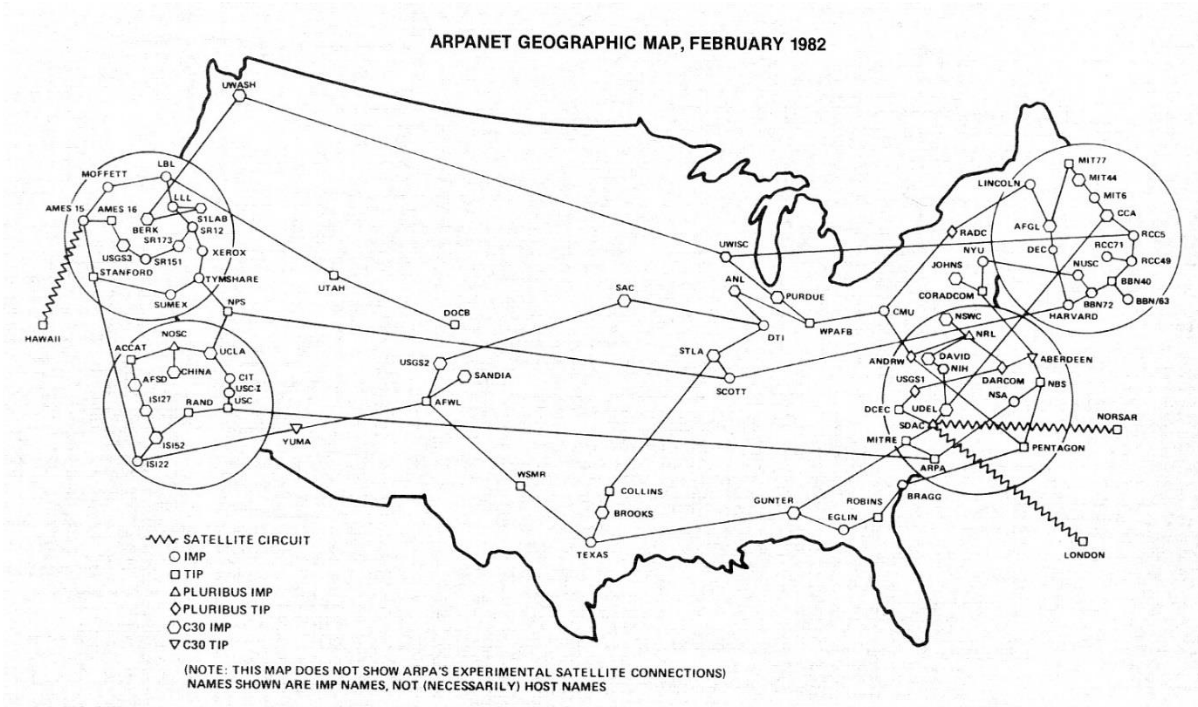
Foiling on-line surveillance: new developments in anonymous communications and their applications

Prof. George Danezis
University College London

http://danez.is
@gdanezis
g.danezis@ucl.ac.uk

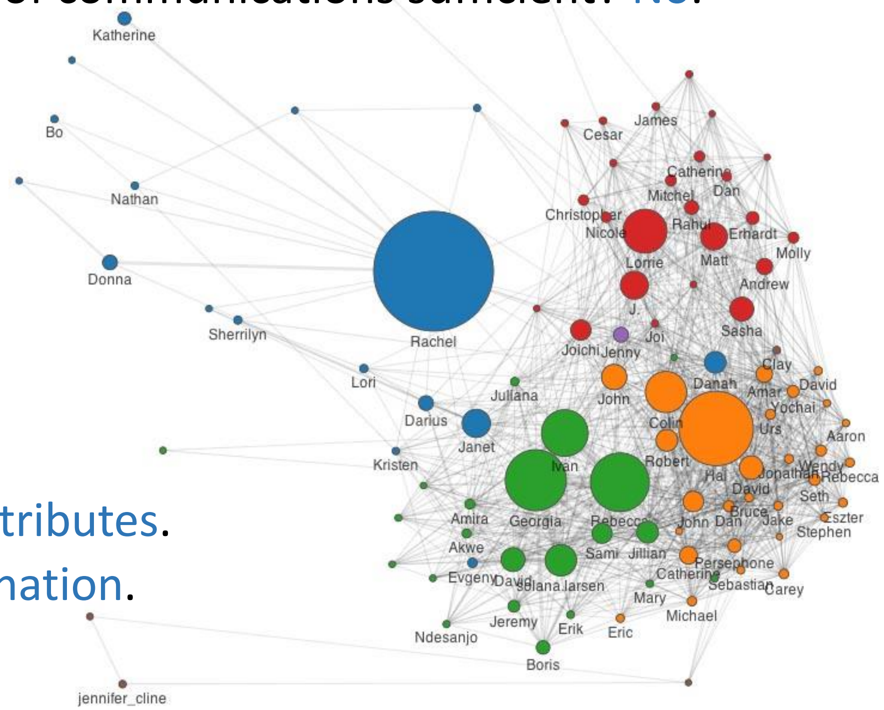
The Internet & Secure Channels



- ARPANET: only meant for unclassified information.
- Deliberate lack of communications security.
- 1990s additions: PGP / SMIME for email; SSL/TLS for TCP.
- Hide the content of communications between two parties.

Meta-data leakage

- Is Encryption protecting the content of communications sufficient? No.
- Meta-data is unprotected:
  - Who talks to whom?
  - How often?
  - At what times?
  - What volumes?
  - In which sequence?
  - What are the groups?
  - From which locations?
- Social Network Analysis.
- Machine Learning to learn private attributes.
- Profiling for price and other discrimination.



‘We Kill People Based on Metadata’

- Gen. Hayden (former director of the CIA & NSA)

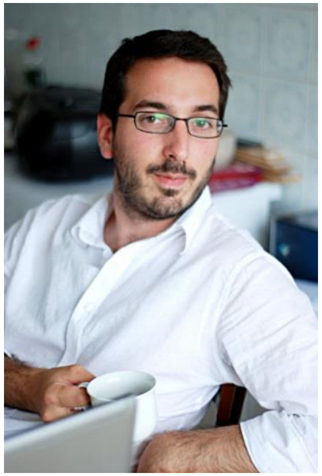
Social Graph: http://www.ethanzuckerman.com/blog/2013/07/03/me-and-my-metadata-thoughts-on-online-surveillance/

This talk: technologies that hide meta-data.

- Understand current trends in anonymous communications research
  - And where next?

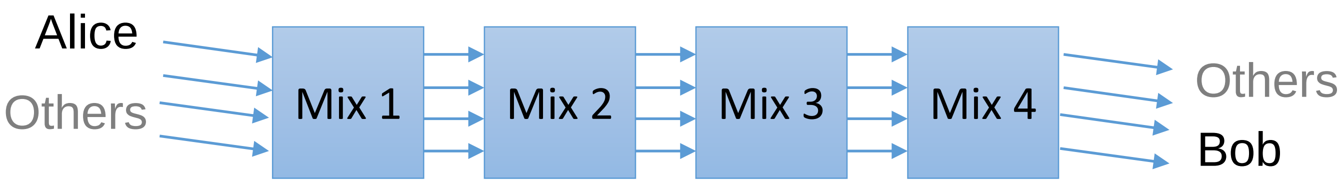
- Three key periods:
  - 79-96: From classic mixes to onion routing.
  - 97-10: The emergence and dominance of Tor.
  - 10-17: Modern mixing beyond Tor.

- A personal journey:
  - 00-04: PhD (Cambridge)
  - 04-09: postdoc (KUL & Microsoft)
  - 09-16: from junior researcher to professor (MSR & UCL).



The Classic mix & DC nets (1980s)

- David Chaum: 1979-1981 proposes mix networks.
- Cryptographic relays hiding input and output correspondence.



- How? Source routing, layered encryption & secret permutation.
- Features:
  (1) Server Anonymity
  (2) Anonymous replies
  (3) Receipts: for reliability.
  (4) Pseudonyms: for persistent communications.
- 1988 – Dinning Cryptographers: Anonymity from Multi-party computation

Chaum, David L. "Untraceable electronic mail, return addresses, and digital pseudonyms." *Communications of the ACM* 24.2 (1981): 84-90.
Chaum, David. "The dining cryptographers problem: Unconditional sender and recipient untraceability." *Journal of cryptology* 1.1 (1988): 65-75.

