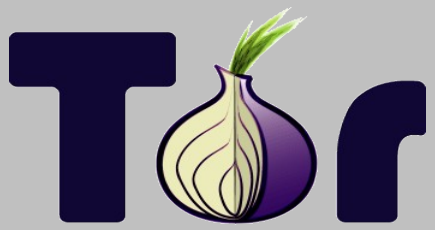
Privacy and Traffic Analysis Resistance for Encrypted Protocols

Mike Perry

The Tor Project



Topics and Goals

Topics:

- Quick Tor Overview
- Application layer privacy
- Traffic Analysis Attacks and Defenses

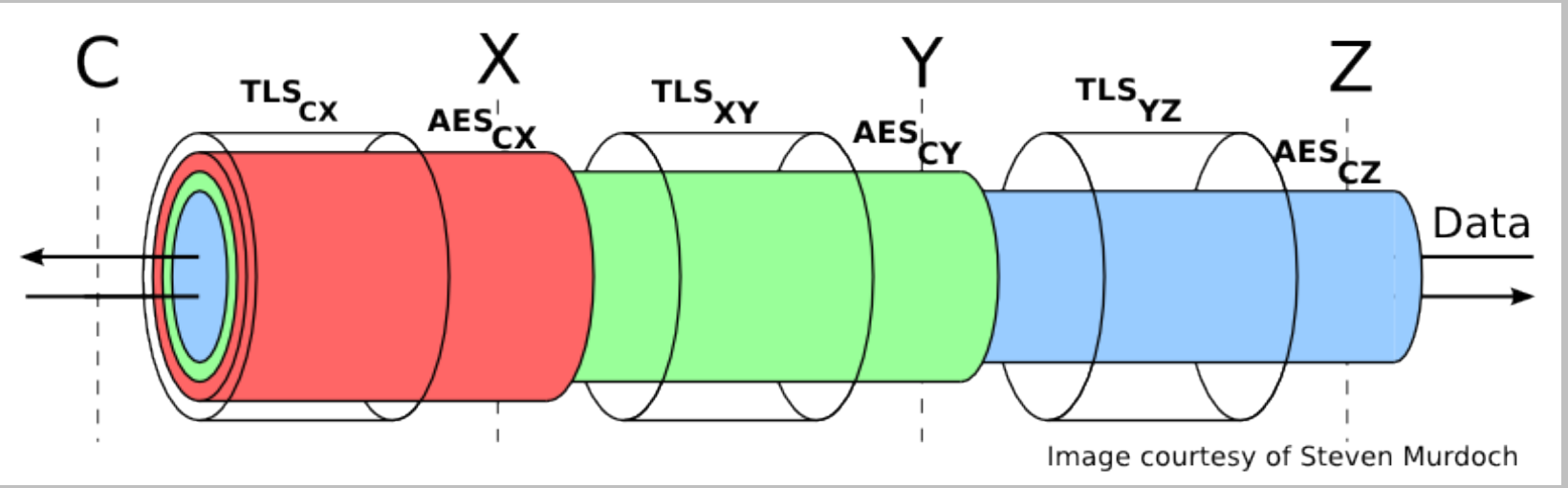
Goals:

- Raise awareness of Tor's threat model
- Spread knowledge of traffic analysis evaluation
- Develop allies and advocates in IETF

Tor Basics

- TCP Overlay Network; Stream abstractions
  - TCP SOCKS Proxy
- ~2 million daily users
  - Not the same users every day!
  - ~1 million users update the browser within 1 week
  - ~5 million Android installs
- Tor is a small non-profit company
  - 20 employees total; $3.5M budget
  - Standards participation is difficult for us

