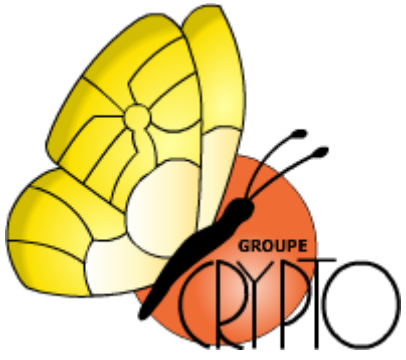
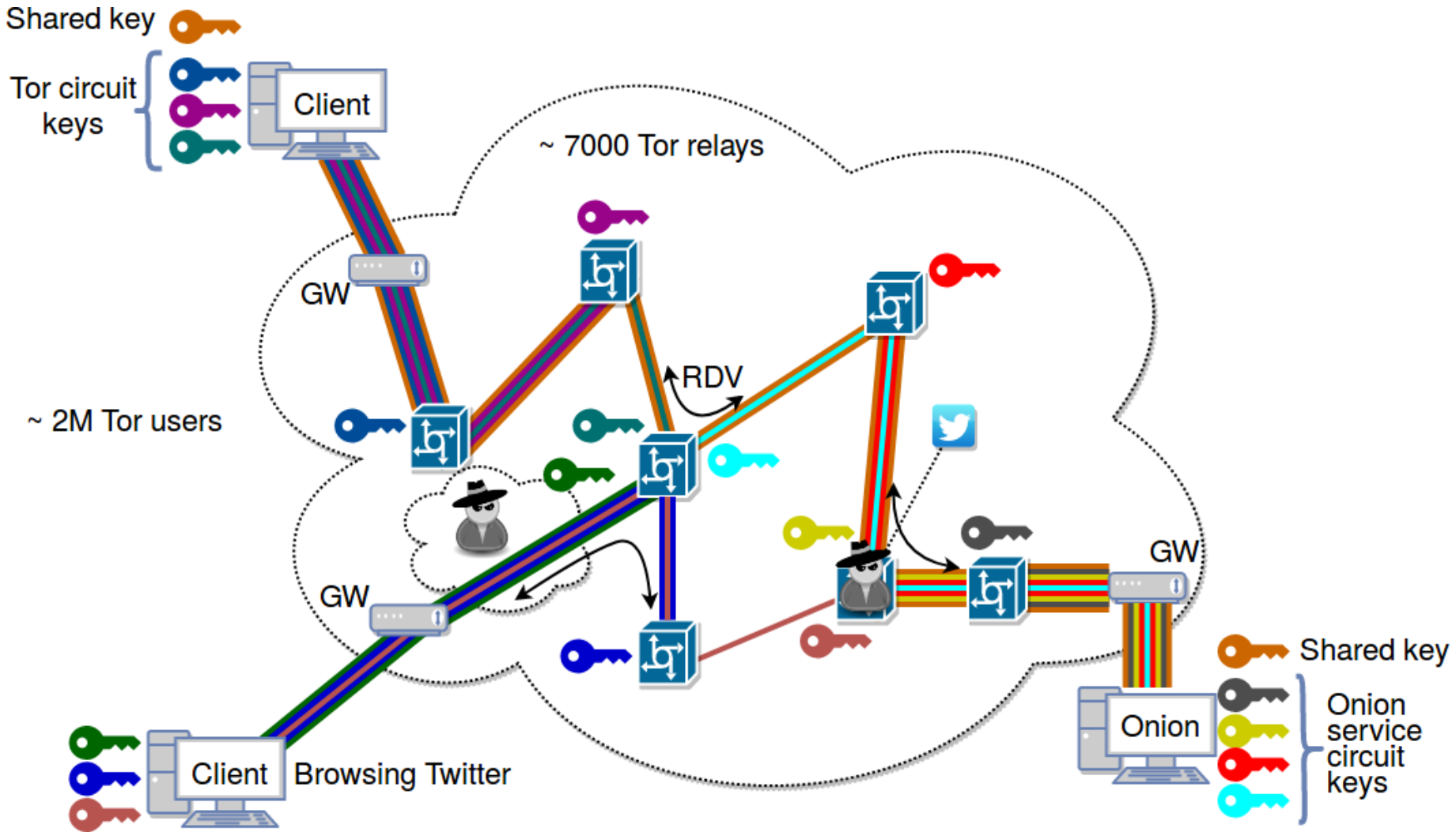
Dropping on the Edge: Flexibility and Traffic Confirmation