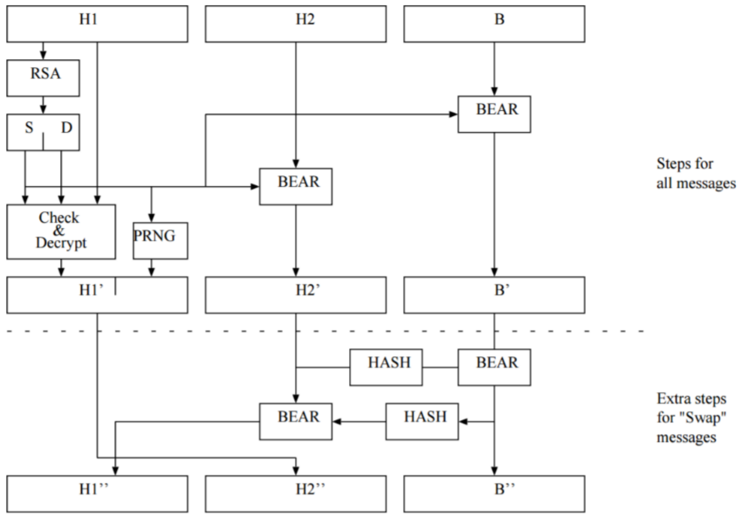
Provable Shuffles & Onion Routing (1990s)

- Provable shuffles for elections:
  - Killer app: casting ballots in electronic elections.
  - Prove that all votes are counted, none added or dropped.
  - Reliance on zero-knowledge proofs and heavy crypto.
  - Architecture: re-encryption, cascades, proofs.
- Mixing email:
  - Systems: Babel, cypherpunk remailer, mixmaster.
  - Architecture: Free route, decryption, mixing.
- Anonymizing streams with Onion Routing for web
  - Relays and layered encryption (like mixes)
  - No mixing, batching or delaying (unlike mixes).
  - Threat model: partial or local adversary.

Stephanie Bayer, Jens Groth: Efficient Zero-Knowledge Argument for Correctness of a Shuffle. EUROCRYPT 2012: 263-280
Ceki Gülcü, Gene Tsudik: Mixing Email with Babel. NDSS 1996: 2-16
Paul F. Syverson, David M. Goldschlag, Michael G. Reed: Anonymous Connections and Onion Routing. IEEE Symposium on Security and Privacy 1997: 44-54

Mixminion & Tor (2002-2004)

- Established designs mature.
- A type II remailer: mixminion (03)
  - "Modern" crypto for layered encryption (RSA-OAEP & AES)
  - Indistinguishable replies: necessary since fewer replies.
  - ~32 nodes acting as relays.
  - Latency: ~30min – 1h. Payload: 30kbytes.



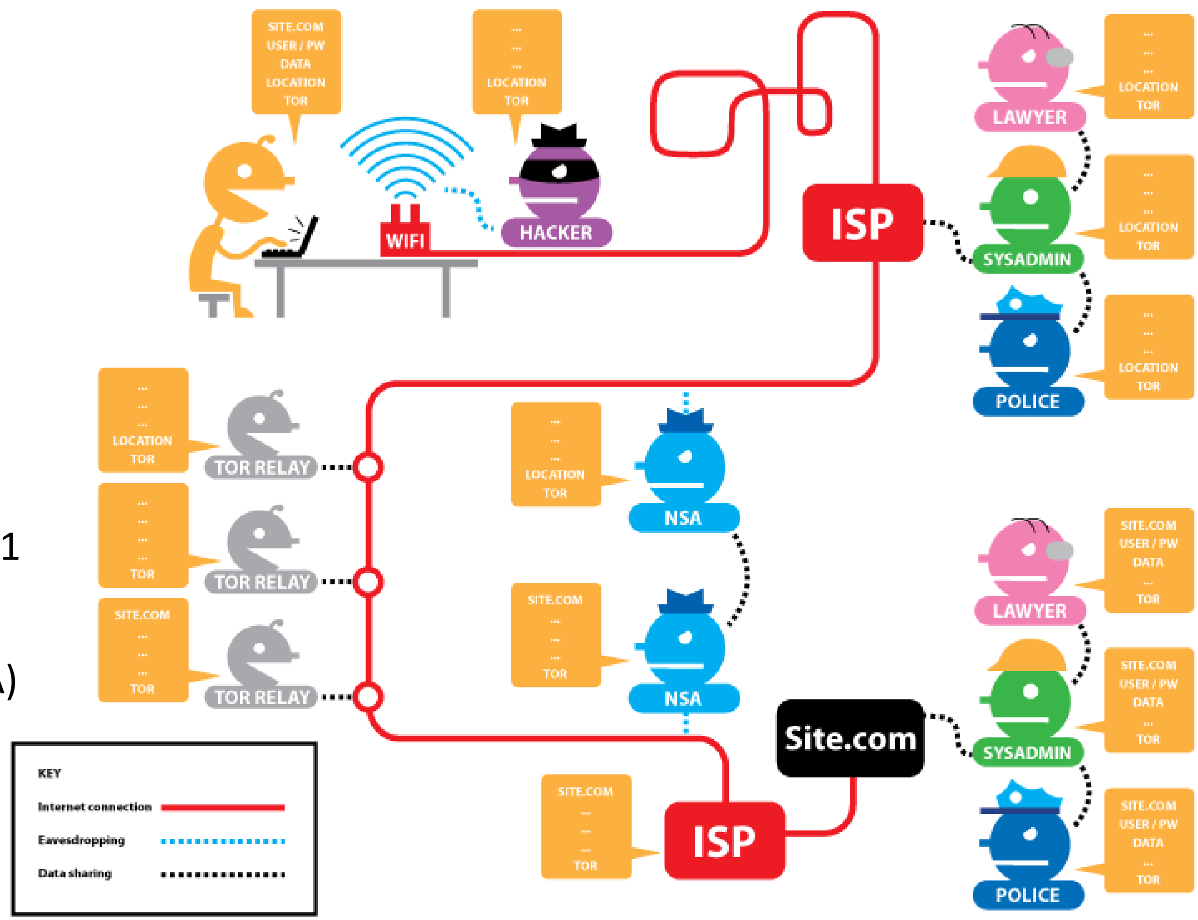
- The second gen. onion router: tor (04)
  - Sequential Ephemeral Diffie-Hellman.
  - All packers transit on the same route.

George Danezis, Roger Dingledine, Nick Mathewson: Mixminion: Design of a Type III Anonymous Remailer Protocol. IEEE Symposium on Security and Privacy 2003: 2-15
Roger Dingledine, Nick Mathewson, Paul F. Syverson: Tor: The Second-Generation Onion Router. USENIX Security Symposium 2004: 303-320

How tor works? (according to the EFF)

- Architecture:
  - Fixed guards
  - 3 relays
  - All cells travel on that path for 10 minutes.
  - No delay or cover traffic.
- Threat model:
  - Adv. Can only observe 1 location.
  - Note the confusion from the graphic! (NSA)

Widespread misunderstanding of the threat model.

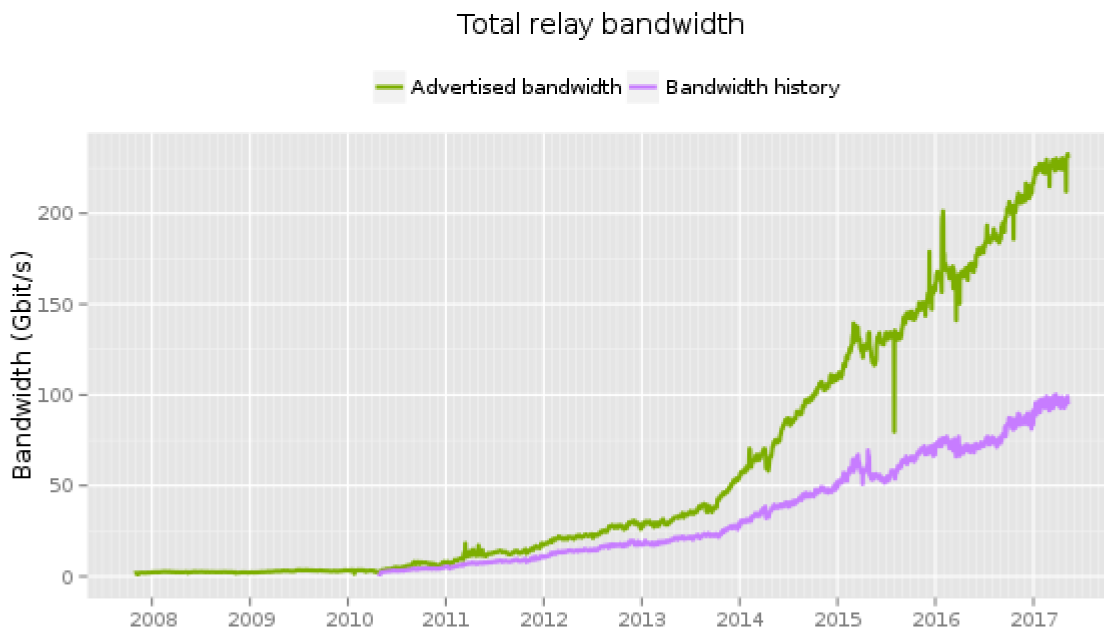


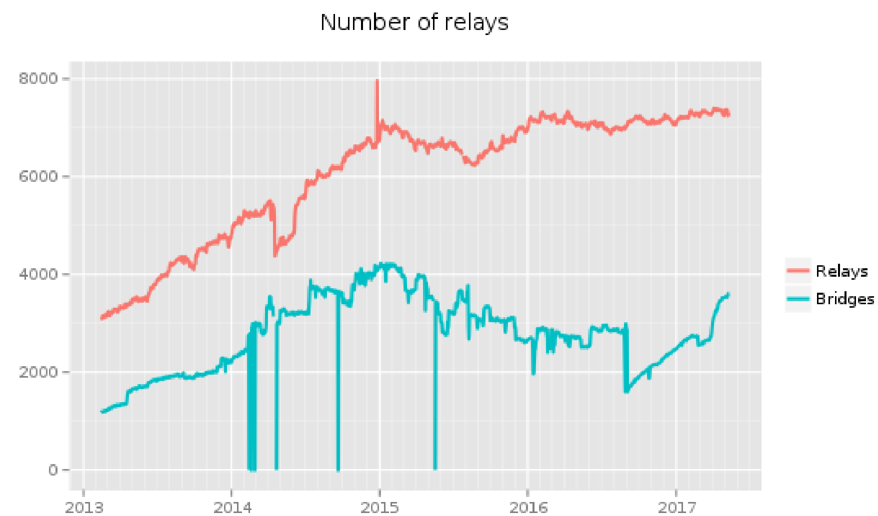
https://www.eff.org/files/tor-https-2.png

Tor wins!

