A potential technique to deanonymise users of the TOR network

[redacted]

OPC-MCR, GCHQ

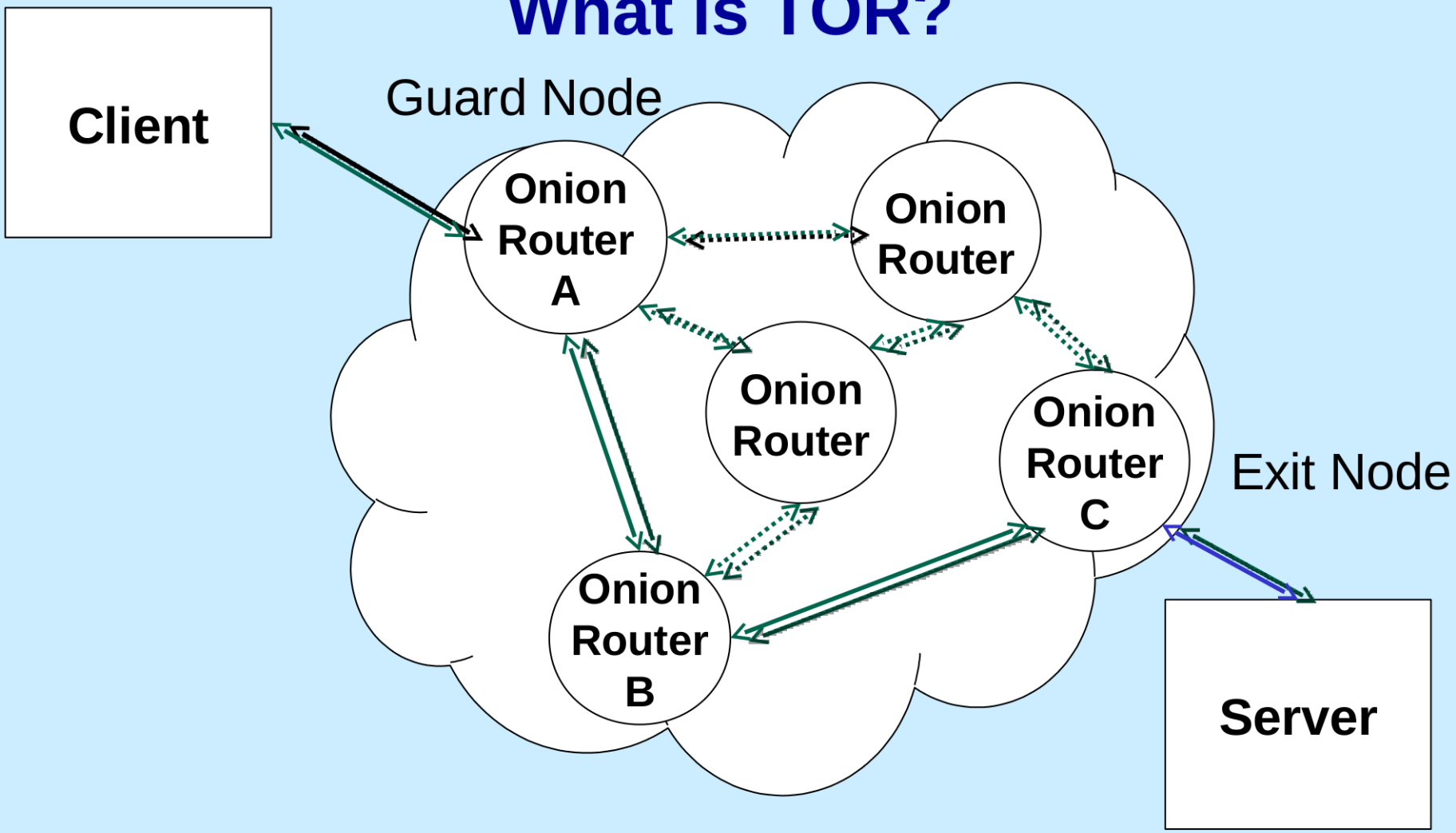
Outline

- TOR and the need for deanonymisation
- Data transformation
- Scoring
- Results
- Current status
- Software

What is TOR?

