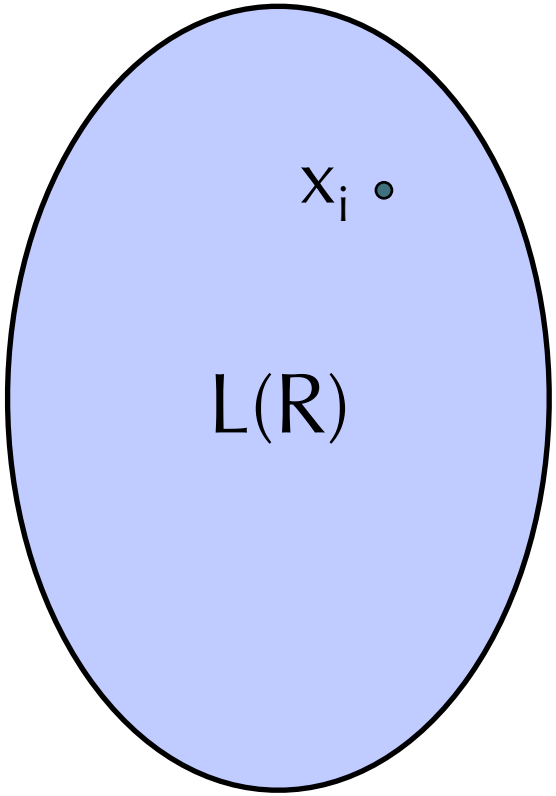
...ing regex-based FTE



How should we realize regex-based FTE?

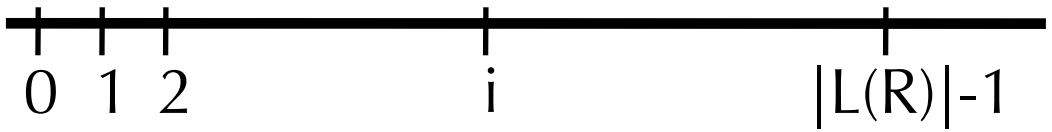
We want: Cryptographic protection for the plain
Ciphertexts in L(R)

