Network Security

Traffic analysis and anonymization

Radboud University, The Netherlands

Spring 2019

A short recap

- Best protection against active and passive attackers: encrypt and authenticate all traffic
- Different security for encryption on different layers:
  - Link-layer: protection between two network "neighbors" (example: WPA2)
  - Network layer: encryption between two nodes (example: IPsec)
  - Transport layer: encryption between client and server process (example: SSL/TLS)
  - Application layer: end-to-end encryption between applications (example: PGP)
- IPsec has two modes of operation: transport and tunneling
- IPsec has two main protocols: authentication headers (AH) and encapsulating security payloads (ESP)
- IPsec's Security Associations (SA) establish unidirectional security relations
- Modern VPN Solution: WireGuard
- TLS uses (long) handshake to agree on cipher suites, establish session keys
- Latest version (TLS 1.3) cleans up various crypto issues

How much web traffic is encrypted?

WIRED

GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY DESIGN OPINION MAG

ENTERPRISE encryption https

Encrypted Web Traffic More Than Doubles After NSA Revelations

BY KLINT FINLEY 05.16.14 | 5:14 PM | PERMALINK

Share 425 Tweet 1,018 +1 143 Share 109 Pin it 6

No crypto

From the article:

> *"Early last year–before the Snowden revelations–encrypted traffic accounted for 2.29 percent of all peak hour traffic in North America, according to Sandvine's report. Now, it spans 3.8 percent. But that's a small jump compared to other parts of the world. In Europe, encrypted traffic went from 1.47 percent to 6.10 percent, and in Latin America, it increased from 1.8 percent to 10.37 percent."*
> *—Klint Finley on wired.com, May 16, 2014.*

. . . update from 2015

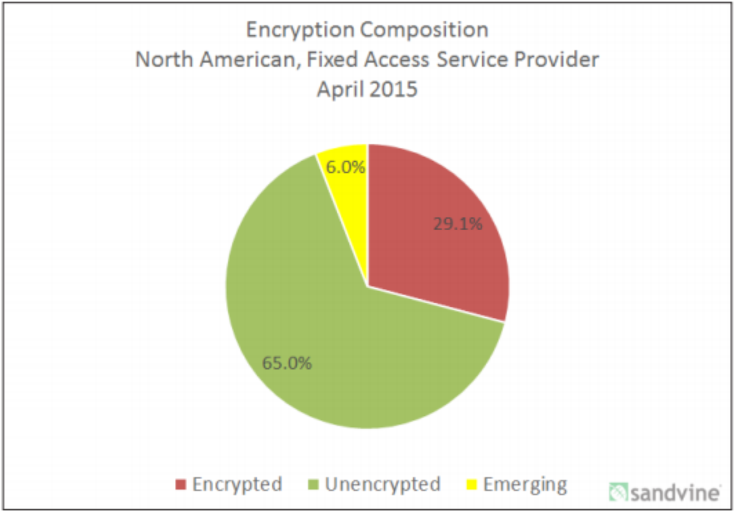


Figure 1 - Encryption Composition - North America, Fixed Access - April 2015

. . . estimated for 2016

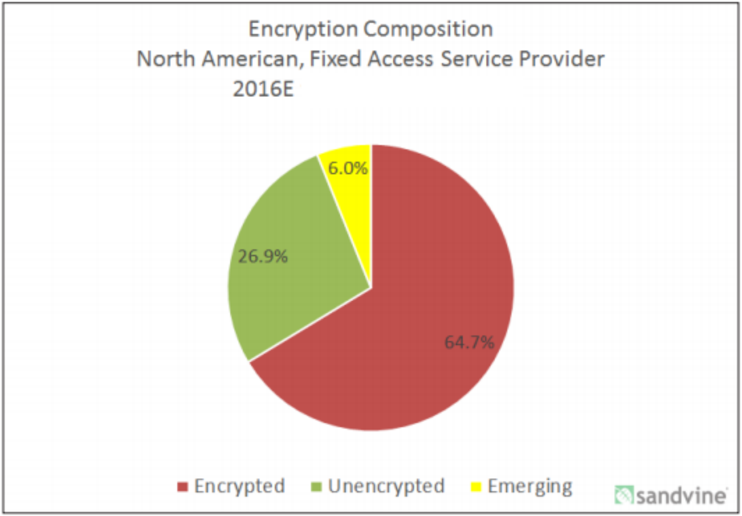
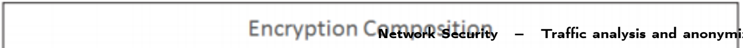


