UCL

Université catholique de Louvain

Correlation attacks on the Tor network

Masters Thesis Defense

Crombé Henri & Declercq Mallory

Supervisor: Pereira Olivier
Co-supervisor: Rochet Florentin

September 5, 2016

Outline

- The Tor network
- Threat model
- Threat analysis
- Conclusion

The Tor network

The Tor network

- Tor stands for “The Onion Routing”.
- Low-latency network designed to anonymize the traffic of TCP-based applications.

→ An adversary cannot discover who is talking to whom if she only observes one end of a circuit.

Tor cells

- Clients and ORs communicate with fixed-size messages called Tor “cells”.
- Tor cells allow to:
    - Send commands to ORs:
        - create or destroy circuits.
    - Relay data through the Tor network
- Cell’s content is encapsulated in layers of encryption, like the layers of an onion.
- The encryption layers are “peeled” by the ORs one layer at a time along a circuit. Only the (current) last hop of a circuit know the actual cell’s content

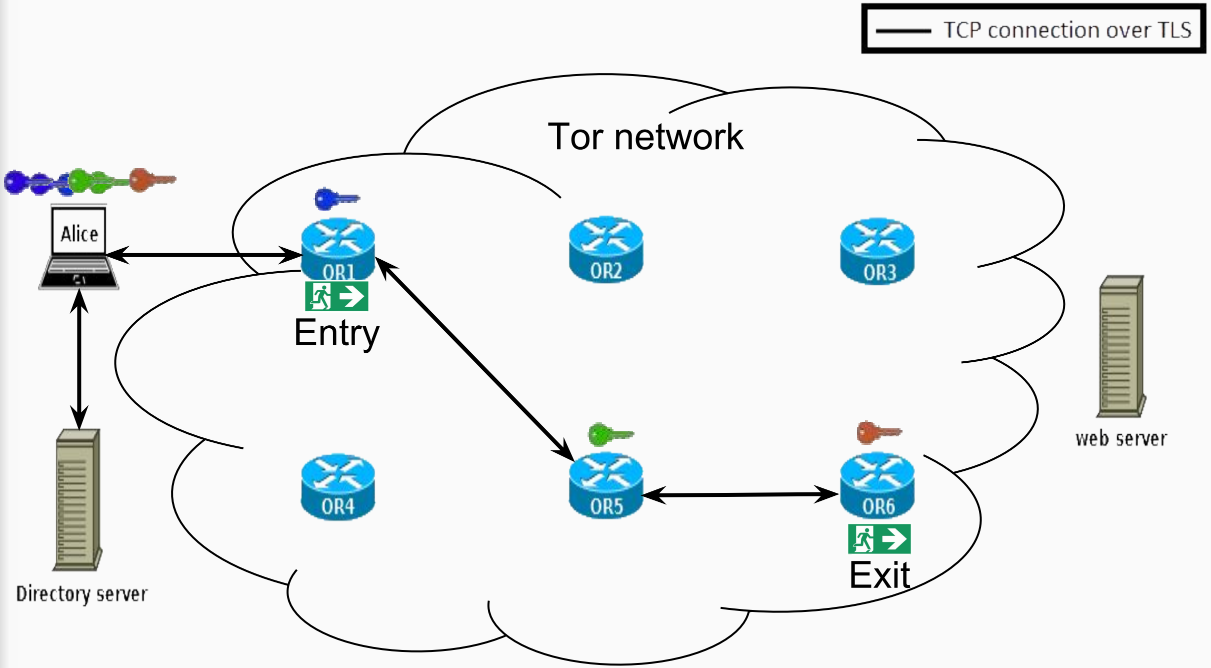
Example of a Web request sent over Tor (I)

Step 1

- Alice's Tor client obtains a list of Tor ORs from a directory server.

Step 2

- Alice's Tor client builds a random 3-hop circuit :
    - Alice negotiates with OR1 to be its Entry point in the Tor network.
    - Then Alice extends the circuit to OR5 using OR1.
    - Finally, Alice finishes the circuit construction by extending the circuit to OR6 (Exit OR).



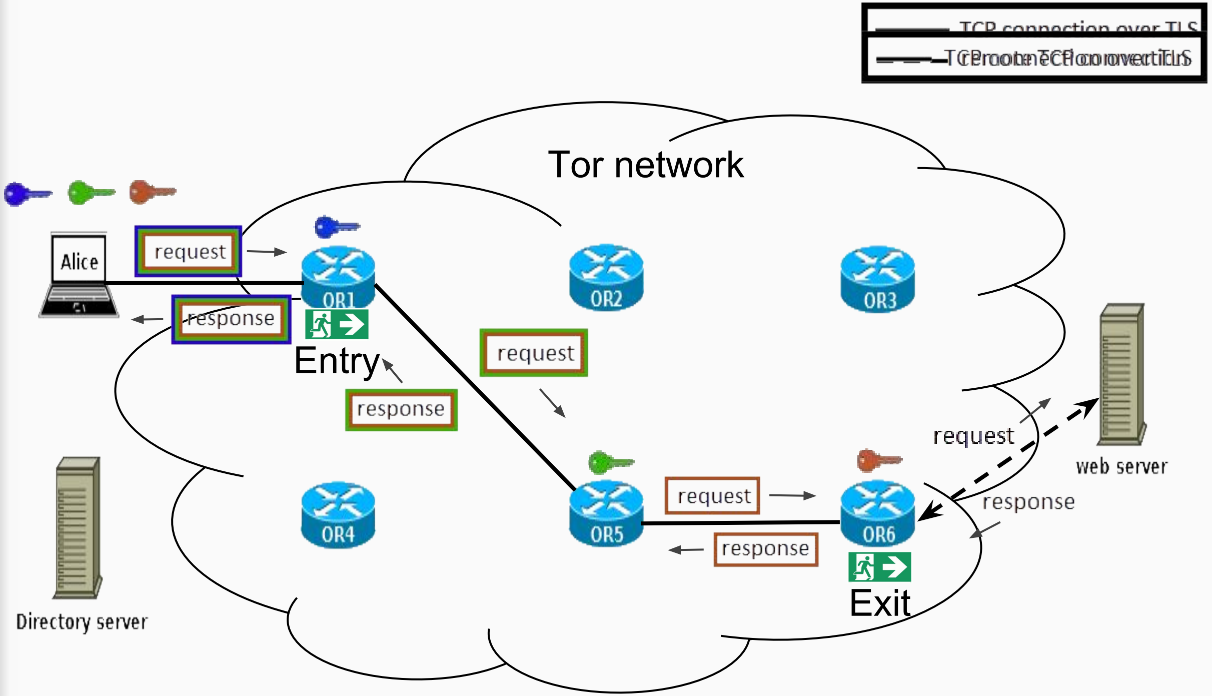
Example of a Web request sent over Tor (II)

Step 3

- Alice instructs the Exit OR to begin a TCP connection with the remote web server.