[Bellare e



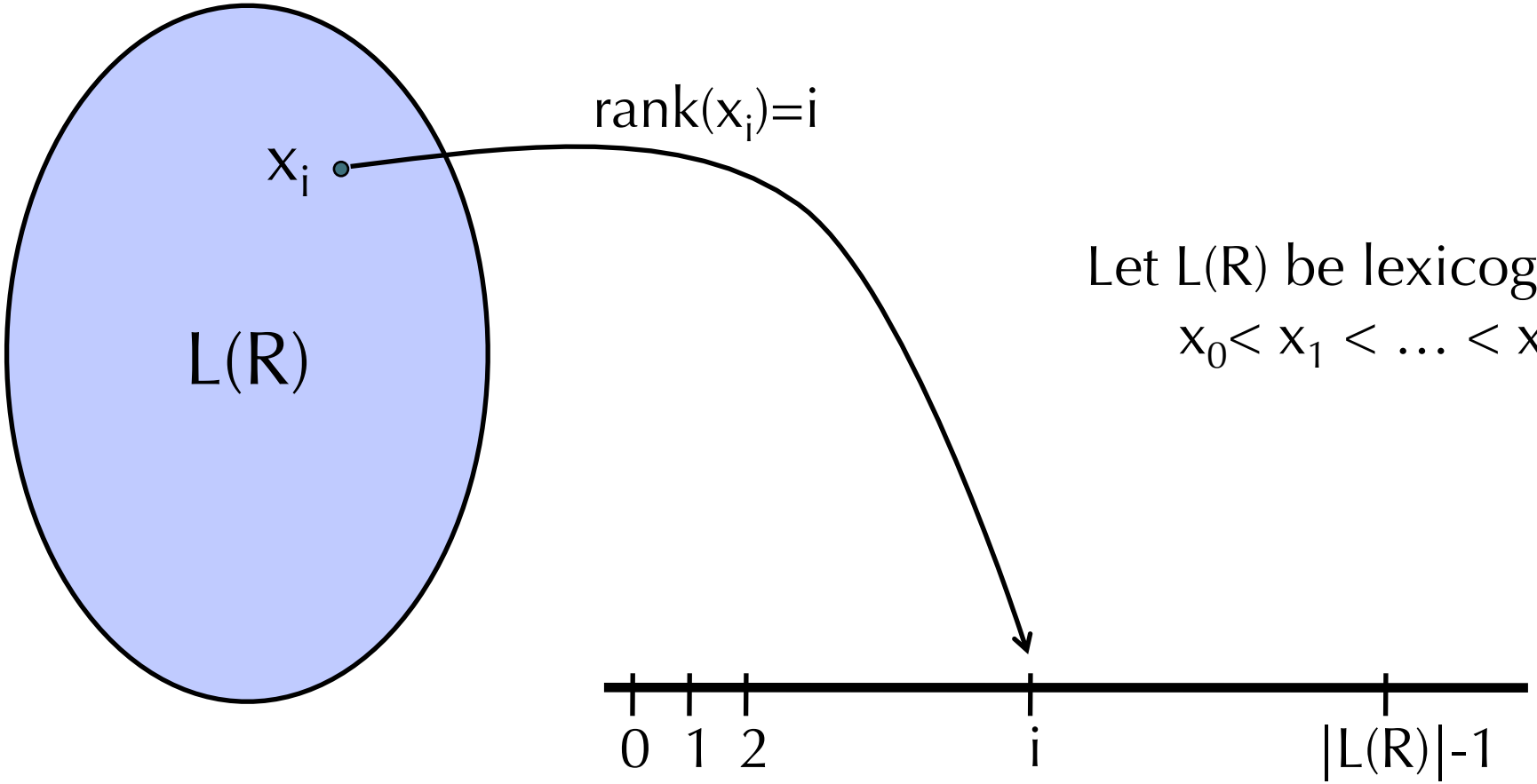
Let L(R) be lexicographically or

$x_0 < x_1 < \ldots < x_i < \ldots < x_{|L(R)|}$



DFA for L(R), there are efficient algorithms

[Bellare e



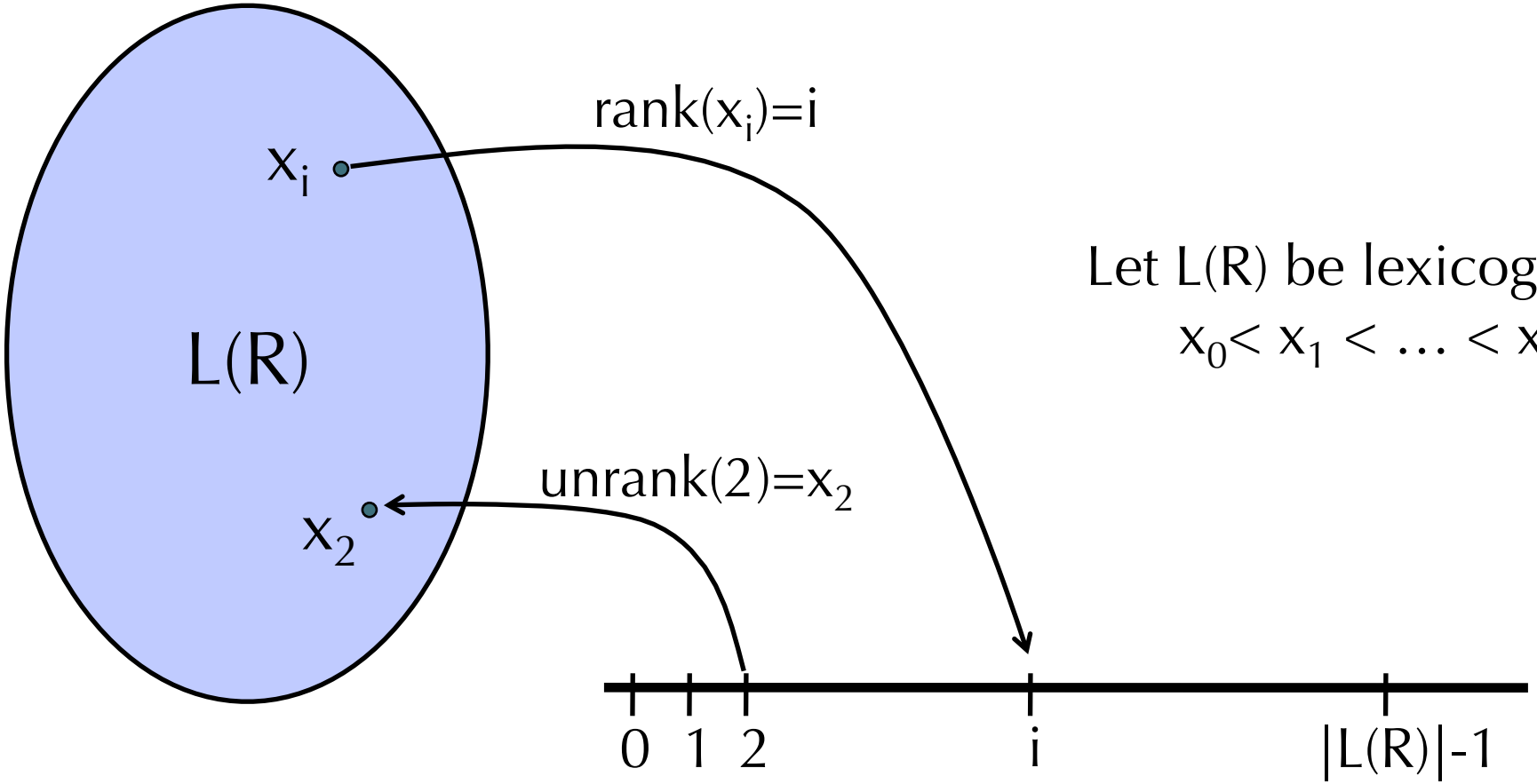
Let L(R) be lexicographically or

$x_0 < x_1 < \ldots < x_i < \ldots < x_{|L(R)|}$

DFA for L(R), there are efficient algorithms

rank: L(R) $\longrightarrow$ {0,1,…,|L(R)|-1}

[Belaire e



Let L(R) be lexicographically or

$x_0 < x_1 < \ldots < x_i < \ldots < x_{|L(R)|}$

