The Past and Future of TLS

Team Pineapple

Schedule

0930-1000: Past Problems and Examples (that still exist)
1005-1035: TLS 1.4 and Beyond
1045-????: Candidate Talk in Rice about TLS Interception!

Next Friday: Potential of Post Quantum in TLS

HTTPS Traffic Analysis
& Website Fingerprinting

In the news...

Outrage grows over **Congress**' **Internet privacy** vote
CNNMoney - Mar 29, 2017
Outrage is growing at Republicans following a controversial vote Tuesday to repeal **Internet privacy** protections that were approved by the ...
How **Congress** can fix **Internet privacy** rule
Opinion - CNN - Mar 29, 2017
View all

Sold out by **Congress** on **internet privacy**
The Denver Post - Apr 3, 2017
Congress has sent President Donald Trump legislation that would kill an online **privacy** regulation, which could allow **internet** providers to sell ...
Trespassing on **Internet privacy**: Our view
Opinion - USA TODAY - Apr 4, 2017
View all

Washington fights for **internet privacy** that **Congress** took away
Crosscut - Apr 5, 2017
On Monday, President Donald Trump signed a law that allows **internet** providers to sell your personal information without your permission.
Message Received: You Want **Privacy** Protections While Surfing ...
KUOW News and Information - Apr 6, 2017
View all

Kai Teoh: Our **internet privacy** is dead. **Congress** sold us out.
The Spokesman-Review - Apr 10, 2017
States have started writing their own legislation to protect broadband **privacy** after

Secure https://techcrunch.com/2017/03/28/house-vote-sj-34-isp-regulations-fcc/

News Startups Mobile Gadgets Enterprise Social Europe Trending Snap Amazon NASA

Congress just voted to let internet providers sell your browsing history

Posted Mar 28, 2017 by Taylor Hatmaker (@tayhatmaker)

Next Story

Less than a week after the Senate voted to empower internet service providers to freely share private user data with advertisers, the House has weighed in, too.

Today in a 215-205 vote on Senate Joint Resolution 34 (H. Res. 230), the House voted to repeal broadband privacy regulations that the Obama administration's FCC introduced in 2016. In a narrower vote than some expected, 15 Republicans broke rank to join the 190 Democrats who voted against the repeal. The FCC rules, designed to protect consumers, required ISPs to seek consent from their customers in order to share their sensitive private data (it's worth noting that ISPs can collect it either way). For consumers, the rollback is a bad deal no matter how you slice it

Do companies use my personal information now?

Yes. Google and Facebook aggregate demographic and other profile data to offer advertisers desirable audiences. "The distinguishing factor here is that consumers choose to use Google and Facebook's services and implicitly agree to trade some privacy for the convenience of their services," Belkoura said. Since customers pay ISPs directly, they should expect "privacy is respected," he said.

https://www.usatoday.com/story/tech/news/2017/04/04/isps-can-now-collect-and-sell-your-data-what-know-internet-privacy/100015356/

ISPs have a unique vantage point that differ from Google and Facebook because they have the ability to capture all network traffic.

And ISPs are not the only ones with access to network traffic:
- Passive eavesdropping
- BGP Hijacking
- Remote Traffic Analysis

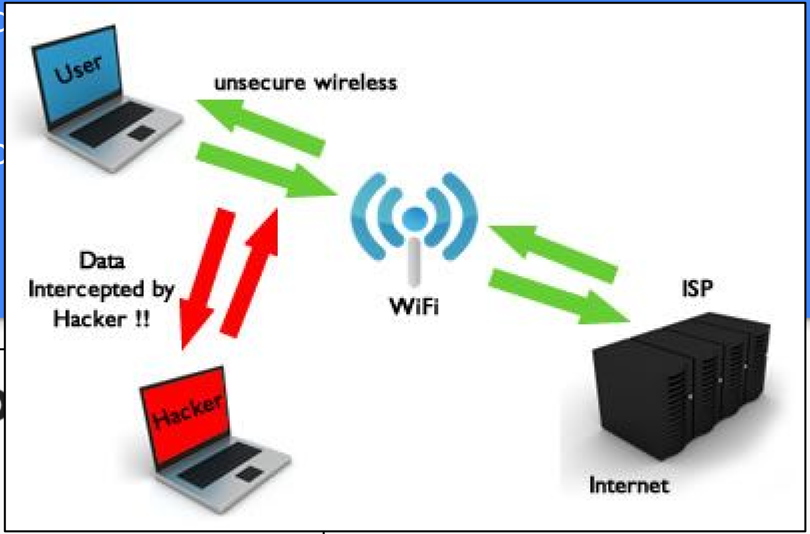
Do companies use my personal information now?

Yes. Google and Facebook aggregate demographic and other profile data to offer advertisers desirable audiences. "The distinguishing factor here is that consumers choose to use Google and Facebook's services and implicitly agree to trade some privacy for the convenience of their services," Belkoura said. Since customers pay ISPs directly, they should expect "privacy is respected," he said.

https://www.usatoday.com/story/tech/news/2017/04/04/isps-can-now-collect-and-sell-your-data-what-know-internet-privacy/100015356/

ISPs have a unique vantage point that differ from Google and
to capture all network traffic.

And ISPs are not the only ones with access to network traffic
- Passive eavesdropping
- BGP Hijacking
- Remote Traffic Analysis



Do companies use my personal info
now?

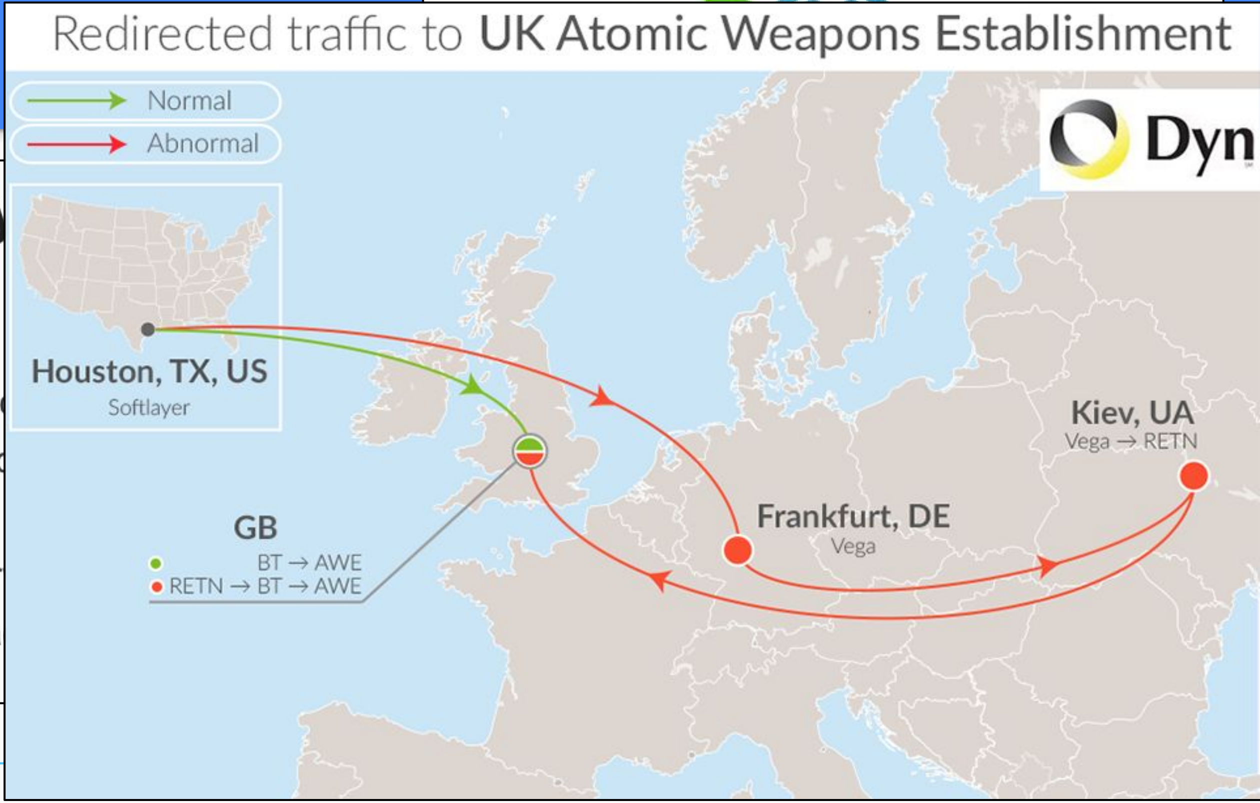
Yes. Google and Facebook aggregate demographic and other profile data to offer advertisers desirable audiences. "The distinguishing factor here is that consumers choose to use Google and Facebook's services and implicitly agree to trade some privacy for the convenience of their services," Belkoura said. Since customers pay ISPs directly, they should expect "privacy is respected," he said.

https://www.usatoday.com/story/tech/news/2017/04/04/isps-can-now-collect-and-sell-your-data-what-know-internet-privacy/100015356/

ISPs have a unique vantage point that differ from Google and
to capture all network traffic.

And ISPs are not the only ones with access to network traffic

- Passive eavesdropping
- BGP Hijacking
- Remote Traffic Analysis



Do companies use my
now?

Yes. Google and Facebook aggregate
advertisers desirable audiences. "The
choose to use Google and Facebook's
privacy for the convenience of their ser
ISPs directly, they should expect "priva

https://www.usatoday.com/story/tech/news/201

ISPs have a unique vantage point that differ from Google and to capture all network traffic.

And ISPs are not the only ones with access to network traffic
- Passive eavesdropping
- BGP Hijacking
- Remote Traffic Analysis

Do companie now?

Yes. Google and Faceb advertisers desirable au choose to use Google a privacy for the convenie ISPs directly, they shou

https://www.usatoday.com/s

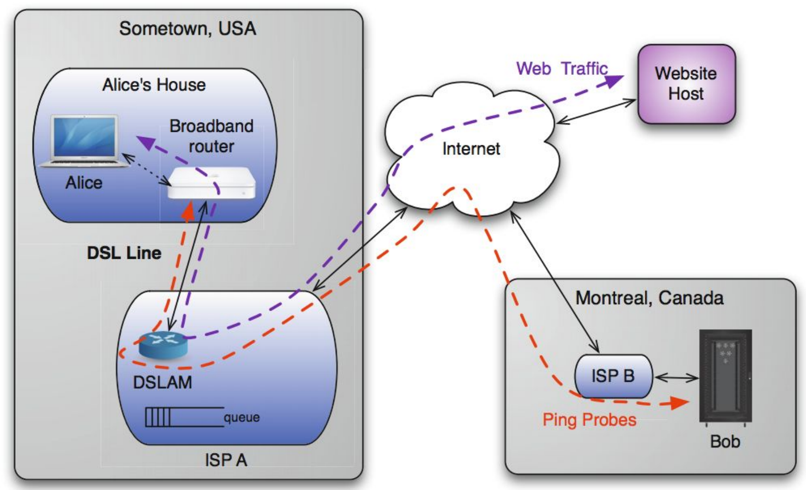


Fig. 1. Queueing side channel. Bob remotely sends probes to Alice's router to infer her activities.

ISPs have a unique vantage
to capture all network traffi

And ISPs are not the only o
- Passive eavesdropping
- BGP Hijacking
- Remote Traffic Analysis

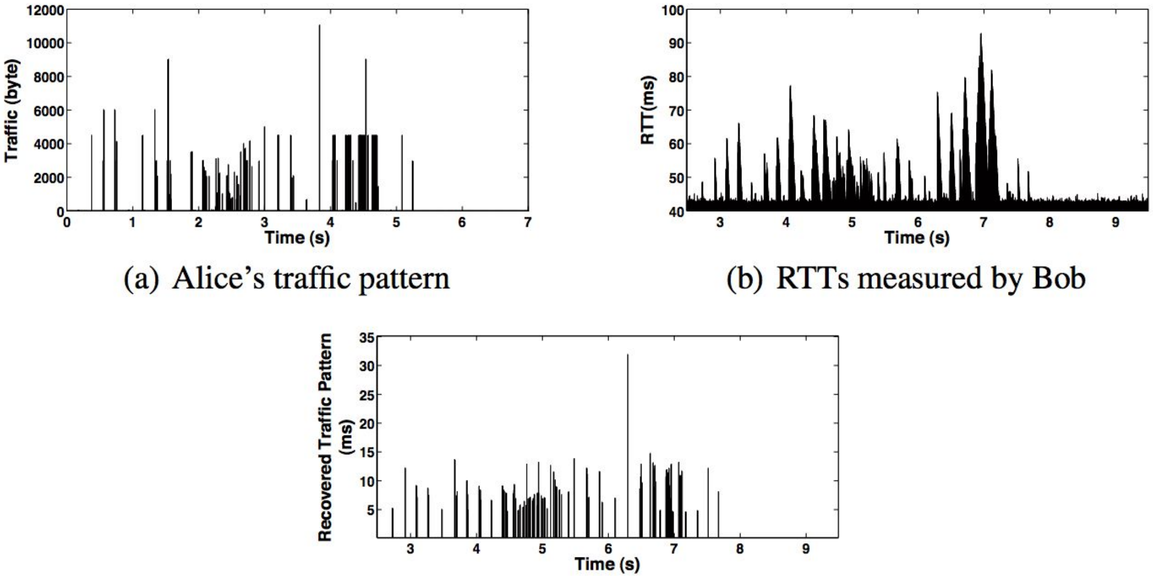
Do companie
now?

Yes. Google and Faceb
advertisers desirable au
choose to use Google a
privacy for the convenie
ISPs directly, they shou

https://www.usatoday.com/s

60 X

Fig. 1. Queueing side channel. Bob remotely sends probes to Alice's router to infer her activities.



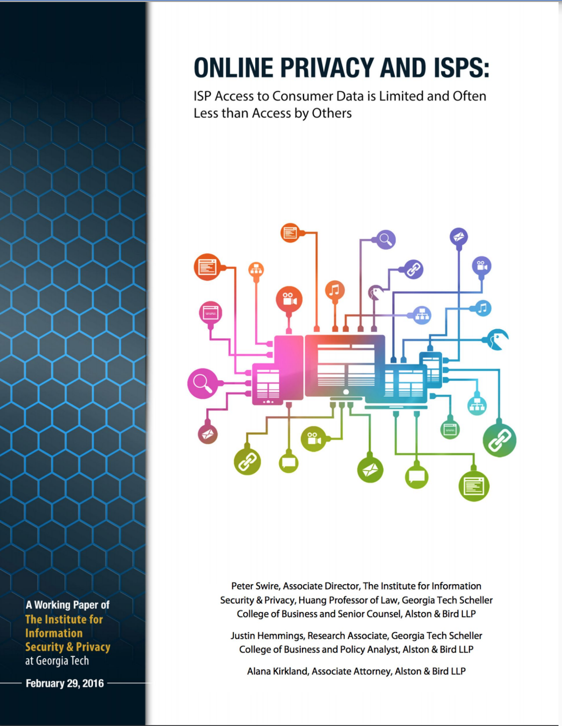
(a) Alice's traffic pattern

(b) RTTs measured by Bob

(c) Recovered traffic pattern

Fig. 2. Real traffic on a DSL vs. probe RTTs. Alice resides in Champaign, IL, while Bob is located in Montereal, Canada.

“This Working Paper is intended to provide information useful to Congress, federal agencies, and the general public in consideration of online privacy issues.” (2016)


ONLINE PRIVACY AND ISPS:
ISP Access to Consumer Data is Limited and Often Less than Access by Others

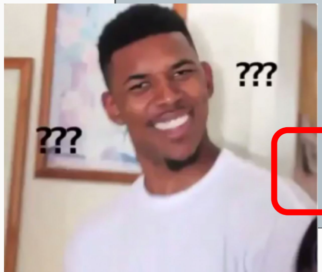
Peter Swire, Associate Director, The Institute for Information Security & Privacy, Huang Professor of Law, Georgia Tech Scheller College of Business and Senior Counsel, Alston & Bird LLP

Justin Hemmings, Research Associate, Georgia Tech Scheller College of Business and Policy Analyst, Alston & Bird LLP

Alana Kirkland, Associate Attorney, Alston & Bird LLP

A Working Paper of
The Institute for Information Security & Privacy
at Georgia Tech

February 29, 2016



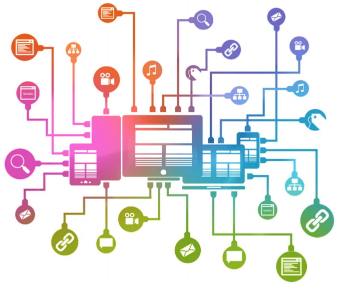
“This Working Paper is intended to provide information useful to Congress, federal agencies, and the general public in consideration of online privacy issues.” (2016)

A Working Paper of
The Institute for Information Security & Privacy at Georgia Tech

February 29, 2016

ONLINE PRIVACY AND ISPS:

ISP Access to Consumer Data is Limited and Often Less than Access by Others



Peter Swire, Associate Director, The Institute for Information Security & Privacy, Huang Professor of Law, Georgia Tech Scheller College of Business and Senior Counsel, Alston & Bird LLP

Justin Hemmings, Research Associate, Georgia Tech Scheller College of Business and Policy Analyst, Alston & Bird LLP

Alana Kirkland, Associate Attorney, Alston & Bird LLP

Chapter 7: Browsers, Internet Video, and E-commerce

This Chapter more briefly examines three additional contexts that are relevant to non-ISP collection of data. Major browsers vary in how extensively they collect user information, but the amount collected can be significant. For instance, most browsers carefully analyze user behavior to suggest search terms while the user is typing and then later use that information to autofill online forms by default. When users are logged-in, their browsing information can be integrated with information from the other contexts engaged in by that browser company. By contrast, ISPs are not developers of any of the major browsers and do not have access to this information.

For Internet video accessed through a browser or a mobile app, the party hosting the video content has the same ability to gain information about the user as any other site hosting content. Third-party ads are served in connection with video content the same as for other content. When Internet video is delivered over a HTTPS connection, the ISP can only see the host domain.

O RLY?

Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow

2010

Shuo Chen
Microsoft Research
Microsoft Corporation
Redmond, WA, USA

Rui Wang, XiaoFeng Wang, Kehuan Zhang
School of Informatics and Computing
Indiana University Bloomington

WEDNESDAY, FEBRUARY 8, 2012

I can still see your actions on Google Maps o SSL

I Know What You Saw Last Minute - The Chrom Browser Case

2016

Ran Dubin
Communication Systems Engineering
Ben-Gurion University of the Negev
Israel

Amit Dvir
Center for Cyber Technologies
Department of Computer Science
Ariel University
Israel

I Know Why You Went to the Clinic: Risks and Realization of HTTPS Traffic Analysis

2014

Brad Miller[1], Ling Huang[2], A. D. Joseph[1], and J. D. Tygar[1]

[1] UC Berkeley
[2] Intel Labs

Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application

2016

Jonathan Muehlstein*, Yehonatan Zion*, Maor Bahumi†, Itay Kirshenb
Ran Dubin‡, Amit Dvir*, Ofir Pele*†
* Center for Cyber Technologies, Department of Computer Science, Ariel Ur
† Center for Cyber Technologies, Department of Electrical and Electronics Engineering
‡ Department of Communication Systems Engineering, Ben-Gurion University o

Identifying HTTPS-Protected Netflix Videos in Real-Time

2017

Andrew Reed, Michael Kranch
Dept. of Electrical Engineering and Computer Science
United States Military Academy at West Point
West Point, New York, USA
{andrew.reed, michael.kranch}@usma.edu

Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow

2010

Shuo Chen
Microsoft Research
Microsoft Corporation
Redmond, WA, USA
shuochen@microsoft.com

Rui Wang, XiaoFeng Wang, Kehuan Zhang
School of Informatics and Computing
Indiana University Bloomington
Bloomington, IN, USA
[wang63, xw7, kehzhang]@indiana.edu

Abstract– With software-as-a-service becoming mainstream, more and more applications are delivered to the client through the Web. Unlike a desktop application, a web application is split into browser-side and server-side components. A subset of the application's internal information flows are inevitably exposed on the network. We show that despite encryption, such a side-channel information leak is a realistic and serious threat to user privacy. Specifically, we found that surprisingly detailed sensitive information is being leaked out from a number of high-profile, top-of-the-line web applications in healthcare, taxation, investment and web search: an eavesdropper can infer the illnesses/medications/surgeries of the user, her family income and investment secrets, despite HTTPS protection; a stranger on the street can glean enterprise employees' web search queries, despite WPA/WPA2 Wi-Fi encryption. More importantly, the root causes of the problem are some fundamental characteristics of web applications: stateful communication, low entropy input for better interaction, and significant traffic distinctions. As a result, the scope of the problem seems industry-wide. We further present a concrete analysis to demonstrate the challenges of mitigating such a threat, which points to the necessity of a disciplined engineering practice for side-channel mitigations in future web application developments.

- WPA, WPA2 do not hide packet sizes

- Web apps leak through:
  - Low entropy input for better interaction (autocomplete, autosuggestion, AJAX, "increasing use of highly interactive and dynamic web interfaces")
  - Stateful communication ("For example, a letter entered in a text box affect all the follow-up auto-suggestion contents")
  - Significant traffic distinctions (^)

Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow

Shuo Chen Rui Wang, XiaoFeng Wang, Kehuan Zhang

2010

M
Mi
R
shuo

Abstract– With software-as-a-service becomi more and more applications are delivered to th the Web. Unlike a desktop application, a we split into browser-side and server-side compone the application's internal information flows exposed on the network. We show that despite e a side-channel information leak is a realistic an to user privacy. Specifically, we found th detailed sensitive information is being leak number of high-profile, top-of-the-line web healthcare, taxation, investment and we eavesdropper can infer the illnesses/medicati the user, her family income and investment HTTPS protection; a stranger on the str enterprise employees' web search queries, desp Wi-Fi encryption. More importantly, the roo problem are some fundamental character applications: stateful communication, low ent better interaction, and significant traffic dis result, the scope of the problem seems ind further present a concrete analysis to d challenges of mitigating such a threat, which points to the necessity of a disciplined engineering practice for side-channel mitigations in future web application developments.

Table I, shows that the sizes of the objects hosted by the same website are so diverse that their standard deviations (often come close or even exceed their means (μ).

Table I. SIZES OF OBJECTS ON FIVE POPULAR WEBSITES

	JPEG		HTML code		Javascript	
(In bytes)	μ	σ	μ	σ	μ	σ
cnn.com	5385	7856	73192	25862	6453	6684
health.state.pa.us	12235	7374	49917	10591	N/A	N/A
medicineNet.com	3931	2239	49313	14472	22530	28184
nlm.nih.gov	11918	48897	22581	15430	4934	5307
WashingtonPost .com	12037	15122	90353	35476	13413	36220

te,

ed

- Significant traffic distinctions (^)

Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow

2010

Shuo Chen
Microsoft Research
Microsoft Corporation
Redmond, WA, USA
shuochen@microsoft.com

Rui Wang, XiaoFeng Wang, Kehuan Zhang
School of Informatics and Computing
Indiana University Bloomington
Bloomington, IN, USA
[wang63, xw7, kehzhang]@indiana.edu

2) "Find a Doctor"

Another useful feature of OnlineHealth[A] is "find a doctor", as shown in Figure 4. By choosing a specialty from the drop-down list and entering a city name (or a zipcode), the user searches the database of OnlineHealth[A] to get a list of doctors matching her desired specialty.

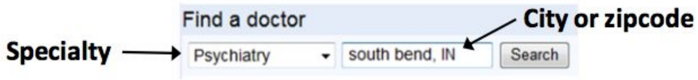


Figure 4: "Find a doctor" feature

We assume that a patient tends to find doctors near her current geographical location. Therefore the input of "city or zipcode" is guessable based on her IP address. When the "search" button is clicked, the web flow vector is $(1507\rightarrow, 270\pm10\rightarrow, \leftarrow582\pm1, \leftarrow x)$. Selection from the drop-down list gives a very-low-entropy input: there are only 94 specialties. We tested all the specialties in "south bend, IN", and found that x was within [596, 1660], i.e., density = 0.089, and every specialty is uniquely identifiable.

A web flow vector v is a sequence of directional packet sizes,

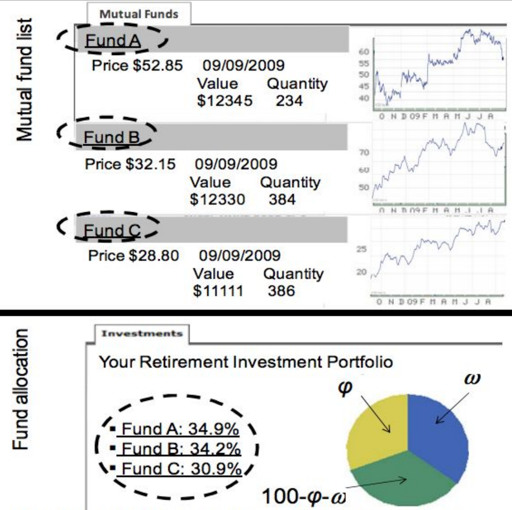
a 50-byte packet from the browser and a 1024-byte packet from the server are denoted by "(50, 1024)".

Side-Channel Leaks in Web Applications: a Reality Today, a Challenge Tomorrow

2010

Shuo Chen
Microsoft Research
Microsoft Corporation
Redmond, WA, USA
shuochen@microsoft.com

Rui Wang, XiaoFeng Wang, Kehuan Zhang
School of Informatics and Computing
Indiana University Bloomington
Bloomington, IN, USA
[wang63, xw7, kehzhang]@indiana.edu



Note: Hyperlinks in the circles navigate to fund detail pages.

Figure 10: Mutual fund list (upper) and allocation (lower)

An image (i.e., <IMG>) on an HTML page is loaded separately from the page. Therefore the size of the image can be identified from the packet size of the response from the server. There are 9 mutual funds available in this type of

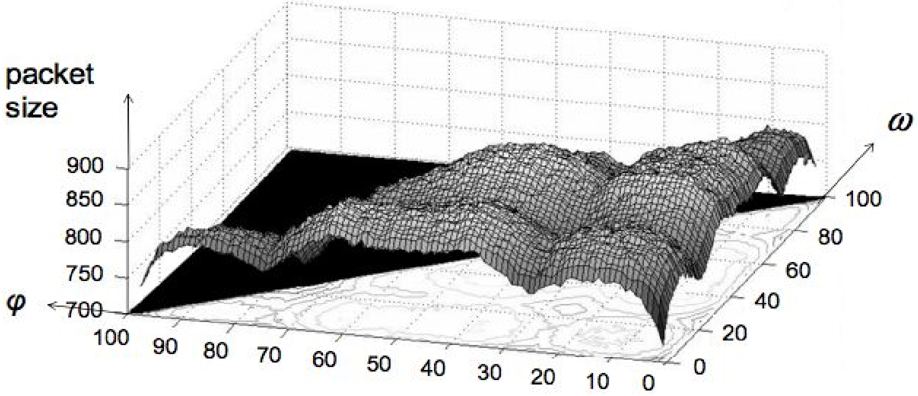
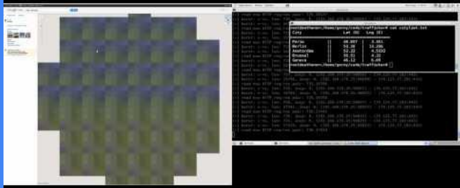


Figure 11: Pie chart sizes plotted on φ and ω axes

WEDNESDAY, FEBRUARY 8, 2012

I can still see your actions on Google Maps over SSL



HTTP Request/Response Pairs (in bytes)	
800 – 13891 820 – 8920	818 – 23910 800 – 6533

Image Size	Coordinate List
12358	(1,2,3); (81,3,12); (144,45,8); …
19771	(43,66,2); (12,55,3); …
9013	(64,22,4); …

- If the user has enabled the overlay images, two images are downloaded for each (x,y,z) location; we need to differentiate between those two request types.

The last issue can be resolved; if you map the HTTP request sizes on a histogram, here's what you get:

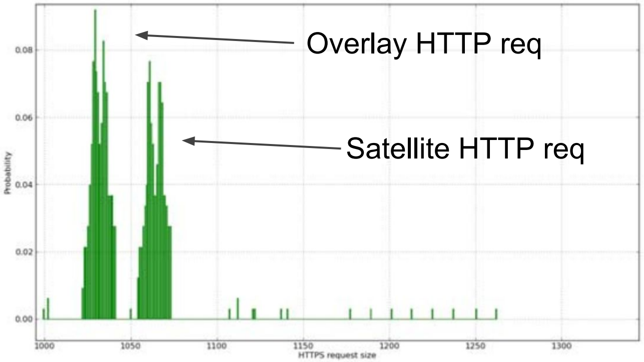
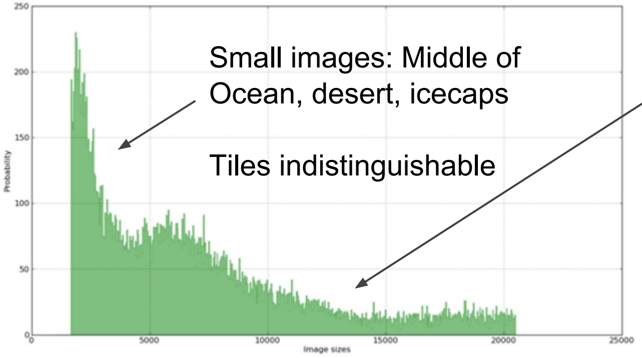


Figure 3

Based on this histogram, we can monitor a connection, create the histogram, and then determine (after a certain amount of time) which requests likely correspond to actual satellite images. All other requests are then ignored. If we continue to look at those remaining HTTP response sizes, the image sizes of the satellite tiles are distributed roughly according to the following graph.



1kb

Figure 4

Large images:
Mtns Rivers Cities,

More unique tiles

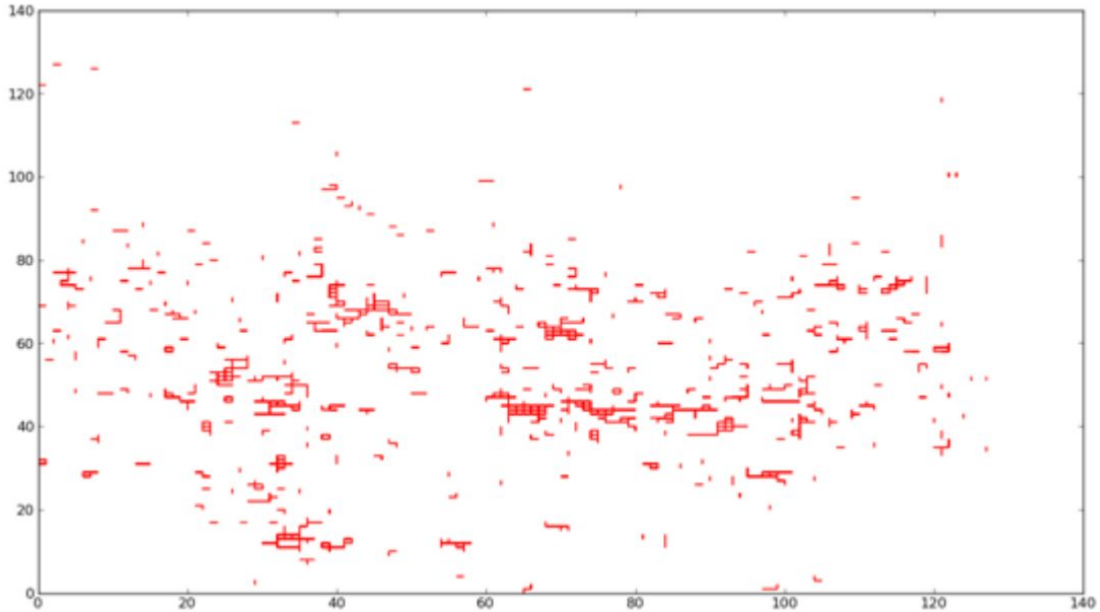


Figure 5

If you look closely, the actual matches occur only at those places were actual rectangles are being formed. If the browser's view window comprises eight tiles, we will get matches for rectangles of at least eight tiles in size (in practice the rectangles will be bigger since Google Maps loads hidden tiles on the edges of the viewport to make smooth scrolling.

Based on the above approach, we can reliably identify a complete zoom and get a bunch of coordinates back for the rectangle. We can then convert these coordinates to latitude-longitude pairs and even use reverse geocoding (not implemented yet) to convert these pairs back to human-readable names. As a result, instead of *48,51; 2,21* we will get *Paris,*

I Know Why You Went to the Clinic: Risks and Realization of HTTPS Traffic Analysis

Brad Miller[1], Ling Huang[2], A. D. Joseph[1], and J. D. Tygar[1]

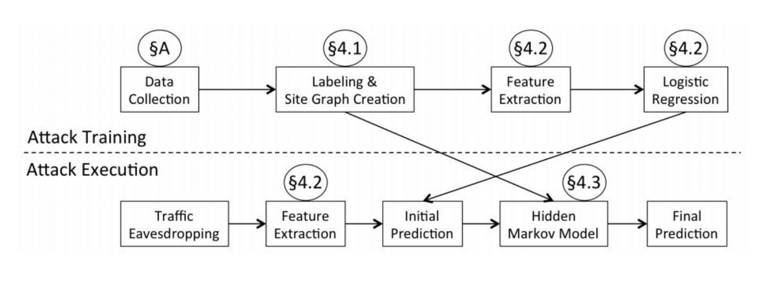
[1] UC Berkeley
[2] Intel Labs

– Novel attack technique capable of achieving 89% accuracy over 500 pages hosted at the same website, as compared to 60% with previous techniques
– Impact of caching and cookies on traffic characteristics and attack performance, affecting accuracy as much as 18%
– Novel defense reducing accuracy to 27% with 9% traffic increase; significantly increased effectiveness of packet level defenses in the HTTPS context

Settings:

- -ISP Snooping
- -Employee Monitoring
- -Surveillance
- -Censorship

Workflow:



Researchers used Machine Learning + Hidden Markov Models to train and identify specific pages within websites like:

ACLU, Bank of America, Legal Zoom, Mayo Clinic Netflix, Planned Parenthood, Wells Fargo YouTube

I Know Why You Went to the Clinic: Risks and Realization of HTTPS Traffic Analysis

Brad Miller[1], Ling Huang[2], A. D. Joseph[1], and J. D. Tygar[1]

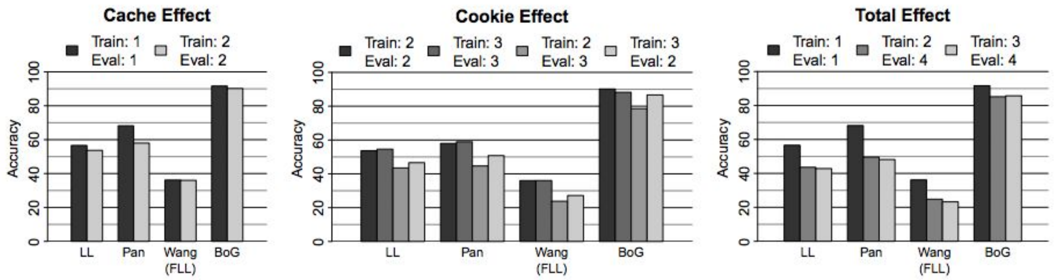
[1] UC Berkeley
[2] Intel Labs

– Novel attack technique capable of achieving 89% accuracy over 500 pages hosted at the same website, as compared to 60% with previous techniques
– Impact of caching and cookies on traffic characteristics and attack performance, affecting accuracy as much as 18%
– Novel defense reducing accuracy to 27% with 9% traffic increase; significantly increased effectiveness of packet level defenses in the HTTPS context

=> Disabling the cache increases unique packet sizes which aids in identification

=> “difference in cookies between training and evaluation conditions will impact accuracy results”

=> effect of cookies & cache can sway accuracy up to 18%



(a) (b) (c)

Mode Number	Cache	Cookie Retention	Browsing Scope
1	Disabled	Fresh VM every 75 samples	Single website
2	Enabled	Fresh VM every 75 samples	Single website
3	Enabled	Same VM for all samples	Single website
4	Enabled	Same VM for all samples	All websites

(d)

I Know What You Saw Last Minute - The Chrome Browser Case

Ran Dubin
Communication Systems Engineering
Ben-Gurion University of the Negev
Israel

Amit Dvir
Center for Cyber Technologies
Department of Computer Science
Ariel University
Israel

Ofir Pele
Center for Cyber Technologies
Department of Computer Science
Department of Electrical and Electronics Engineering
Ariel University
Israel

Ofer Hadar
Communication Systems Engineering
Ben-Gurion University of the Negev
Israel

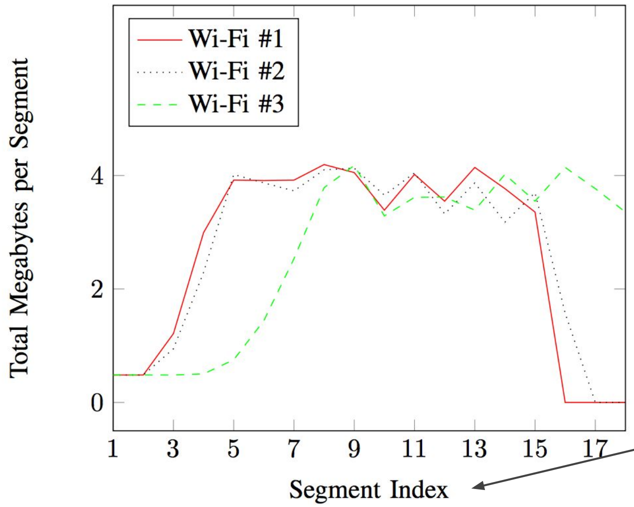


Fig. 3: Total megabytes per segment of three downloads over different Wi-Fi networks of the same video title, all with the same quality representation. Due to network conditions variability, there are differences between the networks.