- Today:
  - Over 7000 relays.
  - Over 200 Gbit/s.
  - About 2M users.
  - About 1 sec latency (median).
  - Tor Project $2M/y

- Hidden Services.
- Bridges.
- Hidden Transports.





The Tor Project - https://metrics.torproject.org/

Why tor won the 2000s-2010s?

- Killer app: the web & TCP abstraction.
  - SOCKS Proxy -> Tor Browser bundle.
  - Email, lists on the decline, plagued by abuse and spam.
  - Hidden (web) Services.

- Interactivity & Usability:
  - Low(er) RTT does not require complex error correction / repetition.
  - Use TCP as substrate – failed connections detected immediately.
  - Can use for email + IM too.
  - "Anonymity loves company!"

- Low latency & cost:
  - Pre-open circuits to minimize crypto overhead.
  - 1-10 seconds (tor) vs. 30-60 mins (mixminion)
  - How? Do not protect against global adversary.

Mix networks have problems: can mixes they really protect against GPA?

Mix problems: Latency

- The problem: Need to break the link between incoming and outgoing messages in a mix, to defeat a Global Passive Observer.

- Only 3 ways – all have a `systems cost’:
  - Delay messages to ensure many messages get `mixed together’. (latency)
  - Inject cover messages to hide path, senders or receivers. (goodput)
  - Drop messages, to hide their meta-data. (reliability)

- Traditional view: prefer delays (latency), since cover (bandwidth) is expensive (2000s!), and we do not know how to deal with drop (unreliability). Exception: ISDN mixes.

High latency was the **most severe mistaken direction** in mix research.

Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner: ISDN-MIXes: Untraceable Communication with Small Bandwidth Overhead. Kommunikation in Verteilten Systemen 1991: 451-463

Mix problems: systems reliability

- `Reliable' mix networks assume a synchronous network model.
- The internet is asynchronous.

- Problems mature mix networks have to handle:
  - Set one of more fixed size for traffic – minimizing waste.
  - Break large messages into chunks.
  - Ensure all chunks are received (acks? FEC?) or retransmit.
  - Ensure the rate of sending does not lead to congestion collapse.
  - Ensure flow control to not overwhelm receiver.

- All of this is harder given very long latencies!
  - Ack based protocols set timers for the Round Trip Time (RTT). Hard!
  - So not only messages were slow to arrive, but they may never arrive.

- Retransmissions eventually lead to de-anonymization!
  - Because of SDA, or corrupt paths …

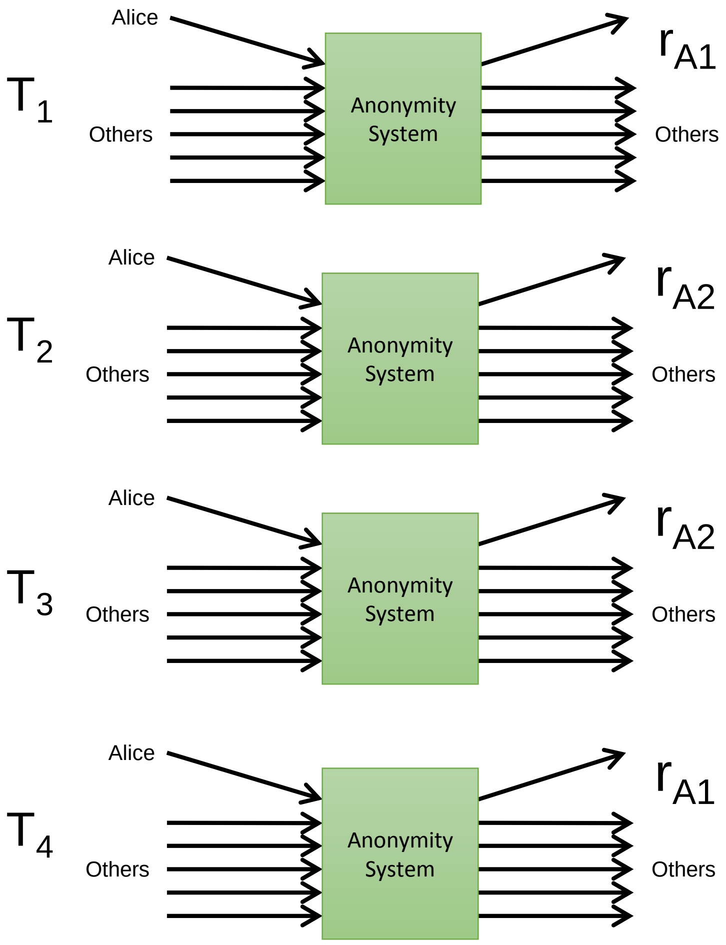
Nikita Borisov, **George Danezis**, Prateek Mittal, Parisa Tabriz: Denial of service or denial of security? ACM Conference on Computer and Communications Security 2007: 92-102

Mix problems: statistical & disclosure attacks

- Tor is not secure against the Global Passive Adversary.
  - Mix networks also not secure in the long term.

- Statistical Disclosure Attacks
  - Alice has few friends $\{r_{A1}, r_{A2}\}$
  - Any anonymity system that mixes together fewer than the whole universe of senders or receivers eventually leaks their relationship.
  - How? Estimate the probability of receiver given Alice sending.

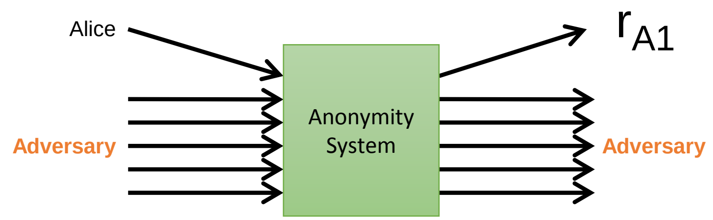
- Key Question: at what rate do anonymity systems leak?



Dakshi Agrawal, Dogan Kesdogan: Measuring Anonymity: The Disclosure Attack. IEEE Security & Privacy 1(6): 27-34 (2003)
George Danezis, Claudia Díaz, Carmela Troncoso: Two-Sided Statistical Disclosure Attack. Privacy Enhancing Technologies 2007: 30-44
George Danezis, Andrei Serjantov: Statistical Disclosure or Intersection Attacks on Anonymity Systems. Information Hiding 2004: 293-308

Mix problems: (n-1) attacks & Sybil attacks

- Mix networks could be totally insecure too!
- How do you know all other messages are from genuine people?
- 2 Attacks:
  - Sybil attacks: adversary pretends to be many senders.
  - (n-1) attacks: the adversary blocks a mix input to only receive a single genuine message.



- How to avoid those? Problematic options:
  - Authenticate users to ensure they are real and genuine.
  - Perform active measurements to detect blocking.
  - Drop messages if they are delayed.
  - Sybil detection based on social graphs.

George Danezis, Len Sassaman: Heartbeat traffic to counter (n-1) attacks: red-green-black mixes. WPES 2003: 89-93
George Danezis, Prateek Mittal: SybilInfer: Detecting Sybil Nodes using Social Networks. NDSS 2009

Mix problems: Epistemic attacks

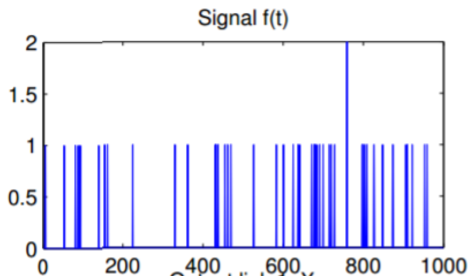
- How to scale up mix networks?
- Problem: all clients need to use the same information to construct paths through relays. Otherwise: attacks based on knowledge of the client (epistemic).
- Consider a user only known a random subset of mix nodes …
- If paths identify clients: then anonymity is not protected. (Leakage).
- Solutions:
  - Download the whole database of routers and routing information. (Bandwidth cost)
  - Privately download parts of it (Private Information Retrieval). (Computationally expensive.)