Tor Path Encryption



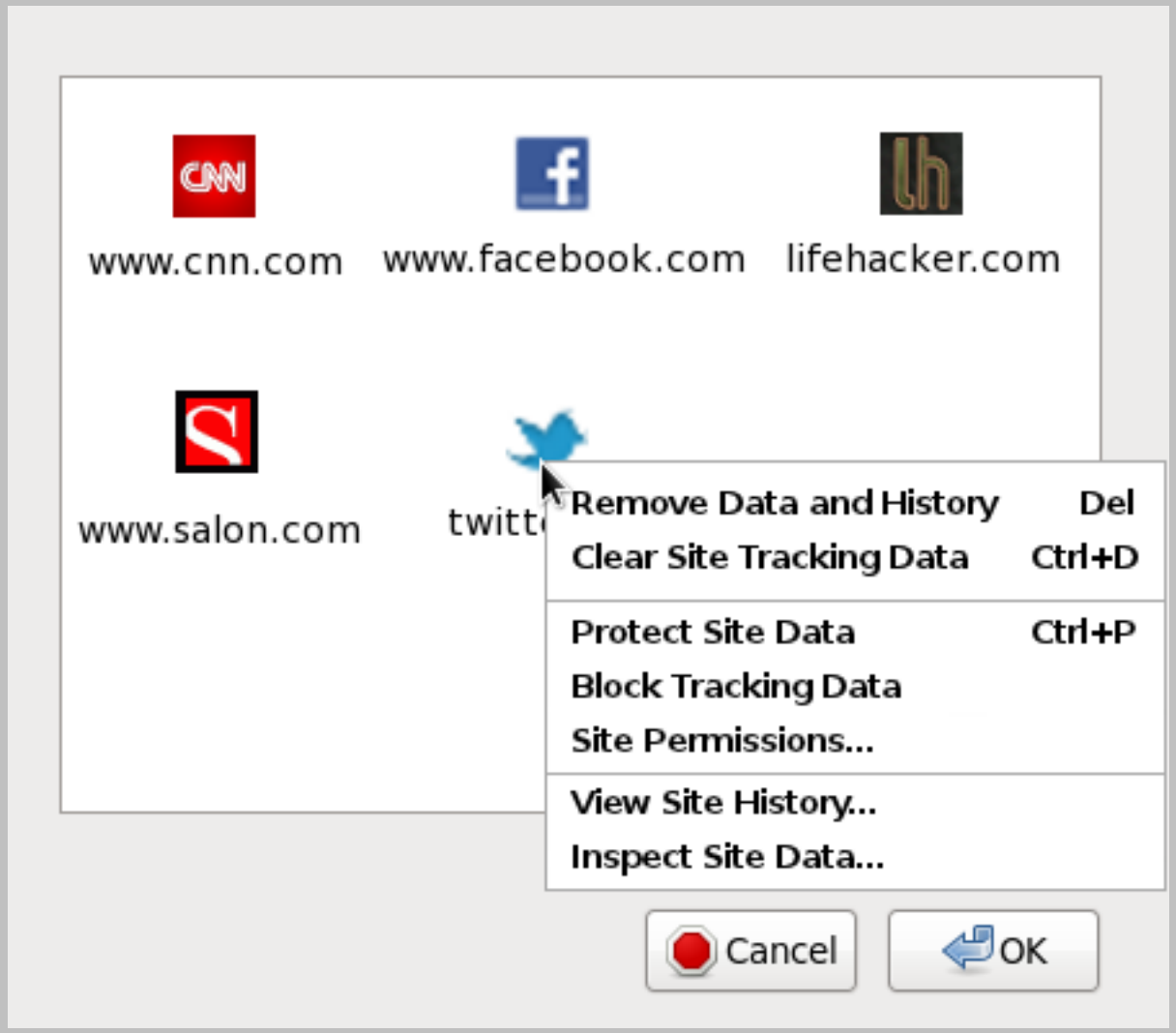
Terminology Normalization

- “Linkability”
  - The ability to associate one user action with another
  - Types: “PBM”; “3rd party”; “Fingerprinting”
- “Fingerprinting” != “Identifier storage”
  - Identifiers are content-accessible browser state (aka “supercookies”)
  - Fingerprinting is any stateless vector
- “First Party Isolation”
  - Bind all content-accessible browser state to the URL bar domain
  - AKA “Double-Keying”