-Dynamic Adaptive Streaming over HTTP (DASH)

-“In DASH, each quality representation is encoded in variable bit rates (VBRs)”

“short segments, typically a few seconds long (2 − 16 seconds), and each segment is encoded several times, each time with a different quality representation”

I Know What You Saw Last Minute - The Chrome Browser Case

Ran Dubin
Communication Systems Engineering
Ben-Gurion University of the Negev
Israel

Amit Dvir
Center for Cyber Technologies
Department of Computer Science
Ariel University
Israel

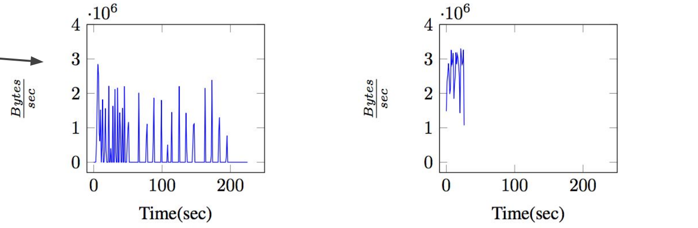
Ofir Pele
Center for Cyber Technologies
Department of Computer Science
Department of Electrical and Electronics Engineering
Ariel University
Israel

Ofer Hadar
Communication Systems Engineering
Ben-Gurion University of the Negev
Israel

"We found that often there are two flows both with audio and video. The short traffic segments contain audio while the longer contain video"

Bits-per-peak
takes TCP retransmission into account

"Audio data and video data can be found in the same 5- tuple flow {protocol, src IP, dst IP, src port, dst port}. In some cases we cannot distinguish between them."



(a) Chrome auto mode over HTTP2. (b) Chrome fixed mode over HTTP2.

Fig. 1: YouTube Costa Rica in 4K - traffic traces from Chrome (Ver 43.0.2357.81) with $HTML5$ player in automatic and fixed quality selection modes.

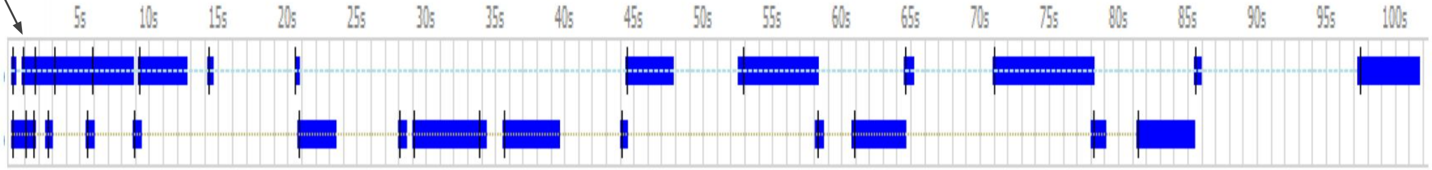
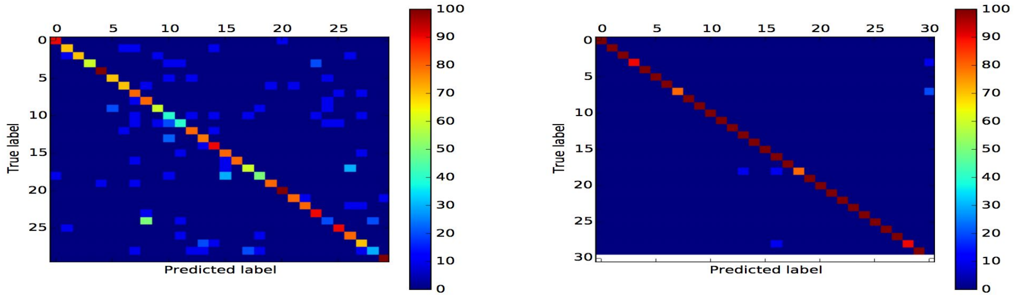


Fig. 2: YouTube Costa Rica 4k auto mode with Chrome. Each horizontal line represents different YouTube flows from the same download. The video quality is $720P$.

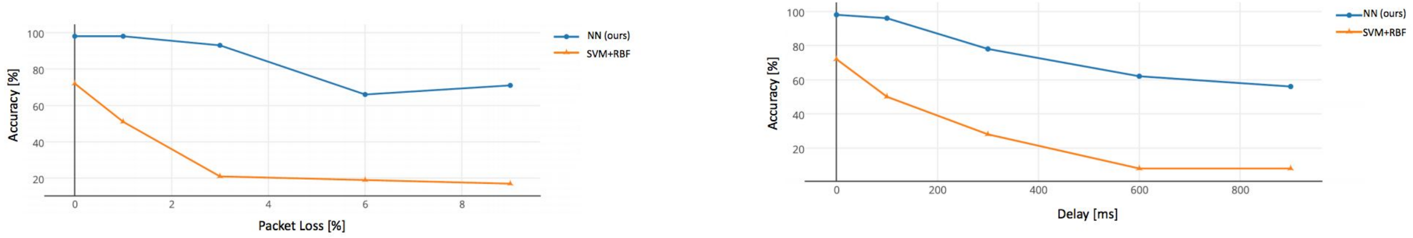
Fig. 4: Proposed solution architecture.



(a) SVM+RBF confusion matrix results.

(b) Our confusion matrix results.

Fig. 7: Classification confusion matrices results.



(a) Accuracy results for additional packet loss percentage

(b) Accuracy results for additional LAN network delay

Fig. 8: Accuracy results for different network conditions.

Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application

Jonathan Muehlstein*, Yehonatan Zion*, Maor Bahumi†, Itay Kirshenboim*†
Ran Dubin‡, Amit Dvir*, Ofir Pele*†
* Center for Cyber Technologies, Department of Computer Science, Ariel University
† Center for Cyber Technologies, Department of Electrical and Electronics Engineering, Ariel University
‡ Department of Communication Systems Engineering, Ben-Gurion University of the Negev

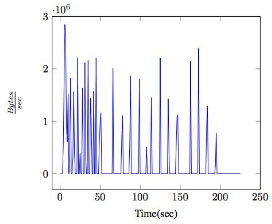


Fig. 2: An example of the bursty behavior of browser traffic.

Selenium crawlers to gather Dataset:

traffic for applications (Youtube, Facebook & Twitter) viewed on different browsers and operating systems

Support Vector Machine

New features => "tried to identify traffic parameters that differentiate between different operating systems and browsers."

Forward packets
Forward total Bytes
Min forward inter arrival time difference
Max forward inter arrival time difference
Mean forward inter arrival time difference
STD forward inter arrival time difference
Mean forward packets
STD forward packets
Backward packets
Backward total Bytes
Min backward inter arrival time difference
Max backward inter arrival time difference
Mean backward inter arrival time difference
STD backward inter arrival time difference
Mean backward packets
STD backward packets
Mean forward TTL value
Minimum forward packet
Minimum backward packet
Maximum forward packet
Maximum backward packet
Total packets
Minimum packet size
Maximum packet size
Mean packet size
Packet size variance

(a) base features

TCP initial window size
TCP window scaling factor
SSL compression methods
SSL extension count
SSL chiper methods
SSL session ID len
Forward peak MAX throughput
Mean throughput of backward peaks
Max throughput of backward peaks
Backward min peak throughput
Backward STD peak throughput
Forward number of bursts
Backward number of bursts
Forward min peak throughput
Mean throughput of forward peaks
Forward STD peak throughput
Mean backward peak inter arrival time diff
Minimum backward peak inter arrival time diff
Maximum backward peak inter arrival time diff
STD backward peak inter arrival time diff
Mean forward peak inter arrival time diff
Minimum forward peak inter arrival time diff
Maximum forward peak inter arrival time diff
STD forward peak inter arrival time diff
Keep alive packets
TCP Maxiumu Segment Size
Forward SSL Version

(b) new features

Features from previous study on Youtube title identification

Predicted labels

Real labels	Windows IExplorer Twitter	Ubuntu Firefox Google-Background	Windows Non-Browser Microsoft-Background	Windows Chrome Twitter	Windows Firefox Twitter	OSX Safari Google-Background	OSX Safari Youtube	Ubuntu Chrome Unknown	Windows Chrome Google-Background	Ubuntu Firefox Twitter	OSX Safari Unknown	Ubuntu Firefox Unknown	Ubuntu Chrome Google-Background	Ubuntu Chrome Twitter	Windows Firefox Google-Background	OSX Safari Twitter	Ubuntu Firefox Youtube	Windows Non-Browser Teamviewer	Ubuntu Chrome Youtube	Windows Non-Browser Dropbox	Windows Chrome Unknown	Ubuntu Chrome Facebook	Windows Firefox Unknown	Ubuntu Firefox Facebook	OSX Chrome Twitter	Windows IExplorer Unknown	Ubuntu Non-Browser Microsoft-Background	Windows IExplorer Google-Background	OSX Chrome Google-Background	OSX Chrome Unknown
Windows IExplorer Twitter	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Google-Background	0	.97	0	0	0	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows Non-Browser Microsoft-Background	0	0	.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows Chrome Twitter	0	0	0	.99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0
Windows Firefox Twitter	0	0	0	0	.98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.02	0	0	0	0	0	0	0
OSX Safari Google-Background	0	0	0	0	0	.92	.04	0	0	0	.02	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OSX Safari Youtube	0	0	0	0	0	.02	.97	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Unknown	0	0	0	0	0	0	0	.84	0	0	0	0	.07	.04	0	0	0	0	.01	0	0	.03	0	0	0	0	0	0	0	0
Windows Chrome Google-Background	0	0	.01	.03	0	0	0	0	.94	0	0	0	0	0	.02	0	0	0	0	0	.01	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Twitter	0	0	0	0	0	0	0	0	0	.95	0	.03	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0
OSX Safari Unknown	0	0	0	0	0	.06	.01	0	0	0	.91	0	0	0	0	.01	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Unknown	0	.02	0	0	0	0	0	0	0	.08	0	.87	0	0	0	0	.01	0	0	0	0	0	0	.03	0	0	0	0	0	0
Ubuntu Chrome Google-Background	0	.07	0	0	0	0	0	.18	0	0	0	0	.73	0	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Twitter	0	.02	0	0	0	0	0	.08	0	0	0	0	.03	.84	0	0	0	0	.01	0	0	.01	0	0	0	0	0	0	0	0
Windows Firefox Google-Background	0	0	0	.01	0	0	0	0	.01	0	0	0	0	0	.97	0	0	0	0	0	0	0	.01	0	0	0	0	0	0	0
OSX Safari Twitter	0	0	0	0	0	0	.06	0	0	0	.03	0	0	0	0	.91	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Firefox Youtube	0	.02	0	0	0	0	0	0	0	.02	0	.02	0	0	0	0	.93	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows Non-Browser Teamviewer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Youtube	0	0	0	0	0	0	0	.07	0	0	0	0	.13	.04	0	0	0	0	.74	0	0	.02	0	0	0	0	0	0	0	0
Windows Non-Browser Dropbox	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Windows Chrome Unknown	0	0	.02	.09	0	0	0	0	.02	0	0	0	0	0	0	0	0	0	0	0	.86	0	0	0	0	0	0	0	0	0
Ubuntu Chrome Facebook	0	0	0	0	0	0	0	.3	0	0	0	0	.04	0	0	0	0	0	0	0	0	.67	0	0	0	0	0	0	0	0
Windows Firefox Unknown	0	0	.06	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.94	0	0	0	0	0	0	0
Ubuntu Firefox Facebook	0	.06	0	0	0	0	0	0	0	.11	0	.28	0	0	0	0	0	0	0	0	0	0	0	.56	0	0	0	0	0	0
OSX Chrome Twitter	0	0	0	0	0	0	0	.13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.75	0	0	0	.06	.06
Windows IExplorer Unknown	.71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.29	0	0	0	0
Ubuntu Non-Browser Microsoft-Background	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Windows IExplorer Google-Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
OSX Chrome Google-Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
OSX Chrome Unknown	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fig. 4: Confusion matrices (rows are ground truth). For most tuples the classification is almost perfect. Exceptions happens mostly between similar tuples and the unknown classes (which can actually be a correct answer that we cannot verify). For example, “Ubuntu Chrome Google-Background” is mistakenly classified as “Ubuntu Chrome Unknown” in 18% of the cases and “Ubuntu Firefox Google-Background” in 7%. The total accuracy is to 96.06%

Identifying HTTPS-Protected Netflix Videos in Real-Time

Andrew Reed, Michael Kranch
Dept. of Electrical Engineering and Computer Science
United States Military Academy at West Point
West Point, New York, USA
{andrew.reed, michael.kranch}@usma.edu

ABSTRACT

After more than a year of research and development, Netflix recently upgraded their infrastructure to provide HTTPS encryption of video streams in order to protect the privacy of their viewers. Despite this upgrade, we demonstrate that it is possible to accurately identify Netflix videos from passive traffic capture in real-time with very limited hardware requirements. Specifically, we developed a system that can report the Netflix video being delivered by a TCP connection using only the information provided by TCP/IP headers.

To support our analysis, we created a fingerprint database comprised of 42,027 Netflix videos. Given this collection of fingerprints, we show that our system can differentiate between videos with greater than 99.99% accuracy. Moreover, when tested against 200 random 20-minute video streams, our system identified 99.5% of the videos with the majority of the identifications occurring less than two and a half minutes into the video stream.

Identifying HTTPS-Protected Netflix Videos in Real-Time

Andrew Reed, Michael Kranch
Dept. of Electrical Engineering and Computer Science
United States Military Academy at West Point
West Point, New York, USA
{andrew.reed, michael.kranch}@usma.edu

Table 2: Database statistics.

Total Videos			Average Length (h:mm:ss)		
All	Movies	Shows	All	Movies	Shows
42,027	3,247	38,780	0:38:54	1:33:30	0:34:17

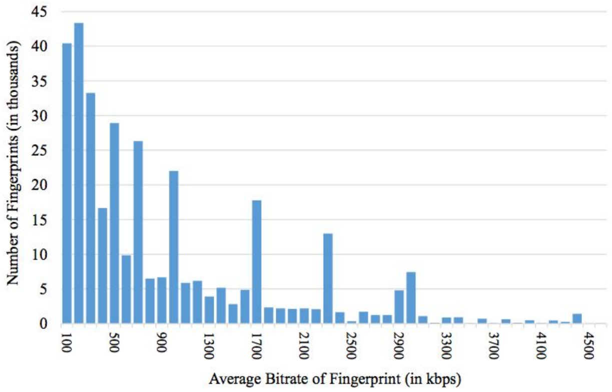


Figure 2: Number of fingerprints by average bitrate. The results are shown in 100 kbps bins. There are 146 fingerprints that exceed 4600 kbps that are not depicted.

"DASH and VBR can produce sequences of video segment sizes (i.e. fingerprints) that are unique for each video"

average of 7.86 fingerprints per video

"Netflix has historically encoded their browser-based videos at 235, 375, 560, 750, 1050, 1750, 2350, and 3000 kbps"

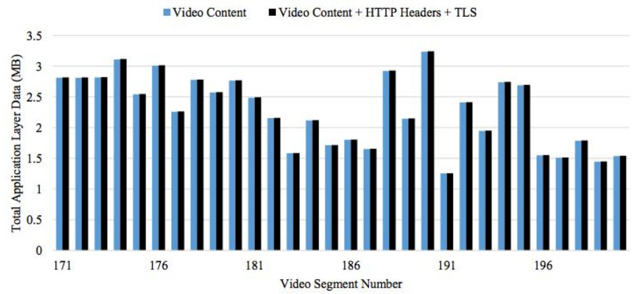


Figure 1: Netflix video overhead due to HTTP headers and TLS (*Home*, 3830 kbps encoding).

Table 1: adudump trace of *Home* (3830 kbps encoding). These are segments 171-180 from Figure 1.

Timestamp	Local PC	Dir.	Netflix Server	Size (B)
1471357732.77583	134.240.17.111.31177	>	198.45.63.167.443	756
1471357736.70148	134.240.17.111.31177	<	198.45.63.167.443	2817667
1471357736.77902	134.240.17.111.31177	>	198.45.63.167.443	756
1471357740.89304	134.240.17.111.31177	<	198.45.63.167.443	2816159
1471357740.97057	134.240.17.111.31177	>	198.45.63.167.443	756
1471357744.45695	134.240.17.111.31177	<	198.45.63.167.443	2822089
1471357744.53453	134.240.17.111.31177	>	198.45.63.167.443	756
1471357748.76052	134.240.17.111.31177	<	198.45.63.167.443	3117490
1471357748.83926	134.240.17.111.31177	>	198.45.63.167.443	756
1471357752.72718	134.240.17.111.31177	<	198.45.63.167.443	2548098
1471357752.80466	134.240.17.111.31177	>	198.45.63.167.443	756
1471357756.87447	134.240.17.111.31177	<	198.45.63.167.443	3014236
1471357756.95195	134.240.17.111.31177	>	198.45.63.167.443	756
1471357760.48768	134.240.17.111.31177	<	198.45.63.167.443	2263764
1471357760.56593	134.240.17.111.31177	>	198.45.63.167.443	756
1471357764.73616	134.240.17.111.31177	<	198.45.63.167.443	2782180
1471357764.81363	134.240.17.111.31177	>	198.45.63.167.443	755
1471357768.73659	134.240.17.111.31177	<	198.45.63.167.443	2577683
1471357768.81421	134.240.17.111.31177	>	198.45.63.167.443	756
1471357772.97218	134.240.17.111.31177	<	198.45.63.167.443	2770492

Identifying HTTPS-Protected Netflix Videos in Real-Time

Andrew Reed, Michael Kranch
Dept. of Electrical Engineering and Computer Science
United States Military Academy at West Point
West Point, New York, USA
{andrew.reed, michael.kranch}@usma.edu

4.3 kd-Tree Search

Similar to [10], we create a 6D key for each 30-ADU window and conduct a range search of the kd-tree to retrieve a shortlist of potential matches. The ranges for each search are as follows:

- **1st Dimension Min** $= \frac{Total\ Received}{1.0019} - (30 * 525\ bytes)$
- **1st Dimension Max** $= \frac{Total\ Received}{1.0017} - (30 * 515\ bytes)$
- **2nd through 6th Dimension Min**: -0.0001
- **2nd through 6th Dimension Max**: +0.0001

Our 1st dimension ranges are based on these two observations of Netflix traffic:

- HTTP headers add ~520 bytes to each video segment.
- TLS overhead adds ~0.18% to the combined video content plus HTTP headers.