George Danezis, Paul F. Syverson: Bridging and Fingerprinting: Epistemic Attacks on Route Selection. Privacy Enhancing Technologies 2008: 151-166
George Danezis, Richard Clayton: Route Fingerprinting in Anonymous Communications. Peer-to-Peer Computing 2006: 69-72

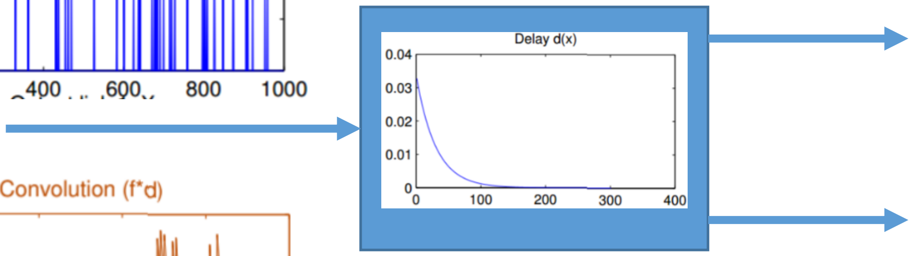
Onion routing & Tor also has problems…

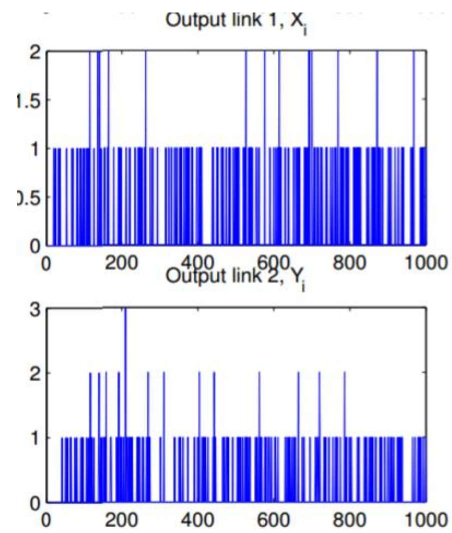
Tor problems: Stream Tracing attacks

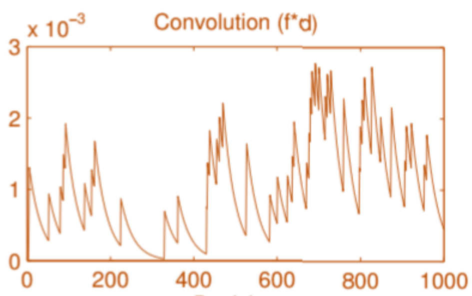
- An adversary can link two points of an anonymous circuit.
- How? Make a model template of output from input, and match.



Tor Router
with delay

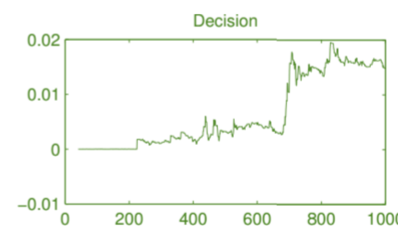






Template: distribution of outputs

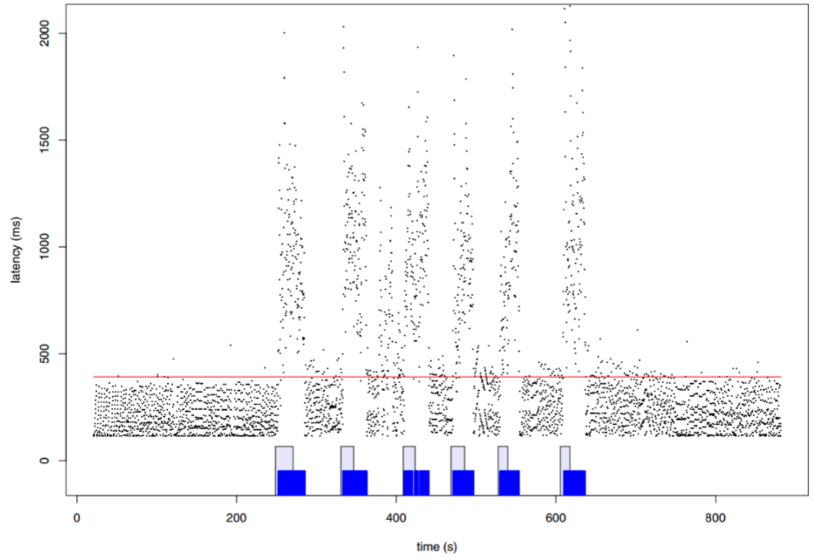
Decision:



George Danezis: The Traffic Analysis of Continuous-Time Mixes. Privacy Enhancing Technologies 2004: 35-50

Tor problems: Indirect load estimation

- Idea:
  - Loop of traffic will be processed on same queue as the target connection.
  - When the target connection has load on it, the delay will be greater.
  - We can use a tor circuit to measure the delay.



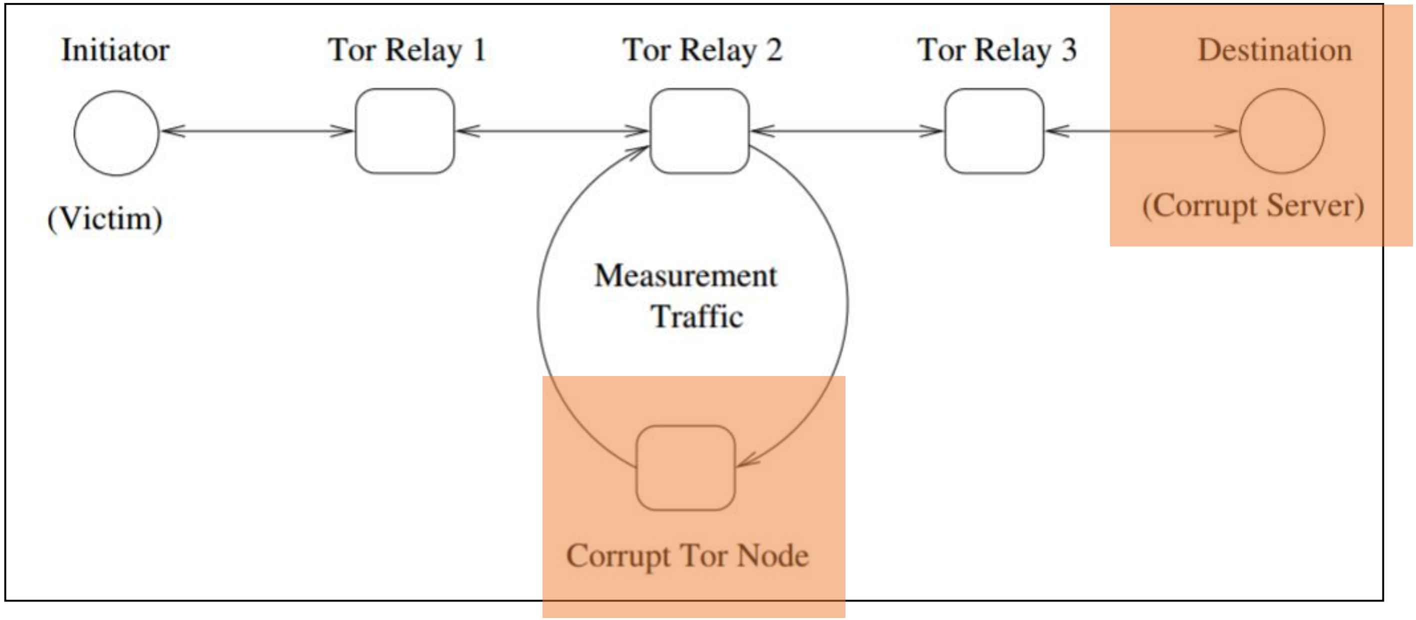
- Illustration:
  - X-asis: time.
  - Blue: injected patterns from server.
  - Dots: observed delay of loop traffic.

Morality: Observing a link does not mean observing everything, but observing anything.

Steven J. Murdoch, **George Danezis**: Low-Cost Traffic Analysis of Tor. IEEE Symposium on Security and Privacy 2005: 183-195

Tor problems: Indirect load estimation

- Global passive adversary is an abstraction.
- Real adversaries only need an estimate of traffic load.
- Possible indirect clogging attacks: inject pattern at corrupt server, and trace through indirect load estimation.



Steven J. Murdoch, **George Danezis**: Low-Cost Traffic Analysis of Tor. IEEE Symposium on Security and Privacy 2005: 183-195

Tor problems: website fingerprinting

- Tor does not significantly disrupt the timing, volume and dynamics of web browsing streams.
- Website fingerprinting uses machine learning to guess which web page is being loaded through tor.
- It works well, even against delaying, cover and other defences.

