GUARD SETS FOR ONION ROUTING

Jamie Hayes - joint work with George Danezis

University College London
j.hayes@cs.ucl.ac.uk

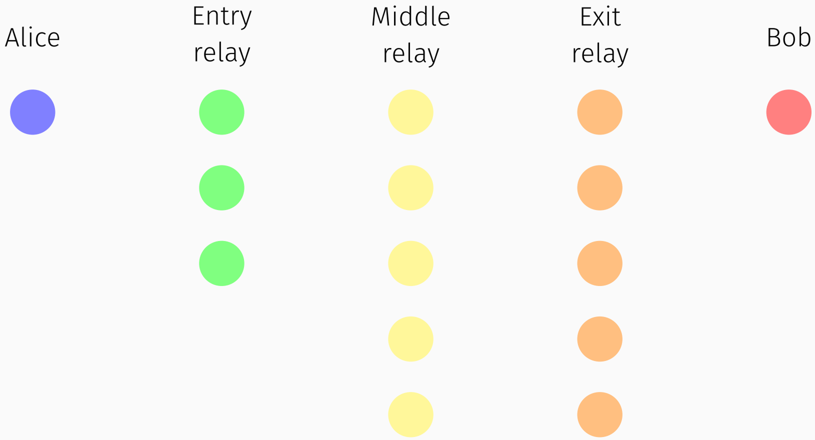
WHY DOES TOR EXIST?

- Encryption conceals the data - not the metadata.
- Tor attempts to hide this metadata by obscuring communication patterns by sending traffic through Tor relays.
- Low latency - trade off between usability and security.

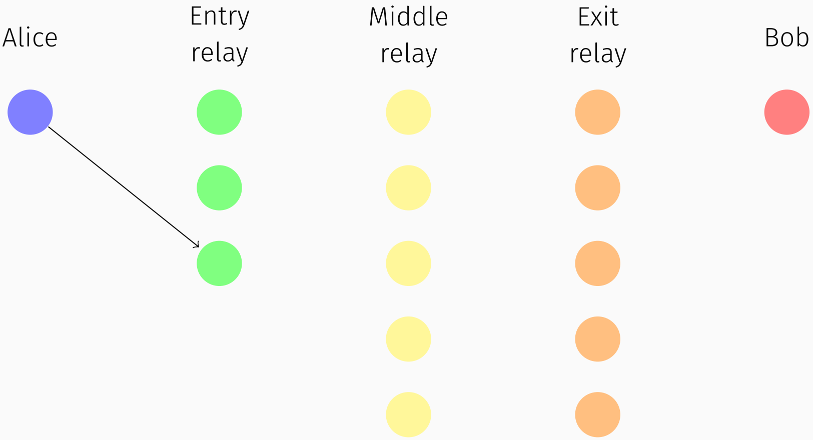
WHAT IS TOR?

- Thousands of volunteers relays contributing to the network.
- Developed in mid 2000’s - estimated 2,000,000 daily users.
- Used for many different reasons - whistle blowers, journalists, activists, military.

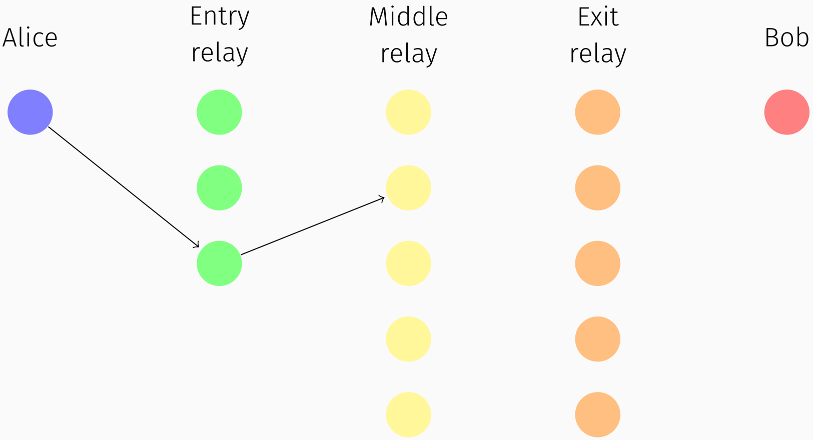
HOW DOES TOR WORK?