Looking Forward

TLS 1.2 Specs

“Note in particular that type and length of a record are not protected by encryption. If this information is itself sensitive, application designers may wish to take steps (padding, cover traffic) to minimize information leakage.”

Web Apps

- Application Specific Padding
- “TLS-level length hiding can be effective if combined with application-level policy”

    Previous research had to “take steps”: modify GnuTLS (TLS 1.2) to implement padding for records

    TLS 1.3 => Record Padding part of specs:

5.4. Record Padding

All encrypted TLS records can be padded to inflate the size of the TLSCiphertext. This allows the sender to hide the size of the traffic from an observer.

When generating a TLSCiphertext record, implementations MAY choose to pad. An unpadded record is just a record with a padding length of zero. Padding is a string of zero-valued bytes appended to the ContentType field before encryption. Implementations MUST set the padding octets to all zeros before encrypting.

Application Data records may contain a zero-length TLSInnerPlaintext.content if the sender desires. This permits generation of plausibly-sized cover traffic in contexts where the presence or absence of activity may be sensitive. Implementations MUST NOT send Handshake or Alert records that have a zero-length

Regarding ISP’s

VPNS, TOR, proxies?

market competition?

Laws & regulations?

Looking Forward

Chaffing and Winnowing: Confidentiality without Encryption
Ronald Rivest (1998)

secure channel →

Alice

constructs 4 packets, each containing one bit of her message and a valid MAC

Serial	Bit	MAC
1	**1**	**234**
2	**0**	**890**
3	**0**	**456**
4	**1**	**678**

insecure channel →

Charles

adds 4 chaff packets with inverted bits and invalid MAC, shown in *italics* (**chaffing**)

Serial	Bit	MAC
1	*0*	*321*
1	**1**	**234**
2	**0**	**890**
2	*1*	*987*
3	**0**	**456**
3	*1*	*543*
4	*0*	*765*
4	**1**	**678**

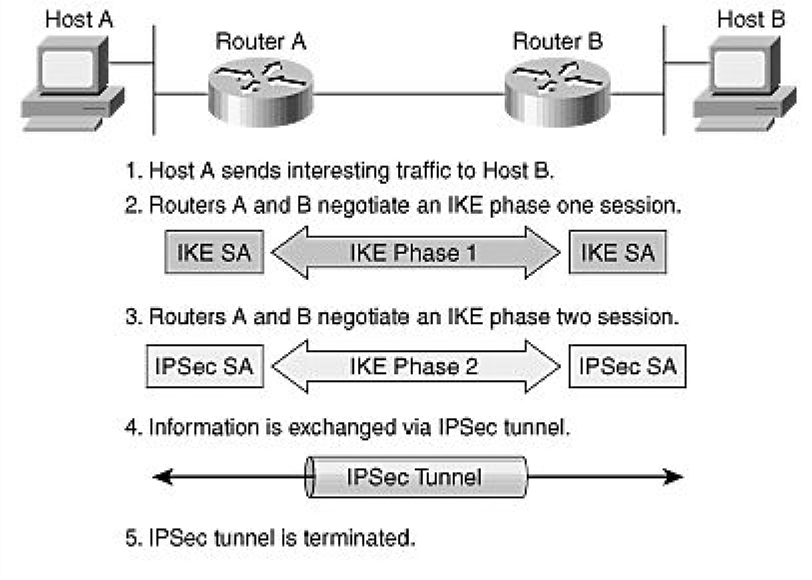
Bob

discards packets with invalid MAC to recover the message (**winnowing**)

In this example, Alice wishes to send the message "1001" to Bob. For simplicity, assume that all even MAC are valid and odd ones are invalid.

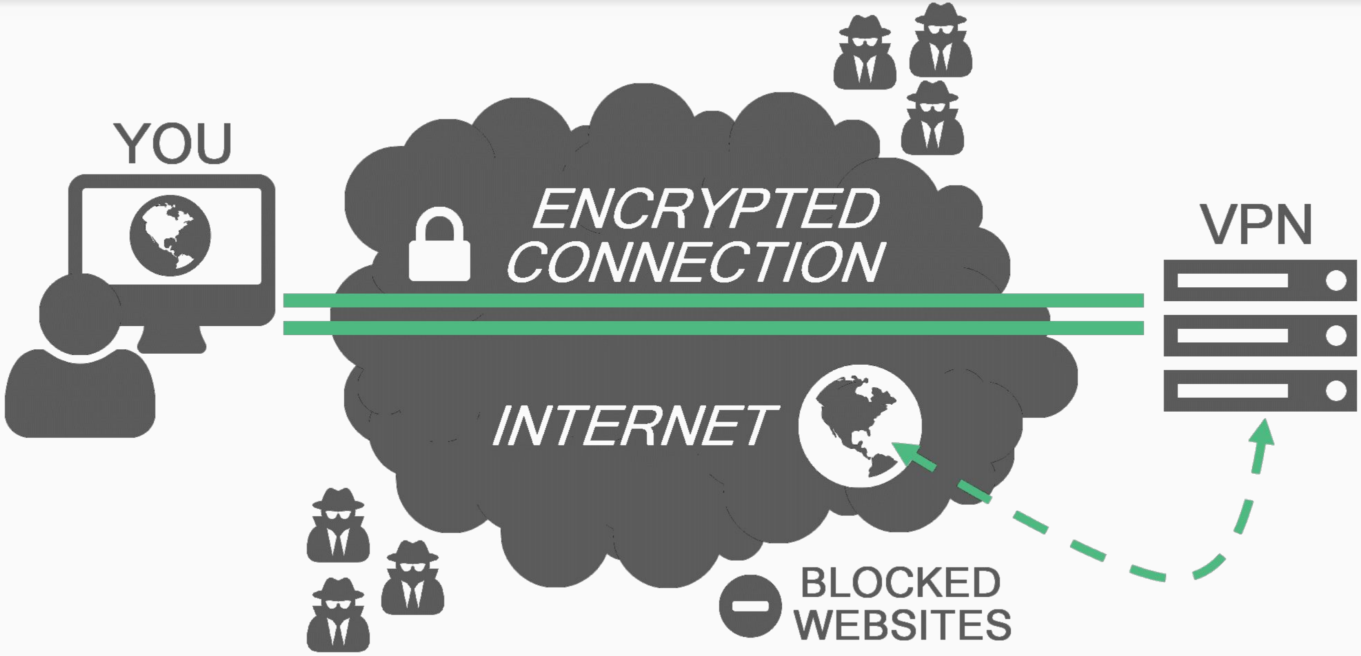
IPv6: Another Security Risk

IPv6 & IPSec



Host A
Router A
Router B
Host B
1. Host A sends interesting traffic to Host B.
2. Routers A and B negotiate an IKE phase one session.
IKE SA
IKE Phase 1
IKE SA
3. Routers A and B negotiate an IKE phase two session.
IPSec SA
IKE Phase 2
IPSec SA
4. Information is exchanged via IPSec tunnel.
IPSec Tunnel
5. IPSec tunnel is terminated.

VPN Services



Too Big or Too Small? The PTB-PTS ICMP-based Attack against IPsec Gateways

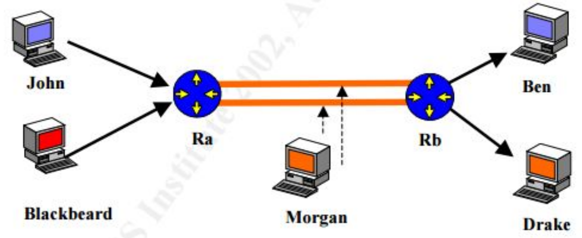
Vincent Roca [1] , Saikou Fall [1] Details

1 PRIVATICS - Privacy Models, Architectures and Tools for the Information Society
Inria Grenoble - Rhône-Alpes, CITI - CITI Centre of Innovation in Telecommunications and Integration of services

Abstract : This document introduces the "Packet Too Big"-"Packet Too Small" Internet Control Message Protocol (ICMP) based attack against IPsec gateways. We explain how an attacker having eavesdropping and packet injection capabilities, from the unsecure network where he only sees encrypted packets, can force a gateway to reduce the Path Maximum Transmission Unit (PMTU) of an IPsec tunnel to the minimum, which can trigger severe issues for the hosts behind this gateway: with a Linux host, depending on the PMTU discovery algorithm in use (i.e., PMTUd versus PLPMTUd) and protocol (TCP versus UDP), the attack either creates a Denial of Service or major performance penalties. This attack highlights two fundamental problems, namely: (1) the impossibility to distinguish legitimate from illegitimate ICMP packets coming from the untrusted network, and (2) the contradictions in the way Path MTU is managed by some end hosts when this Path MTU is below the minimum packet size any link should support because of the IPsec encapsulation. Status of This Memo This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

Document type : Other publications
Work in Progress document of the IPSECME (IP Security Maintenance and Extensions) of the IETF (In.. 2016, pp.16

Domain :
Computer Science [cs] / Cryptography and Security [cs.CR]
Computer Science [cs] / Networking and Internet Architecture [cs.NI]



(Diagram adapted from Tzvetkov, VPN Attacks)

Cut-And-Past Attack:
This attack will only be possible on two networks that use IPSEC as a tunnel between the two routers that link the networks. There is also a requirement that the attacker has access to a second machine in each of the two networks.

The attack works by Morgan sniffing a legitimate encrypted packet from John to Ben. Morgan also sniffs a planned packet sent from Blackbeard to Drake. Morgan copies encrypted data from John's packet into a packet from Blackbeard to Drake. Router B is tricked into decrypting Johns packet for Ben and sending it to Drake. This exploit is not as straightforward as it may appear, as there are some other requirements relating to the sequence numbers used in IPSEC packets and ensuring that John's genuine

packets don't reach Router B before the false packets do. IPSEC includes various replay-attack protection methods that would make this attack a little more difficult to successfully carry out in a real world situation.

Session Hijacking:
Similar to the previous attack, Blackbeard could have created packets that are intended to arrive at Ben as if they were sent from John. Instead of stealing Johns packet and asking Router b to decrypt it for Drake, Morgan now pastes Blackbeards data into John's packet and it is decrypted by Rb and sent to Ben as though it came from John.

These attacks are much more complicated to conduct in practice, as sequence numbers and other authentication issues must be overcome. Despite this, the attacks appear feasible.

Results

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies 2015; 2015 (1):77–91

Vasile C. Perta*, Marco V. Barbera, Gareth Tyson, Hamed Haddadi[1], and Alessandro Mei[2]

A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN clients

All VPN services surveyed rely on the correct configuration of the operating system's routing table. Worryingly, no attempt is made to secure this operation. -

Provider	Countries	Servers	Technology	DNS	IPv6-leak	DNS hijacking
Hide My Ass	62	641	OpenVPN, PPTP	OpenDNS	Y	Y
IPVanish	51	135	OpenVPN	Private	Y	Y
Astrill	49	163	OpenVPN, L2TP, PPTP	Private	Y	N
ExpressVPN	45	71	OpenVPN, L2TP, PPTP	Google DNS, Choopa Geo DNS	Y	Y
StrongVPN	19	354	OpenVPN, PPTP	Private	Y	Y
PureVPN	18	131	OpenVPN, L2TP, PPTP	OpenDNS, Google DNS, Others	Y	Y
TorGuard	17	19	OpenVPN	Google DNS	N	Y
AirVPN	15	58	OpenVPN	Private	Y	Y
PrivateInternetAccess	10	18	OpenVPN, L2TP, PPTP	Choopa Geo DNS	N	Y
VyprVPN	8	42	OpenVPN, L2TP, PPTP	Private (VyprDNS)	N	Y
Tunnelbear	8	8	OpenVPN	Google DNS	Y	Y
proXPN	4	20	OpenVPN, PPTP	Google DNS	Y	Y
Mullvad	4	16	OpenVPN	Private	N	Y
Hotspot Shield Elite	3	10	OpenVPN	Google DNS	Y	Y

Table 1. VPN services subject of our study

Leaks

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies 2015; 2015 (1):77–91

Vasile C. Perta*, Marco V. Barbera, Gareth Tyson, Hamed Haddadi[1], and Alessandro Mei[2]

A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN clients

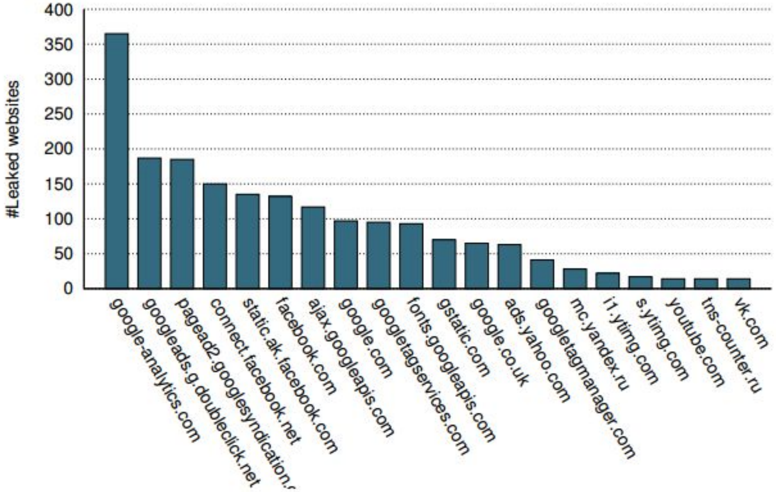


Fig. 3. Top third-parties that leak IPv4-only websites through the `Referer` header. 92% of the Alexa top 1K IPv4-only websites embed objects of at least 1 of these third parties.

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies 2015; 2015 (1):77–91

Vasile C. Perta*, Marco V. Barbera, Gareth Tyson, Hamed Haddadi[1], and Alessandro Mei[2]

A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN clients

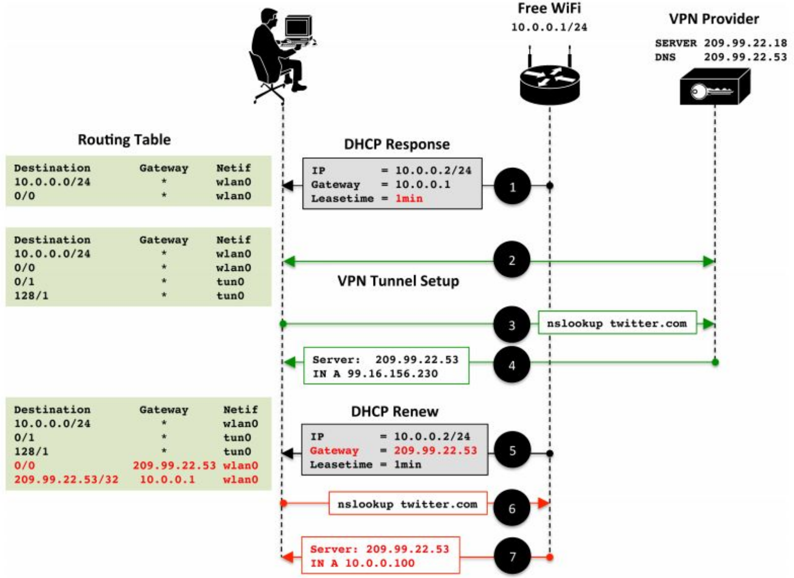


Fig. 5. Hijacking the DNS through a route injection attack (OpenVPN tunnels)

DEFAULTS...

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies 2015; 2015 (1):77–91

Vasile C. Perta*, Marco V. Barbera, Gareth Tyson, Hamed Haddadi[1], and Alessandro Mei[2]

A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN clients

The simplest scenario is where the VPN client does not change the victim's default DNS configuration (e.g., HideMyAss over OpenVPN). In this case, subverting DNS queries is trivial. The access point can simply use DHCP to set the victim's DNS server to one that it manages itself. The adversary will then receive all DNS queries generated by the host.

DE GRUYTER OPEN Proceedings on Privacy Enhancing Technologies 2015; 2015 (1):77–91

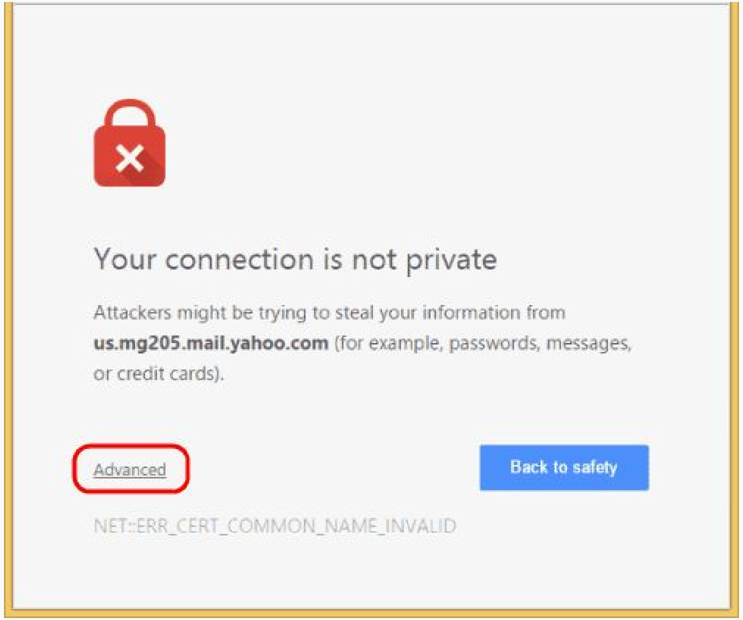
Vasile C. Perta*, Marco V. Barbera, Gareth Tyson, Hamed Haddadi[1], and Alessandro Mei[2]

A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN clients

5.4 Attack feasibility

Both versions of the DNS hijacking attack we presented require the adversary to control the DHCP server used by the victim host (*e.g.,* the WiFi router). We do not deem this assumption to be particularly restrictive, as it falls within the typical threat model of commercial VPN services (*e.g.,* securing communications in an untrusted wireless network).