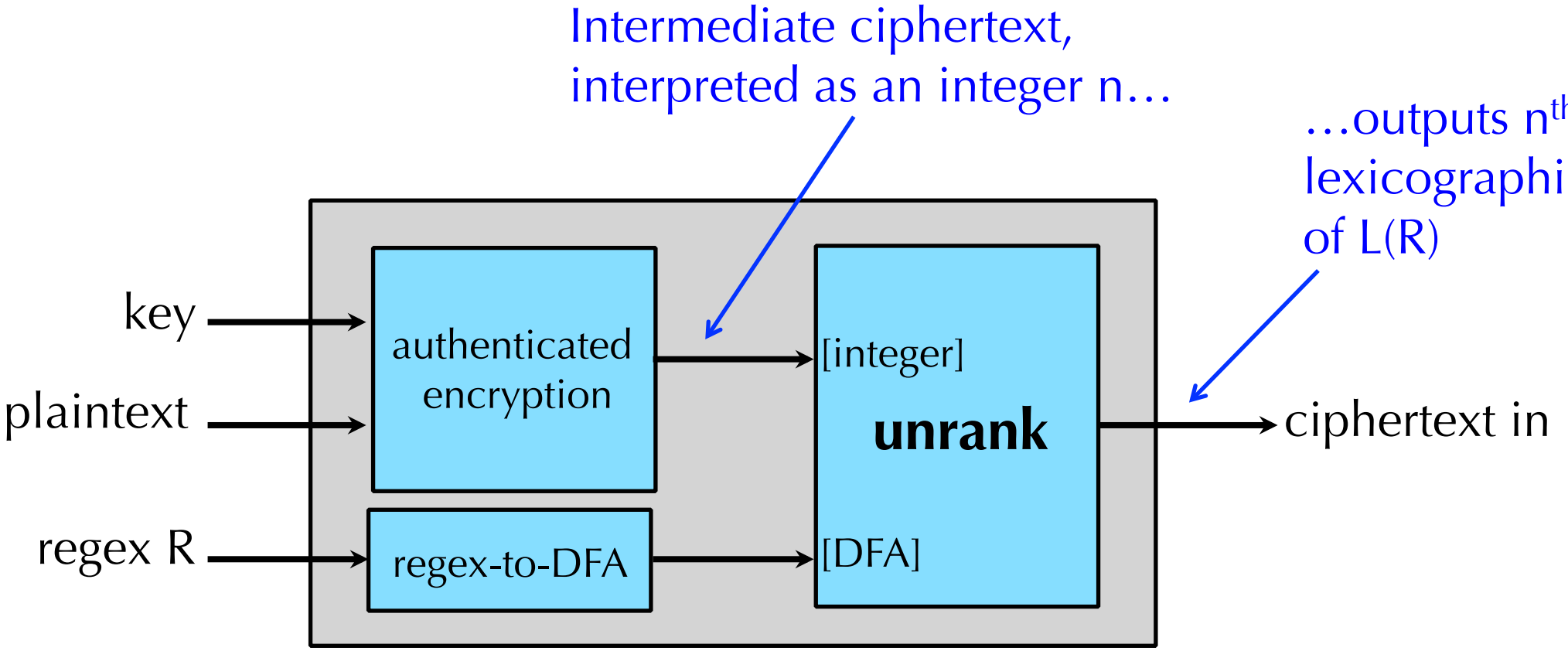
DFA for L(R), there are efficient algorithms

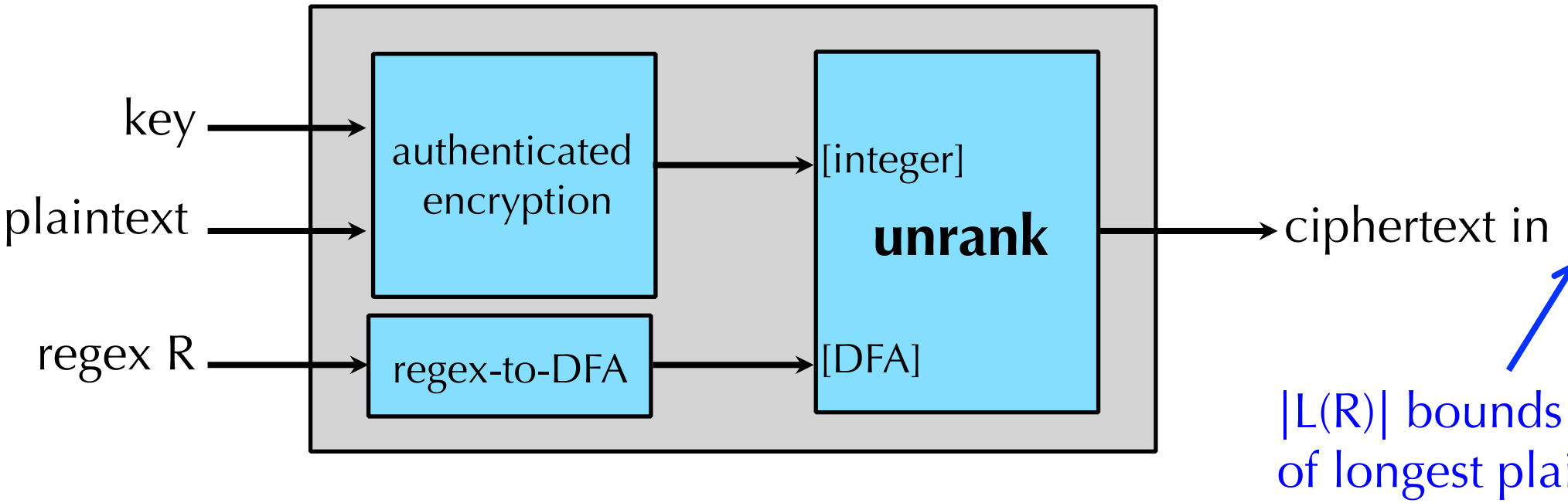
rank: L(R) $\longrightarrow$ {0,1,…,|L(R)|-1}

rank: {0,1,…,|L(R)|-1} $\longrightarrow$ L(R)

h that rank(unrank(i)) = i

With precomputed table
rank, unrank are *O*(n)


Intermediate ciphertext,
interpreted as an integer n…
…outputs n
lexicographi
of L(R)
key
plaintext
regex R
authenticated
encryption
regex-to-DFA
[integer]
unrank
[DFA]
ciphertext in



g very large languages leads to:

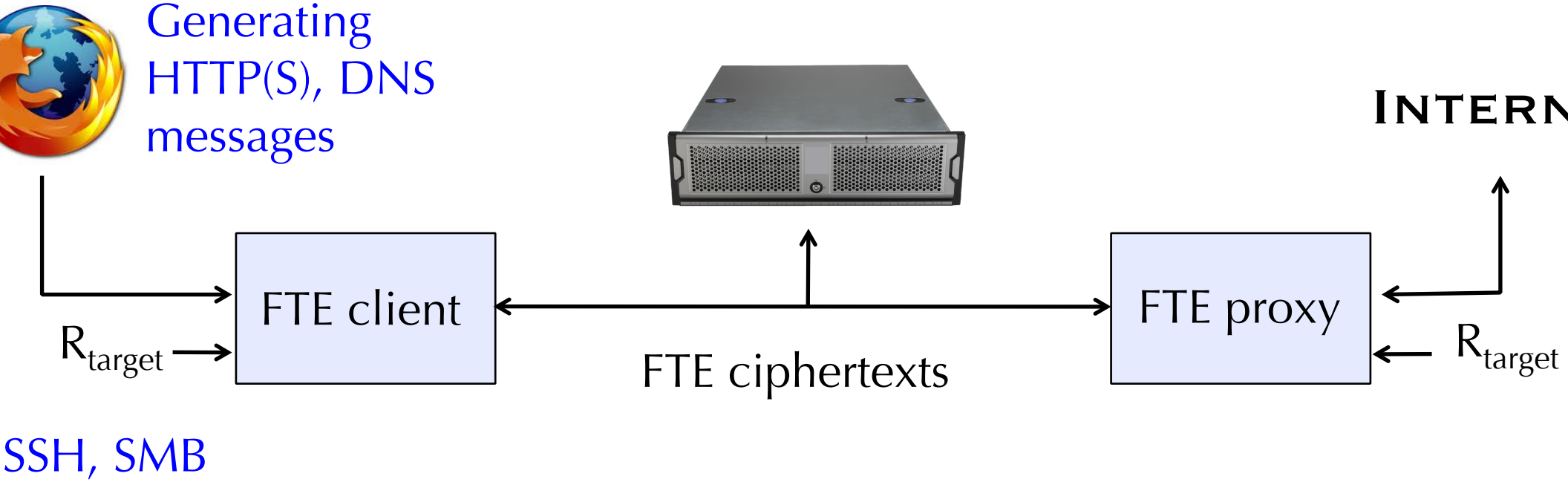
rge tables – naively, (#DFA states) x (length of longest plain

tency issues – waiting for long plaintext to buffer

nking, and using unrank(C_1), unrank(C_2), unrank(C_3), leads

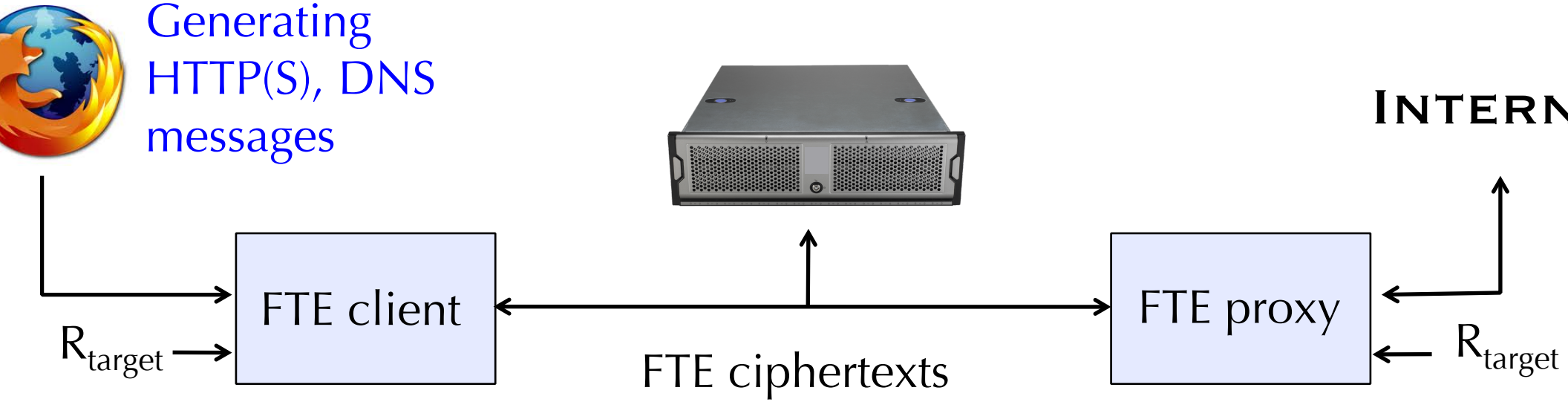
e case: Browsing the web through an FTE tunn

FTE "wins" if the DPI classifies the stream it se
as the target protocol



e case: Browsing the web through an FTE tunn

FTE "wins" if the DPI classifies the stream it se
as the target protocol



ch "target" format, we visited each of the Alexa Top 50 websites fi

DPI

gex

	appid	l7-filter	YAF	DPI-X
appid-http				
l7-http				
yaf-http1 **yaf-http2**				
appid-ssh				
l7-ssh				
yaf-ssh1 **yaf-ssh2**				
appid-smb				
l7-smb				
yaf-smb1 **yaf-smb2**				

gex

DPI

	appid	l7-filter	YAF	DPI-X
appid-http	**1.0**	0.0	1.0	1.0
l7-http	0.0	**1.0**	0.16	1.0
yaf-http1 **yaf-http2**	0.0 0.0	0.0 0.0	**1.0** **1.0**	1.0 1.0
appid-ssh	**1.0**	0.32	1.0	1.0
l7-ssh	0.16	**1.0**	0.16	1.0
yaf-ssh1 **yaf-ssh2**	1.0 1.0	0.21 0.31	**1.0** **1.0**	1.0 1.0
appid-smb	**1.0**	1.0	1.0	1.0
l7-smb	0.0	**1.0**	0.38	1.0
yaf-smb1 **yaf-smb2**	0.0 0.0	0.04 0.04	**1.0** **1.0**	1.0 1.0