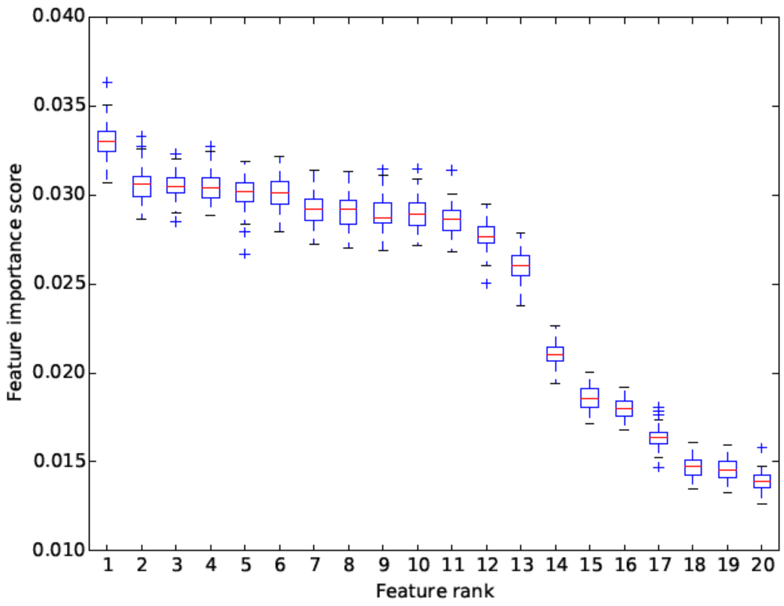
Defenses	This work	k-NN [39]	CUMUL [28]	Bandwidth overhead (%)
No defense	0.91 $\pm$0.01	0.91 $\pm$0.03	0.91 $\pm$0.04	0
Morphing [40]	**0.90** $\pm$0.03	0.82 $\pm$0.06	0.75 $\pm$0.07	50 $\pm$ 10
Decoy pages [27]	**0.37** $\pm$0.01	0.30 $\pm$0.06	0.21 $\pm$0.02	130 $\pm$ 20
Adaptive Padding [31]	**0.30** $\pm$0.04	0.19 $\pm$0.03	0.16 $\pm$0.03	54
BuFLO [12]	**0.21** $\pm$0.02	0.10 $\pm$0.03	0.08 $\pm$0.03	190 $\pm$ 20
Tamaraw [35]	**0.10** $\pm$0.01	0.09 $\pm$0.02	0.08 $\pm$0.03	96 $\pm$ 9

- Note: they also work great against TLS/SSL!

Jamie Hayes, **George Danezis**: k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. USENIX Security Symposium 2016: 1187-1203

What features allow fingerprinting?

Random forest classifier allows for feature importance analysis.



№	Feature Description
1.	Number of incoming packets.
2.	Number of outgoing packets as a fraction of the total number of packets.
3.	Number of incoming packets as a fraction of the total number of packets.
4.	Standard deviation of the outgoing packet ordering list.
5.	Number of outgoing packets.
6.	Sum of all items in the alternative concentration feature list.
7.	Average of the outgoing packet ordering list.
8.	Sum of incoming, outgoing and total number of packets.
9.	Sum of alternative number packets per second.
10.	Total number of packets.
11-18.	Packet concentration and ordering features list.
19.	The total number of incoming packets stats in first 30 packets.
20.	The total number of outgoing packets stats in first 30 packets.

And many more problems …

- Traffic analysis:
  - Sampling attacks
  - IX, AS sampling & BGP rerouting attacks
  - +Many mix attacks: DoS & epistemic attacks (do not matter because no GPA.)
- Tor is both too much and too little:
  - Too little: real adversaries can gain near GPA capabilities, or enough to break Tor. The Snowden revelations confirm this.
  - Too much: if it is trivial to link two points simpler design is possible:
    (1) No need for multiple layers of encryption.
    (2) A single hop security is all you get after a long time.

In conclusion: Tor is great if you want to hide from a relatively weak adversary. Not so great against more powerful adversaries.

Can the NSA / GCHQ break tor?

- Mixed evidence from Snowden Leaks and FBI successes:
  - GCHQ deck of slides on working group to tackle tor ("tor stinks" deck).
  - "Egotistical Giraffe/Goat" tools – exploits in tor bundle.
  - XKEYSCORE rules for extracting bridges and tracking downloads.
  - GCHQ paper on stream tracing.
- FBI is suspiciously successful at finding Hidden Services:
  - Success ascribed to op-sec failures – plausible.
  - On the other hand if success was guided by traffic analysis, it would also be "parallel constructed" as op-sec failure.

As of 2011 (Snowden documents) GCHQ had all the necessary infrastructural, mathematical, and operational tools to routinely break tor. Whether it did is a **matter of policy and other choices, not lack of capability**.

However, **tor is still the best systematic protection available** to individuals and legitimate organizations.

What next after Tor?

Measuring privacy degradation

- Problem: Tor is weak (stream tracing) and mix networks are weak (Statistical disclosure). But one is weaker. How do we measure anonymity?
- Define metrics for anonymity, and anonymity degradation.
  - Rely on probability theory to capture the uncertainty introduced by the system vis-à-vis an adversary.
  - Example: the entropy over the distribution of receivers.
- How to compute those probabilities?
  - Hard: large traces of adversary observations.
  - Complex constraints.
  - One way: Metropolis-Hastings Markov chain Monte Carlo (it took 7 years!)

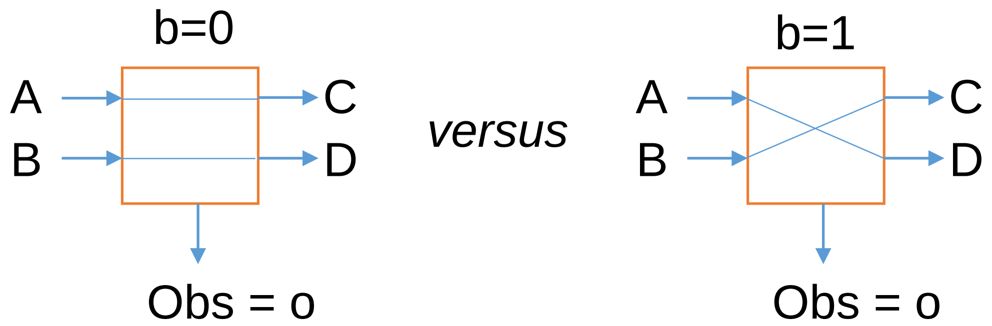
Our ability to build robust mix networks depends on correctly measuring their leakage. All of them leak. The question is: how much?

Carmela Troncoso, **George Danezis**: The bayesian traffic analysis of mix networks. ACM Conference on Computer and Communications Security 2009: 369-379
Andrei Serjantov, **George Danezis**: Towards an Information Theoretic Metric for Anonymity. Privacy Enhancing Technologies 2002: 41-53

Anoa Anonymity notions

- Define anonymity as three (ε,δ) differentially private mechanisms.
  - Adapted.

- Relationship anonymity – define two settings:



- For security parameters (ε,δ) it should hold that:

$$\forall o.\, \Pr[Obs = o |\, b = 0] \leq e^{\epsilon} \Pr[Obs = o | b = 1] + \delta$$

- Or for the special case :

$$\frac{\Pr[Obs = o |\, b = 0]}{\Pr[Obs = o | b = 1]} \leq e^{\epsilon} \equiv L(o)$$ (notation)

Properties of Anoa definition

- Small (ε,δ) are better.
  - ε > 0 – denotes the degree of bias introduced in the posterior belief in *b* no matter what the prior.
  - 0 < δ < 1 – is the likelihood of a catastrophic event.

- It composes under multiple correlated communications.
  - Adversary observes many rounds of the same relationships.
  - Naïve composition: sum ε and sum δ.
  - Downside: terrible bound, may lead to excluding perfectly good systems.

- Philosophical question:
  - Should we be looking at the worse case ε or the average ε. (in the coin tosses of the security mechanism)?
  - Note the $\forall a$ in the definition

In defence of an average ε metric (1)

- Argument for the worse case ε (largest).
  - This is a security metric.
  - Thus we must capture the observation for which the adversary gets the most information.

- However consider multiple runs of the protocol with ε=1, and the adversary observes for concrete observations o_0, o_1, o_2, o_3:

  With $L(o_0) = e^{-0.2}$, $L(o_1) = e^{0.1}$, $L(o_2) = e^{0.1}$, $L(o_3) = e^{0.1}$

- What is the overall $L(o = (o_0, o_1, o_2, o_3))$?
  - $L(o = (o_0, o_1, o_2, o_3)) = e^{0.1}$ (ie. $e^{(-0.2+0.1+0.1+0.1)}$) (1)
  - Much lower than e^4. (ie. $e^{4 \times \varepsilon}$) which is the possible maximum.
  - Eq. (1) Approaches the 4 x mean ε. The more observations the closest it gets.
  - What about the maximum? As more observations come in, the deviation from the mean becomes cryptographically small!