Figure 2 · Encryption Composition · North America, Fixed Access · 2016 Estimate



. . . and for 2019

NSS LABS

RESEARCH LIBRARY | CLOUD PLATFORM LOGIN | FREE TRIAL | MY ACCOUNT

SECURITY TEST | CLOUD PLATFORM | RESEARCH & ADVISORY | COMPANY | RESOURCES

About NSS | Analysts | Executive Team | Board of Directors | **News**

NSS Labs Predicts 75% of Web Traffic Will Be Encrypted by 2019

SSL/TLS encrypted internet traffic grew 90% year over year from July 2015 to July 2016.

AUSTIN, Texas – November 9, 2016 - NSS Labs, Inc., the world's leading cyber security product research, testing, and advisory company, today released new research examining the usage of Secure Socket Layer (SSL) and Transport Layer Security (TLS) encryption. SSL/TLS enables secure transmissions of private data over the internet, including credit card details, passwords and sensitive personal information. Enterprises use SSL/TLS to encrypt their traffic in order to address multiple issues including controlling access, confidentiality and reducing exposure to protocol–specific attacks (e.g. Heartbleed).

As part of on-going research and analysis, NSS Labs found that HTTPS (SSL/TLS encrypted) internet traffic grew over 90% year over year, with more than 40.5% of websites encrypting traffic by default in July 2016 vs. 21.3% in July 2015. Unsurprisingly, **97%** of surveyed enterprises are seeing an increase in encrypted web traffic. NSS predicts this trend to continue with 75% of all web traffic to be encrypted by 2019.

Key findings include:

- More non-enterprise traffic is encrypted than enterprise traffic, depending on region, type of content, etc.
- Over 40% of the most visited websites are encrypted by default; less than 10% have HTTPS properly applied (source: Trustworthy Internet Movement).
- Encryption does not protect us against all threats

The perfect world (crypto-wise)

Imagine a world in which . . .

- . . . all Internet traffic is encrypted and authenticated,
- . . . e-mails are all PGP encrypted and signed,
- . . . everybody is using cipher suites that offer high security,
- . . . all trusted parties are trustworthy,
- . . . crypto implementations are correct and secure,
- . . . applied cryptographers have trouble finding a job.

What *does* an attacker see?

EU's Data Retention Directive (annulled 2014)

"Member States shall ensure that the categories of data specified in Article 5 are retained for periods of not less than six months and not more than two years from the date of the communication."

From Article 5:

- data necessary to trace and identify the source of a communication
- data necessary to identify the destination of a communication
- data necessary to identify the date, time and duration of a communication
- data necessary to identify the type of communication
- data necessary to identify users' communication equipment or what purports to be their equipment
- data necessary to identify the location of mobile communication equipment

Encrypting and authenticating content does not prevent any of this!

What can you do with "meta data"?

"Metadata absolutely tells you everything about somebody's life. If you have enough metadata you don't really need content... [It's] sort of embarrassing how predictable we are as human beings."

—Stewart Baker, former general counsel of the NSA

"We kill people based on metadata."