Step 4

- Alice's Tor client creates a cell containing the GET request and sends it to the web server.



Threat model

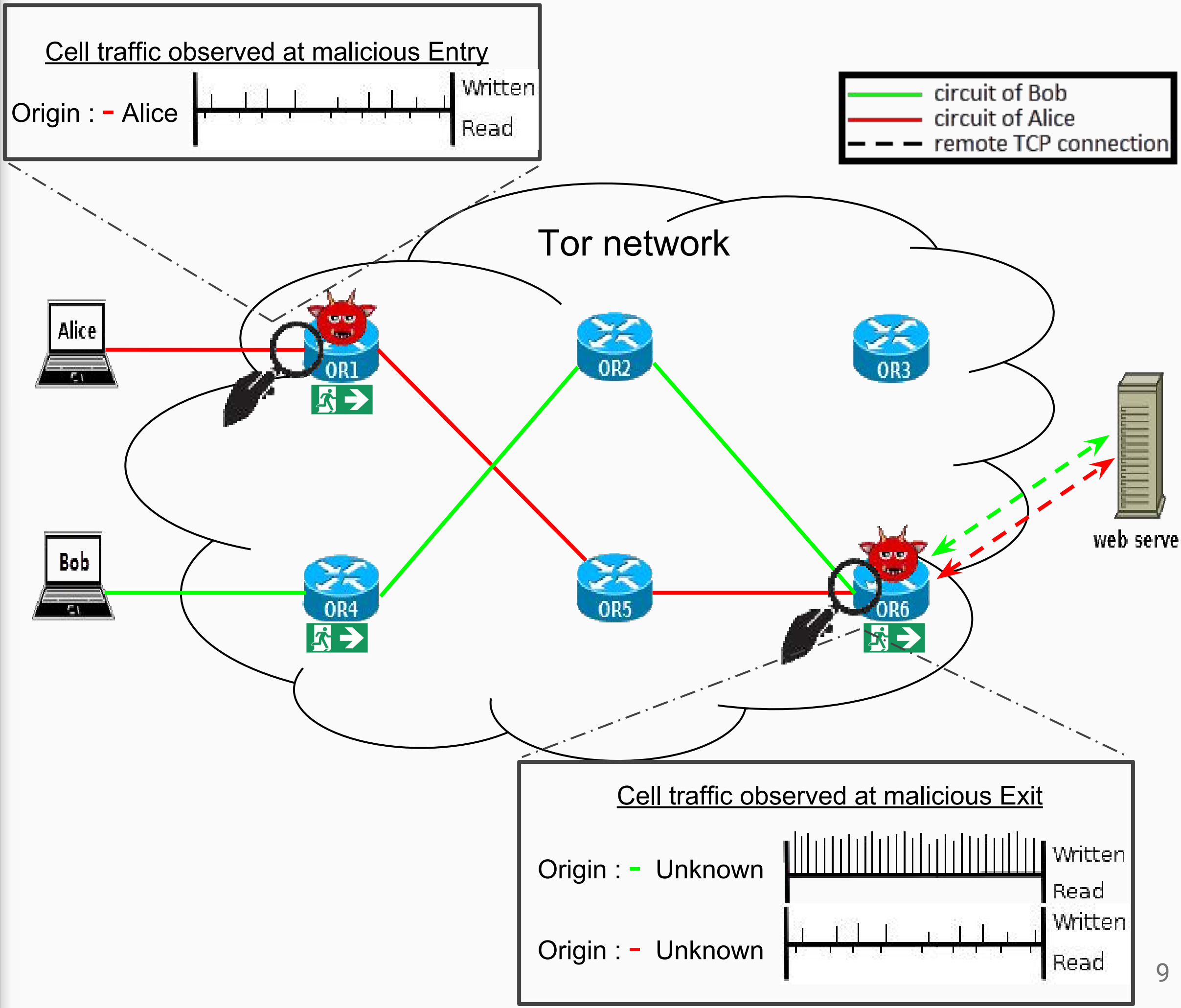
End-to-end correlation attacks

Threat model : Example

- If an adversary controls the entry node OR1, she knows :
  - Alice’s IP address
  - Cell traffic pattern between OR1 and Alice

- If the adversary controls the exit node OR6, she knows :
  - Green/Red circuits are linked to the Green/Red exit connections
  - Cell traffic patterns of the Green/Red circuits
  - Content exchanged on circuits and connections

- If she succeeds to match the entry pattern of Alice with its corresponding exit pattern at the exit, she is able to infer with whom a given user interacts (Alice <-> web server).



Cell traffic observed at malicious Entry

Origin : 🟥 Alice

🟩 circuit of Bob

🟥 circuit of Alice

– – remote TCP connection

Tor network

Cell traffic observed at malicious Exit

Origin : 🟩 Unknown

Origin : 🟥 Unknown

In this context ...

- We analyze the effectiveness of end-to-end correlation attacks against :
  - Different types of clients :
    - Web surfing
    - Bulk download
    - IRC
    - SSH
  - With different correlation techniques :
    - Basic approach : Analyze traffic timings and determine the proportion of time the circuits have been active/inactive at the same moment (window of 1 second).
    - Packet counting : Focus on counting the number of cells transferred over the lifetime of the circuits and evaluate similarity using distance measurements
    - Cross-correlation : Measure similarity between Entry and Exit traffic by computing the cross-correlation coefficient between the two traffic patterns.

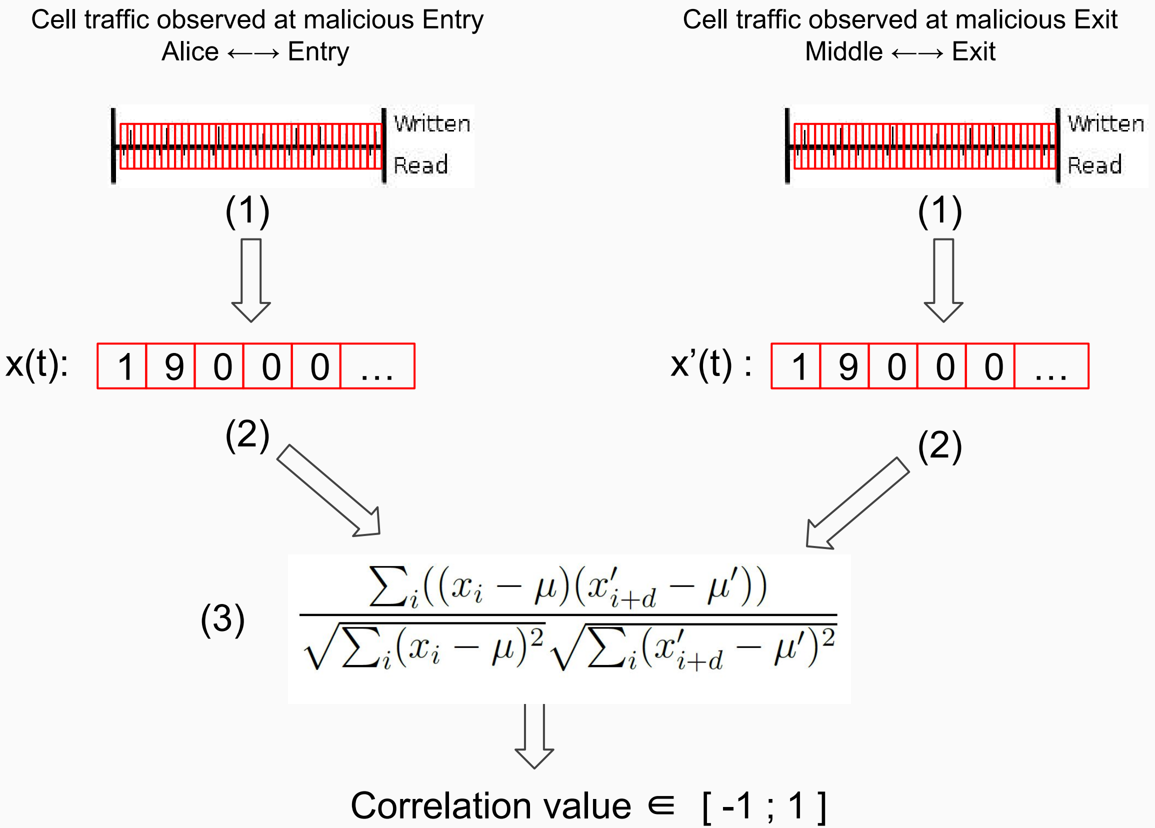
Example of traffic correlation technique :