- "The Onion Router"
- Hides source of traffic by passing encrypted versions of your internet traffic between multiple TOR routers
- Notation:
  - "Client" – the initiator of communication
  - "Guard node" – the TOR router the client contacts
  - "Exit node" – the TOR router that relays your traffic to the final destination (with no extra encryption so this link can be exploited by SIGINT system)

What is TOR?



Who uses TOR?

- TOR was created by the US government and is now maintained by the Electronic Frontier Foundation (EFF)
- EFF will tell you there are many pseudo-legitimate uses for TOR
- We're interested as bad people use TOR, in particular:
  - Terrorists
  - Paedophiles

Aim

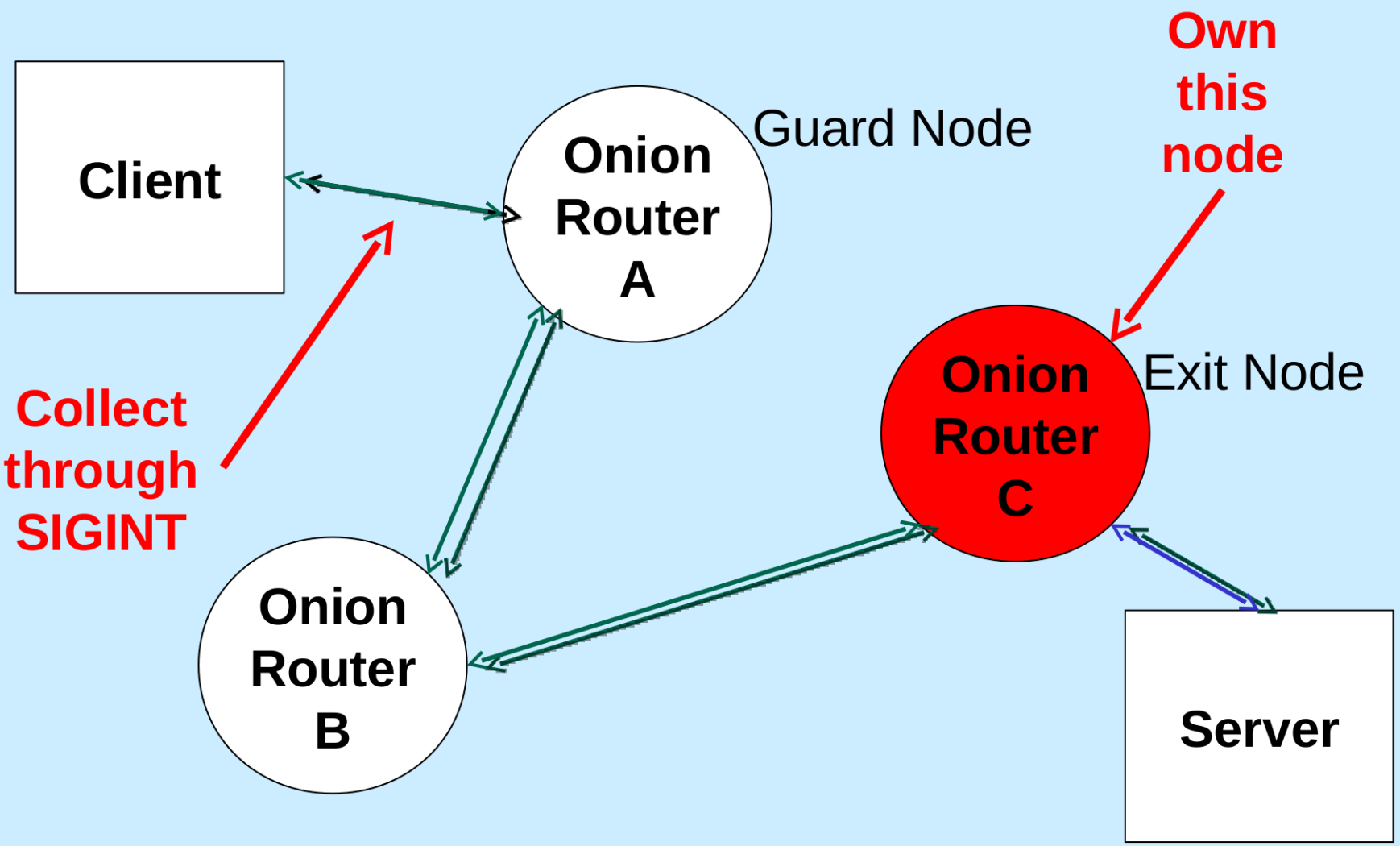
- **Find client IP address associated with TOR exit node traffic**
- Attack based on externals – specifically packet timings
  - Strong crypt is being used

