Protocols for anonymity

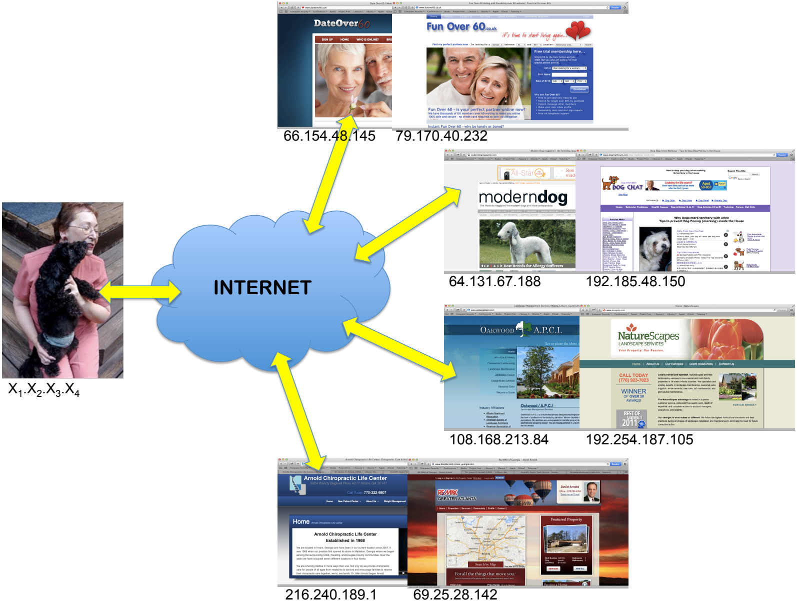
Myrto Arapinis
School of Informatics
University of Edinburgh

March 17, 2014

Context

- The Internet is a public network:
  - network routers see all traffic that passes through them
- Routing information is public:
  - IP packet headers contain source and destination of packets
- Encryption does not hide identities:
  - encryption hides payload, but not routing information

Routing information can reveal who you are!



INTERNET

$X_1.X_2.X_3.X_4$

66.154.48.145

79.170.40.232

64.131.67.188

192.185.48.150

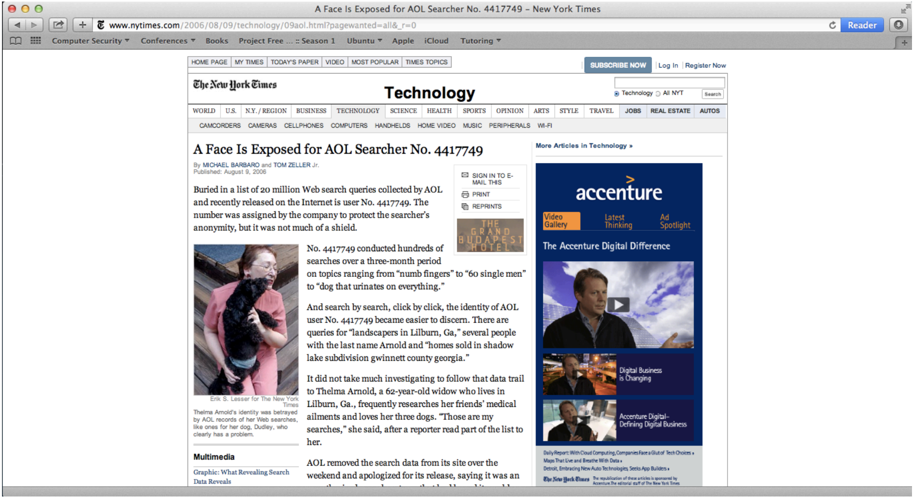
108.168.213.84

192.254.187.105

216.240.189.1

69.25.28.142

Routing information can reveal who you are!



A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER Jr.
Published: August 9, 2006

Buried in a list of 20 million Web search queries collected by AOL and recently released on the Internet is user No. 4417749. The number was assigned by the company to protect the searcher's anonymity, but it was not much of a shield.

Erik S. Lesser for The New York Times

Thelma Arnold's identity was betrayed by AOL records of her Web searches, like ones for her dog, Dudley, who clearly has a problem.

No. 4417749 conducted hundreds of searches over a three-month period on topics ranging from "numb fingers" to "60 single men" to "dog that urinates on everything."

And search by search, click by click, the identity of AOL user No. 4417749 became easier to discern. There are queries for "landscapers in Lilburn, Ga," several people with the last name Arnold and "homes sold in shadow lake subdivision gwinnett county georgia."