Cross-correlation coefficient

(1) Slice traffic pattern into windows of one second

(2) Determine how many cell(s) have been sent during each window

(3) Compute correlation value between the two retrieved patterns



Cell traffic observed at malicious Entry
Alice ⟵⟶ Entry

Cell traffic observed at malicious Exit
Middle ⟵⟶ Exit

x(t): | 1 | 9 | 0 | 0 | 0 | … |

x'(t) : | 1 | 9 | 0 | 0 | 0 | … |

$$\frac{\sum_i((x_i-\mu)(x'_{i+d}-\mu'))}{\sqrt{\sum_i(x_i-\mu)^2}\sqrt{\sum_i(x'_{i+d}-\mu')^2}}$$

Correlation value ∈ [-1 ; 1]

Adversary model

- Unrealistic relay-adversary that controls the entire network
- Can observe and record the cell traffic of every circuits at every ORs
- Know which circuits have effectively been used by the clients

→ 2 scenarios :

1. Global Adversary : can differentiate the type of traffic flowing through each circuits
   - ➢ This scenario allows to observe the efficiency of each correlation technique against different types of traffic separately
2. Blind adversary : cannot differentiate the type of traffic flowing through the circuits
   - ➢ Allows to observe the efficiency of the correlation techniques as-if the adversary were performing on the live network

Threat analysis

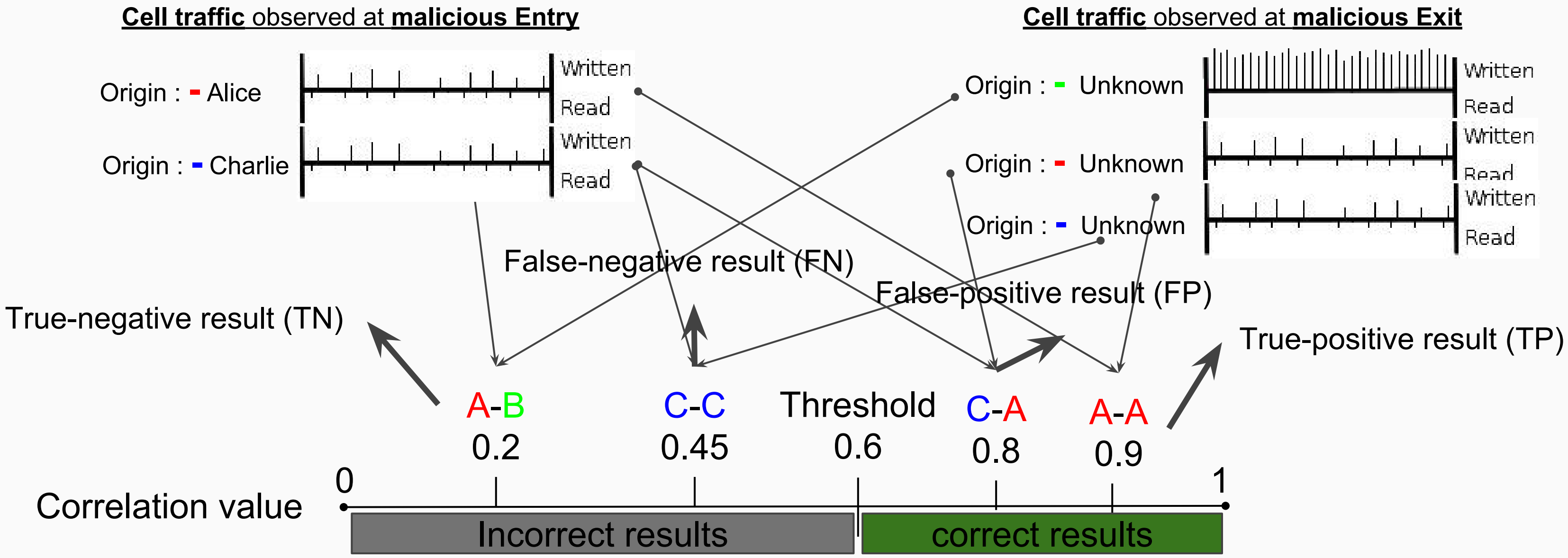
Experimental set-up

- The developed attacks have been set up on reduced-scale private Tor networks simulated with Shadow.
- Shadow is a network simulator that allows to accurately simulates Tor network topologies (Clients, ORs, directory servers, etc…)

→ Implementation of a Shadow pluggin allowing to replay TCP traces over the simulated network → More traffic behaviors can be simulated

→ Tor circuit construction protocol have been altered to unveil circuits anonymity

Evaluation methodology



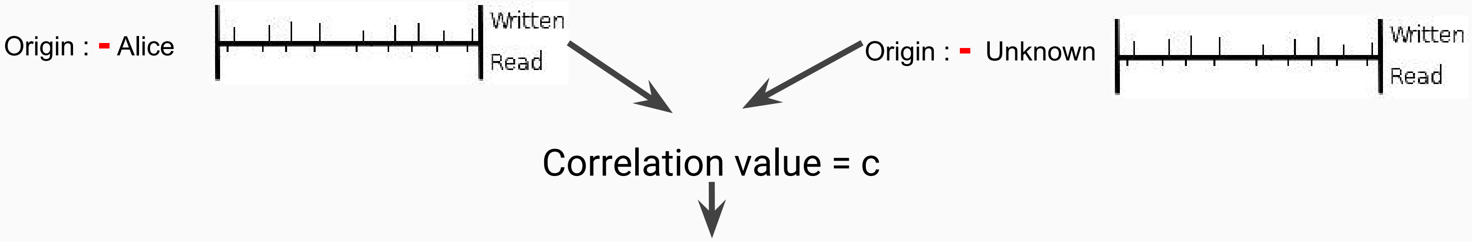
Evaluation methodology (II)
FPR & FNR

- False-positive rate = $FPR = \frac{FP}{FP + TN}$
  - Measures the proportion of false-positives found among all the unrelated circuits (Entry and Exit traffic pattern doesn't belong to the same circuit).
  - A low FPR is desired by the adversary.
- False-negative rate = $FNR = \frac{FN}{FN + TP}$
  - Measures the proportion of traffic patterns that belong to the same circuit and that have not been correctly matched (correlation value < threshold)
  - A low FNR is desired by the adversary.