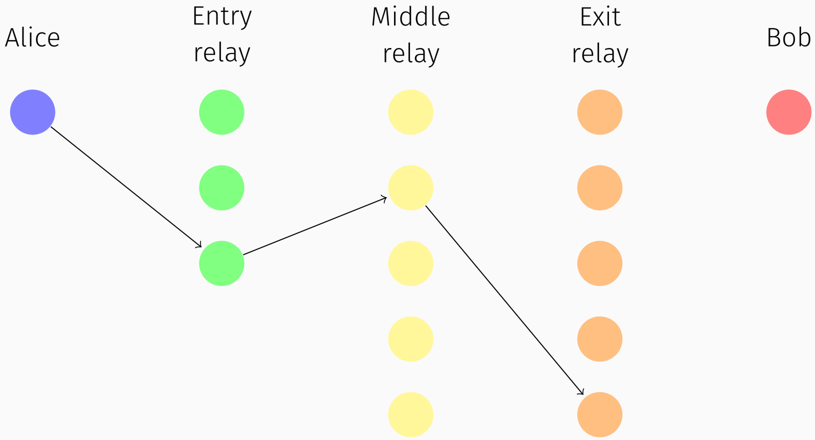
HOW DOES TOR WORK?



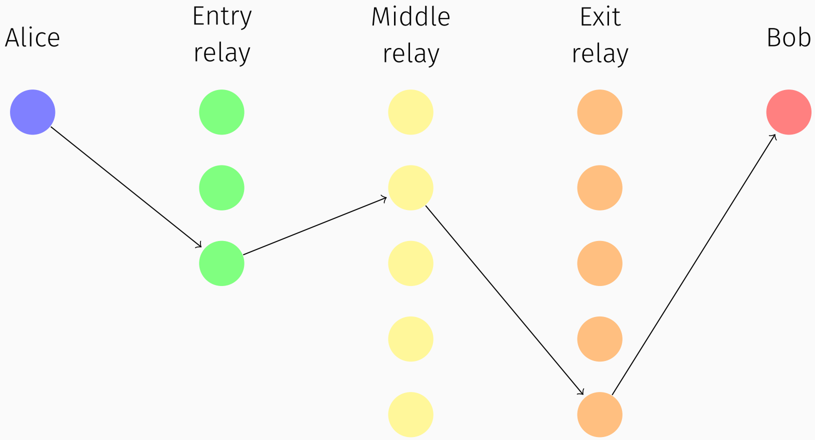
HOW DOES TOR WORK?



HOW DOES TOR WORK?



HOW DOES TOR WORK?



PREDECESSOR ATTACK

- Connecting to the majority of the network in a short amount of time is bad.

Example

- Adversary controls 50 out of 1000 relays.
- Without permanent entry relay –> pick random entry and exit in circuit - probability of both being adversary controlled is 0.25%. After 100 separate connections probability of profiling is 25%.
- Probability that you have been profiled increases with each connection!

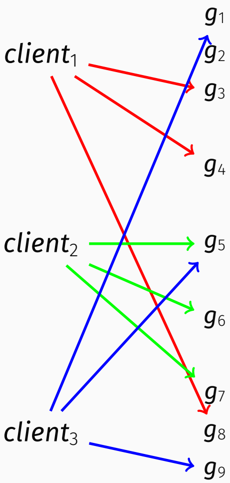
WHAT ARE ENTRY GUARDS?

- With permanent entry relay –> pick random entry (assume exit is adversary controlled). Chance of being profiled is 5%.
- Three stable relays with the guard flag that on startup Tor chooses for the client. Post 2015 three guards become one.
- Guards are used for 2-3 months (9 months for one guard).
- If a client has been unlucky and chosen an adversary guard they can ”escape” it - never rotating guards would lead to load imbalancing.
- Guard relays have a higher startup cost.

WEAKNESSES

- **Fingerprinting attack** - three guards uniquely identify a client. Less so with one guard - but still a problem.

- **Statistical disclosure attack** - even if the identity of the guards does not in itself uniquely determine the user, a bigger possible set of users is preferable to a smaller set of users.



WEAKNESSES

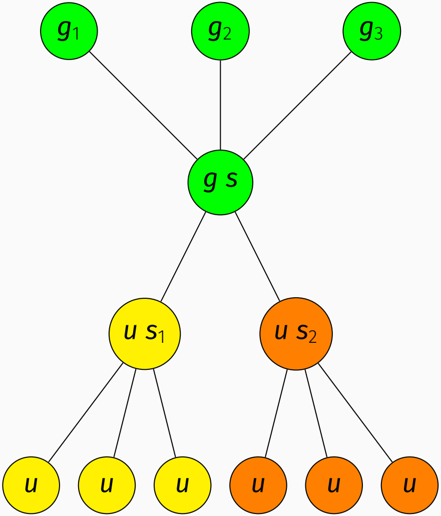
- Three guards - new guard relays underused. Bandwidth allocates a large fraction for use as a guard but only a few clients will rotate to it.
- Decreasing rotation period leads to more compromise but better spread of load.
- One guard better for load balancing - unused bandwidth used for middle and exit relay.
- But anonymity sizes of new guards still bad.
- Slow rate of rotation to new guards facilitates attacks!