Aim

- We'll make our task easier by assuming we own the exit node being used
  - Allows us to see all the traffic associated with a TOR circuit
  - Demultiplex traffic by (unknown) user

Side note: Circuit tracing

- One suggestion was to track packets through each hop in the TOR network
- We experimented with spotting all links in circuits created by GCHQ
- Visibility was too low to be a sensible approach
  - 13 out of 8294 potential inter-TOR-router links were seen
- We will directly correlate:
  - exit node traffic, and
  - traffic between client and guard node

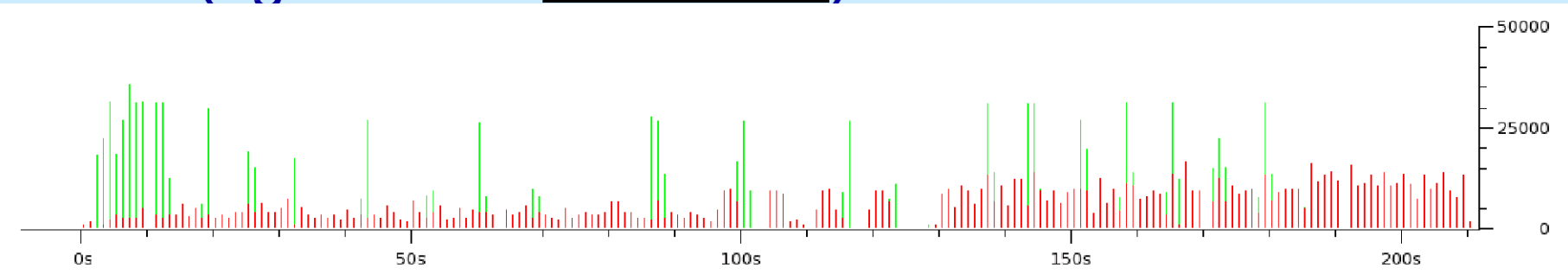


Test data collection

- Used the standard “TOR button” web-browser package to access TOR
- Made minor changes to ensure we could collect exit node traffic
  1. “News”: Search for news, visit news websites
  2. “TOR”: Browse the TOR website and then use a privacy checking website
     - Split into 2a and 2b as TOR changed circuit mid-way through
  1. “Download”: visit to SlashDot followed by downloading a large PDF file.
  2. “Forum”: Search on Google followed by browsing a PC technical help forum.