Probability of matching related circuits

- Probability of correctly matching an Entry (I) and an Exit (J) cell traffic pattern that belong to the same circuit.

$$P(I = J|I \sim J) = \frac{(1 - FNR) * P(I = J)}{(1 - FNR - FPR) * P(I = J) + FPR}$$



Origin : ▬ Alice

Origin : ▬ Unknown

Correlation value = c

If c ≥ threshold :

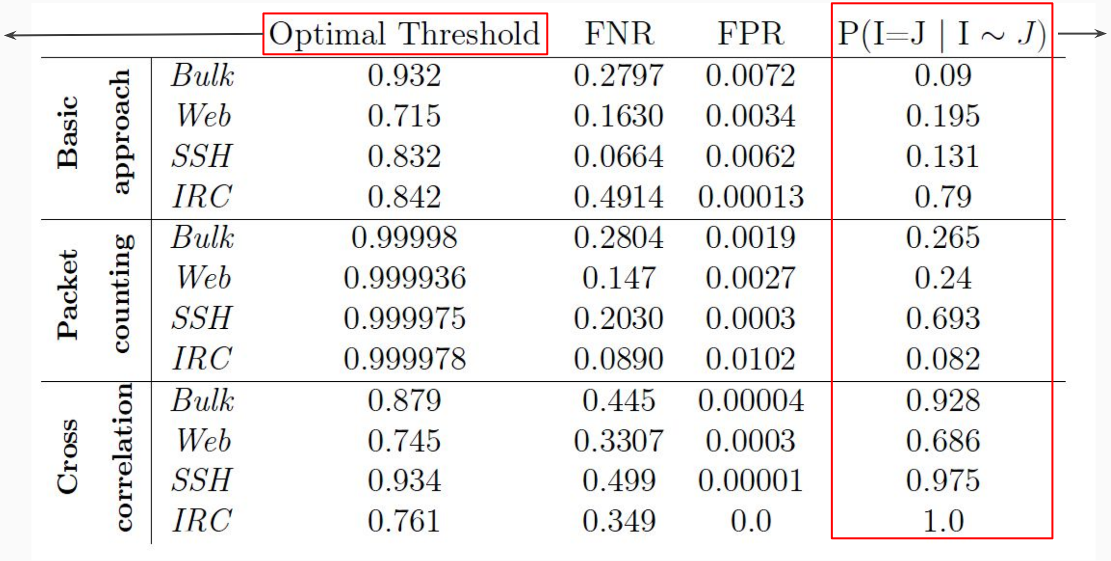
What is the probability that the two traffics belong to the same circuit ?

→ P(I=J | I~J)

- Defined by Levine *et al.* in their paper *Timing attacks in low-latency mix-based systems.*

Global Adversary : results

Threshold that maximizes P(I=J | I~J) for each client type and correlation technique.

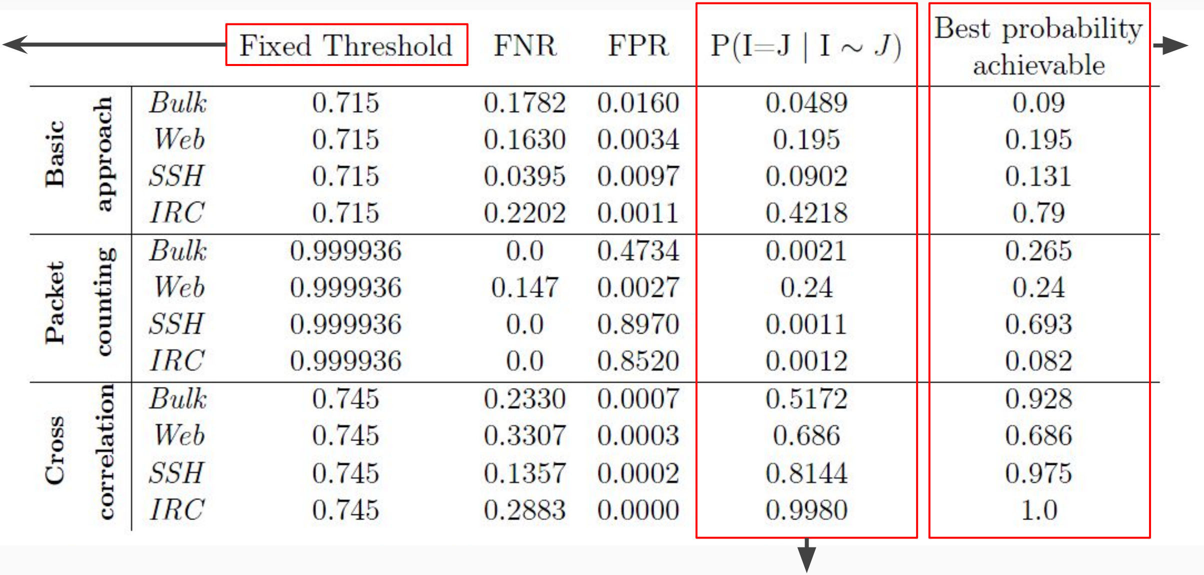


		Optimal Threshold	FNR	FPR	$P(I{=}J \mid I \sim J)$
Basic approach	*Bulk*	0.932	0.2797	0.0072	0.09
	Web	0.715	0.1630	0.0034	0.195
	SSH	0.832	0.0664	0.0062	0.131
	IRC	0.842	0.4914	0.00013	0.79
Packet counting	*Bulk*	0.99998	0.2804	0.0019	0.265
	Web	0.999936	0.147	0.0027	0.24
	SSH	0.999975	0.2030	0.0003	0.693
	IRC	0.999978	0.0890	0.0102	0.082
Cross correlation	*Bulk*	0.879	0.445	0.00004	0.928
	Web	0.745	0.3307	0.0003	0.686
	SSH	0.934	0.499	0.00001	0.975
	IRC	0.761	0.349	0.0	1.0

Probability of correctly matching an Entry (I) and an Exit (J) cell traffic pattern that belong to the same circuit.

Blind Adversary : results

The Threshold is fixed such that it maximizes the probability of correctly matching Web clients.