It did not take much investigating to follow that data trail to Thelma Arnold, a 62-year-old widow who lives in Lilburn, Ga., frequently researches her friends' medical ailments and loves her three dogs. "Those are my searches," she said, after a reporter read part of the list to her.

AOL removed the search data from its site over the weekend and apologized for its release, saying it was an

Multimedia

Graphic: What Revealing Search Data Reveals

Your IP address is your ID

Your IP address leaves behind digital tracks that can be used to identify you and invade your privacy

Anonymity

Definition (ISO/IEC standard 15408)

A user may use a service or resource without disclosing the users identity.

Anonymity

Definition (ISO/IEC standard 15408)

A user may use a service or resource without disclosing the users identity.

⟶ this can be achieved by hiding one's activities among others' similar activities

- Dinning cryptographers
- Crowds
- Chaum's mix
- Onion routing

Three-party dinning cryptographers (3DC) protocol

Three cryptographers are having dinner. Either NSA paid for the dinner, or one of the cryptographers. They want to know if it is the NSA that paid, but without revealing the identity of the cryptographer that paid in the case the NSA did not pay.

3DC protocol:

1. Each cryptographer flips a coin and shows it to his left neighbor:
   - each cryptographer will see his own coin and his right neighbor's
2. Each cryptographer announces whether the two coins he saw are the same. If he is the payer, he lies
3. odd number of "same" ⇒ the NSA paid
   even number of "same" ⇒ one of the cryptographers paid
   - only the payer knows he is the one who paid

Superposed sending

- 3DC protocol generalises to any group size *n* (nDC)
- Sender wants to anonymously broadcast a message *m*:
  1. for each bit of the *m*, every user generates a random bit and sends it to his left neighbor
     - every user learns two bits: his own, and his right neighbor's
  2. each user (except the sender) announces (own_bit XOR neighbor's_bit)
  3. the sender announces (own_bit XOR neighbor's_bit XOR message_bit)
  4. XOR of all announcements = message_bit
     - every randomly generated bit occurs in this sum twice (and is canceled by XOR)
     - message_bit occurs only once

Limitations of the DC protocol

The DC protocol is impractical:

- Requires pair-wise shared secret keys (secure channels) between the participants (to share random bits)
- Requires large amounts of randomness

Crowds

[M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions". ACM Transactions on Information and System Security.]

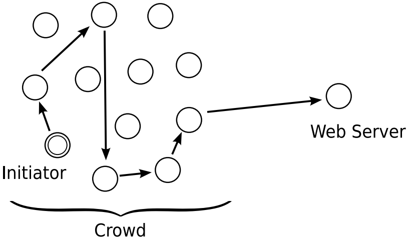
Idea: randomly route the request through a crowd of users

Crowds

[M. K. Reiter and A. D. Rubin, “Crowds: anonymity for Web transactions”. ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of m users; c out of m users may be corrupted

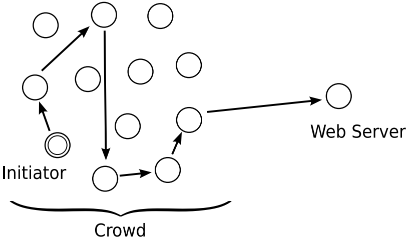


Crowds

[M. K. Reiter and A. D. Rubin, “Crowds: anonymity for Web transactions”. ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of *m* users; *c* out of *m* users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:

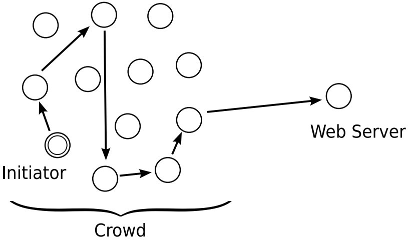


Crowds

[M. K. Reiter and A. D. Rubin, “Crowds: anonymity for Web transactions”. ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of *m* users; *c* out of *m* users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:
  1. the initiator selects a forwarder from the crowd and sends him his request

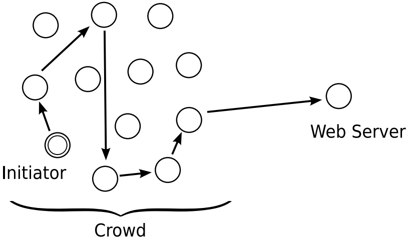


Crowds

[M. K. Reiter and A. D. Rubin, “Crowds: anonymity for Web transactions”. ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of m users; c out of m users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:
  1. the initiator selects a forwarder from the crowd and sends him his request
  2. a forwarder delivers the request directly to the server with probability $1 - p_f$; he forwards the request to a randomly selected new forwarder from the crowd with probability p_f; the new forwarder repeats the procedure