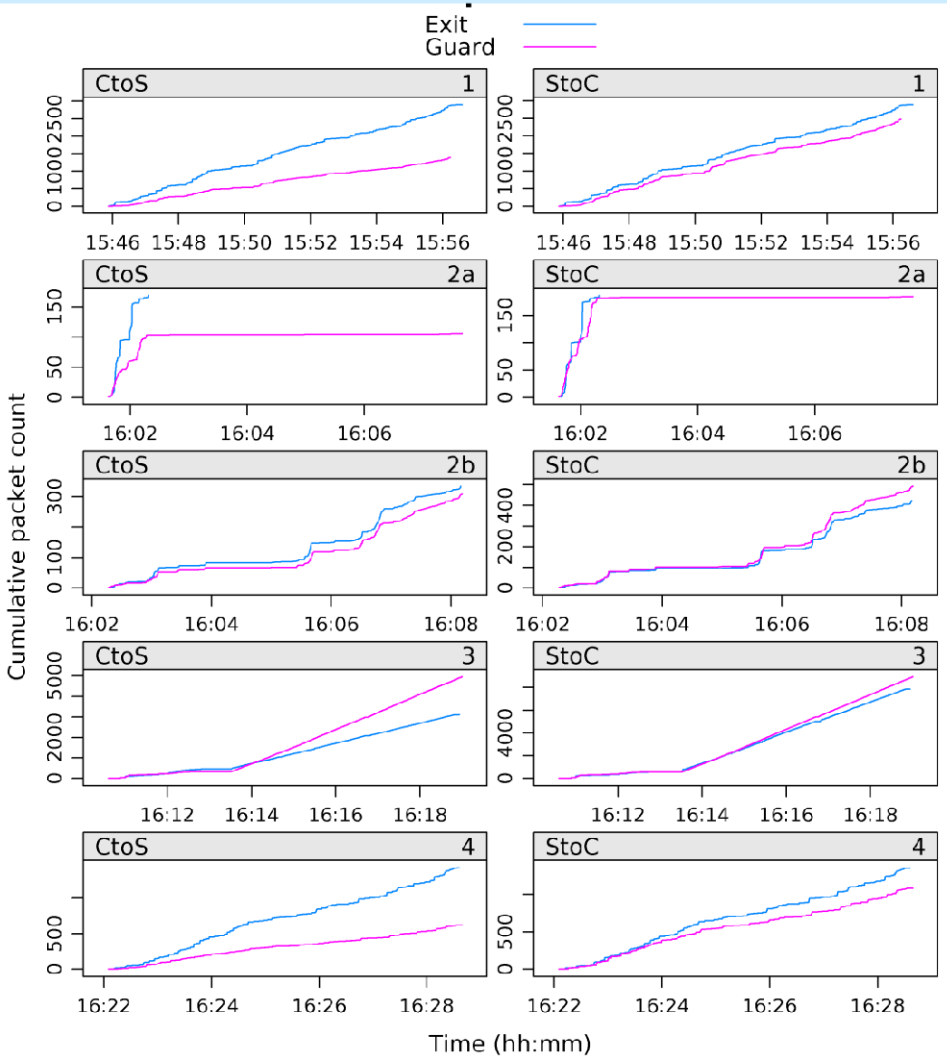
Flattening of timing patterns

- [redacted] (ICTR-NE) observed that TOR can flatten out timing patterns
- TOR uses a rate-limiting store-and-forward procedure at each TOR router
- Graph shows bytes of exit node traffic in green and client traffic in red whilst downloading a 1MB file (figure from [redacted])



Cumulative packet counts

- Our new insight is to use cumulative packet counts
- Hope packets are approximately preserved
  - Approximate as TOR repacketises data
- See strong correlation



Scoring: basic idea

- An idea of [redacted]
  - Bin time into intervals
  - For each interval get a pair (E_i, G_i)
    - Cumulative exit node packets upto time i
    - Cumulative guard node packets upto time i
  - Measure the correlation between these pairs
- We use 1s time-windows
  - Easy for the SIGINT system
  - Seems to work