			Fixed Threshold	FNR	FPR	$P(I=J \mid I \sim J)$	Best probability achievable
Basic	approach	*Bulk*	0.715	0.1782	0.0160	0.0489	0.09
		Web	0.715	0.1630	0.0034	0.195	0.195
		SSH	0.715	0.0395	0.0097	0.0902	0.131
		IRC	0.715	0.2202	0.0011	0.4218	0.79
Packet	counting	*Bulk*	0.999936	0.0	0.4734	0.0021	0.265
		Web	0.999936	0.147	0.0027	0.24	0.24
		SSH	0.999936	0.0	0.8970	0.0011	0.693
		IRC	0.999936	0.0	0.8520	0.0012	0.082
Cross	correlation	*Bulk*	0.745	0.2330	0.0007	0.5172	0.928
		Web	0.745	0.3307	0.0003	0.686	0.686
		SSH	0.745	0.1357	0.0002	0.8144	0.975
		IRC	0.745	0.2883	0.0000	0.9980	1.0

Probability of correctly matching an Entry (I) and an Exit (J) cell traffic pattern that belong to the same circuit when an optimal threshold is fixed for each client type and correlation technique.

Probability of correctly matching an Entry (I) and an Exit (J) cell traffic pattern that belong to the same circuit when the threshold is fixed to maximize the probability of matching web clients.

Conclusion

Lessons learned

- Basic approach and packet-counting techniques are not suitable for a blind adversary
- Cross-correlation technique is the most effective technique :
  - ~ 3x better than basic approach and packet-counting.
  - When fixing a unique threshold that maximizes the chance of catching Web clients :
    - IRC and SSH traffic are still highly vulnerable.
    - Bulk traffic is harder to correlate but still vulnerable (more FP -> 2x less efficient).

→ All kind of traffic is at risk.

Questions & answers

Backup slides

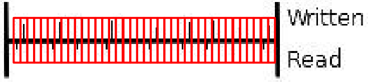
Traffic correlation technique : Basic approach

(1) Slice traffic pattern into windows of one second

(2) Determine when the cell were sent through the circuit

(3) Compute correlation value between the two pattern

Cell traffic observed at malicious Entry
Alice ⟷ Entry

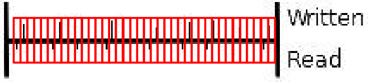


(1)

E(t) :	1	1	0	0	0	…

(2)

Cell traffic observed at malicious Exit
Middle ⟷ Exit



(1)

E’(t) :	1	1	0	0	0	…

(2)

(3) $$\frac{\sum E(t) \times E'(t)}{\sum E(t)}$$

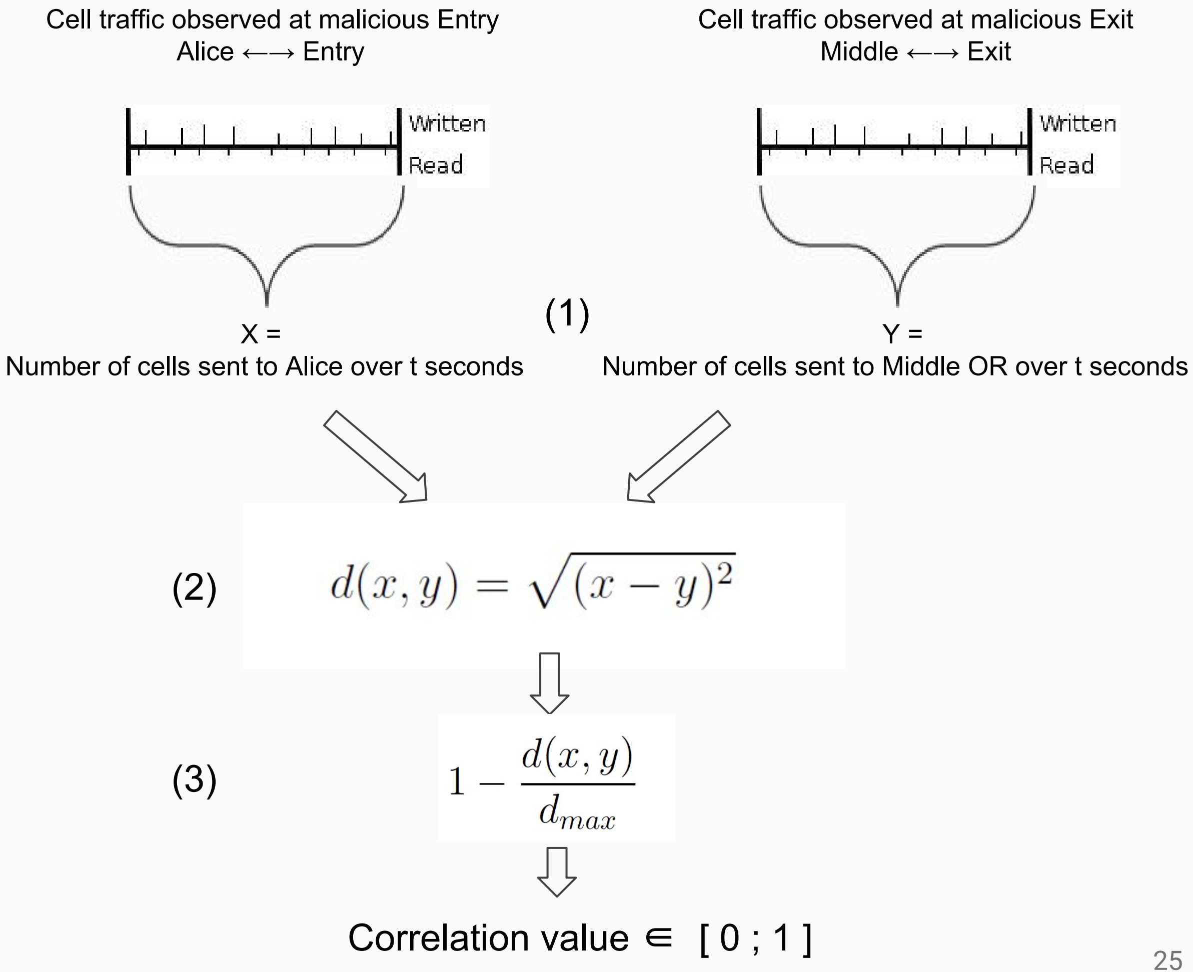
Correlation value ∈ [0 ; 1]

Traffic correlation technique : Packet-counting

(1) Count the number of cells sent over the circuit over period of time

(2) Compute the distance between the two counts

[(3) Normalize measurements with max distance observed among all correlated circuits]



Cell traffic observed at malicious Entry
Alice ⟵⟶ Entry

Cell traffic observed at malicious Exit
Middle ⟵⟶ Exit

(1)

X =
Number of cells sent to Alice over t seconds

Y =
Number of cells sent to Middle OR over t seconds

(2)