In defence of an average ε metric (2)

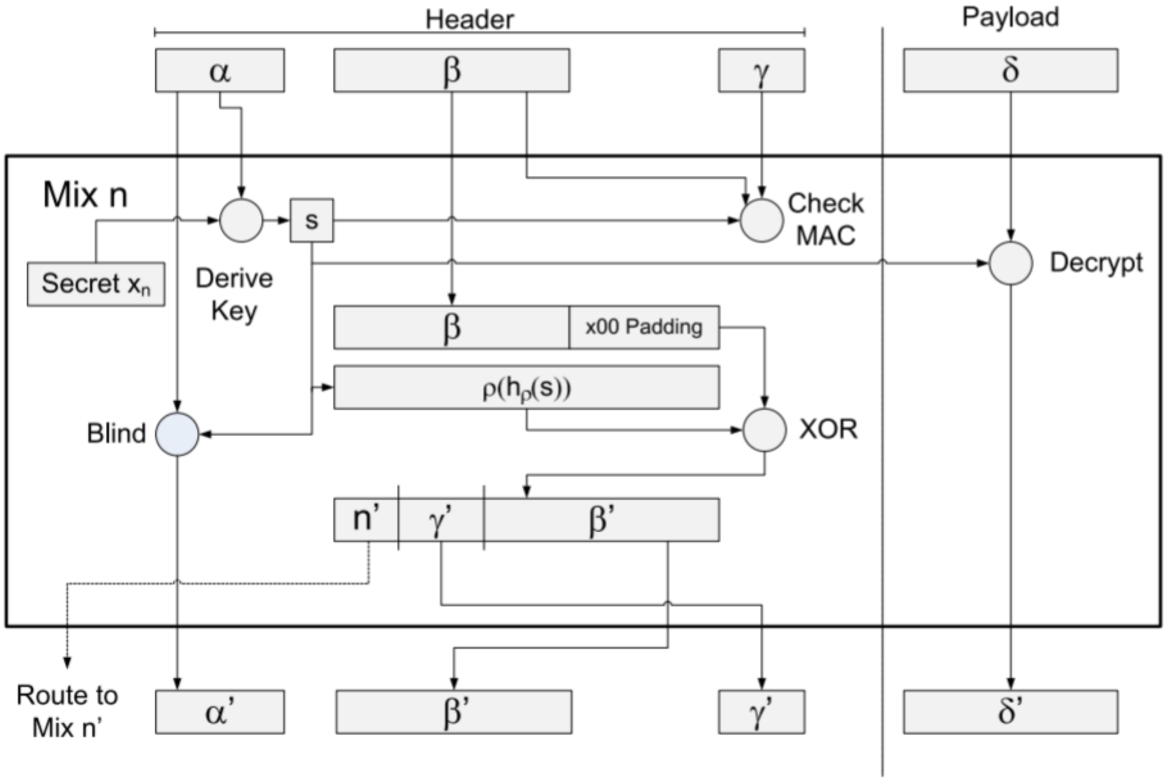
- Morality of the story:
  - The mean ε seems much more informative about the mechanisms under composition.
  - An adversary will unlikely beat the mean ε over multiple experiments (multiple attacks) or multiple observations.
- Good news – Monte Carlo evaluation of anonymity:
  - Mean ε is much easier to compute experimentally (through Monte Carlo).
  - Perform the experiment multiple times and estimate the probability distribution of the mean ε. And the probability of encountering untypical samples – which you can fold into the probability δ.

- In the experimental section of our latest works we consider the mean ε, and the results are stunningly different from the worse case!

Sorting out the crypto: the Sphinx format

- Clients pack messages in layers of encryption.
- Each mix decrypts a layer.
- Many features needed: unlinkability, resistance to active attacks, indistinguishable replies, no leakage of path length, path position, etc. Many ways of getting it wrong.
- Sphinx does it (provably) right, and everyone may use it.

Do not reinvent your own mix network crypto.



George Danezis, Ian Goldberg: Sphinx: A Compact and Provably Secure Mix Format. IEEE Symposium on Security and Privacy 2009: 269-282

Understanding indistinguishability of streams

- Why can tor streams be traced?
  - Two different web browsing streams look very different.
  - On-off periods.
  - Great variability of packet rates and volumes in general TCP.

- Traffic streams that are regular can be confused with each other, hampering tracing.

- Key applications: Voice-over-IP and instant messaging.
  - Constant rate traffic, or very low volumes.
  - **Drac design: create a bed of regular traffic in a close nit social network.**
  - Indistinguishability of calls “within” network.
  - Anonymity of calls to “far” nodes in the network.

George Danezis, Claudia Díaz, Carmela Troncoso, Ben Laurie: Drac: An Architecture for Anonymous Low-Volume Communications. Privacy Enhancing Technologies 2010: 202-219

Preventing mass surveillance & embedding anonymity at the network level

- Tor is too small to argue that it cannot be subject to a `global passive adversary'
- However, if the whole internet was `anonymized' then a GPA would indeed be difficult to instantiate.
- Mass surveillance resistance: there is no trivial bit string on the network that may act as a stream identifier, or betray a connection between a sender and receiver.
- Forces an adversary to record traffic, and perform statistical traffic analysis.

- HORNET: can route anonymized streams at 93 Gb/s!
  - Turn all routers into onion routers.
  - Minimize any per-flow state to scale up to many cores.
  - Still susceptible to stream tracing.

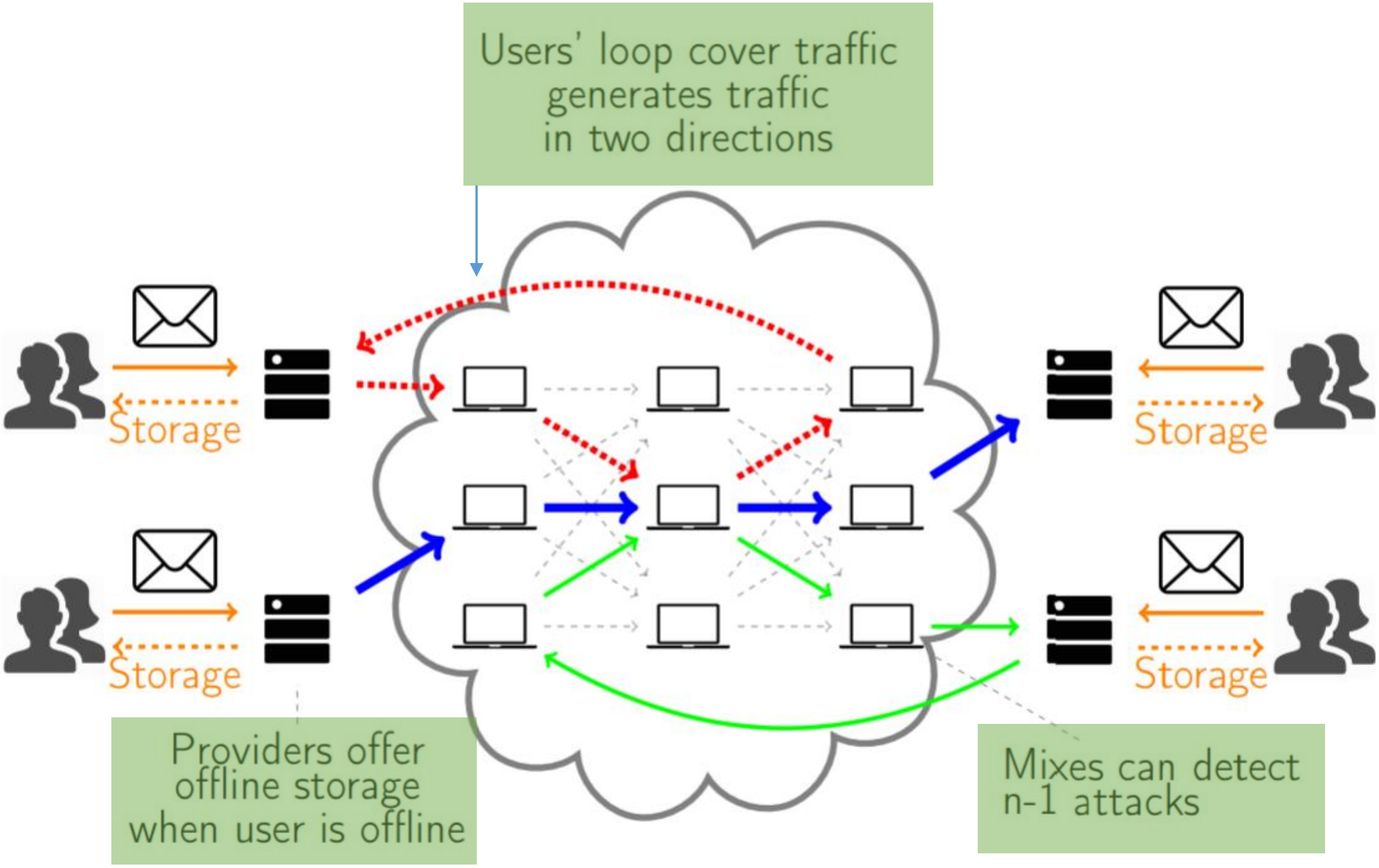
Chen Chen, Daniele Enrico Asoni, David Barrera, **George Danezis**, Adrian Perrig: HORNET: High-speed Onion Routing at the Network Layer. ACM Conference on Computer and Communications Security 2015: 1441-1454

Modern mixnets : loopix

- The Loopix Anonymity System.
- 3rd party anonymity only.
- Providers for access control.
- UDP transport & loss.
- Very low latency mixing (1.5 sec latency)
- Cover traffic in loops from clients and mixes.
- Variant of SG-mix (exp. Delay)
- Active (n-1) detection.
- Lean mathematical foundation to help analysis of leakage.