Scoring: refinements

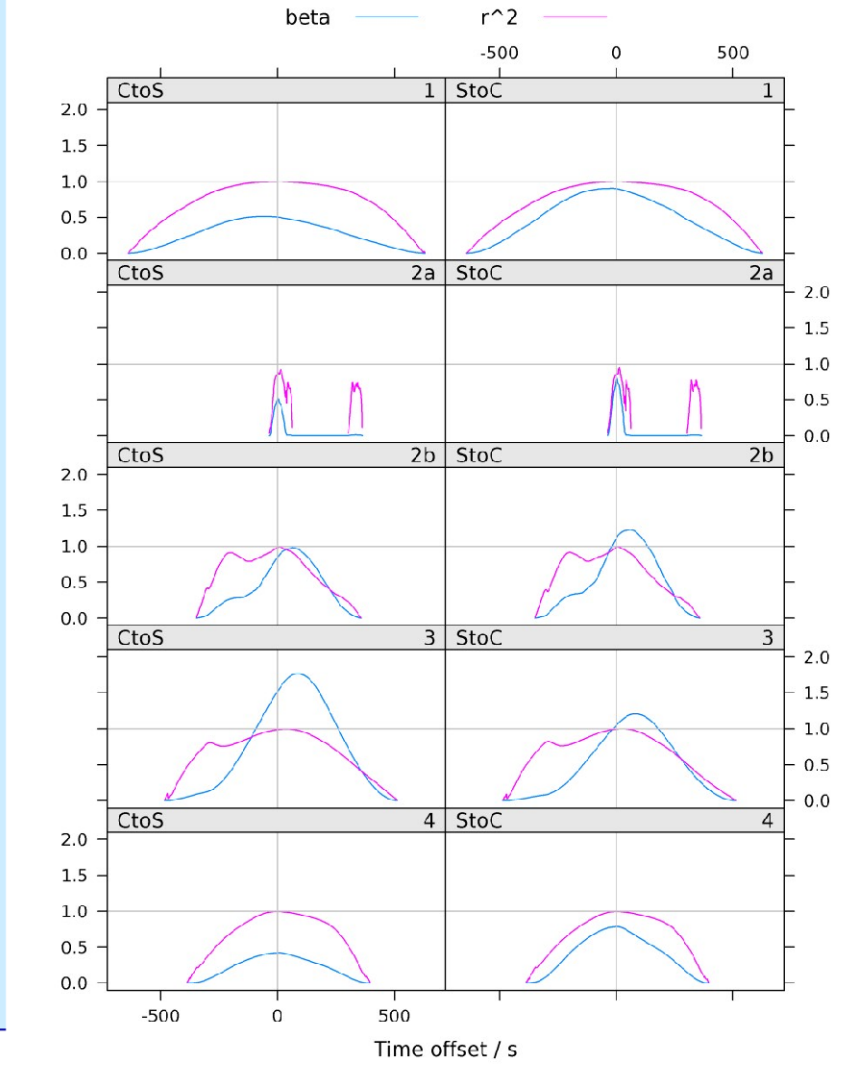
- We also expect counts to be similar
  - Fit a linear model
    - $G_i = \alpha + \beta E_i$
  - Only accept sessions where $\frac{1}{2} < \beta < 2$

Scoring: refinements

- There may be an unknown time-offset
  - Traffic takes time to relay through the TOR network
  - SIGINT clocks may not be synchronised
  - We slide the traces against each other and find the best match
  - Truncate to exit node trace (we know that it is a complete TOR circuit)

Self-comparison

- We show how the score behaviour as a function of time slide
- See high correlation (pink) at small time offset
- Also generally see β (blue) in a sensible range