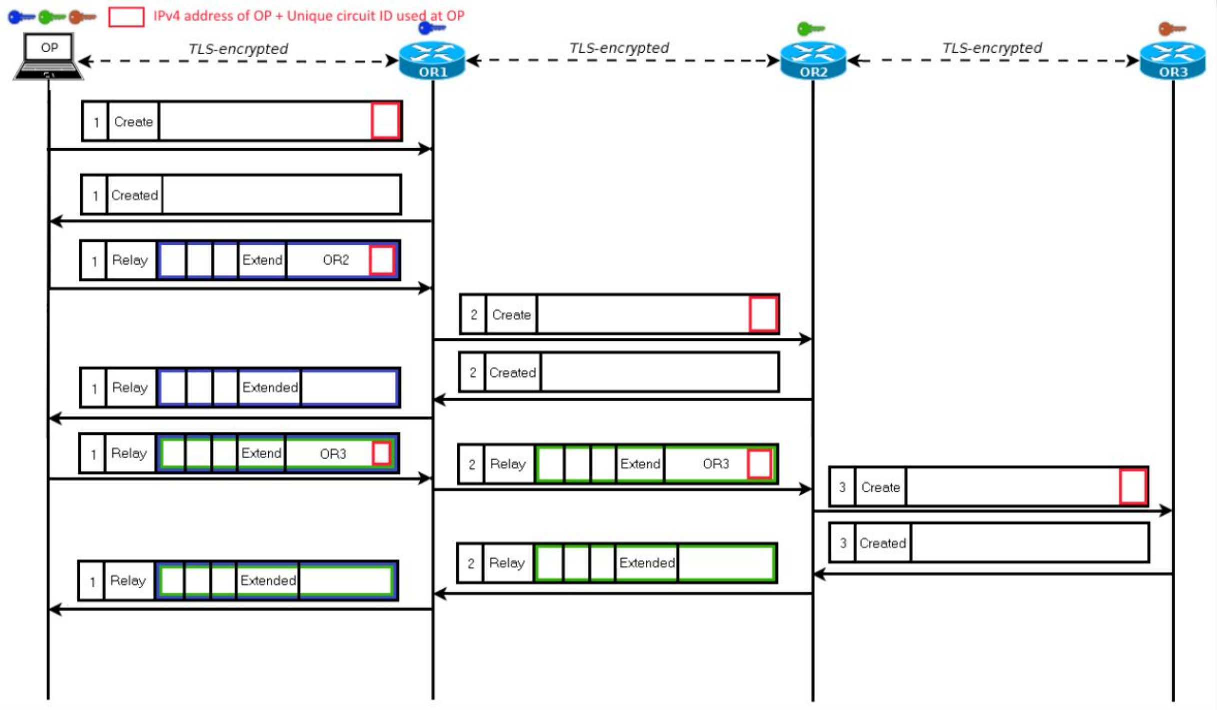
$$d(x,y) = \sqrt{(x-y)^2}$$

(3)

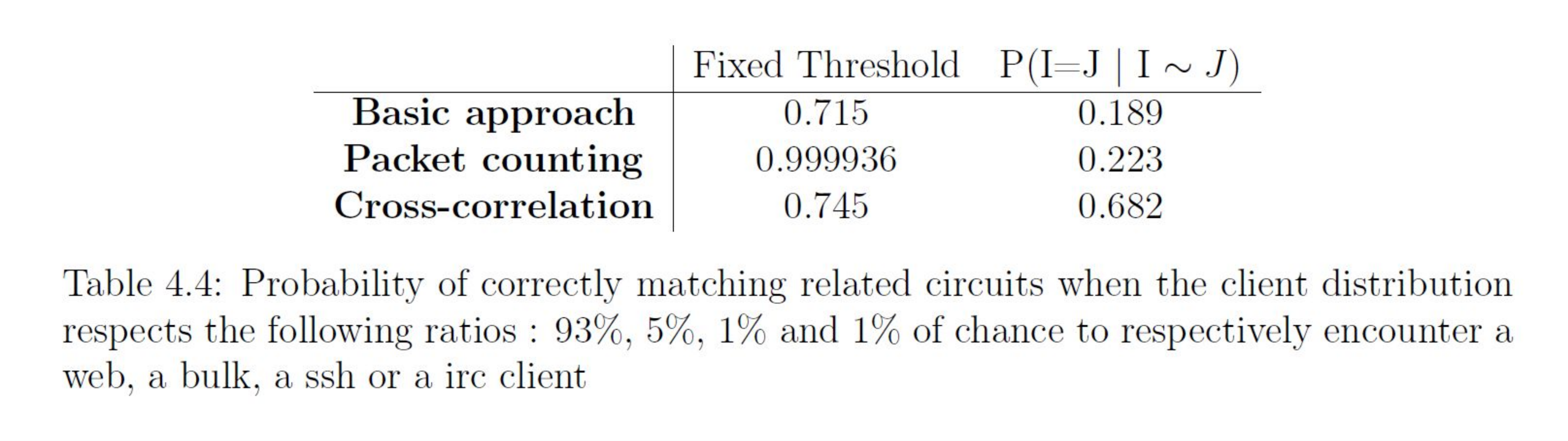
$$1 - \frac{d(x,y)}{d_{max}}$$

Correlation value ∈ [0 ; 1]

Circuit mapping



Probability of correctly matching Entry and Exit traffic patterns when threshold is uniquely fixed for all clients

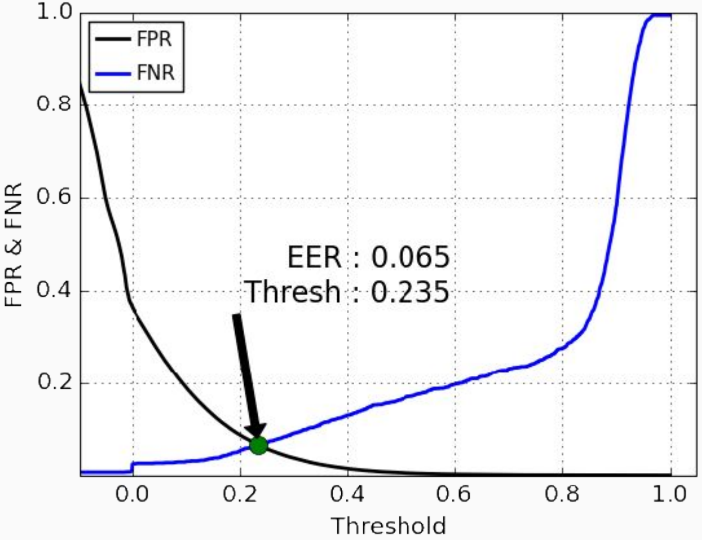


	Fixed Threshold	$\mathrm{P(I{=}J} \mid \mathrm{I} \sim J)$
Basic approach	0.715	0.189
Packet counting	0.999936	0.223
Cross-correlation	0.745	0.682