—Michael Hayden, former director of the NSA and the CIA

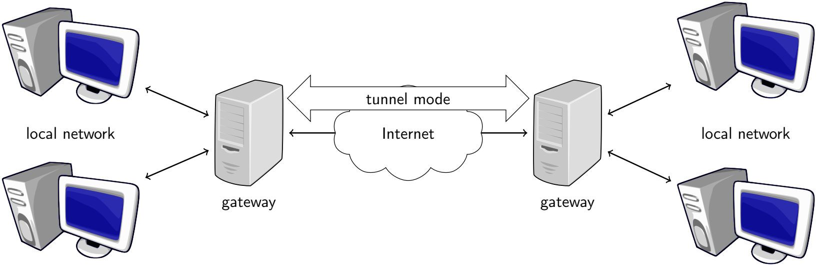
Is "metadata" all an attacker gets?

- Common assumption: an attacker sees only traffic data ("meta data")
- Example, interview with Jimmy Wales (Wikipedia founder):

  "***You've said that you're going to start encrypting communications on Wikipedia as a result***...

  We have done. It's not completely finished yet but the only thing that GCHQ, hopefully, can see is that you're looking at Wikipedia. They can't see which article you're reading. It's not the government's business to know what everybody is reading."
- Small experiment: 10 Wikipedia pages, load one at random through HTTPS
- Attacker sniffs the network, tries to figure out which one
- Not that hard:
  - 10 webpages have different amount of pictures
  - 10 webpages have different (length of) URLs to pictures
  - Attacker can count bytes of requests to image server
  - This is not the only thing an attacker sees: number of requests, delays, same for replies...

IPsec ESP in tunnel mode



- Everything between the gateways has the *gateways' addresses*
- Nodes behind the gateways are indistinguishable
- RFC 2406 calls this "limited traffic flow confidentiality"
- Problem 1: Does not help against state-level attacker who can request gateway's logfiles
- Problem 2: Potentially small *anonymity set*

Anonymizing proxies

- Somewhat similar idea (without crypto): use a proxy server
- Typically: application-specific proxies (e.g., HTTP proxies)
- Requests to websites come from proxy
- All users behind the proxy are indistinguishable
- Various problems:
  1. Single point of failure against state-level attackers
  2. Proxy somewhere in the Internet: easy to correlate ingoing/outgoing traffic
  3. No crypto protection to the proxy
- Can add crypto to the proxy (e.g., OpenVPN Service)
- That still does not solve problems 1 and 2

Mix Networks

- Idea for anonymous electronic mail by Chaum, 1981: mixing networks
- Assume that Alice wants to anonymously send message M to Bob
- Uses intermediate computer called *mix* and public keys
  - K_B of Bob, and
  - K_M of the mix
- Sends to mix: $K_M(R_1, K_B(R_0, M), B)$ for random R_0, R_1
- Mix collects many such mails, decrypts to $(K_B(R_0, M), B)$
- Sends mails in lexicographic order to receivers
- Receiver Bob decrypts and obtains M
- Achieves anonymity if encrypted messages are indistinguishable
- Very important: never repeat input and output!
- Has high communication latency (wait for enough messages)

Return Addresses

- Now Alice can send mail to Bob, how about replies?
- Need a way for Bob to reply without revealing Alice's address/identity
- Alice includes a return address in her message encrypted to Bob:

$$K_M(R_1, A_X), K_X$$

- R_1, K_X are random one-time symmetric keys
- A_X is Alice's real address
- Bob can send response M as