WHAT WE WANT

- Instantly populate new guards - optimal spread of load.
- No churn.
- Remove possibility of unique guard history.
- Large sets of clients on guards.
- Easy in static environment but Tor is dynamic. A lot of clients and relays leave and join the network - maintaining load balance over time is difficult!

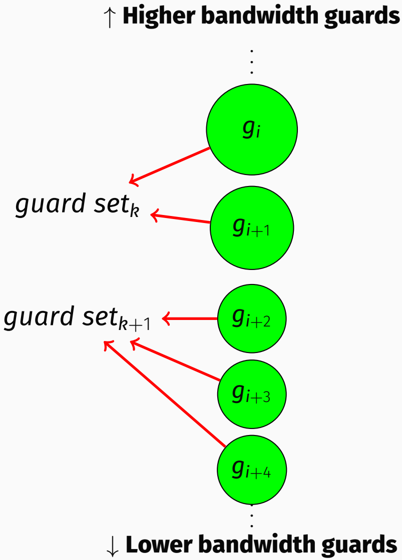
GUARD SETS

- Put guards and users in to sets.
- Better protection against fingerprinting and disclosure attacks.
- Improved reliability and security when single guards are temporarily unavailable - less churn.
- The provision of more, and more uniform, bandwidth to each client as compared with the single guard proposal.



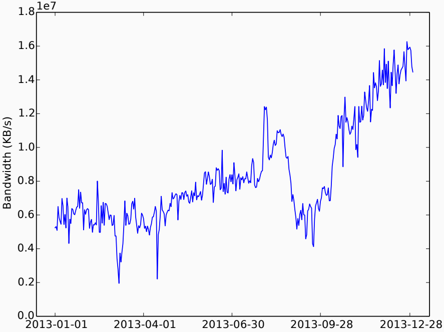
HOW ARE GUARD SETS FORMED?

- Initially list all relays with guard flags in descending order of bandwidth.
- Choose a threshold at which to create guard sets. We chose 40MB/s, and set a deletion threshold at 20MB/s.
- Cycle through list splitting guards in to guard sets, creating guard sets with equal bandwidth.

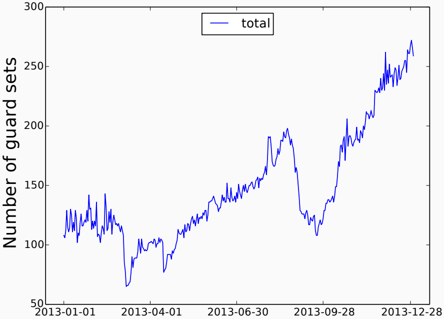


NUMBER OF GUARD SETS THROUGHOUT 2013

Rate of churn of total guard set bandwidth mirrors rate of churn of guard sets.



- Total guard set bandwidth



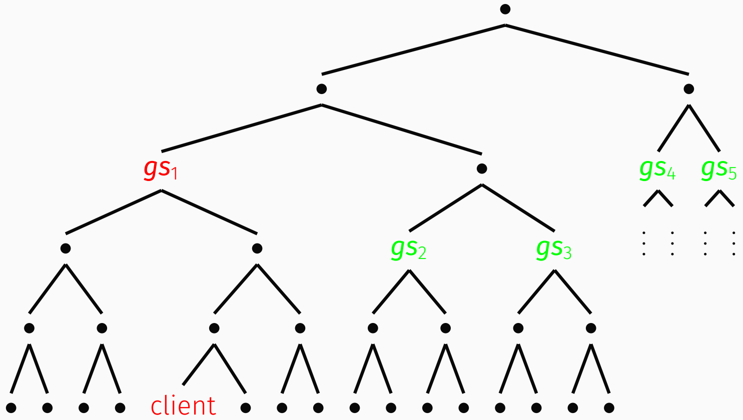
- Number of guard sets

HOW ARE ASSIGNMENTS MANAGED?

- Use a binary tree for assignments.
- Authority assigns guard set positions in the tree and manages guard - guard set assignments.
- Guard sets sit on an intermediate layer.
- Clients are assigned to a random leaf.
- Clients use the guard set associated with this leaf.