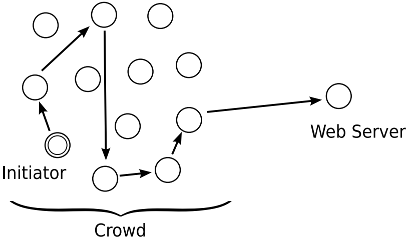


Crowds

[M. K. Reiter and A. D. Rubin, “Crowds: anonymity for Web transactions”. ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

- a crowd is a group of m users; c out of m users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:
  1. the initiator selects a forwarder from the crowd and sends him his request
  2. a forwarder delivers the request directly to the server with probability $1 - p_f$; he forwards the request to a randomly selected new forwarder from the crowd with probability p_f; the new forwarder repeats the procedure
  3. the response from the server follows same route in opposite direction

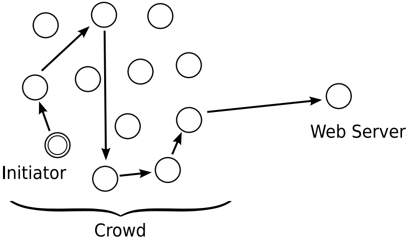


Crowds

[M. K. Reiter and A. D. Rubin, “Crowds: anonymity for Web transactions”. ACM Transactions on Information and System Security.]

Idea: randomly route the request through a crowd of users

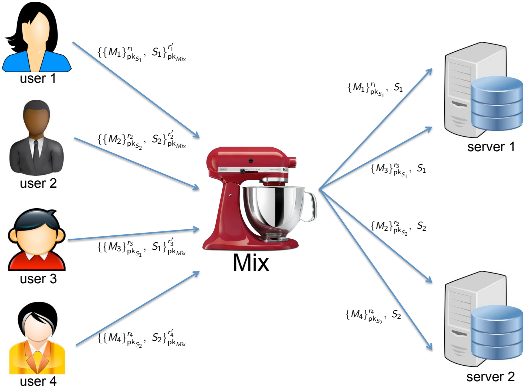
- a crowd is a group of m users; c out of m users may be corrupted
- an initiator that wants to request a webpage creates a path between him and the server:
  1. the initiator selects a forwarder from the crowd and sends him his request
  2. a forwarder delivers the request directly to the server with probability $1 - p_f$; he forwards the request to a randomly selected new forwarder from the crowd with probability p_f; the new forwarder repeats the procedure
  3. the response from the server follows same route in opposite direction



Crowd IS NOT resistant against an attacker that sees the whole network traffic!

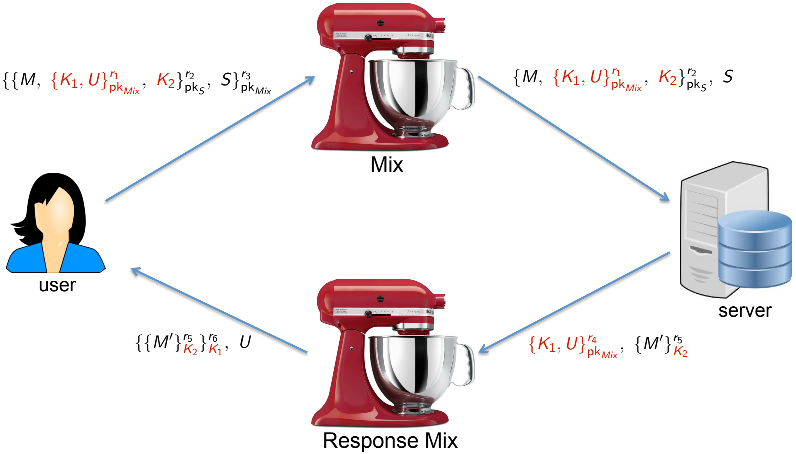
Chaum's mix

[D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", Communications of the ACM, February 1981.]