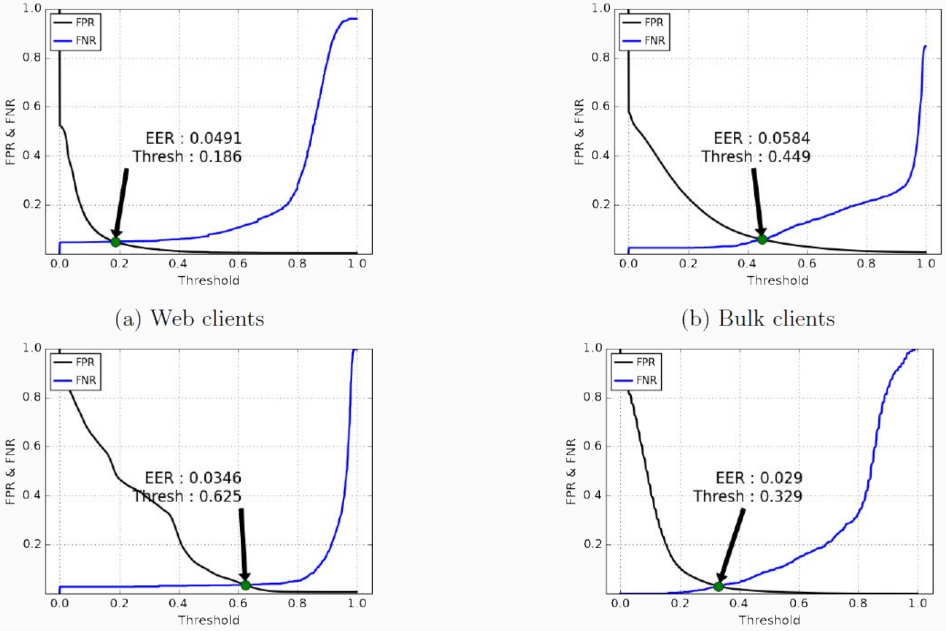
Table 4.4: Probability of correctly matching related circuits when the client distribution respects the following ratios : 93%, 5%, 1% and 1% of chance to respectively encounter a web, a bulk, a ssh or a irc client

Evaluation methodology
FPR & FNR

- FPR and FNR obtained with the cross-correlation technique for the bulk traffic:



Results: basic approach



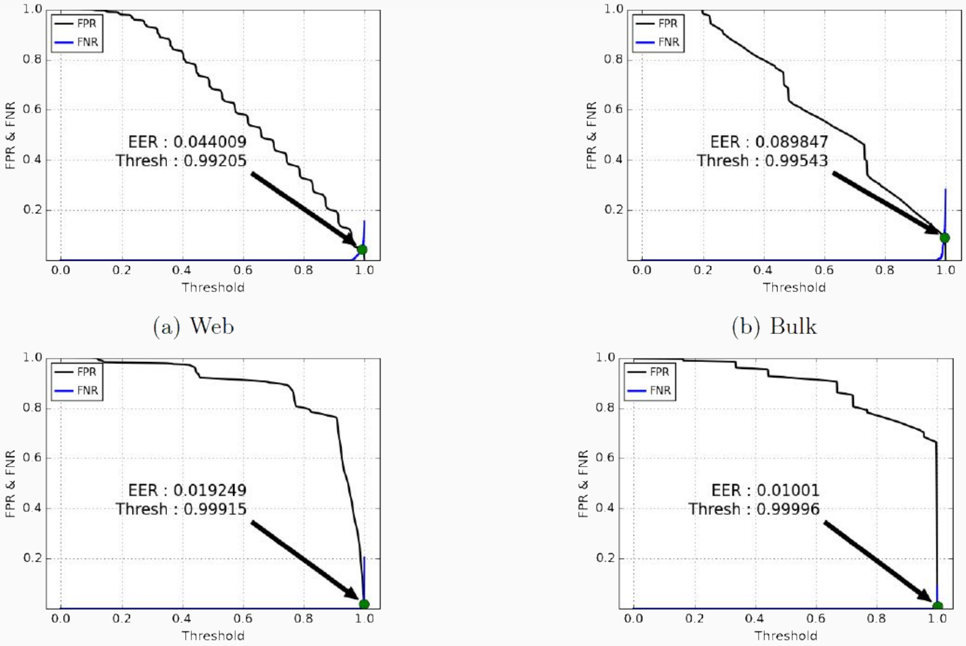
(a) Web clients

(b) Bulk clients

(c) Ssh clients

(d) Irc clients

Results: packet counting



EER : 0.044009
Thresh : 0.99205

(a) Web

EER : 0.089847
Thresh : 0.99543

(b) Bulk

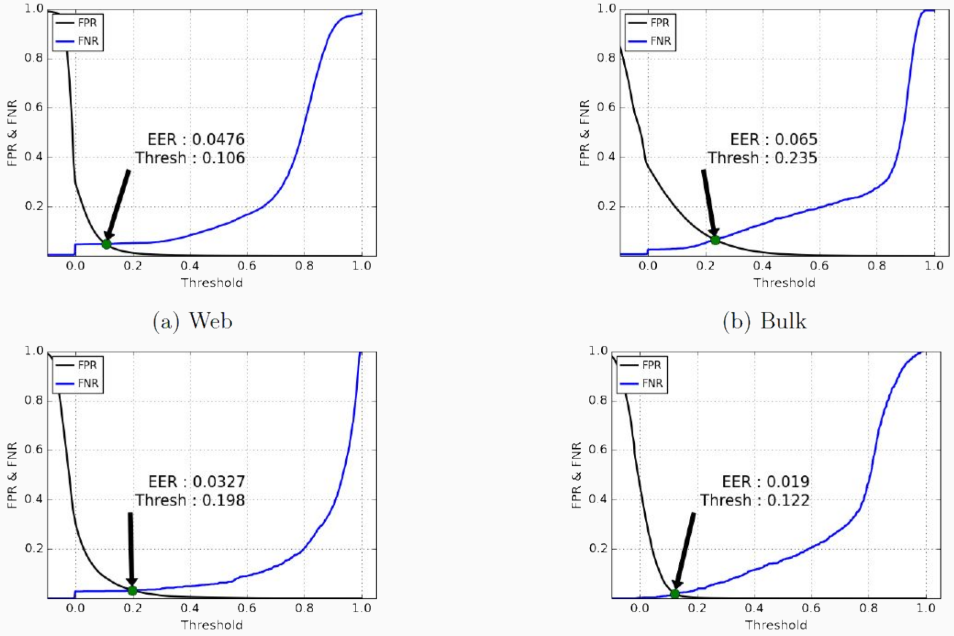
EER : 0.019249
Thresh : 0.99915

(c) SSH

EER : 0.01001
Thresh : 0.99996

(d) Irc

Results: cross-correlation



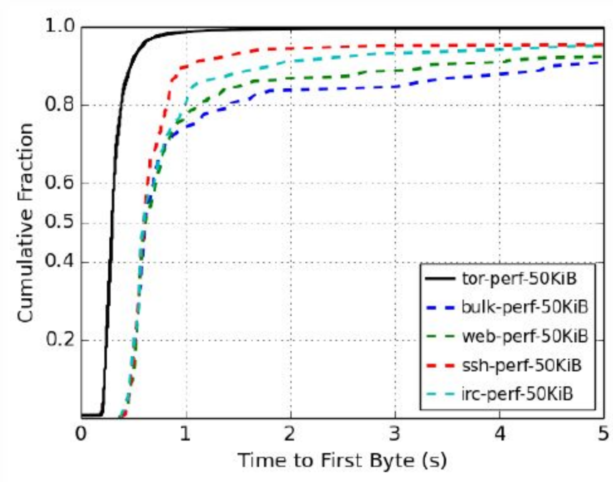
(a) Web

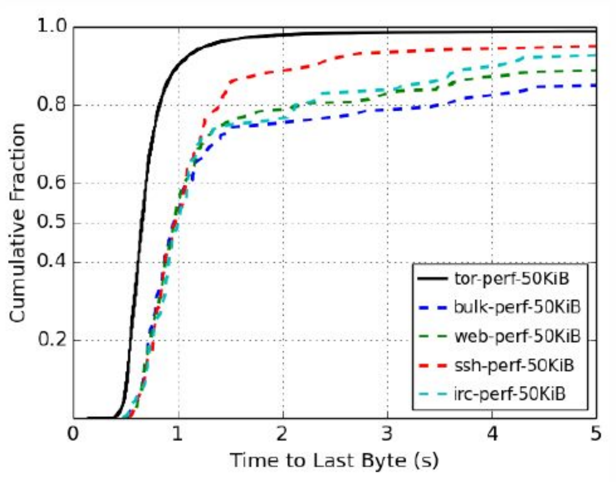
(b) Bulk

(c) SSH

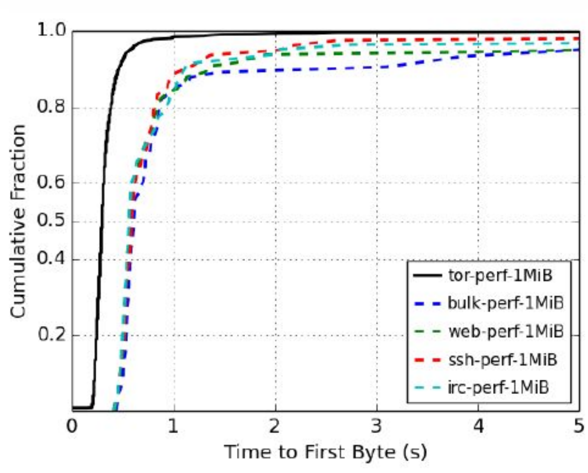
(d) Irc

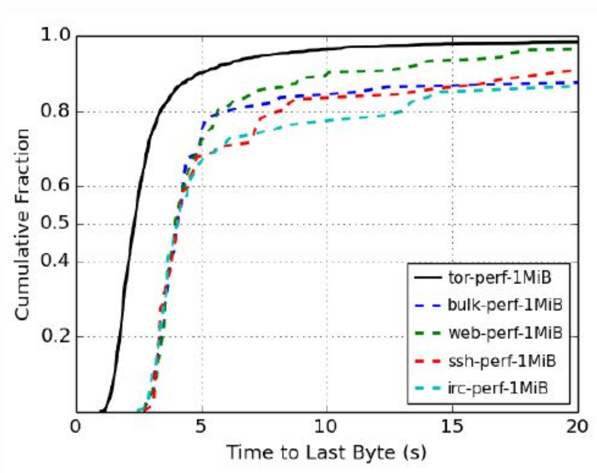
Time to first byte & time to last byte (I)
50 KiB



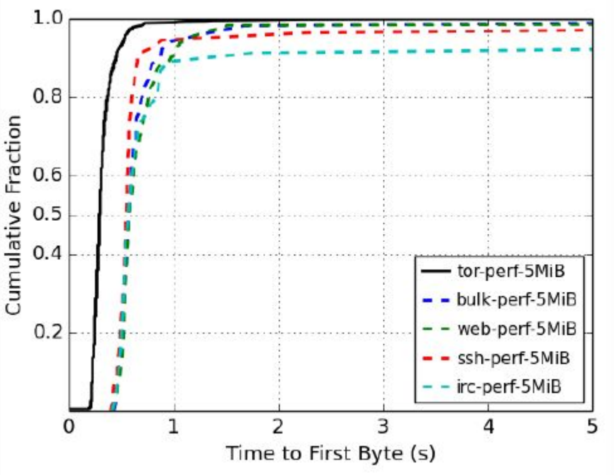


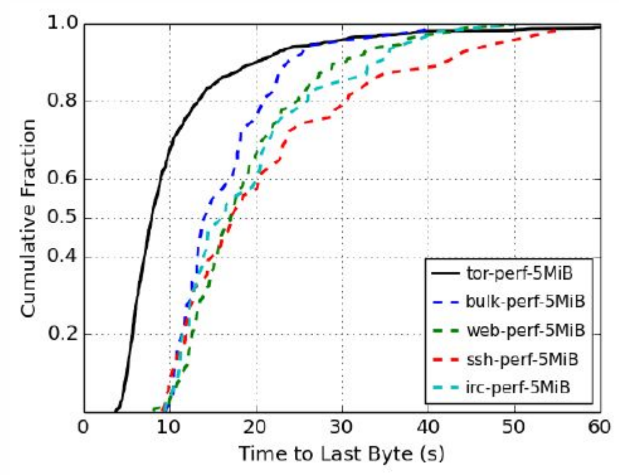
Time to first byte & time to last byte (II)
1 MiB





Time to first byte & time to last byte (III)
5 MiB





Global Adversary

- Unrealistic adversary model.
- The adversary monitors each onion routers.
- She has some prior knowledge about :
  - Type of traffic flowing through a circuit.
  - Time when a circuit is first used.
  - Cell traffic patterns of each circuit on each OR.

→ Make use of those information to perform an efficient end-to-end correlation attack.

→ Determine best achievable performance per client types and correlation techniques

Blind Adversary

- As-if the adversary operates on the live Tor network
- No information about the type of traffic flowing through the circuits

→ Should fix a unique threshold for all kind of clients ! How ?

→ 95% of the traffic exiting Tor is assigned to Web surfing …

→ Then, adversary should fix the threshold such that it maximizes its chance of correctly matching Web client circuits.

Experimental set-up (II)

- Three-step attack:
  - Step 1: make use of the Tor controller interface to generate logs containing relevant information about the circuit.
    - The amount and type of cell transferred every second per circuit.
  - Step 2: production of results via a Python script in charge of :
    - extracting the data saved in the Tor controller logs to generate the pattern of each circuit.
    - Compute the correlation between each entry traffic pattern and all exit traffic patterns.
  - Step 3:
    - Make the results for a given threshold