Abstract Privacy and Anonymity Issues

- Traffic integrity and confidentiality
- Linkability sources
  - State management (supercookies/identifiers)
  - Browser fingerprinting
- Traffic analysis
  - Traffic fingerprinting
  - Correlation
  - Confirmation
  - Route manipulation and analysis

First Party Relationships



Identifier Storage in HTTP/2

- Alternative-Services Header caching
- ALPN and NPN successes cached to govern initial connection counts
- Server PUSH response caching

Identifier Storage in QUIC

- Superset of HTTP/2, plus:
  - 0-RTT state caching
  - Discovery and Alternate-Protocol state
  - 64bit connection-id (for third parties)
  - Congestion window information?

Tor's View of Fingerprinting

- Sources of fingerprinting in order of concern:
  1. End-user configuration details
  2. Device and hardware characteristics
  3. Operating System vendor and version differences
  4. User behavior
  5. Browser vendor and version differences (ignored)
- Fingerprinting is dependent on user base size

Fingerprinting examples

- QUIC
  - Timestamps in ACK, NONC
  - Local link property inference?
  - Congestion control properties/behavior?
- HTTP/2
  - Couldn't find anything other than browser version fingerprinting issues (which we ignore)..
  - (TCP fingerprinting out of scope because Tor terminates TCP)