DPI

	appid	l7-filter	YAF	DPI-X
appid-http	**1.0**	0.0	1.0	1.0
l7-http	0.0	**1.0**	0.16	1.0
yaf-http1	0.0	0.0	**1.0**	1.0
yaf-http2	0.0	0.0	**1.0**	1.0
appid-ssh	**1.0**	0.32	1.0	1.0
l7-ssh	0.16	**1.0**	0.16	1.0
yaf-ssh1	1.0	0.21	**1.0**	1.0
yaf-ssh2	1.0	0.31	**1.0**	1.0
appid-smb	**1.0**	1.0	1.0	1.0
l7-smb	0.0	**1.0**	0.38	1.0
yaf-smb1	0.0	0.04	**1.0**	1.0
yaf-smb2	0.0	0.04	**1.0**	1.0

gex

ince these all have 1.0 on the diagonals,
ve made “intersection” regexs for HTTP, SSH, SMB,

gex

DPI

	appid	l7-filter	YAF	DPI-X
appid-http	1.0	0.0	1.0	**1.0**
l7-http	0.0	1.0	0.16	**1.0**
yaf-http1	0.0	0.0	1.0	**1.0**
yaf-http2	0.0	0.0	1.0	**1.0**
appid-ssh	1.0	0.32	1.0	**1.0**
l7-ssh	0.16	1.0	0.16	**1.0**
yaf-ssh1	1.0	0.21	1.0	**1.0**
yaf-ssh2	1.0	0.31	1.0	**1.0**
appid-smb	1.0	1.0	1.0	**1.0**
l7-smb	0.0	1.0	0.38	**1.0**
yaf-smb1	0.0	0.04	1.0	**1.0**
yaf-smb2	0.0	0.04	1.0	**1.0**

!

sclassification rates with manual/learned rege

nually, using RFCs and
ossible) DPI source code.

DPI

gex

	appid	l7-filter	YAF	DPI-X	bro	nProbe
manual-http						
manual-ssh						
manual-smb						
learned-http						
learned-ssh						
learned-smb						

ia simple technique) from traffic that
ed by the DPI.

sclassification rates with manual/learned rege

DPI

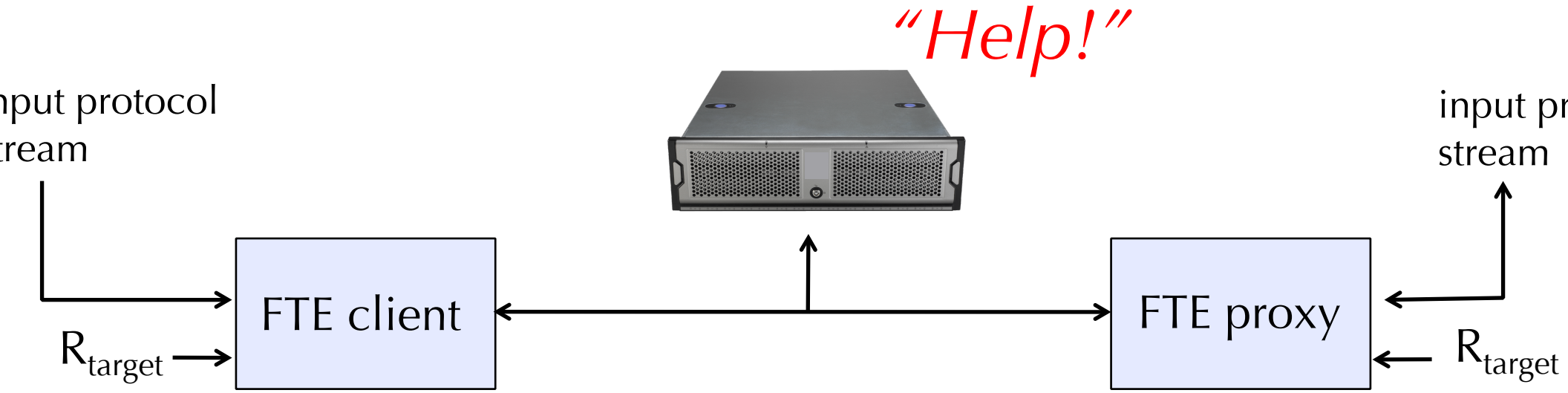
gex

	appid	l7-filter	YAF	DPI-X	bro	nProbe
manual-http						
manual-ssh						
manual-smb						
learned-http						
learned-ssh						0.0
learned-smb						