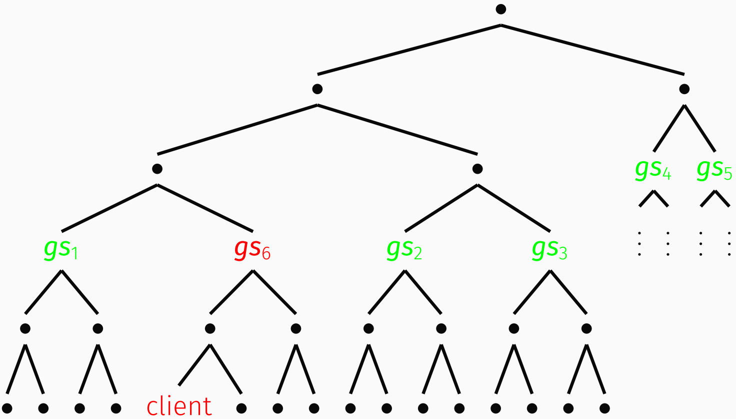
HOW TO ADD A GUARD SET TO THE TREE

Flip fair coin at each branch until we reach a guard set, then push down a layer



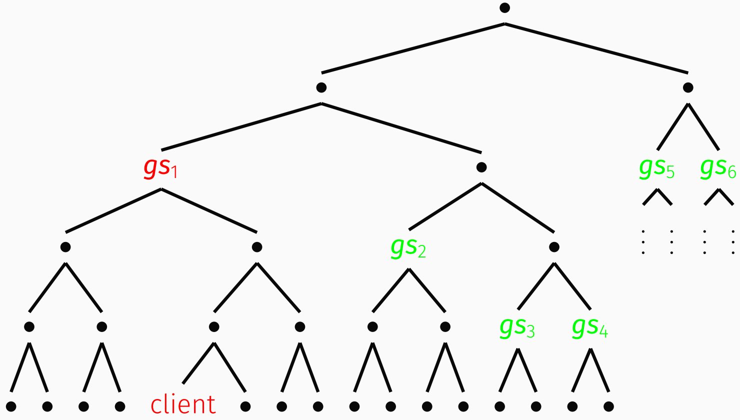
HOW TO ADD A GUARD SET TO THE TREE

Flip fair coin at each branch until we reach a guard set, then push down a layer



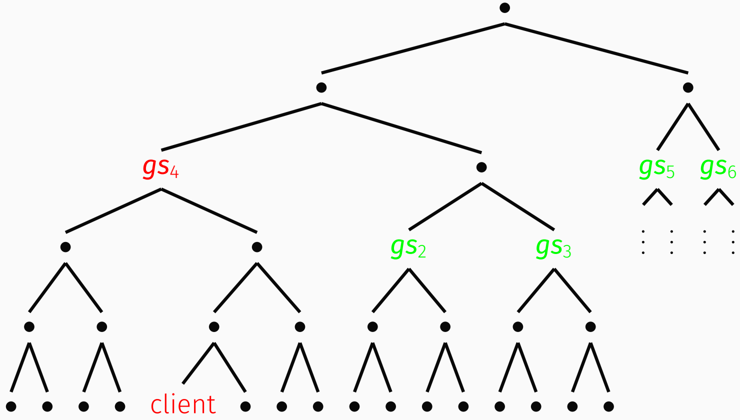
HOW TO REMOVE A GUARD SET FROM THE TREE

Choose the right most guard set (with a common ancestor) for replacement



HOW TO REMOVE A GUARD SET FROM THE TREE

Choose the right most guard set (with a common ancestor) for replacement



HOW TO COMPUTE GUARD SETS

How to compute and manage guard sets given a consensus document.

HOW TO COMPUTE GUARD SETS

How to compute and manage guard sets given a consensus document.

- retrieve guard set positions in tree.

HOW TO COMPUTE GUARD SETS

How to compute and manage guard sets given a consensus document.

- retrieve guard set positions in tree.
- update needy guard sets.

HOW TO COMPUTE GUARD SETS

How to compute and manage guard sets given a consensus document.

- retrieve guard set positions in tree.
- update needy guard sets.
- remove guard sets that are below deletion threshold.

HOW TO COMPUTE GUARD SETS

How to compute and manage guard sets given a consensus document.

- retrieve guard set positions in tree.
- update needy guard sets.
- remove guard sets that are below deletion threshold.
- create new guard sets from available bandwidth.

HOW TO COMPUTE GUARD SETS

How to compute and manage guard sets given a consensus document.

- retrieve guard set positions in tree.
- update needy guard sets.
- remove guard sets that are below deletion threshold.
- create new guard sets from available bandwidth.
- add new guard sets to tree.