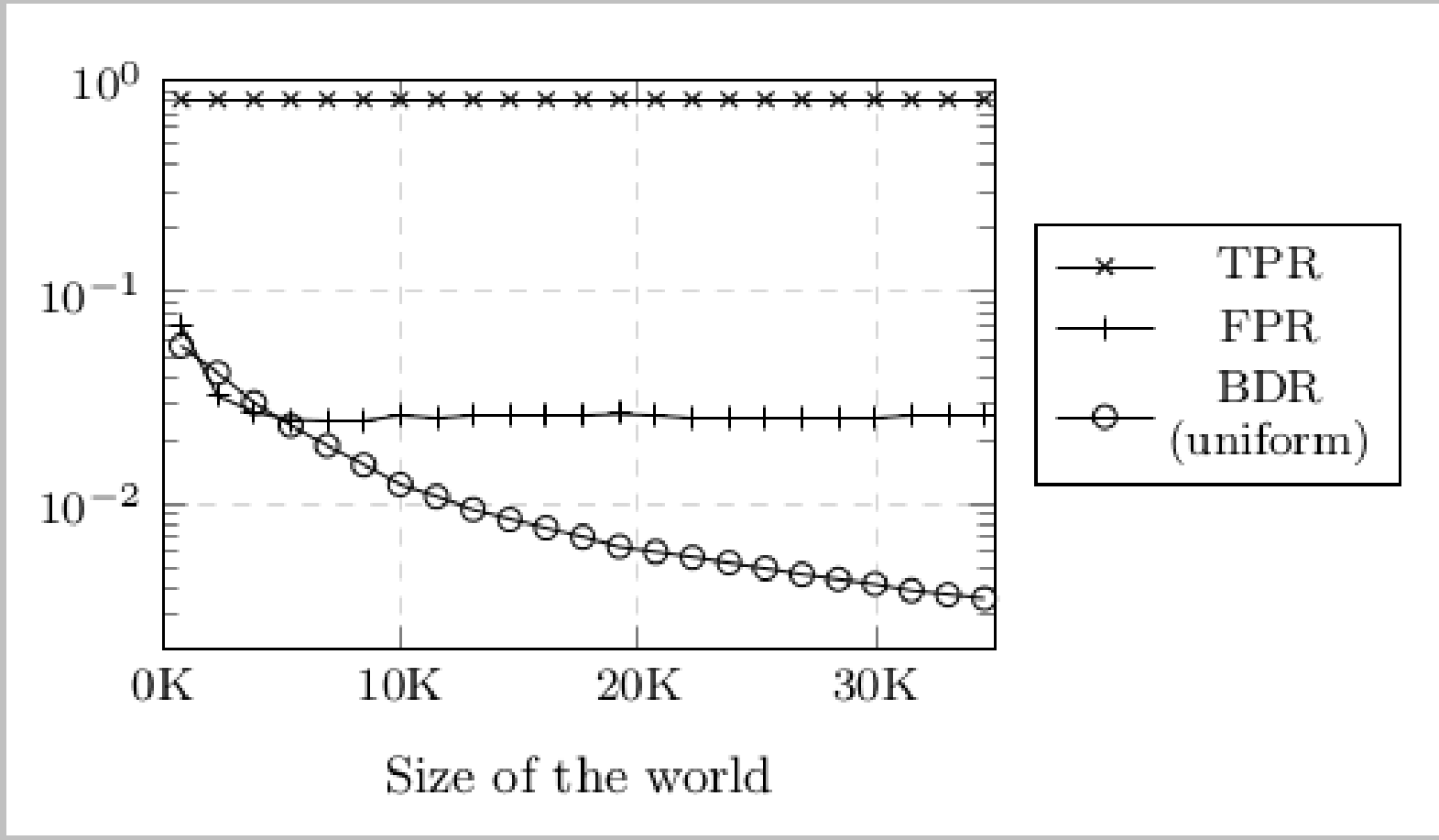
Traffic Analysis

- Confirmation and Correlation (aka end-to-end)
- VBR audio fingerprinting
  - ~256bits of padding mitigates many cases
  - CBR is a sure-shot (but not WebRTC default!)
- Website Traffic Fingerprinting
  - TLS: 'Side-Channel Leaks in Web Applications'
    - Padding ~256bytes mitigates many cases
  - Very sensitive to base rate: More pages → less accuracy and less padding
  - Tor's 512 byte cell size helps

Evaluating Attacks and Defenses

- Effectiveness is a function of the “World Size”
  - Base Rate Fallacy and VC Dimension
- Closed vs Open World
  - Truly closed worlds may not exist
  - Browser cache, AJAX, changing content...
- Valid metrics:
  - Bayesian Detection Rate (aka Precision)
  - Receiver Operating Characteristic AUC
  - P-ROC AUC (sensitive to world-size)
  - Interclass and Intraclass variance

Effects of the Base Rate Fallacy



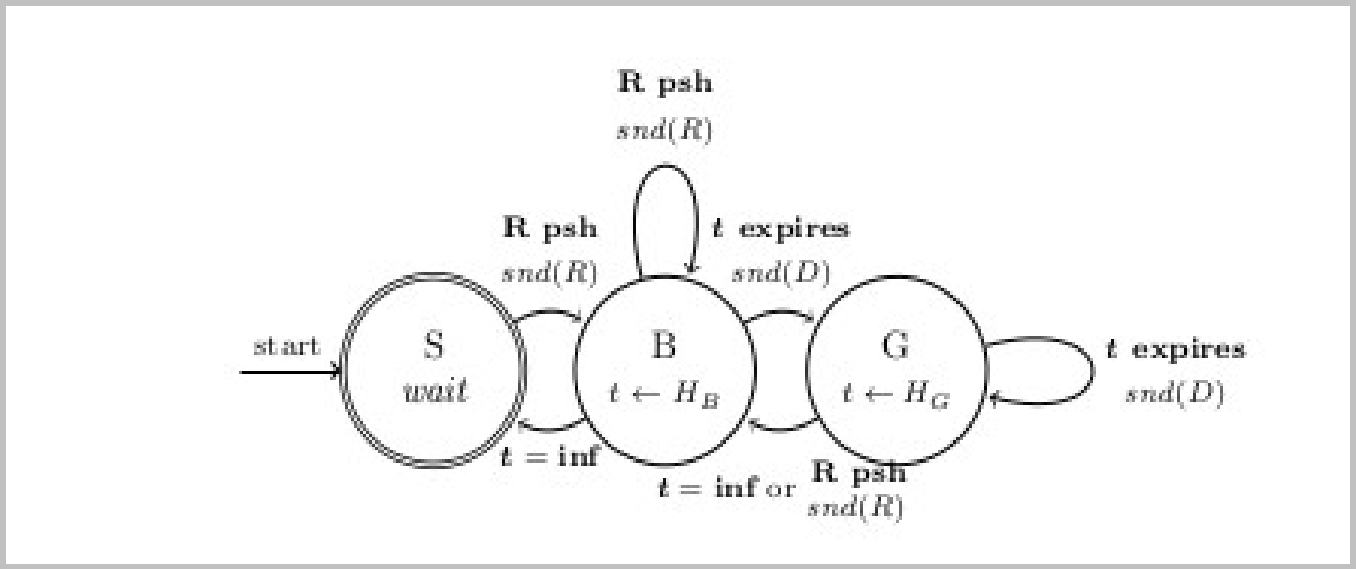
https://www.eecs.berkeley.edu/~sa499/papers/ccs-webfp-final.pdf