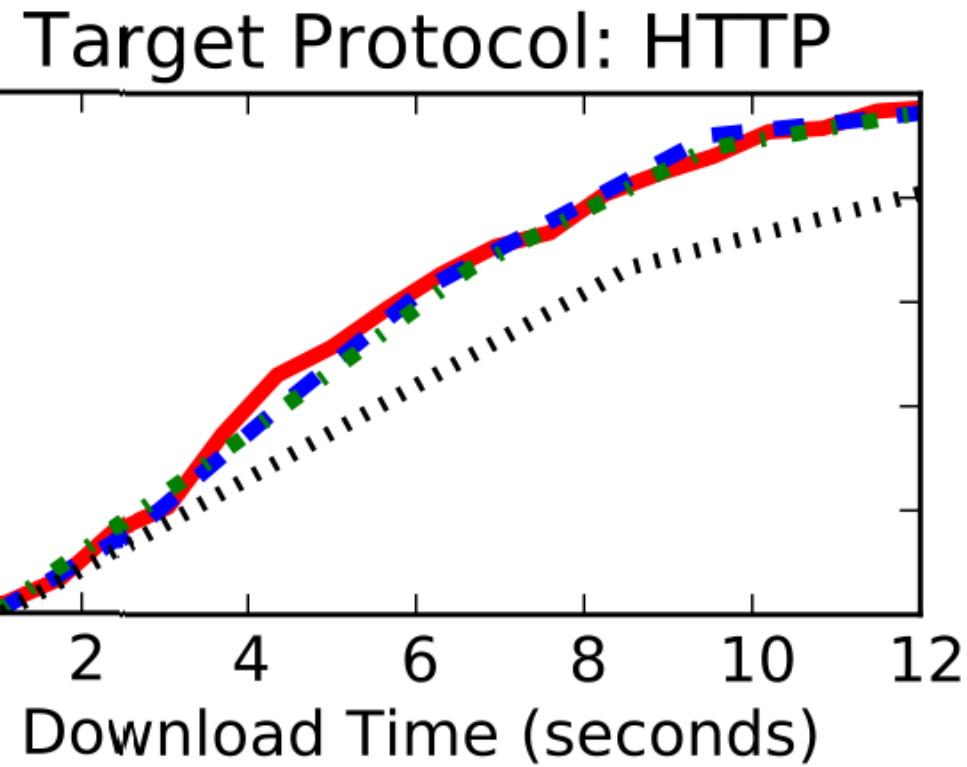
1.0

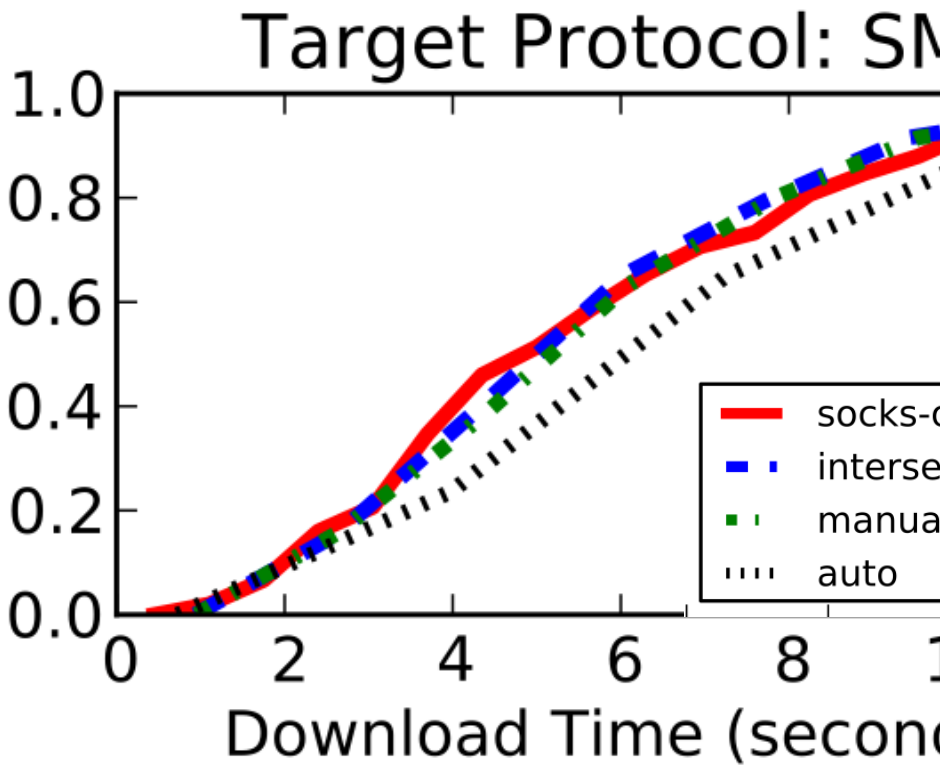
(except this,
explain in t

Punchline: regex-based FTE can make real DPI say whatever we want it to.