user 1
user 2
user 3
user 4
$\{\{M_1\}^{r_1}_{\mathrm{pk}_{S_1}},\ S_1\}^{r'_1}_{\mathrm{pk}_{Mix}}$
$\{\{M_2\}^{r_2}_{\mathrm{pk}_{S_2}},\ S_2\}^{r'_2}_{\mathrm{pk}_{Mix}}$
$\{\{M_3\}^{r_3}_{\mathrm{pk}_{S_1}},\ S_1\}^{r'_3}_{\mathrm{pk}_{Mix}}$
$\{\{M_4\}^{r_4}_{\mathrm{pk}_{S_2}},\ S_2\}^{r'_4}_{\mathrm{pk}_{Mix}}$
Mix
$\{M_1\}^{r_1}_{\mathrm{pk}_{S_1}},\ S_1$
$\{M_3\}^{r_3}_{\mathrm{pk}_{S_1}},\ S_1$
$\{M_2\}^{r_2}_{\mathrm{pk}_{S_2}},\ S_2$
$\{M_4\}^{r_4}_{\mathrm{pk}_{S_2}},\ S_2$
server 1
server 2

- message padding and buffering to avoid time correlation attacks
- dummy messages are generated by the mixes themselves to prevent an attacker sending $n-1$ messages to a mix with capacity n, allowing him to then link the sender of the n^{th} message with its recipient

Anonymous return addresses



$\{\{M, \{K_1, U\}^{r_1}_{\mathrm{pk}_{Mix}}, K_2\}^{r_2}_{\mathrm{pk}_S}, S\}^{r_3}_{\mathrm{pk}_{Mix}}$

Mix

$\{M, \{K_1, U\}^{r_1}_{\mathrm{pk}_{Mix}}, K_2\}^{r_2}_{\mathrm{pk}_S}, S$

user

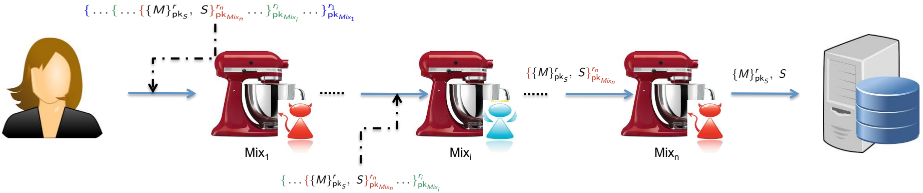
server

$\{\{M'\}^{r_5}_{K_2}\}^{r_6}_{K_1}, U$

$\{K_1, U\}^{r_4}_{\mathrm{pk}_{Mix}}, \{M'\}^{r_5}_{K_2}$

Response Mix

Mix cascade


$\{\ldots\{\ldots\{\{M\}^{r}_{\mathrm{pk}_S},\ S\}^{r_n}_{\mathrm{pk}_{Mix_n}}\ldots\}^{r_i}_{\mathrm{pk}_{Mix_i}}\ldots\}^{r_1}_{\mathrm{pk}_{Mix_1}}$
$\{\ldots\{\{M\}^{r}_{\mathrm{pk}_S},\ S\}^{r_n}_{\mathrm{pk}_{Mix_n}}\ldots\}^{r_i}_{\mathrm{pk}_{Mix_i}}$
$\{\{M\}^{r}_{\mathrm{pk}_S},\ S\}^{r_n}_{\mathrm{pk}_{Mix_n}}$
$\{M\}^{r}_{\mathrm{pk}_S},\ S$
Mix_1
Mix_i
Mix_n

- messages are sent through a sequence of mixes
- some of the mixes may be corrupted
- a single honnest mix guarantees anonymity against an attacker controlling the whole network provided it applies:
  - message padding
  - buffering
  - dummy messages

Limitations of Chaum's mixnets

- Asymmetric encryption is not efficitent
- Dummy messages are innefficient
- Buffering is not efficient