Defenses Tor Has Considered

- Pipeline Randomization
- HTTPOS
- Traffic Morphing
- Tamaraw
- Walkie-Talkie
- CS-BuFLO
- ALPaCA
- Adaptive Padding

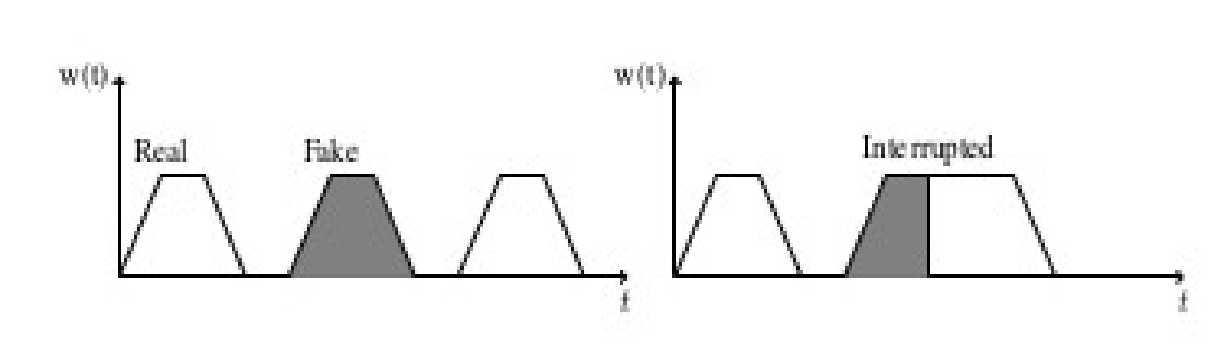
Adaptive Padding State Machines

- Two two-state state machines on each endpoint (one per direction)
- One state specifies histograms for sending padding after non-padding, the other specifies probability of sending successive padding.