Dogan Kesdogan, Jan Egner, Roland Büschkes: Stop-and-Go-MIXes Providing Probabilistic Anonymity in an Open System. Information Hiding 1998: 83-98
Ania Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, **George Danezis**: The Loopix Anonymity System. CoRR abs/1703.00536 (2017)

The loopix architecture



Loopix details and design choices (Q&A)

- Q: Why do you use an exponential delay per message instead of batching?
  A: The memoryless property allows for easy analysis. Poisson arrivals are not necessary.

- Q: What kind of cover traffic you use?
  A: Sender and mixes send loops to themselves; users send drop packets; those are substituted up to a point with real traffic. This offer sender unobservability.

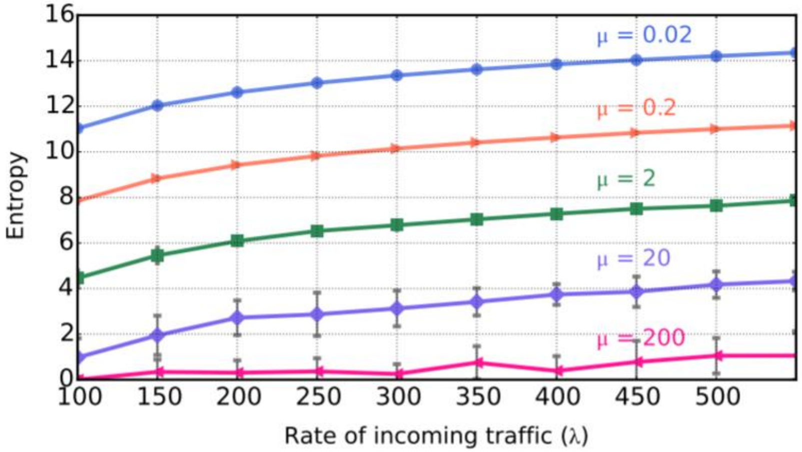
- Q: Why do you use a UDP transport?
  A: We are not interested in retransmitting a lot of classes of traffic, including the cover traffic. So UDP avoids delaying the latest real messages to ensure every piece of cover is delivered.

Loopix details and design choices (Q&A)

- Q: What topology do you use?
  A: Stratified network, with each layer of mixes feeding messages to the next layers. The path goes from user to provider to stratified to provider to user.
- Q: What are providers for?
  A: They buffer messages at the end of paths to support offline delivery. They do admission control to avoid Sybil attacks.
- Q: Why are mixes sending cover traffic?
  A: Mixes measure the amount of cover traffic returning to them to estimate whether they are under a n-1 attack. If they are they may deploy drop messages.
- Q: Is cover traffic not too expensive as the system grows?
  A: Well, as there are more users the "natural" traffic not under the control of the adversary may also grow. The amount of cover traffic necessary (or traffic unknown to the adversary) is a measure of the system topology not the number of users.

An evaluation of loopix anonymity

- Two key security parameters:
  - Overall rate of messages not controlled by the adversary.
  - Cover drop messages, loop messages of honest users.
  - Real messages if the adv. Knows nothing about them.
  - Exponential delay at the mix.



- Illustration:
  - (Simulation results).
  - X-axis: rate of messages.
  - Lines: delay (lower mu is higher delay).
  - Y-axis: anonymity measure.

Loopix open questions

- Fragmentation & classes of traffic.
  - Big messages need to be split in small packets – but more messages more leakage.
  - But big packets lead to large overhead for small messages.
  - Multiple parameter sets would be distinguishable.
  - And may need to be delayed by a different amount.
- Reliable transmission.
  - Need to a system of acks and retransmits.
  - But retransmits leak (composition).
  - And the end-point may not be on-line right now (unreliable RTT).
- Efficient directory authorities.
  - Users need to learn of the topology privately, or naïve PIR.
  - Key question: can we leverage loopix to get cheaper PIR?
- Private and dynamic parameter adaptation.
  - How users chose the rates of cover traffic and user-specified delay?
  - Collective statistics on number of users, and volumes needs to be secure.

Scaling private lookups with anonymity

- Remember epistemic attacks: how can you distribute privately all the routing and keying information necessary to build circuits or paths?
- Private Information Retrieval (PIR) – good primitive but expensive.
- Solution: use an anonymity system to make PIR cheaper.

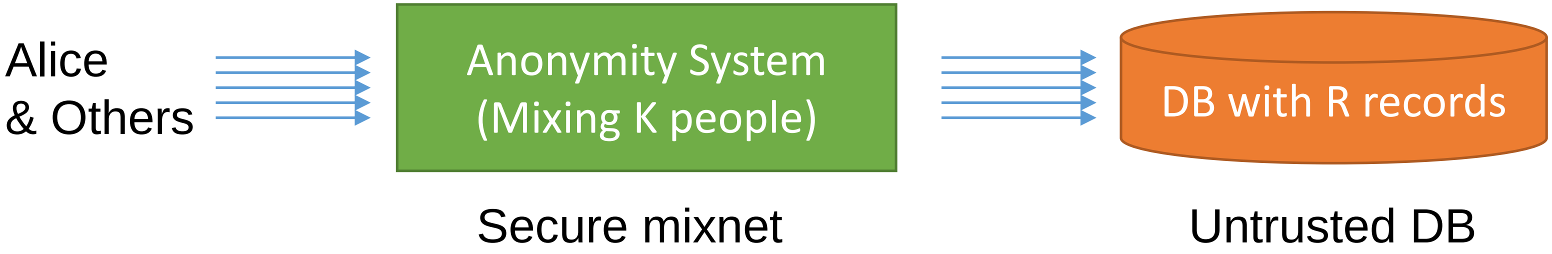
- This is the killer app of tor – private web browsing is a flexible application of PIR.
- So is it trivial to use an anonymity system to do PIR?

Raphael R. Toledo, **George Danezis**, Ian Goldberg: Lower-Cost ∈-Private Information Retrieval. PoPETs 2016(4): 184-201 (2016)
Ania Piotrowska, Jamie Hayes, Nethanel Gelernter, **George Danezis**, Amir Herzberg: AnoNotify: A Private Notification Service. IACR Cryptology ePrint Archive 2016: 466 (2016)

Reminder: PIR & trivial anonymity solution

- PIR: private information retrieval.
- Public database, private lookups.
- Trivial solution: download full database.

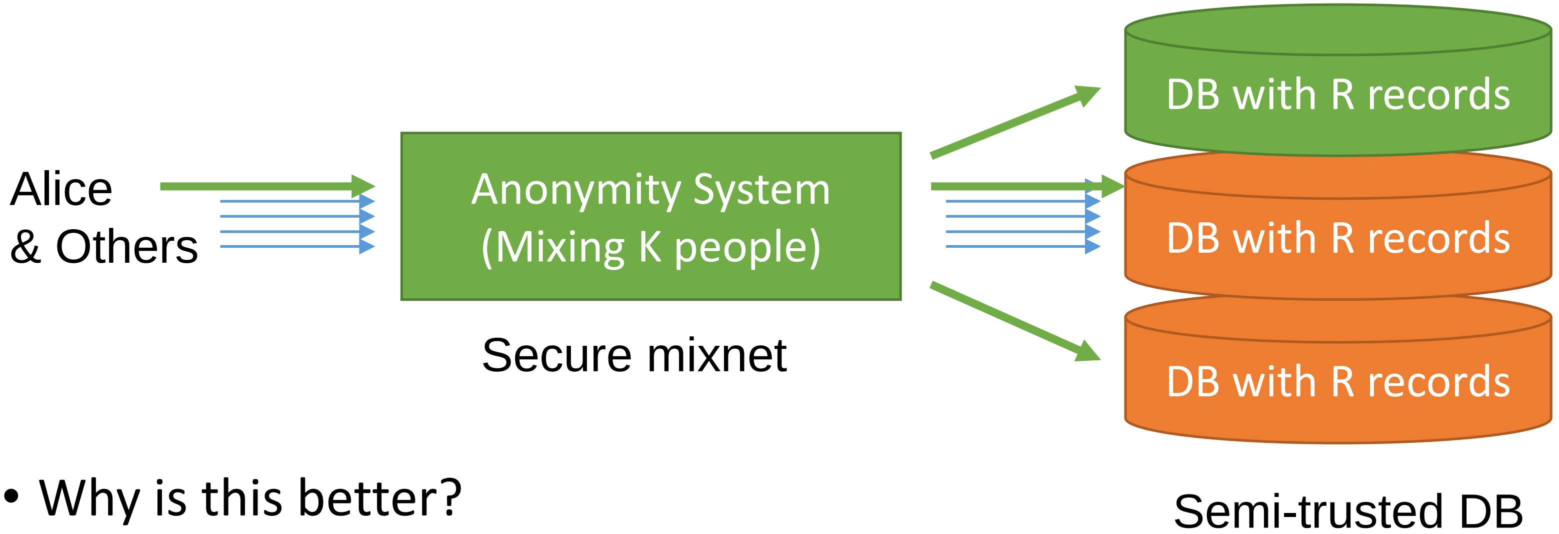
- A broken design:



- If K < R then at least one record is not accessed.
  - Cryptographic game: adversary provides two challenge records and wins if they guess which was accessed.
  - Thus the adversary may exclude the possibility that this was the record accessed by Alice – catastrophic failure.