Onion routing

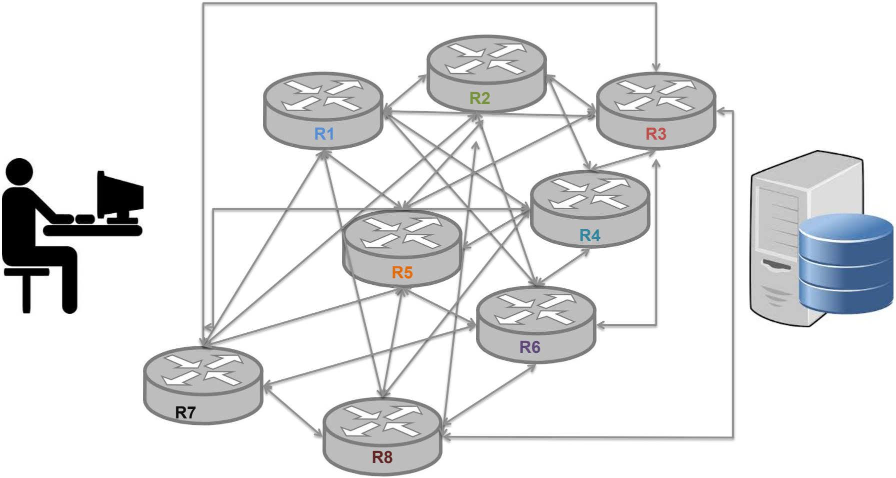
[R. Dingledine, N. Mathewson, and P. F. Syverson: "Tor: The Second-Generation Onion Router", USENIX Security Symposium 2004]

Idea: combine advantages of mixes and proxies

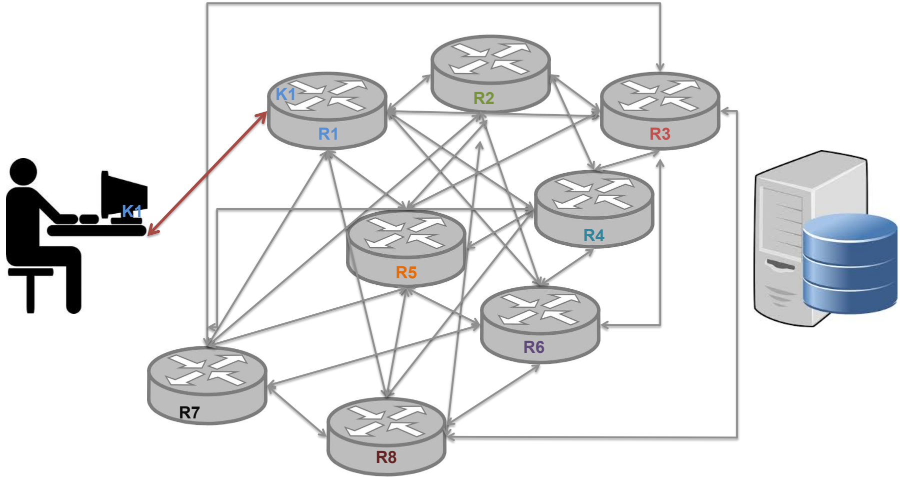
- use public-key crypto only to establish circuit
- use symmetric-key crypto to exchange data
- distribute trust like mixes