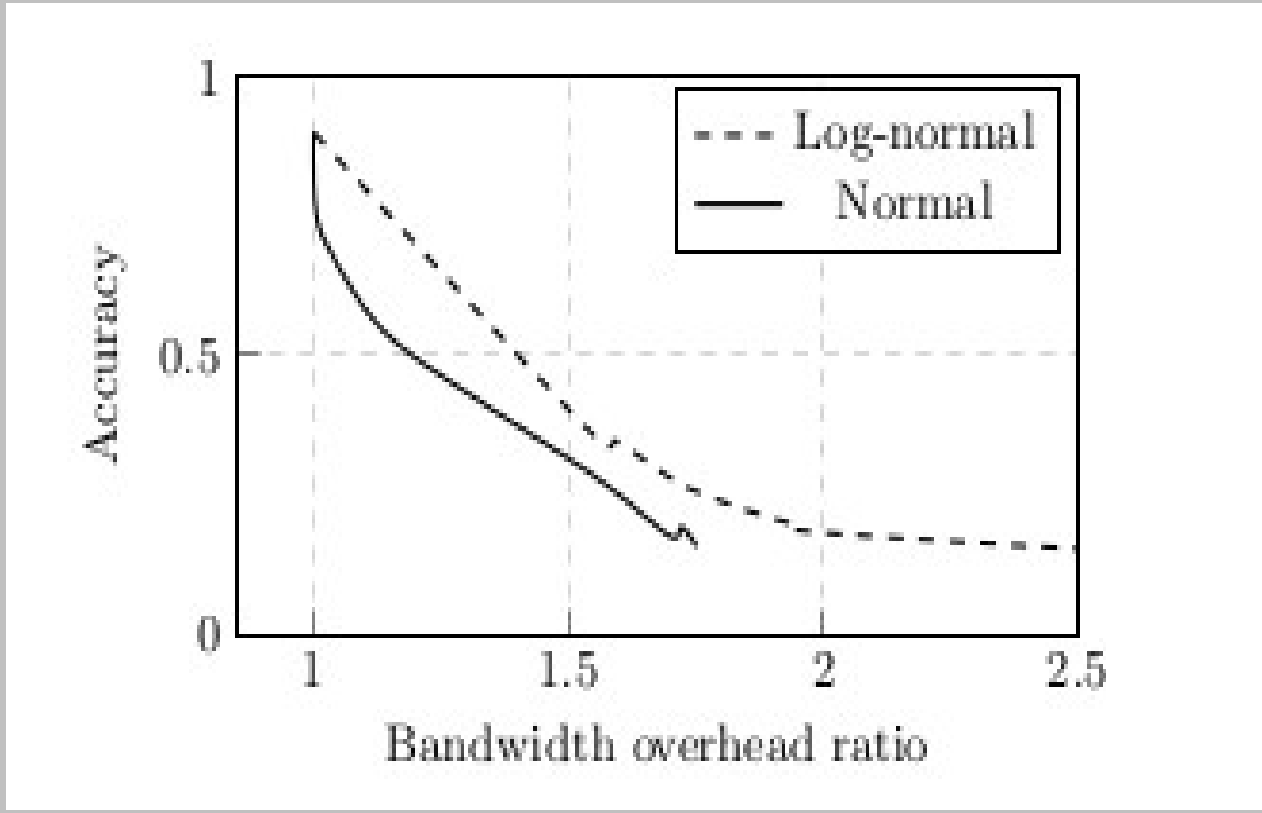
Adaptive Padding Token Removal

- Tokens are removed when either padding or non-padding is sent
  - Shapes traffic towards target distribution w/ minimal overhead



Adaptive Padding Overhead

- 0-60% overhead (tunable). ***No latency cost.***
  - Tradeoff “sweet spots” at ~5% and 25%



Citations and Related Work

http://www.cs.unc.edu/~fabian/papers/tissec2010.pdf

https://www.eecs.berkeley.edu/~sa499/papers/ccs-webfp-final.pdf

https://arxiv.org/pdf/1512.00524.pdf

https://www.petsymposium.org/2014/papers/Miller.pdf

https://www.freehaven.net/anonbib/cache/morphing09.pdf

https://security.cs.georgetown.edu/~msherr/papers/muffler.pdf

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/WebAppSideChannel-final.pdf

http://www0.cs.ucl.ac.uk/staff/G.Danezis/papers/k-fingerprinting.pdf

https://www.petsymposium.org/2017/papers/issue2/paper54-2017-2-source.pdf

Thanks

Mike Perry <mikeperry@torproject.org>

C963 C21D 6356 4E2B 10BB 335B 2984 6B3C 6836 86CC

https://www.torproject.org/projects/torbrowser/design/