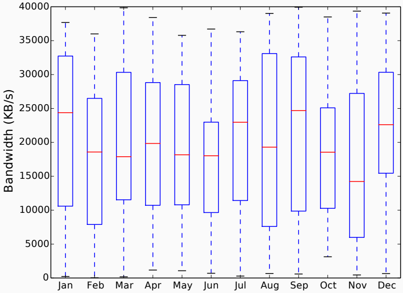
BINARY TREE REMAINS BALANCED

- Throughout 2013 difference in guard set layers didn't diverge.
- No large difference in number of clients of different guard sets. Anonymity sets are uniform.
- Load on guard sets is similar.

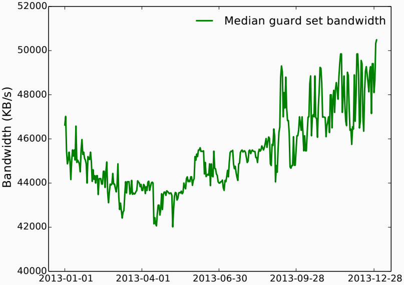
Example

Initialising on 1 January 2013 produces 108 guard sets, given 2.75 million users this creates 108 user sets of size 25463. By end of 2013 at worst there may exist some user sets of size 795, meaning there will always be at least 795 clients with the same guard history.

GUARD BANDWIDTH IS FULLY USED



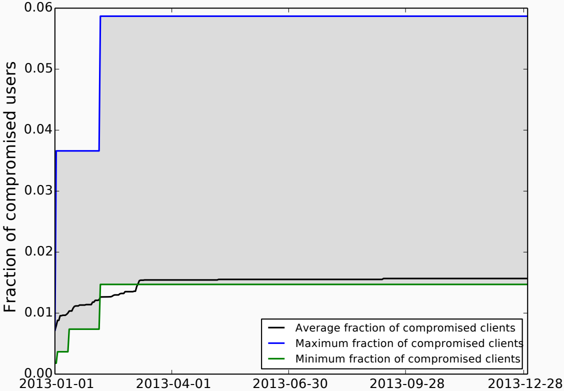
- Start of 2013 - 0.46%.
- End of 2013 - 0.031%.
- Biased towards smaller bandwidth relays, which do not make a large contribution to total bandwidth on the network.



- Guard set bandwidth remains steady over time.

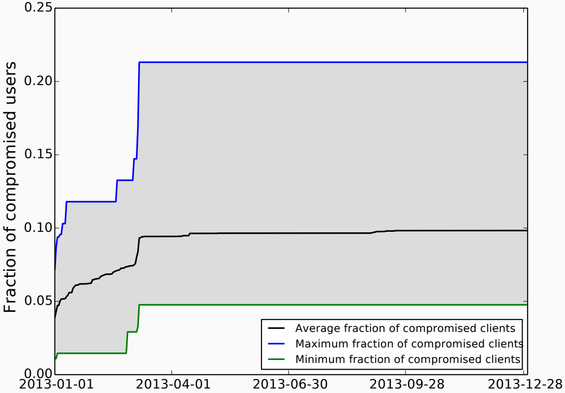
ADVERSARY - SINGLE GUARD

Fraction of compromised users given an adversary controlling one guard. With 1 million users and 100 repeats.



ADVERSARY - 1% GUARD BANDWIDTH

Fraction of compromised users given an adversary controlling 1% of guard bandwidth. With 1 million users and 100 repeats.



CONCLUSION

Guard sets provide a scalable solution for Tor.

CONCLUSION

Guard sets provide a scalable solution for Tor.

- **Stable** - Due to relay stability and the way we created guard sets, guard set deletion is a rare occurrence. Low churn limits the potential for predecessor attack. Shared history, even under failure, eliminates fingerprinting attack.

CONCLUSION

Guard sets provide a scalable solution for Tor.

- **Stable** - Due to relay stability and the way we created guard sets, guard set deletion is a rare occurrence. Low churn limits the potential for predecessor attack. Shared history, even under failure, eliminates fingerprinting attack.
- **Fair** - Clients can expect the same performance no matter their choice of guard set.

CONCLUSION

Guard sets provide a scalable solution for Tor.

- **Stable** - Due to relay stability and the way we created guard sets, guard set deletion is a rare occurrence. Low churn limits the potential for predecessor attack. Shared history, even under failure, eliminates fingerprinting attack.
- **Fair** - Clients can expect the same performance no matter their choice of guard set.
- **Large** - All guard sets serve roughly equal sized user sets, and a large number of users at any time. This prevents statistical attacks on the basis of discovering a user's guards.

THANKS!

Questions?

j.hayes@cs.ucl.ac.uk

@_jamiedh