A second, more restrictive requirement is to know the IP address of DNS server in use by the VPN at the victim host. To tackle this, the adversary could passively monitor the client-side IP of the VPN tunnel. This would reveal the VPN service used, which could then be mapped to the relative DNS server (*e.g.,* column "DNS" in Table 1). Note that the mapping may need to take into consideration location too, as we observed some providers to use different DNSes in different servers.



More Traffic Analysis!

AHHHHH


Harvard University
@Harvard
Follow

Alert: Unconfirmed reports of explosives at four sites on campus: Science Center, Thayer, Sever, and Emerson. Evacuate those buildings now.

Reply Retweet Favorite More

846 RETWEETS 27 FAVORITES

9:14 AM - 16 Dec 13

Two Days Later

CAPTURE EVERYTHING

Client Hellos

	Client to Server	Server to Client	Discarded
Unfiltered	9547378	3776313	99.226%
Handshake & Client Hello Filter	51766	59	2.859%
1st Byte TLS Version	51677	3	0.005%
1st Byte TLS Version (Record)	51677	0	0.000%

Problems

Usage

Distinguishing between clients on the fly!

(Anti)Forensics!

Intrusion detection!

Shitware detection!

Homogenous platform verification!

Honeypots!

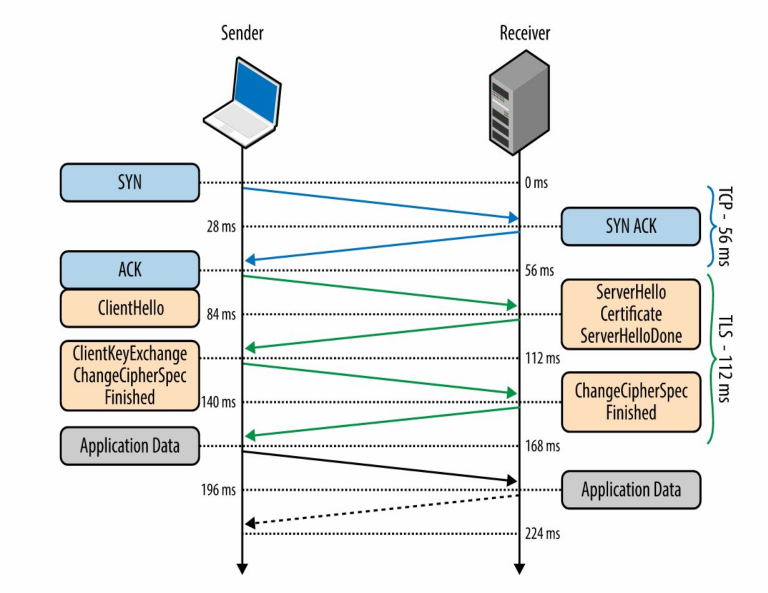
Solutions?

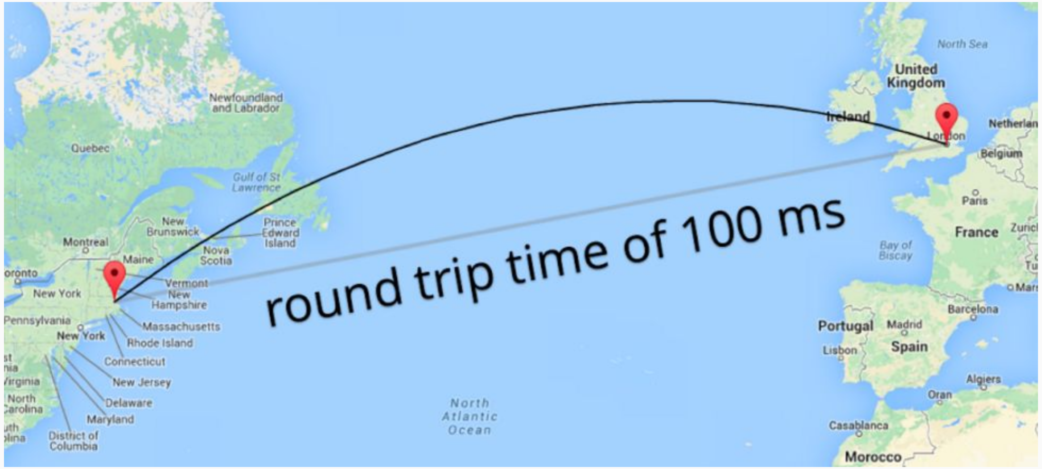
Do less.

QUIC (Quick UDP Internet Connections)

Motivation

How do you make the web faster?





Assuming you have very fast Internet…
Then maybe we do not need to change anything

Not everybody could take fast internet for granted

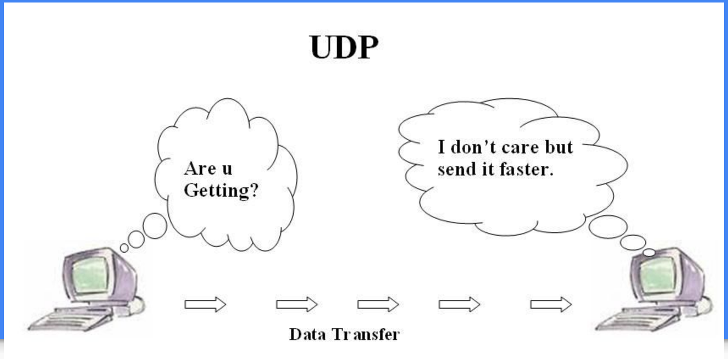
Solution: QUIC (Quick UDP Internet Connection)

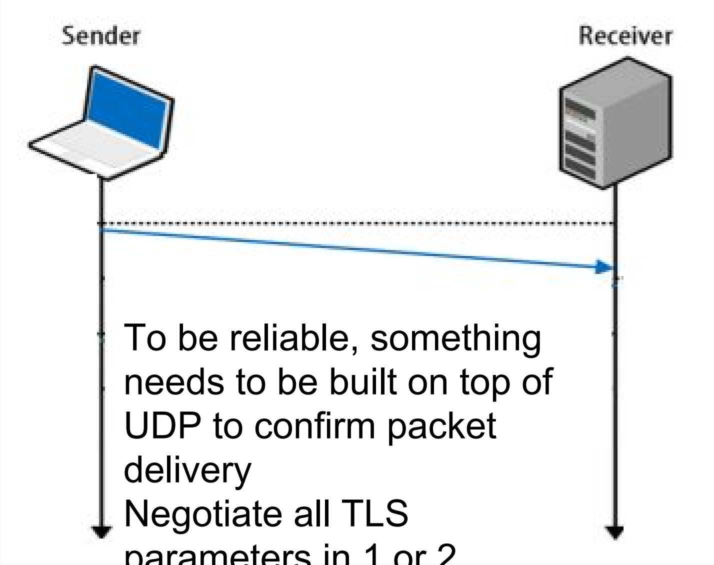
Experimental transport layer network protocol

Jim Roskind at Google in 2012

Reduces latency and runs in user-space

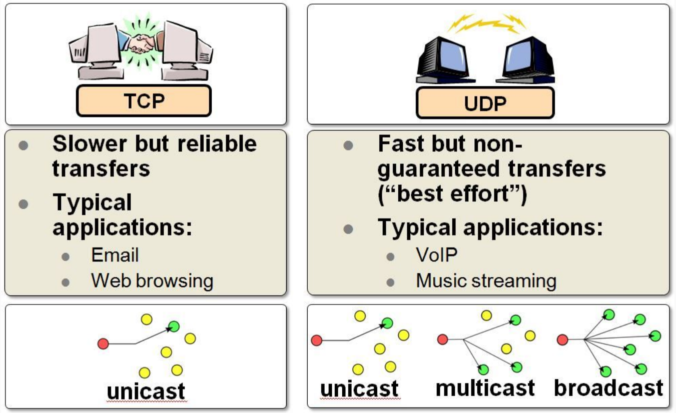
Background - UDP





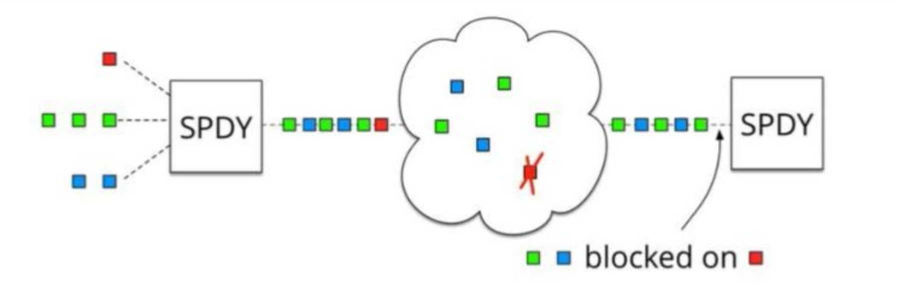
To be reliable, something needs to be built on top of UDP to confirm packet delivery
Negotiate all TLS parameters in 1 or 2 packets

UDP is TCP's wild cousin, a "fire and forget protocol"
A message is assumed to have arrived, so the network uses less time to validate packets.

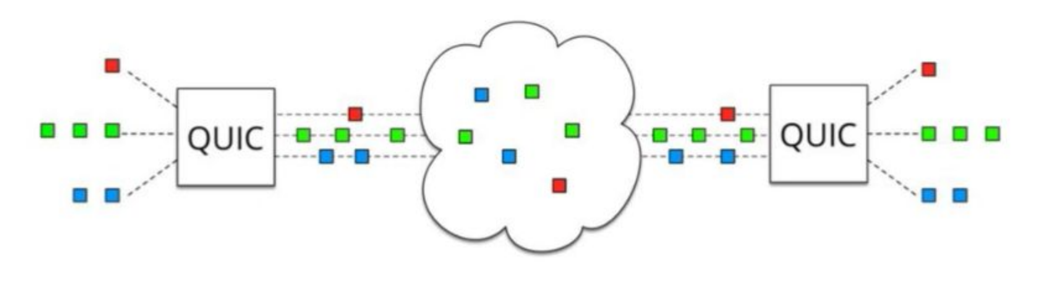


Why is UDP faster?

TCP: The order in which TCP packets are processed matters

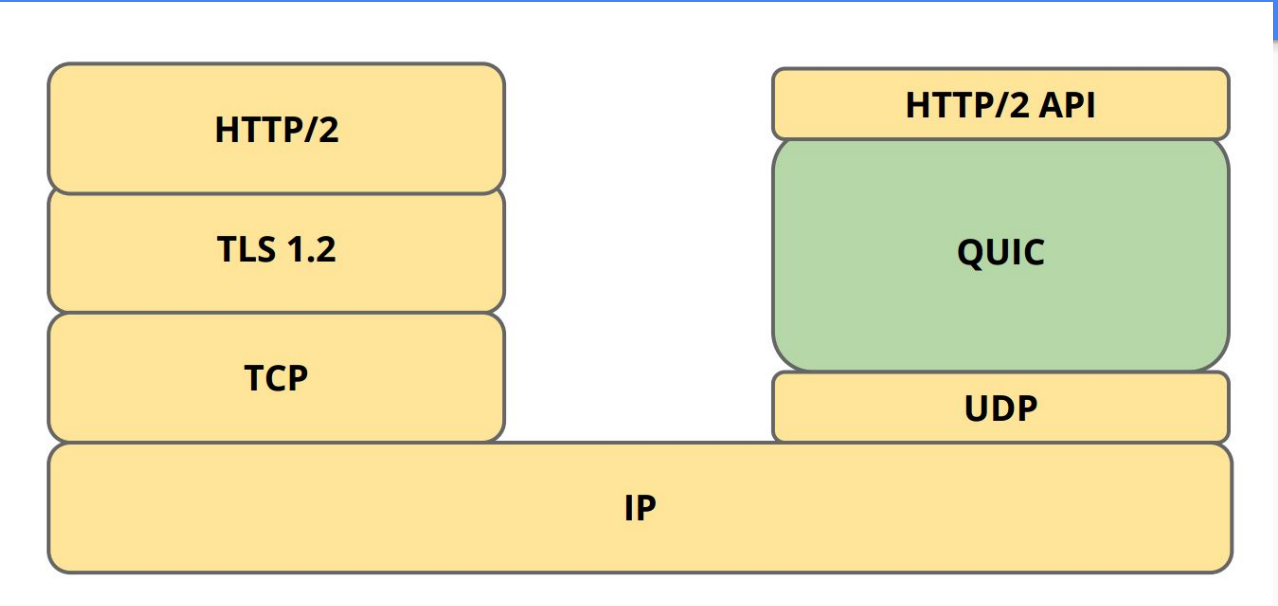


UDP: is not dependent on the order in which packets are received



Forward Error Correction:
10% Overhead

How does Quic fit in?



Requires server/client collaboration and support

If you are King Google, you can do this

Client + Server support

Chromium 29 (Aug 2013) and Opera 16

[DEMO] chrome://net-internals/#quic

`chrome://net-internals/#events&q=type:QUIC_SESSION%20is:active`

Google servers and community projects (libquic, goquic)

Host	Secure	Version	Peer address	Connection UID	Active stream count
36.docs.google.com:443	true	QUIC_VERSION_30	[2a00:1450:4013:c00::bd]:443	2708254184554045987	1
apis.google.com:443	true	QUIC_VERSION_30	[2a00:1450:400e:802::200e]:443	5189742635553804178	0
clients4.google.com:443	true	QUIC_VERSION_30	[2a00:1450:400e:802::200e]:443	5174608782190849431	0
i.ytimg.com:443	true	QUIC_VERSION_30	[2a00:1450:4013:c01::8a]:443	10559272118787914470	0
plus.google.com:443	true	QUIC_VERSION_30	[2a00:1450:400e:801::200e]:443	2461447815203244151	0
r18---sn-5hne6ned.googlevideo.com:443	true	QUIC_VERSION_30	[2a00:1450:401c:f::17]:443	14426173135210551355	0
s.ytimg.com:443	true	QUIC_VERSION_30	[2a00:1450:4013:c01::65]:443	814538457547024801	0
ssl.google-analytics.com:443	true	QUIC_VERSION_30	[2a00:1450:4007:80b::2008]:443	16111488254187388150	0
ssl.gstatic.com:443	true	QUIC_VERSION_30	[2a00:1450:400e:801::2003]:443	13147793992039561928	0
www.google.be:443	true	QUIC_VERSION_30	[2a00:1450:400c:c04::5e]:443	4019955848903944504	0
www.youtube.com:443	true	QUIC_VERSION_30	[2a00:1450:400e:801::200e]:443	1993955220975030604	0
yt3.ggpht.com:443	true	QUIC_VERSION_30	[2a00:1450:400e:801::2001]:443	12318925982785982092	0

InterPlanetary File System (IPFS)