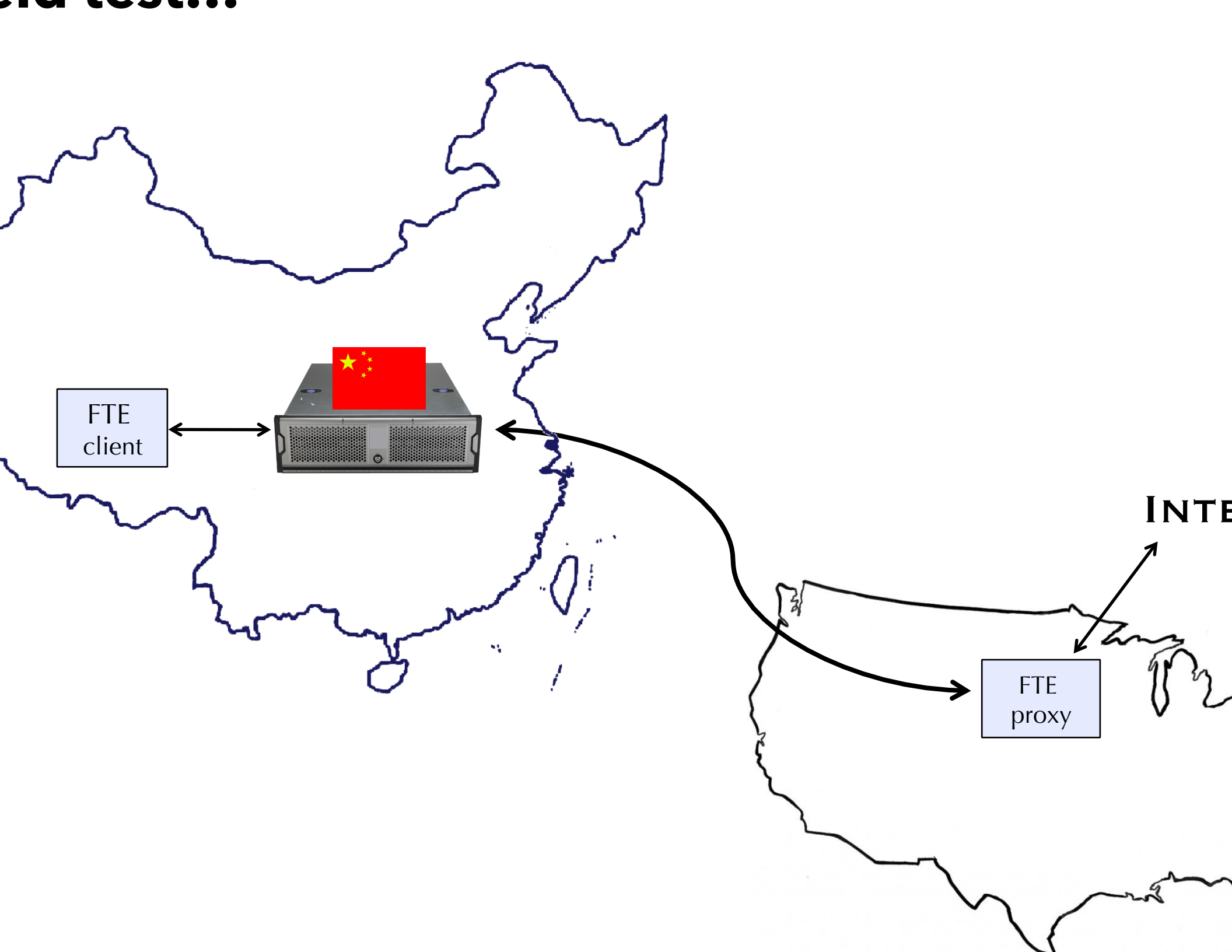
Web-browsing performance





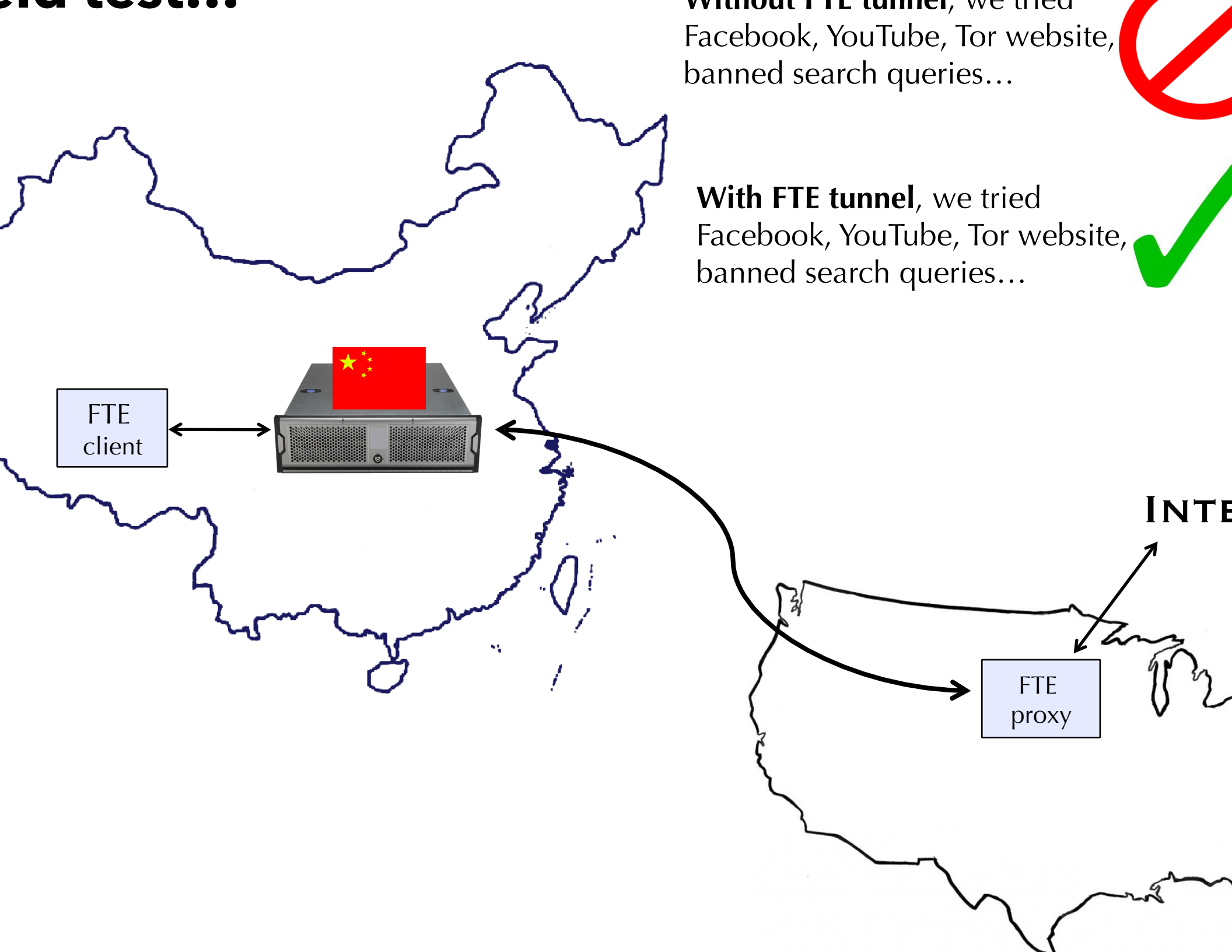
chline: FTE or SSH tunnel result in the sam user web-browsing experience

ld test...



Without FTE tunnel, we tried Facebook, YouTube, Tor website, banned search queries…