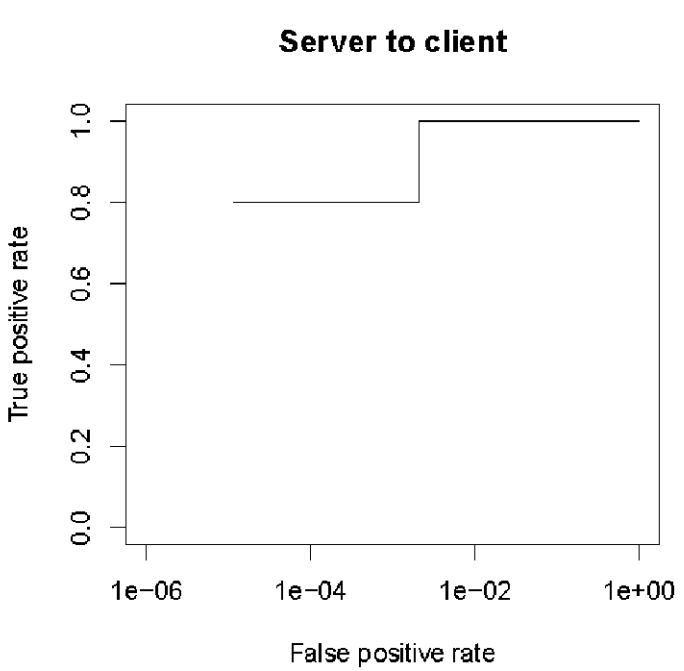


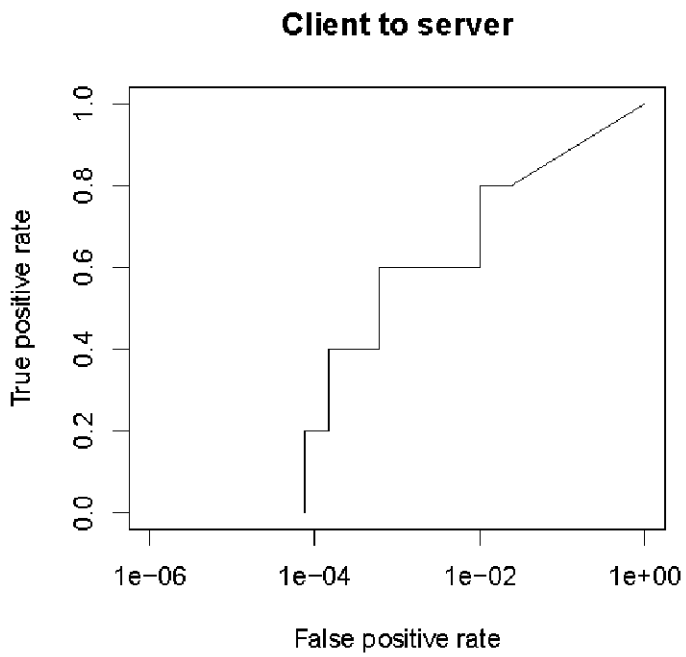
False positives

- Want an algorithm with very low false positive rate
- Used 2 hours of (timestamp, source IP, destination IP) tuples captured from 4 10G internet bearers
- Filtered to tuples between a guard node and a non-TOR node
- Allow time to arbitrarily slide +/- 2 hours
  - In real redeployment one would restrict this slide
- Allow us to plot ROC curves for the technique

False positives

- Linear – log ROC curve plot
- Server-to-client good
  - We miss the very short “2a” session with no false-positives
  - Threshold r^2=0.998
    - High as comparing increasing functions
- Client-to-server direction – many false positives
  - There’s less structure in data as less data flows in this direction when web-browsing





A larger experiment

- We want to find some false hits to understand worst case accuracy for the server-to-client direction
- Let's open the aperture very wide
  - 2027 bearer hours of logs with any time slide
  - Filter to all traffic involving a TOR node
    - Not just likely guard-to-client traffic as before
- We find some false hits (540) but rate is assessed to be low enough.
- 92% of false hits are against the big download session which has little structure

The next step

- We are collecting the required logs of packet times with TOR guard nodes in SIGINT
- GTE / JTRIG have adapted some TOR exit nodes we own to collect the required exit node data
  - We are keen to engage with others with exit nodes too
- Then run the attack
  - Expect to basically work
  - Some extra work might be required to only allow queries on sessions with enough structure
    - Need the bulk data first to progress this question

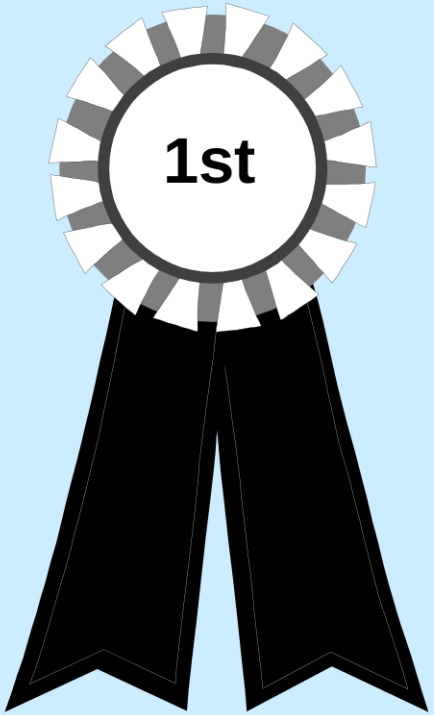
R package

- An R package can be downloaded from [redacted]
- Includes algorithm and the collected web-browsing data
- Recommend R packages for sharing analytics, can contain:
  - R / C / Fortran code
  - Example data
  - Runnable examples and documentation
  - Unit tests

Conclusion

- Have shown a potential externals-based deanonymisation attack for TOR
  - Requires SIGINT collection of guard-to-client packet times
  - Requires TOR collection from exit nodes we own
- Hope to get this running live at GCHQ soon

- Full paper and software available from [redacted]



Questions?

- Work by [redacted]
- [redacted]