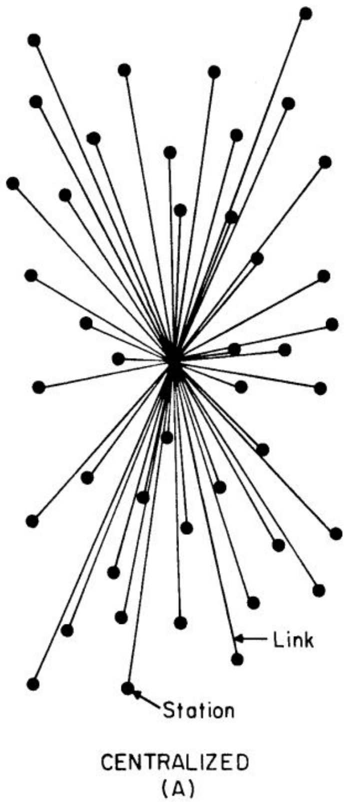


FIG. 1 — Centralized, Decentralized and Distributed Networks

http://

IPFS

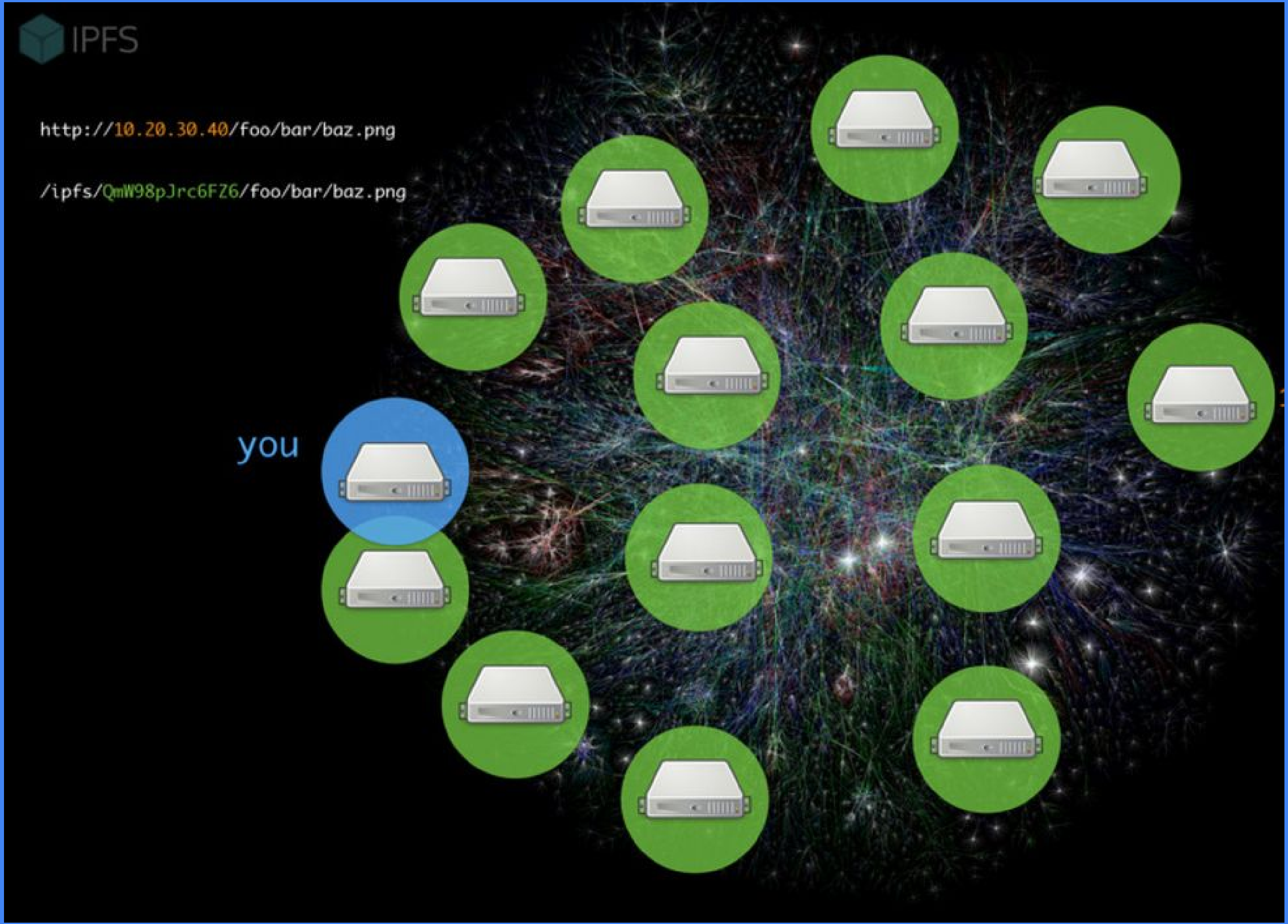
Lo
Di
Ba
Su
Ul
Pe

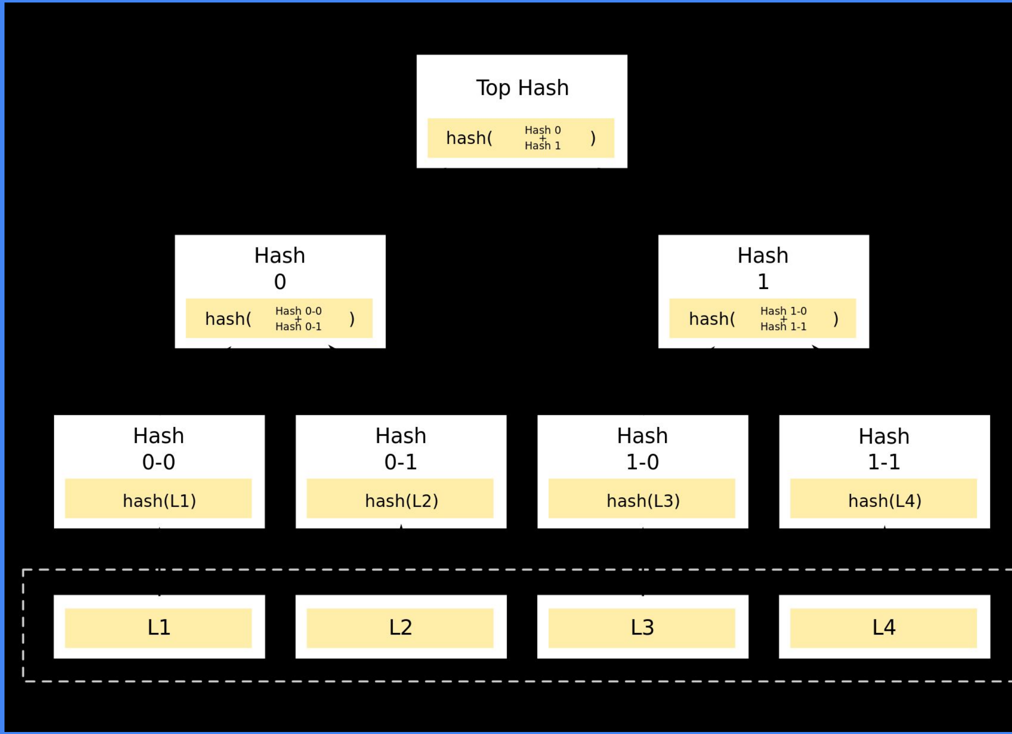
```
http:/                    z.png

http:/                    z.png
```

What do we want?

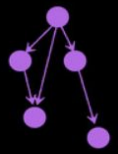
Offline
Smarter
Distributed
Permanent
Safer
Faster


IPFS
http://10.20.30.40/foo/bar/baz.png
/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png
you


Top Hash
hash(Hash 0 + Hash 1)
Hash 0
hash(Hash 0-0 + Hash 0-1)
Hash 1
hash(Hash 1-0 + Hash 1-1)
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4

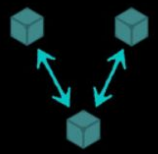
the model

merkledag



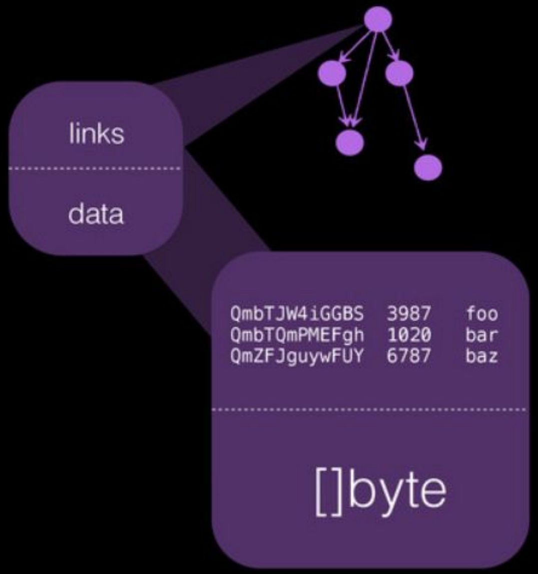
(content)

IPFS nodes



(programs)

in IPFS
data forms a dag

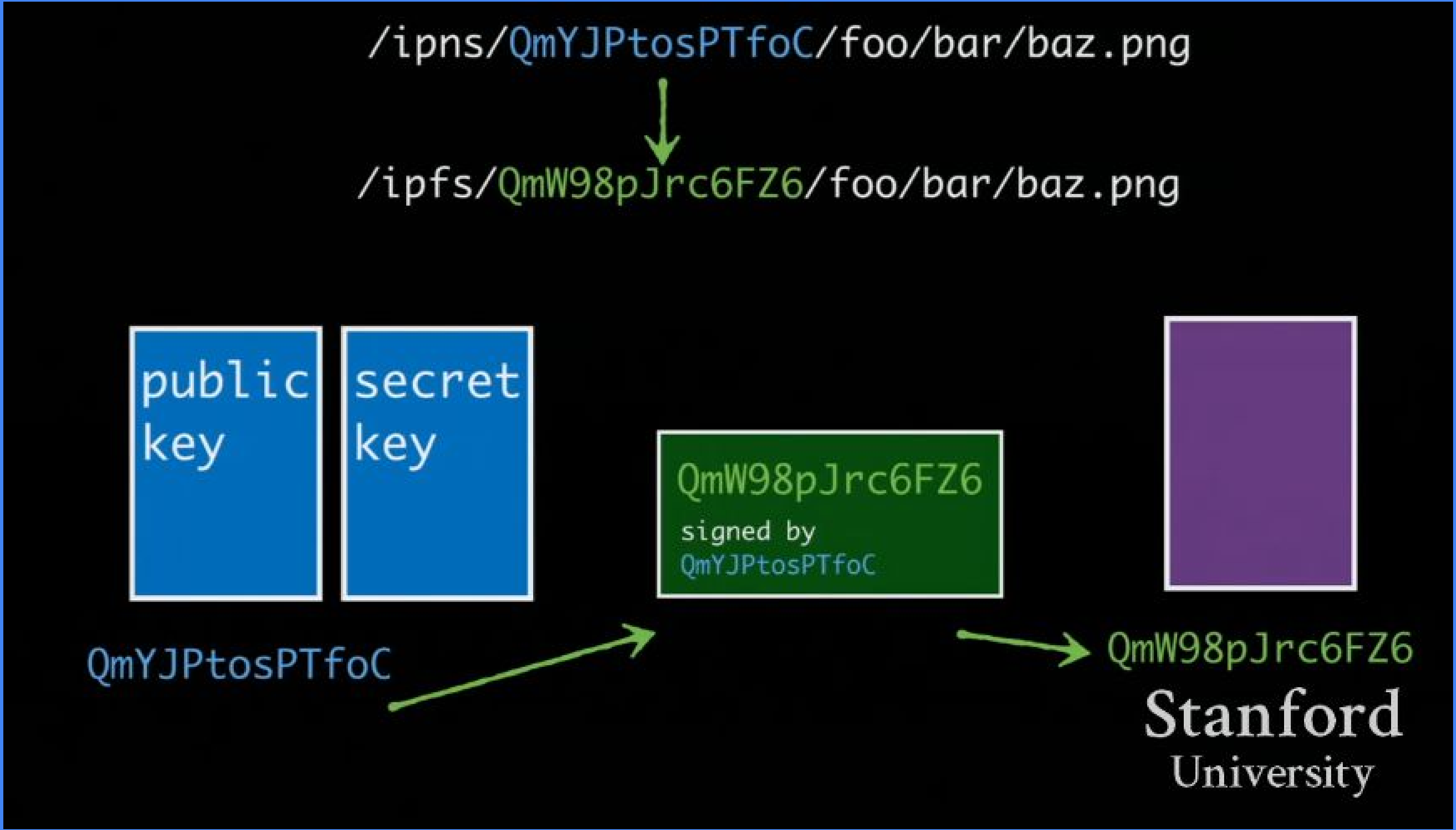


[(hash, size, name), …]

anything you want

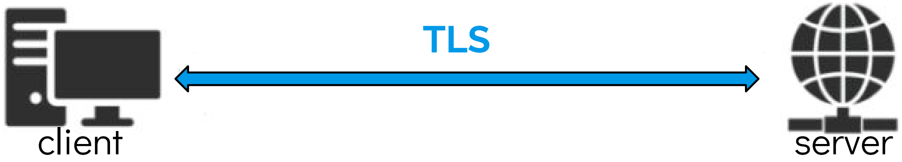
nodes have
links and data

Mutability?

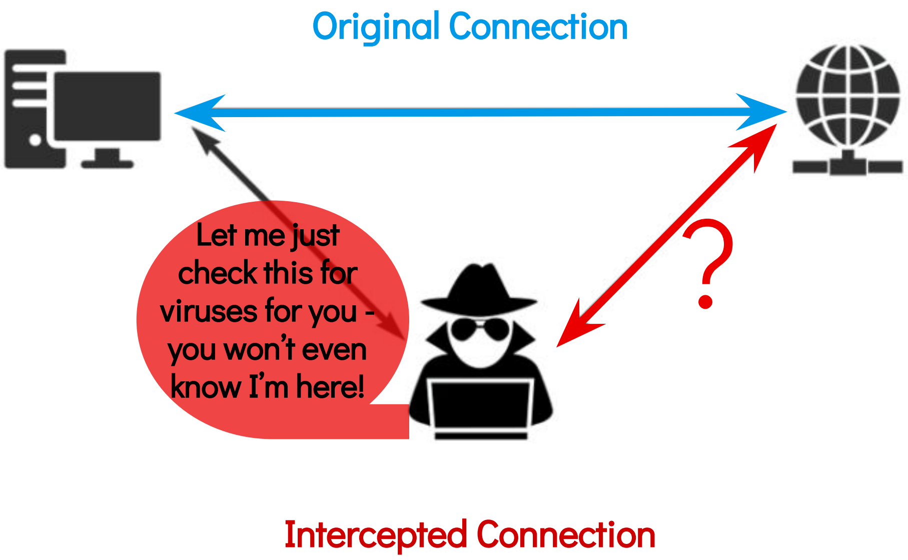

/ipns/QmYJPtosPTfoC/foo/bar/baz.png
/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png
public key
secret key
QmW98pJrc6FZ6
signed by
QmYJPtosPTfoC
QmYJPtosPTfoC
QmW98pJrc6FZ6
Stanford University

Multi-Context TLS

Bethlehem Naylor

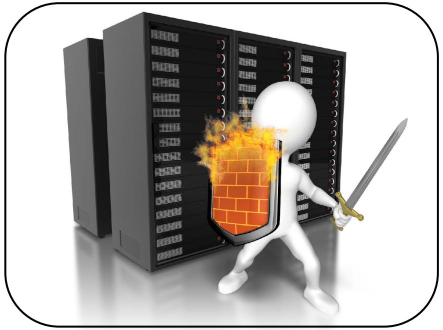


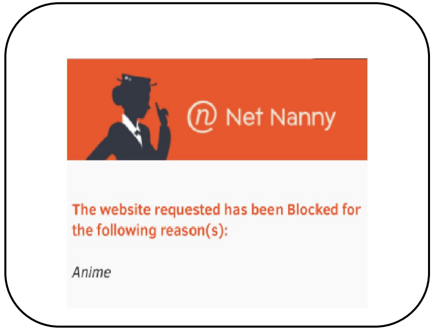
TLS protocol secures communication between **exactly two parties**



in reality: most connections are augmented along their path by middleboxes

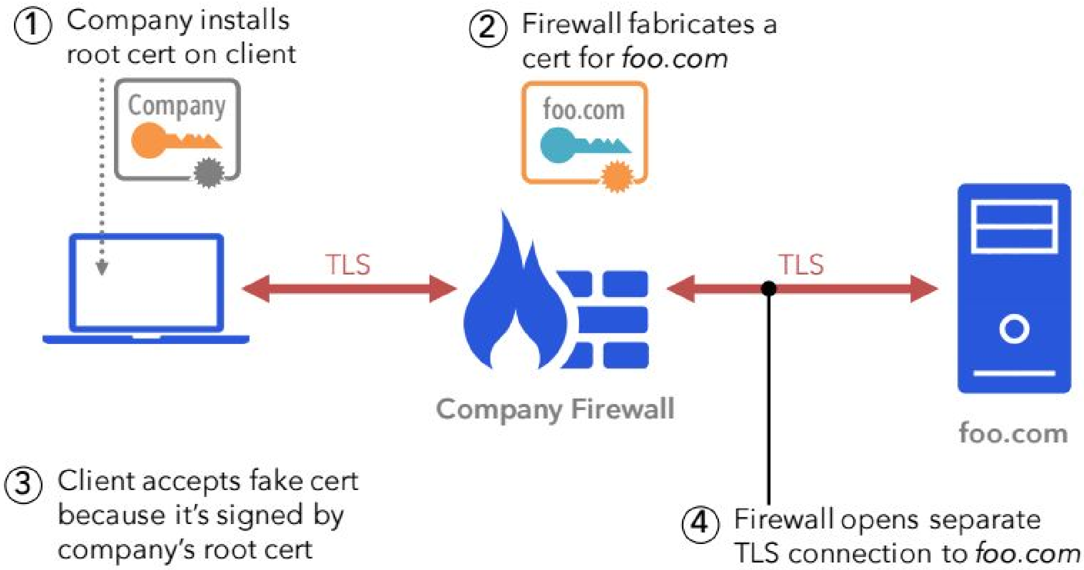
Are middleboxes the enemy?





middleboxes are ~~a necessary evil~~ integral, useful, and here to stay

Are middleboxes the enemy?