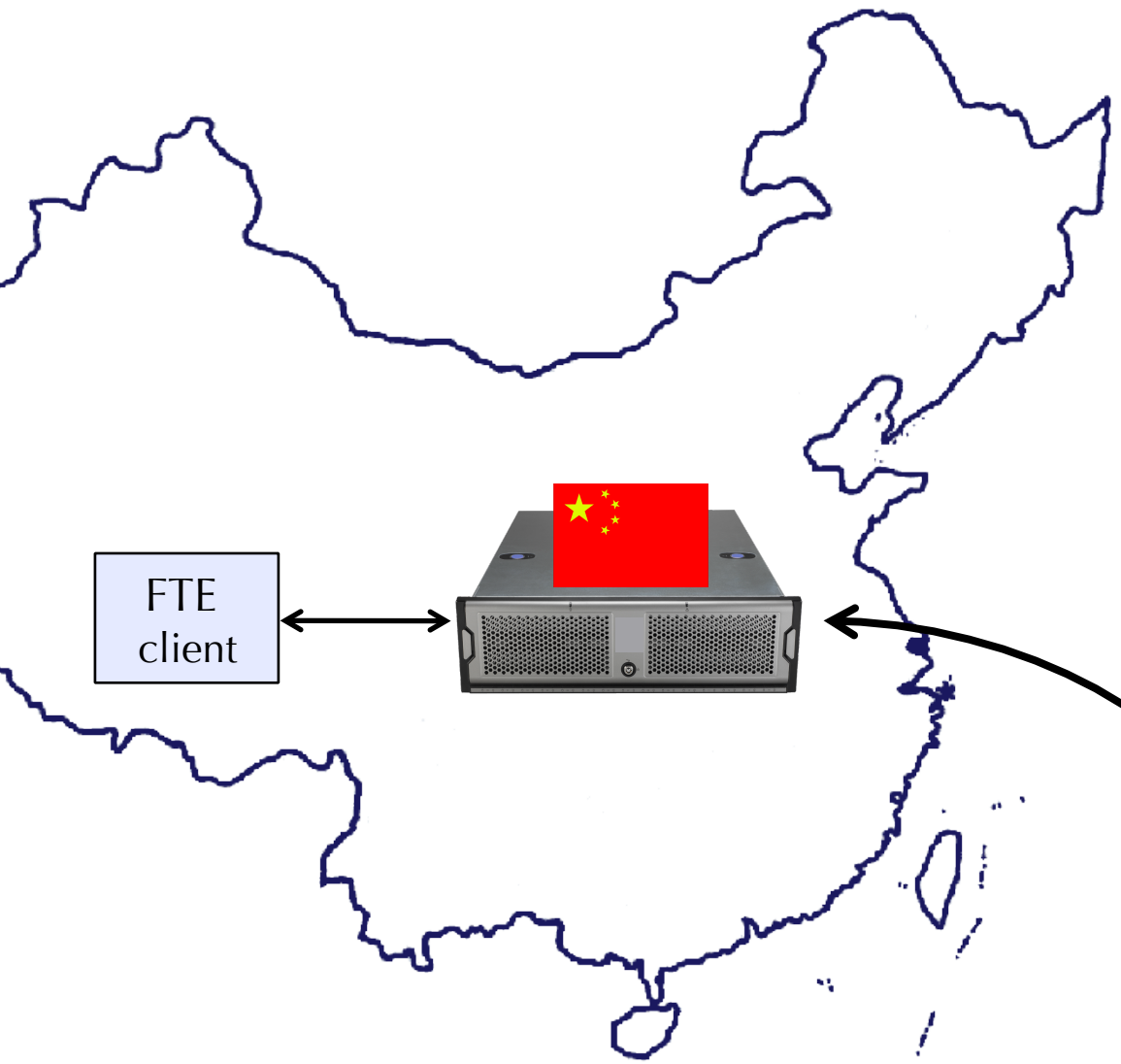
With FTE tunnel, we tried Facebook, YouTube, Tor website, banned search queries…

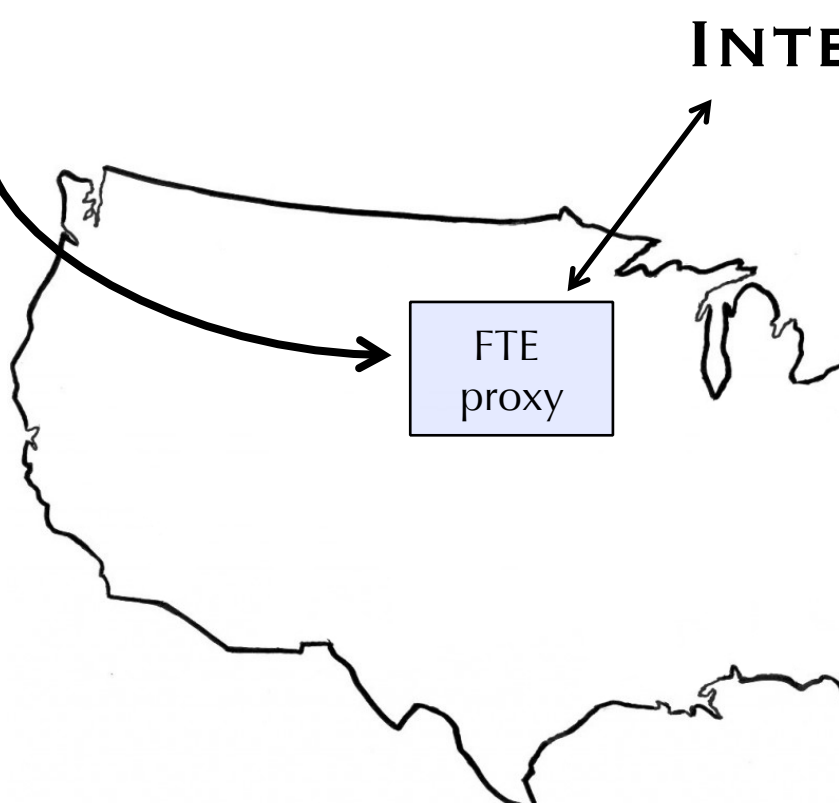


ld test...

Without FTE tunnel, we tried Facebook, YouTube, Tor website, banned search queries…

With FTE tunnel, we tried Facebook, YouTube, Tor website, banned search queries…





TE to download Tor bundle:

hout FTE: "active blacklisting" attack on proxy
ough FTE: no problems

rious tests every 5 minutes for one month,

FTE is open source,
runs on multiple platforms/OS,
and **fully integrated with** 

We even have a nice website:
https://fteproxy.org/

Get it, run it, help us make it better!