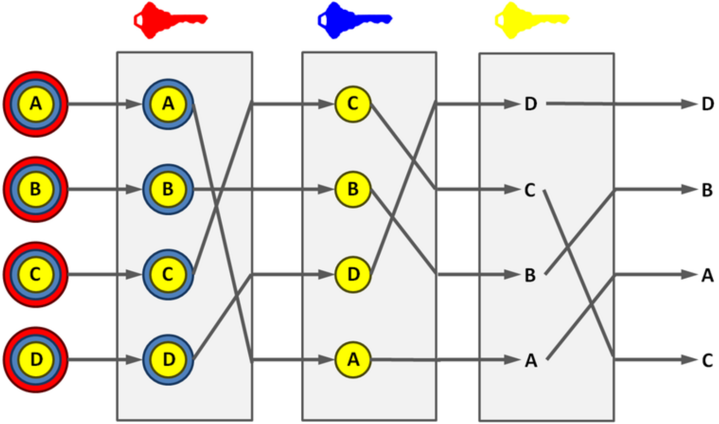
$$K_M(R_1, A_X), K_X(R_0, M)$$

- Mix receives and recovers R_1, A_X, sends to Alice

$$R_1(K_X(R_0, M))$$

- Only Alice can decrypt, because only she knows both K_X and R_1

Cascading Mixes



Mix Nets vs. Anonymizing proxies

Mix Nets

\+ No single point of failure (with cascading)
\+ Inbound/output-traffic analysis does not de-anonymize
\+ Generally good anonymity
– Slow public-key cryptography (at least in vanilla mix nets)
– Long latency

Anon. Proxies

\+ Low latency
\+ No overhead from slow crypto
– Single point of failure
– Inbound/output-traffic analysis de-anonymizes
– Fairly weak anonymity

Idea of Tor (The Onion Router): Combine advantages:

- Use cascade of "proxies", called *Tor relays* or *Tor nodes*
- Use fast symmetric crypto instead of asymmetric crypto

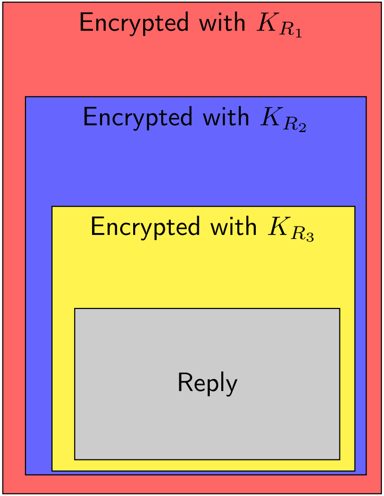
Onion Routing and Tor

- Assume that user shares symmetric keys with three *relays*:
  - Entry relay R_1 (key K_{R_1})
  - Middle relay R_2 (key K_{R_2})
  - Exit relay R_3 (key K_{R_3})
- Wants to anonymously send request to `www.wikileaks.org`
- Prepares packet as follows:
  - Write dest. `www.wikileaks.org`, encrypt with K_{R_3}
  - Write dest. R_3 encrypt with K_{R_2}
  - Write dest. R_2 encrypt with K_{R_1}
- Send this packet to R_1

Onion Routing and Tor

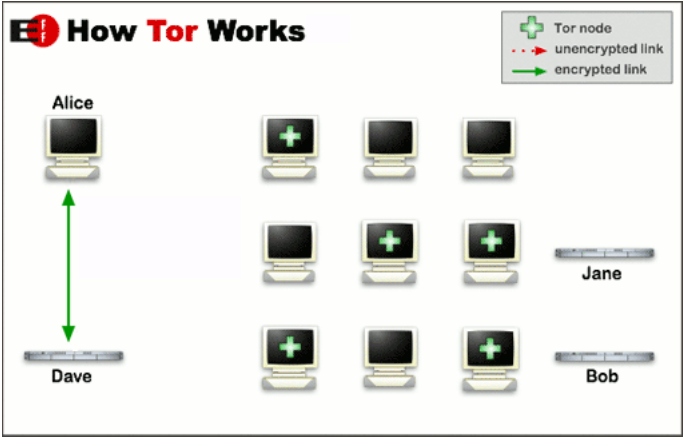
- R_1 receives packet, removes encryption with K_{R_1}
- Sees next destination: R_2, forwards
- R_2 receives packet, removes encryption with K_{R_2}
- Sees next destination: R_3, forwards
- R_3 receives packet, removes encryption with K_{R_3}
- Sees next destination: `www.wikileaks.org`, sends request