But does not defend against attackers that controle the hole network

TOR circuit setup

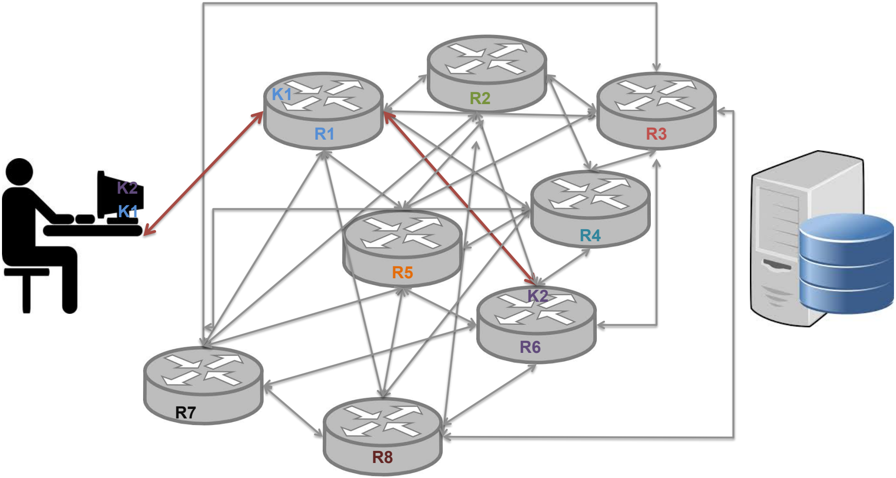


TOR circuit setup



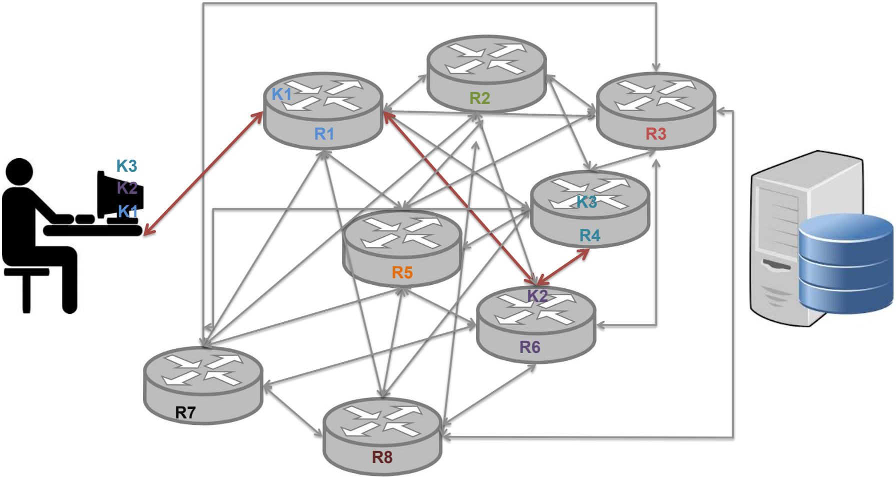
- client establishes session key **K1** and circuit with Onion Router **R1**

TOR circuit setup



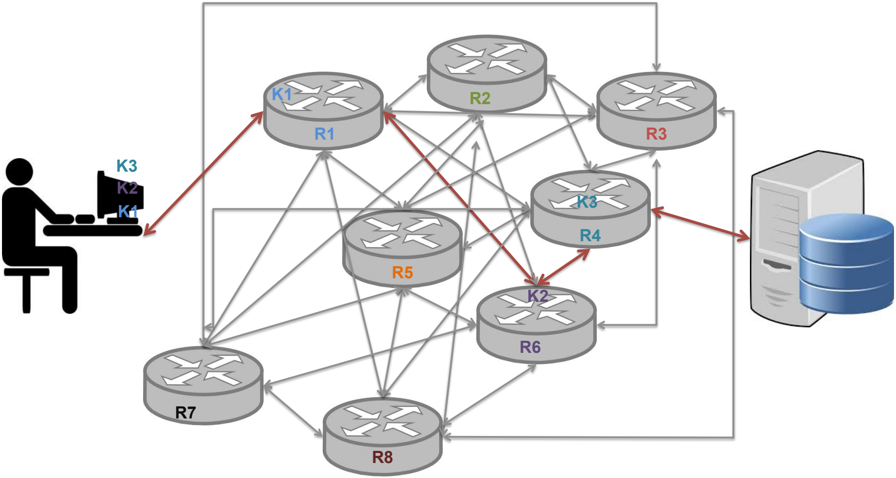
- client tunnels through that circuit to extend to Onion Router **R6**

TOR circuit setup



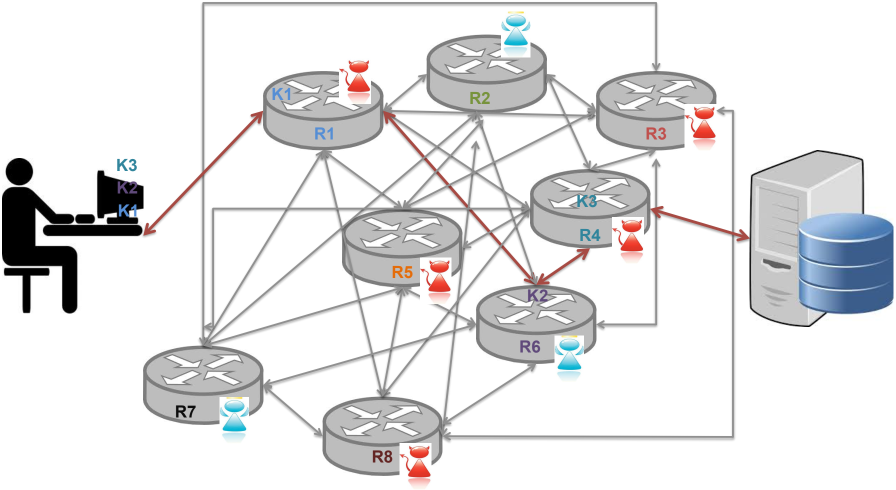
- client tunnels through that extended circuit to extend to Onion Router **R4**

TOR circuit setup



- client applications connect and communicate of established TOR circuit

TOR circuit setup



a single honnest Onion Router on the TOR circuit guarantees anonymity against an attacker controlling some Onion Routers