middleboxes break TLS

TLS

identity authentication

data secrecy & integrity

+

middlebox support

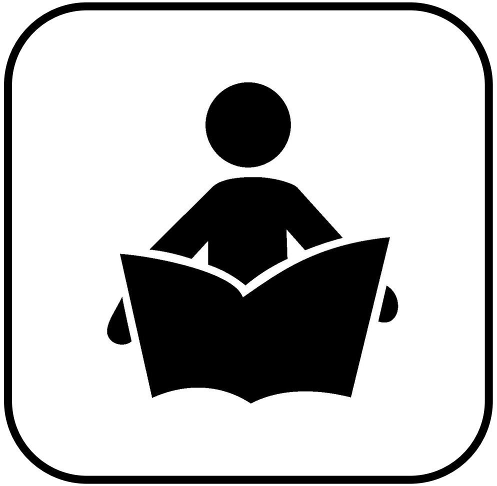
+least privilege

=

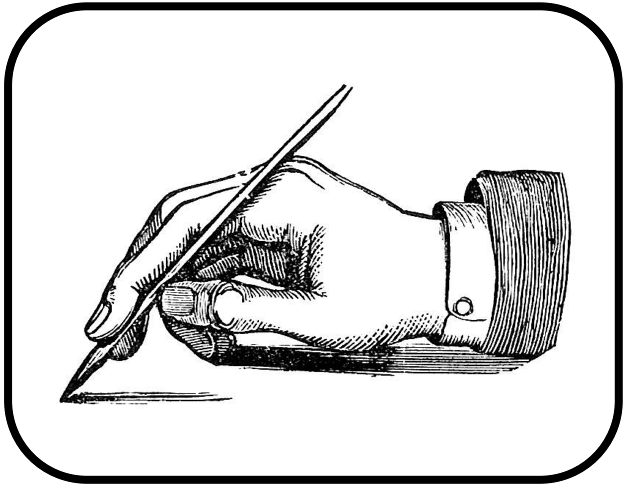
+endpoint agreement

multi-context TLS

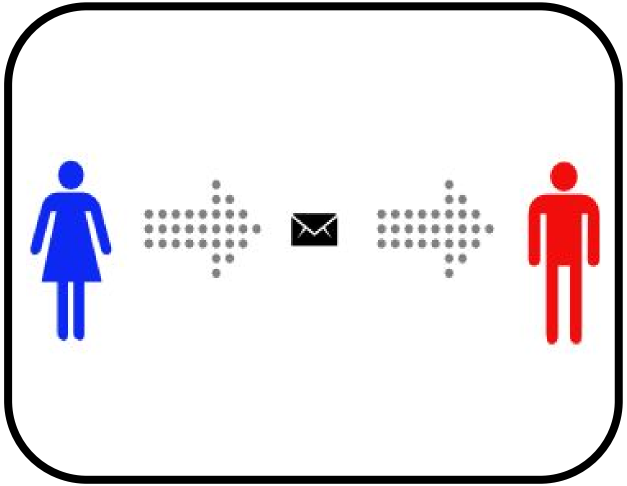
Least Privilege



Readers



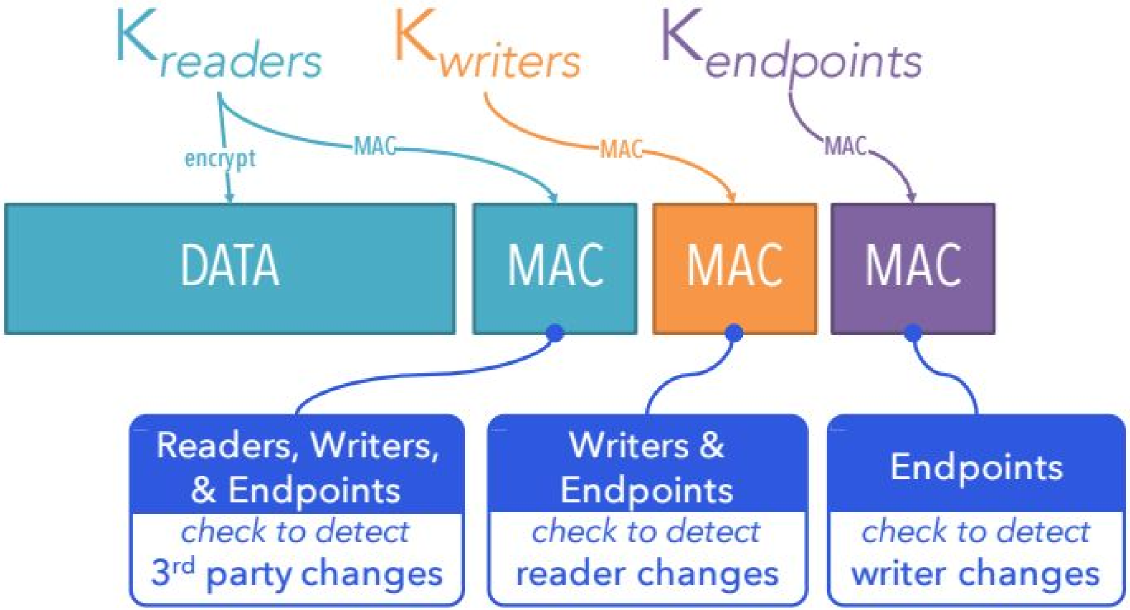
Writers



Endpoints

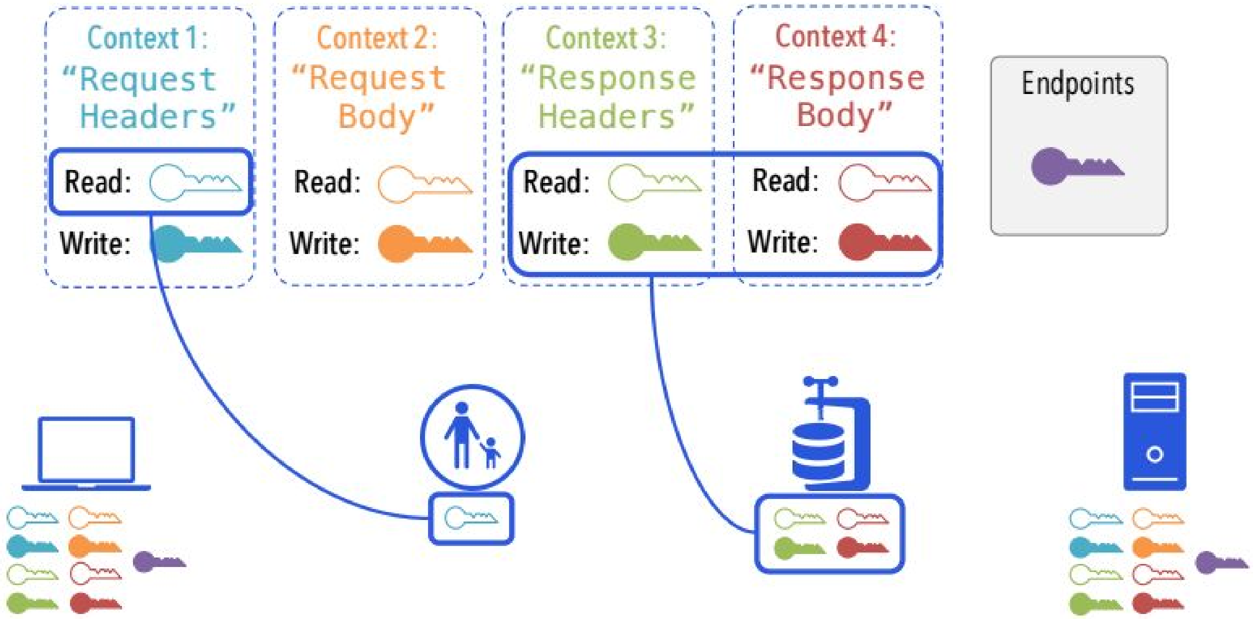
3 different access levels

Least Privilege



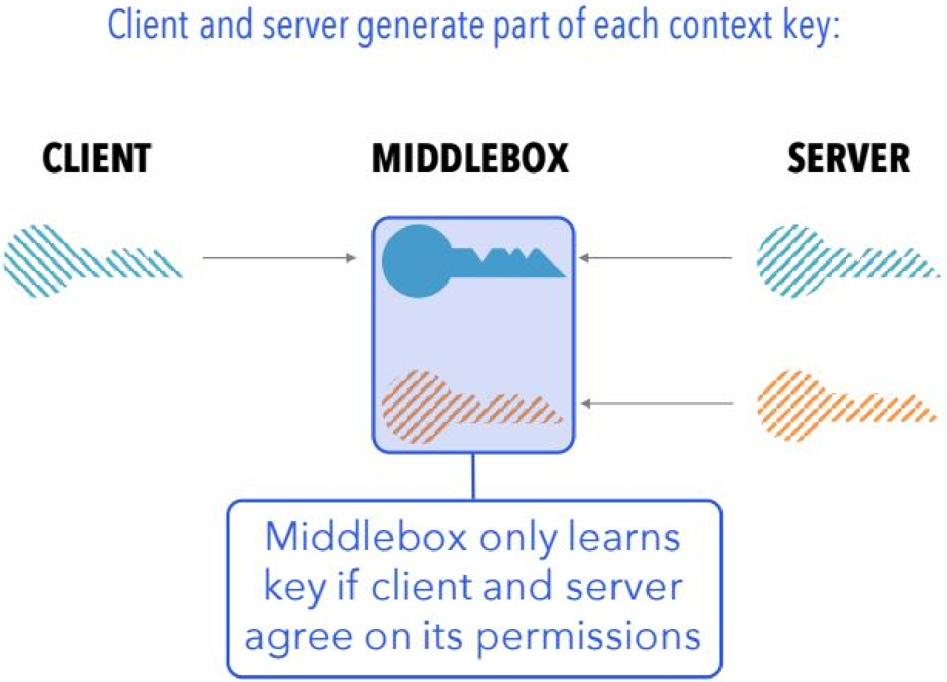
3 different encryption keys that grant 3 different access levels

Encryption Contexts



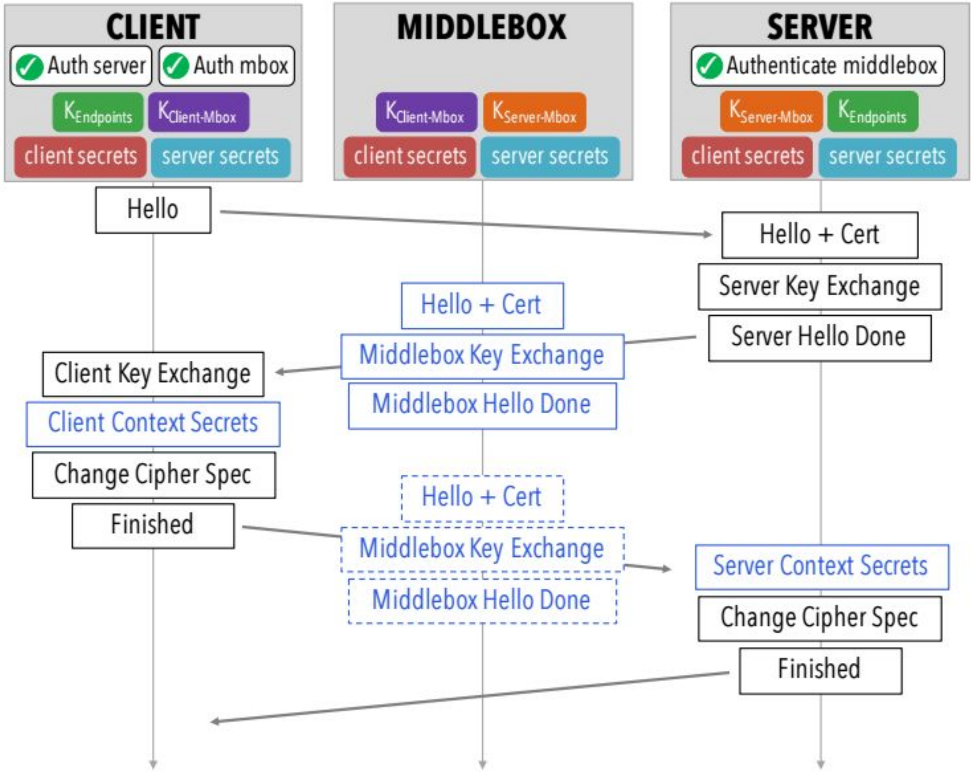
readers & writers receive **minimal access necessary** to do their jobs

Encryption Contexts

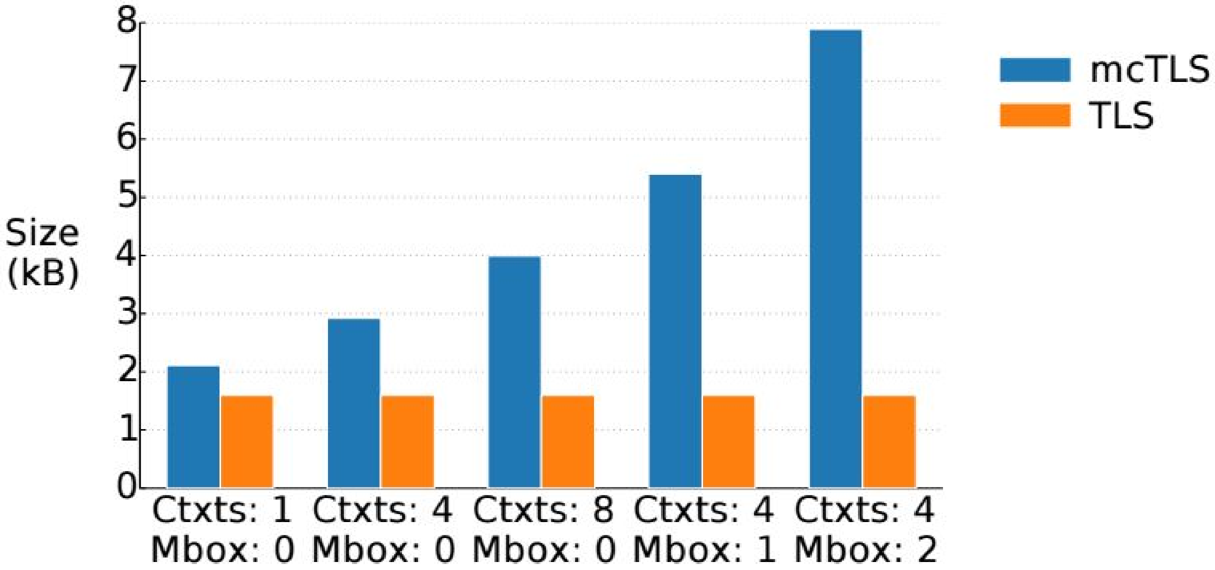


Client and server explicitly grant consent to use middleboxes

Handshake Protocol

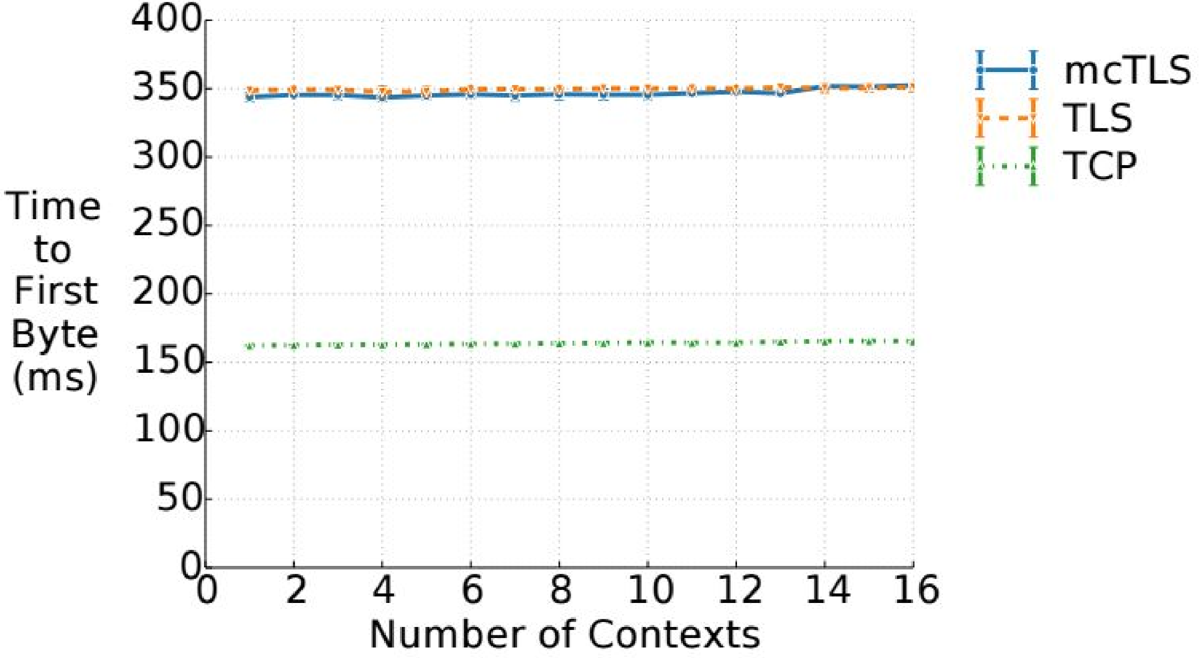


mcTLS: Performance



mcTLS increases handshake size

mcTLS: Performance



"You can try but it's pretty small in here...the water's going cold and the good soap is gone."

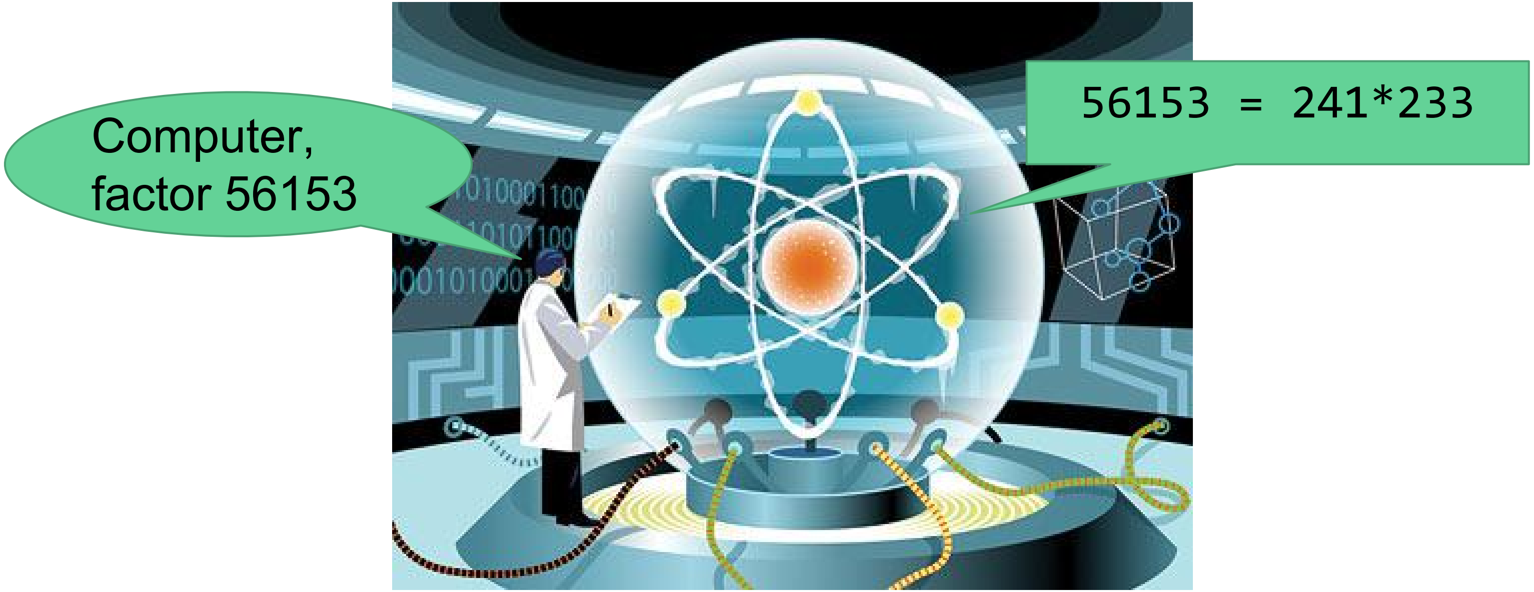
POST QUANTUM

STOP HERE FOR NOW!