Reply from `www.wikileaks.org`



- R_3 receives reply from `www.wikileaks.org`
- R_3 encrypts with K_{R_3}, sends to R_2
- R_2 encrypts with K_{R_2}, sends to R_1
- R_1 encrypts with K_{R_1}, sends to Tor client
- Client removes all encryption, obtains reply

Establishing a Circuit



Request listing of Tor nodes from directory server (DS) Pick entry, middle, and exit node; obtain their public keys from DS

Attacks against Tor, part I

- Tor offers anonymity up to the transport layer
- Tor cannot offer application-level anonymity
- Example: I connect through Tor to a website and enter on that website:

  "My name is Peter Schwabe, I live in the Netherlands, my IP address is 83.163.166.232."
- Various Bittorrent clients do precisely this: send the IP address as part of application data
- Conclusion: Bittorrent over Tor isn't a good idea
- Browsers are easily identifiable, see Panopticlick by EFF
- Conclusion: Use the Tor browser (modified Firefox)
- Tor re-uses an existing circuit for new TCP connections for 10 minutes
- Leaking your IP address to Bittorrent may also de-anonymize your browser session (bad apple attack)!

Attacks against Tor, part II

- Tor provides anonymity as long as *not all three* relays attack *together*
- Possible attack: control all three relays on a path
- Anybody can run Tor relays, so can, for example, the NSA
- Let's assume that NSA runs 1% of the Tor relays
- Each circuit has a $1/1,000,000$ chance to be fully controlled by NSA
- Possible solution: longer circuits (problem: slower, less reliable)
- Better solution: more non-NSA relays

Correlation attacks

- Tor is aiming at low latency (for web browsing etc.)
- Tor does not wait for traffic to do mixing
- (Timing) correlation attack is still possible:
  - Think of the whole Tor network as one big proxy
  - Correlate traffic going into and out of this proxy
- Conclusion by Felix von Leitner (Fefe) on Aug 5, 2013:

  "Tor ist tot. Tor basiert auf der Annahme, dass der Gegner nicht in der Lage ist, das gesamte Internet zu überwachen." "Tor is dead. Tor is based on the assumption, that the opponent does not have the whole Internet under surveillance"
- Very controversial discussion ensued... see http://blog.fefe.de/?ts=af0134f5

“Tor stinks”

- Snowden leaked NSA slides “Tor stinks” from 2007
- Quotes from these slides:

  “We will never be able to de-anonymize all Tor users all the time.”

  *“With manual analysis we can de-anonymize a **very small fraction** of Tor users, however **no** success de-anonymizing a user in response to a TOPI request/on demand.”*

Tor as censorship circumvention

- Various countries filter Internet traffic by destination address
- Most prominent example: Golden Shield Project ("Great Firewall of China")
- Firewalls and gateways cannot see the true destination of Tor traffic
- Tor is a powerful tool to circumvent censorship (e.g., in China)
- Can also use Tor to circumvent country filters:
  - Need an IP address in the US: use Tor with US exit node
  - Need access to a specific paper: use Tor with exit node in some university

Bridges and pluggable transports

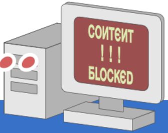
- Easy solution for censors:
  - Obtain list of Tor nodes from directory server
  - Block access to the Tor network (all relays)
- Solution: Tor bridges (entry nodes that are not in the public list)
- Obtain IP address of a bridge by
  - visiting `https://bridges.torproject.org/`
  - writing e-mail to bridges@torproject.org
- Censors can also block Tor by identifying Tor traffic
- Tor traffic is relatively easy to identify:
  - Disguised as HTTPS traffic, but
  - uses random domain names
  - has a characteristic packet-size distribution
- Solution: fully disguise Tor traffic as other traffic
- Pluggable Transport API allows communication between ofuscator and Tor client

★ YOU ★

Can Help Protect

Freedom Of Speech

Online...



Run a TOR RELAY Today!

★ ★ TorProject.org ★ ★