Option 1: Anonymous dummy requests

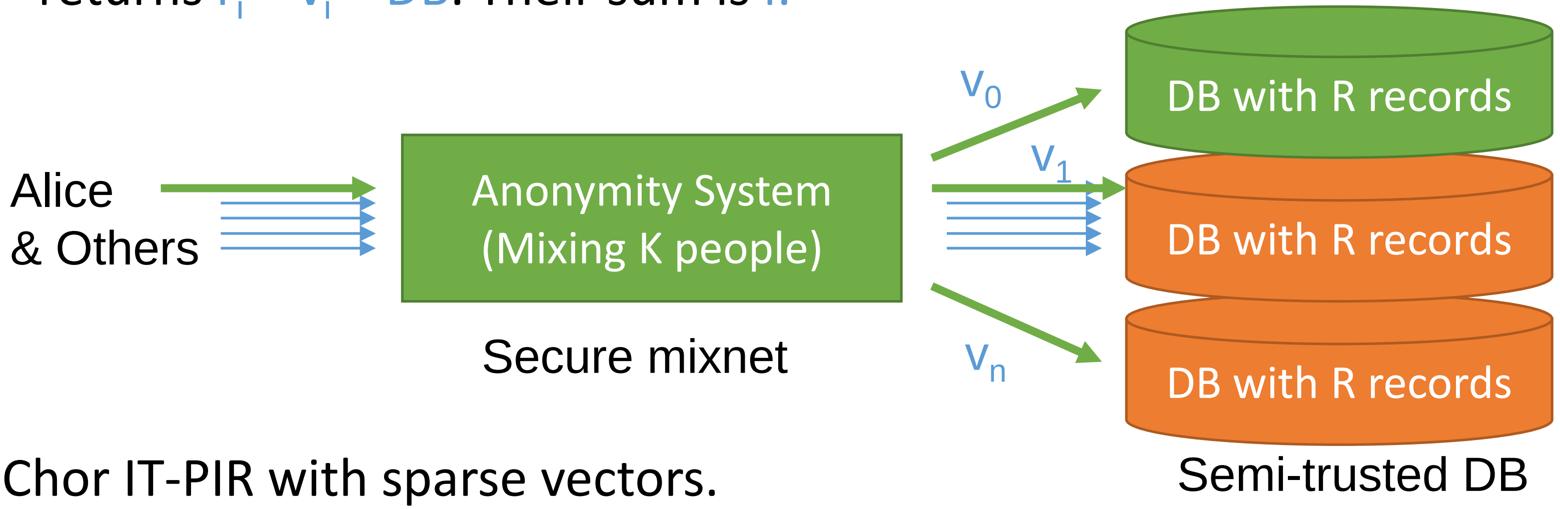
- Alice sends the request for the record, along with some dummy requests across semi-trusted servers.



- Why is this better?
  - At least one of the servers honest cannot be observed.
  - Adversary cannot be sure if any record was accessed on that server.
  - Non-catastrophic leakage.
  - Then: anonymity system amplifies the uncertainty of the adversary!

Option 2: Light PIR

- Alice sends the sparse binary vectors $v_0, v_1, \ldots, v_n$, one to each DB server. With the property $v_0 + v_1 + \ldots + v_n \mod 2 = I(r)$. Each DB returns $r_i = v_i \bullet DB$. Their sum is r.



Chor IT-PIR with sparse vectors.

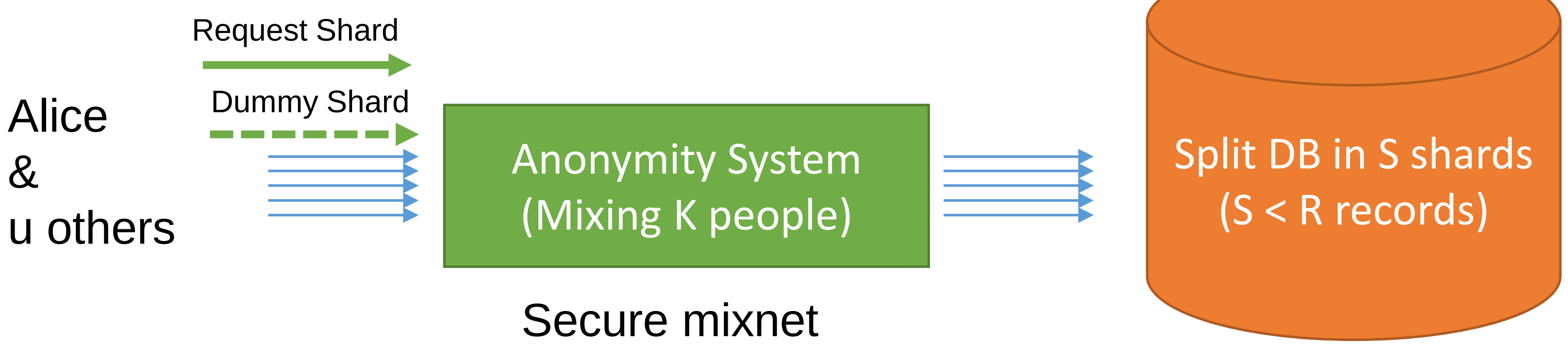
- Less costly to communicate the vectors. Less to compute the returns.
- Security based on at least one honest DB.
- BUT: leakage, all records are no more equally likely given a view of the adversary.
- However, an anonymity system increases the adversary's confusion.

Option 3: Sharded trivial PIR.

- Can we do it with untrusted infrastructure? Yes.

Each user sends a request to a shard and a dummy shard.

When a shard is requests it is fully downloaded (trivial PIR)



- Result:
  - If the users u >> S then the probability is that all shards are downloaded.
  - Leakage, but non-catastrophic.
  - Crucial dependence on anonymity system to mix dummies, requests, etc.

Scaling private lookups with anonymity

- You can use an anonymity system for efficient PIR!
- However, they leak: it is **key to understand rate of leakage** to make use of relaxed notions of PIR leveraging anonymity systems.

- Relevance to anonymous communications:
  - Users need to retrieve directory information.
  - For each node: position, keys, address, parameters.
  - For each user: keys, provider, address.
  - Cheaper to do with PIR using the anonymity system.
  - Sharded PIR: allows the retrieval of large number of records (so does PIR).
  - Untrusted directories (for privacy).

Raphael R. Toledo, **George Danezis**, Ian Goldberg: Lower-Cost ∈-Private Information Retrieval. PoPETs 2016(4): 184-201 (2016)
Ania Piotrowska, Jamie Hayes, Nethanel Gelernter, **George Danezis**, Amir Herzberg: AnoNotify: A Private Notification Service. IACR Cryptology ePrint Archive 2016: 466 (2016)

Private analytics

- How to collect data to tune the anonymity network or provision?
- Releasing detailed statistics from each mix or relay can facilitate traffic analysis.

- Solution: use multi-party computation to collect statistics:
  - Privex: collect simple weighted sums, means and variances from counters at relays.
  - Crux: collect sketches of distirbutions, to compute medians, quantiles, and percentiles.

- Open Question: can we leverage the anonymity system to collect private statistics more efficiently? Under what security definition?

Tariq Elahi, **George Danezis**, Ian Goldberg: PrivEx: Private Collection of Traffic Statistics for Anonymous Communication Networks. ACM Conference on Computer and Communications Security 2014: 1068-1079
Luca Melis, **George Danezis**, Emiliano De Cristofaro: Efficient Private Statistics with Succinct Sketches. NDSS 2016

Taming abuse

- Anonymity revocation is a bad idea!
  - Key argument: bad people will use something else, good people will lose privacy.
  - Black box revocation mechanisms: not robust. Trace honest users, miss dishonest users. Ability to frame users.
  - White box tracing: increases complexity of protocols significantly.
- What abuse?
  - Unwanted communications from anonymous parties: spam, threats, abuse, doxing.
  - Unwanted services: drugs markets, illicit material sites (hidden services)
- Providers & 3rd party anonymity:
  - Alice and Bob know each other, but 3rd parties cannot tell they communicate.
  - Strengthening of channel security.
  - Vulnerability to communication partner.
  - Ability to have strong authentication within channel.

Doctrine: provide GPA resistant 3rd party anonymity, and partial adversary resistant full sender / receiver anonymity.

George Danezis, Len Sassaman: How to Bypass Two Anonymity Revocation Schemes. Privacy Enhancing Technologies 2008: 187-201

Where next?

- Reliability: Loopix provides unreliable transport. Traffic analysis resistant flow / congestion control, Acks & retransmits. Malicious mixes.
- Efficiency: Can we make cryptography cheaper? IM messages have 160 bytes of payload. Core internet routers shift many GB/s of traffic.
- Economics: mix service operations cost, users will eventually have to pay. Provider model is one possible direction.
- Analytics: network management, provisioning, payments, and grant reporting require analytics. How to do those safely?

In Conclusion:

- Mix networks are the future of strong anonymity: low-latency, cover traffic, active defences, and providers for payment and Sybil resistance.
- Key to deploying solutions is understanding leakage to compare systems. They all leak, but at different rates.