PQC: An Introduction

Attacking Public-key Crypto

- Diffie-Hellman Key Exchange: given g^x, find x
- RSA Encryption / Signatures: given $n = p \cdot q$, find p and q
- Shor's algorithm breaks both in polynomial time



Post-Quantum Cryptography

- Many schemes resist attacks from quantum computers
    - Secret-key cryptography
    - Lattice-based cryptography
    - Hash-based cryptography
    - Code-based cryptography
    - Multivariate-quadratic-equations cryptography
    - Meet-privately-in-a-sealed-vault cryptography

Post-Quantum Cryptography

- Many schemes resist attacks from quantum computers
  - Secret-key cryptography
  - Lattice-based cryptography
  - Hash-based cryptography
  - Code-based cryptography
  - Multivariate-quadratic-equations cryptography
  - Meet-privately-in-a-sealed-vault cryptography
- Why don’t we use them?
  - Efficiency
  - Confidence
  - Usability

NIST PQC (http://csrc.nist.gov/groups/ST/post-quantum-crypto/)

- The National Institute of Standards and Technology (NIST) is looking to standardize quantum-resistant public-key crypto schemes
- Evaluation criteria
  - Security
  - Cost
    - Key, ciphertext, signature sizes
    - Computational efficiency
  - Simplicity
- Timeline
  - Submit your proposal by November 30
  - 3–5 years of public scrutiny
  - 2 years of writing standards

NewHope: TLS with PQC

Quantum Computers vs. TLS

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

Post-Quantum Key Exchange

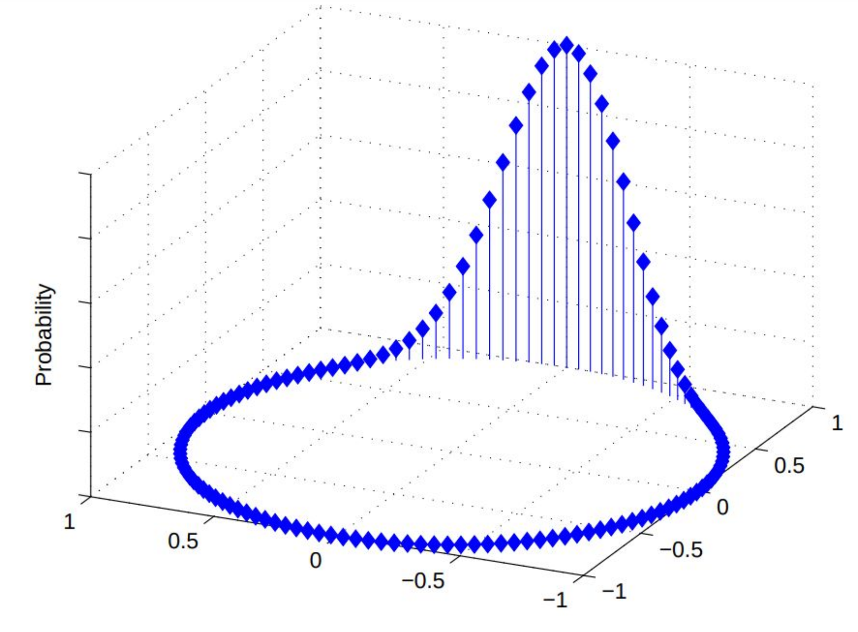
TLS_**RLWE**_ECDSA_WITH_AES_128_GCM_SHA256

(Ring Learning With Errors)

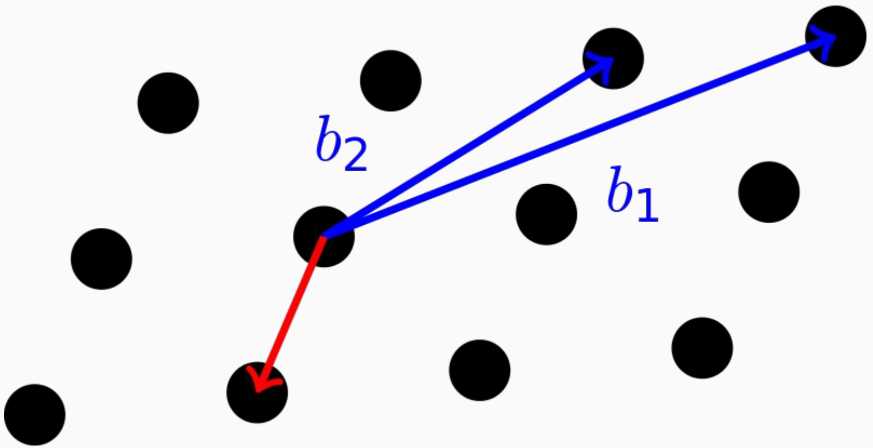
Post-quantum key exchange for the TLS protocol from the ring learning with errors problem
Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila
http://eprint.iacr.org/2014/599.pdf

Ring Learning with Errors

- Given (a, $a \cdot s + e$), find s
- a, s, e are complex integers: $n + mi$
  - Modulo prime $q = 2^{32} - 1$
- The error e is small
- Decision problem: distinguish between (a, $a \cdot s + e$) and (a, b) for random b

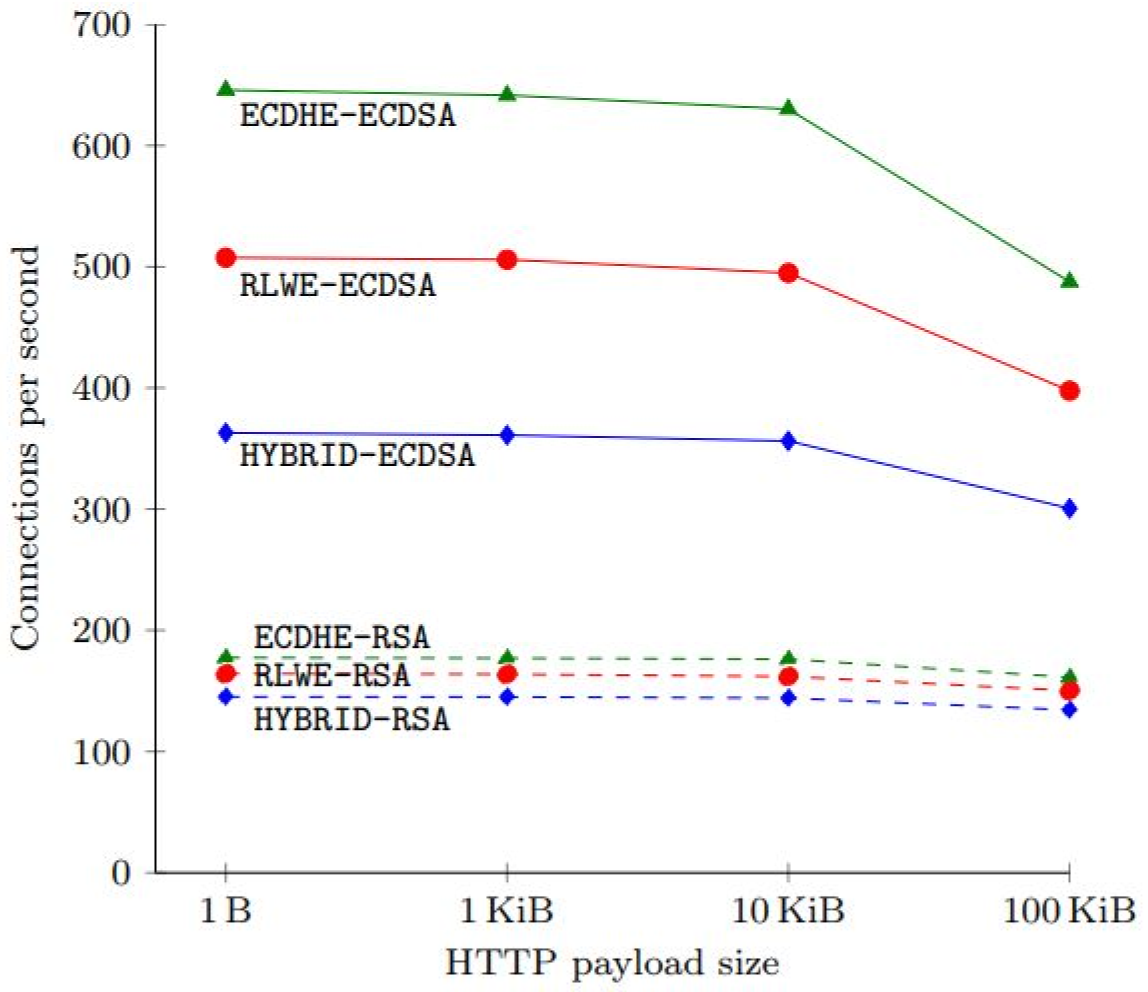


RLWE reduces to the Shortest Vector Problem



Operation	Client constant-time	Server constant-time	Client non-constant-time	Server non-constant-time
R-LWE key generation	0.9	1.7	0.6	1.3
R-LWE Bob shared secret	0.5	(1.1)	0.4	(0.9)
R-LWE Alice shared secret	(0.1)	0.4	(0.1)	0.4
Total R-LWE runtime	1.4	2.1	1.0	1.7
EC point mul., `nistp256`	0.4	0.7	—	—
Total ECDH runtime	0.8	1.4	—	—
RSA sign, 3072-bit key	(3.7)	8.8	—	—
RSA verify, 3072-bit key	0.1	(0.2)	—	—

Table 2: Average runtime in milliseconds of cryptographic operations using `openssl speed` Numbers in parentheses are reported for completeness, but do not contribute to the runtime in the client and server's role in the TLS protocol.



Connections per second
700
600
500
400
300
200
100
0
ECDHE-ECDSA
RLWE-ECDSA
HYBRID-ECDSA
ECDHE-RSA
RLWE-RSA
HYBRID-RSA
1 B
1 KiB
10 KiB
100 KiB
HTTP payload size

A New Hope

“We more than **double** the security parameter, **halve** the communication overhead, and speed up computation by more than **a factor of 8** in a portable C implementation and by more than **a factor of 27** in an optimized implementation targeting current Intel CPUs”

Post-quantum key exchange – a new hope
Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe
https://eprint.iacr.org/2015/1092.pdf

	BCNS [22]	Ours (C ref)	Ours (AVX2)
Generation of **a**		43 440[a] (43 607)[a]	37 470[a] (36 863)[a]
NTT		55 360	8 448
NTT^{-1}		59 864[b]	9 464[b]
Sampling of a noise polynomial		32 684[c]	5 900[c]
HelpRec		14 608	3 404
Rec		10 092	2 804
Key generation (server)	≈ 2 477 958	258 246 (258 965)	88 920 (89 079)
Key gen + shared key (client)	≈ 3 995 977	384 994 (385 146)	110 986 (111 169)
Shared key (server)	≈ 481 937	86 280	19 422

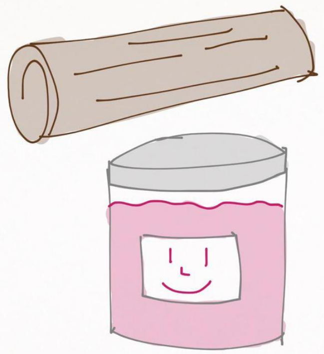
[a] Includes reading a seed from `/dev/urandom`

[b] Includes one bit reversal

[c] Excludes reading a seed from `/dev/urandom`, which is shared across multiple calls to the noise generation

Preventing Backdoors

- Given (a, $a \cdot s + e$), find s
- "for standardization purposes, a single *a* value should be generated in a verifiably random, 'nothing up my sleeve' manner" - BCNS

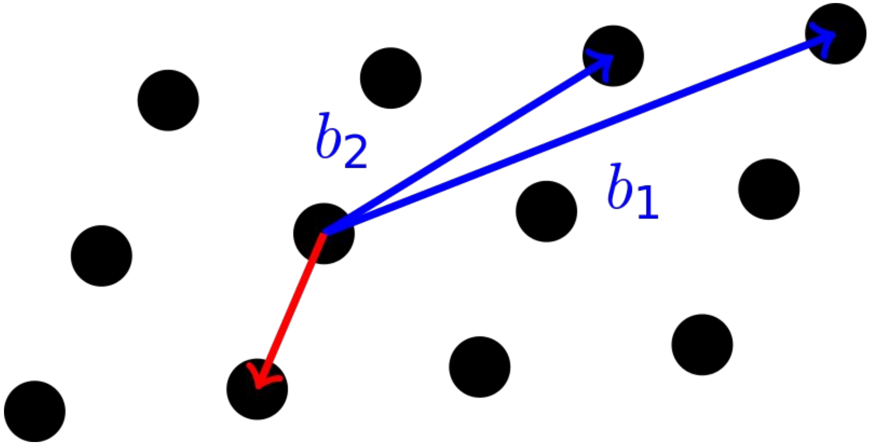


Google’s Results

- Combine existing ECDHE with New Hope
- “Although the median connection latency only increased by a millisecond, the latency for the slowest 5% increased by 20ms and, for the slowest 1%, by 150ms.”
- “we did not find any unexpected impediment to deploying something like NewHope”

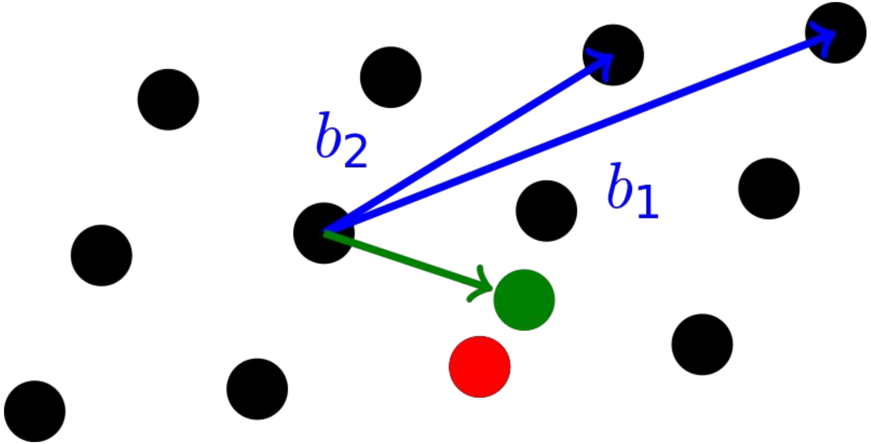
Alternative PostQuantum

What we’ve seen so far



Shortest Vector Problem

Closest Vector Problem



But What About Everything Else?

- **Secret-key cryptography**
- ~~Lattice-based cryptography~~
- **Hash-based cryptography**
- **Code-based cryptography**
- **Multivariate-quadratic-equations cryptography**

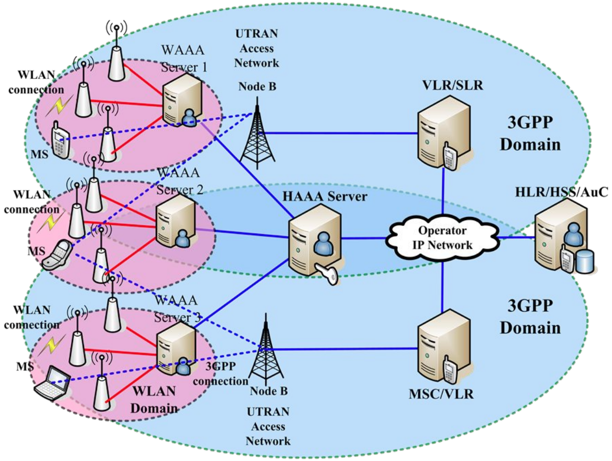
Secret-Key Cryptography (Symmetric)

Stream Ciphers vs. **Block Ciphers**

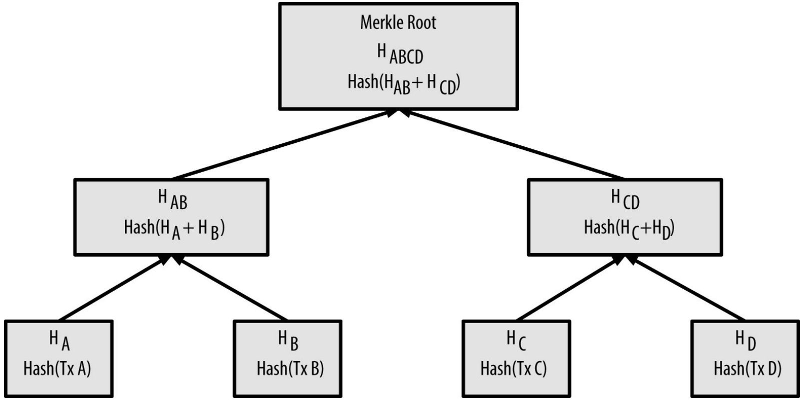
Twofish, Serpent, **AES** (Rijndael), Blowfish, CAST5, Kuznyechik, RC4, 3DES, Skipjack, Safer+/++ (Bluetooth), and IDEA

Symmetric Key Management (Kerberos & 3GPP)

Benefit: Widespread already, just expand!



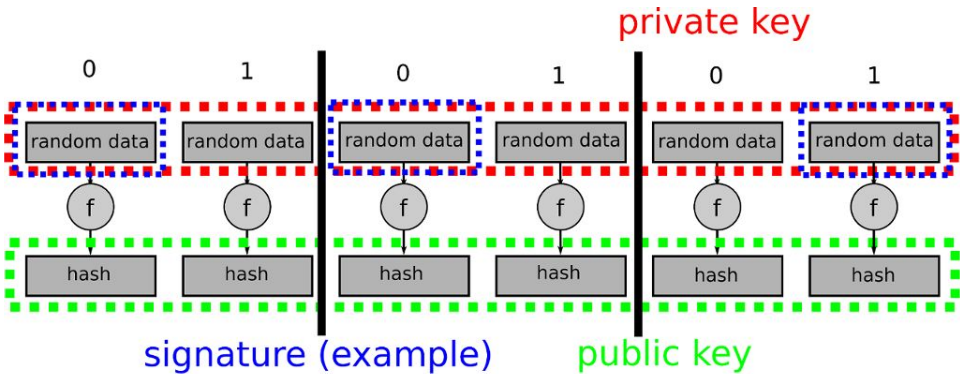
Hash-Based Cryptography



Lamport-Diffie and Merkle Trees

Before PQ: little interest from limit on number of signatures

Chaining!



Benefit: Provable Reductions!

Drawback: Security?

Oh No! - Collisions

HASH	BITS	CPB	'89	'90	'91	'92	'93	'94	'95	'96	'97	'98	'99	'00	'01	'02	'03	'04	'05	'06	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16	'17
MD2	128	638	[21]																					[illegible]							
Snefru-2	128	?		[3]	[19]																										
MD4	128	4.0		[22]					[20]																						
RIPEMD	128	?		[23]														[7]													
MD5	128	5.1				[24]												[7]													
HAVAL-256-3	256	?				[25]											[11]														
SHA-0	160	?					[26]		[†]											[27]											
GOST	256	?						[28]														[14]									
SHA-1	160	18							[29]										[15]								[51]				[53]
RIPEMD-160	160	17								[30]														[illegible]							
Tiger	192	6.2								[31]																					
Panama	512	2.5										[33]				[34]					[35]										

Code-Based Cryptography

McEliece and Niederreiter

Example:
Public Key is dt ✖ n matrix K. Messages are n-bit strings of “weight t” (n-bit strings having exactly t bits set to 1). Encrypt message m by multiplying K by m. Receiver creates a “hidden Goppa code” to decrypt

Benefit: Extremely efficient key generation, encryption, and decryption

Drawback: Long public keys

European Commission

Multivariate-Quadratic Cryptography

Rainbow, HIdden Field Equations (HFE), UOV Cryptosystems,

Sequence of polynomials and variables with coefficients. Each polynomial required to have a degree of at most 2, with no squared terms.

Verify signatures with standard hash function (but then why not hash-based?)

Shorter Public Keys!

Drawback: Efficient but lots of exploitable mathematical structure

Supersingular Isogeny Diffie-Hellman (SIDH)

Diffie-Hellman broken by quantum computers on *general grounds*, no matter the implementation chosen.

Need a model with: exponentially many subgroups, ways to identify quotients up to isomorphism, resolve how receivers decrypt message without knowing the encryption function

Supersingular elliptic curve: very large and non-commutative ring

Benefits: Forward Secrecy & Small Keys (3072 public)

Going from Here

Need to improve **efficiency**, build **confidence**, and improve **usability** of PQC

Efficiency: So far, no O(*b*)-bit signatures, O(*b*)-bit keys, polynomial signing, and polynomial verification in one PQ algorithm.

Confidence: Need to gain familiarity with PQC and PQ cryptanalysis

Usability: Need software implementations (with correctness and speed **BUT** without timing and other side channel leaks)