in Onion Routing Protocols

Florentin Rochet and Olivier Pereira

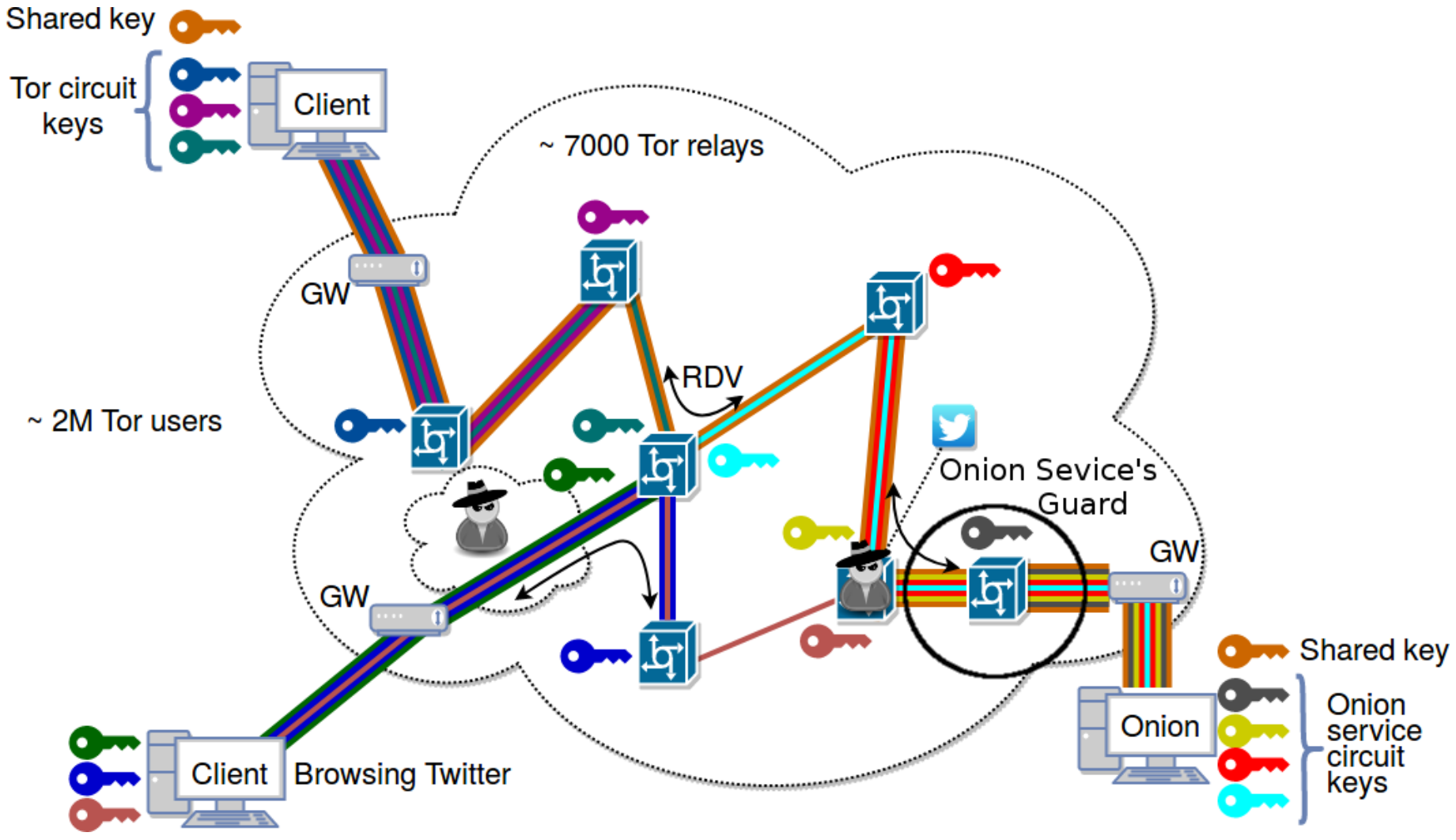
UCLouvain Crypto Group, Belgium



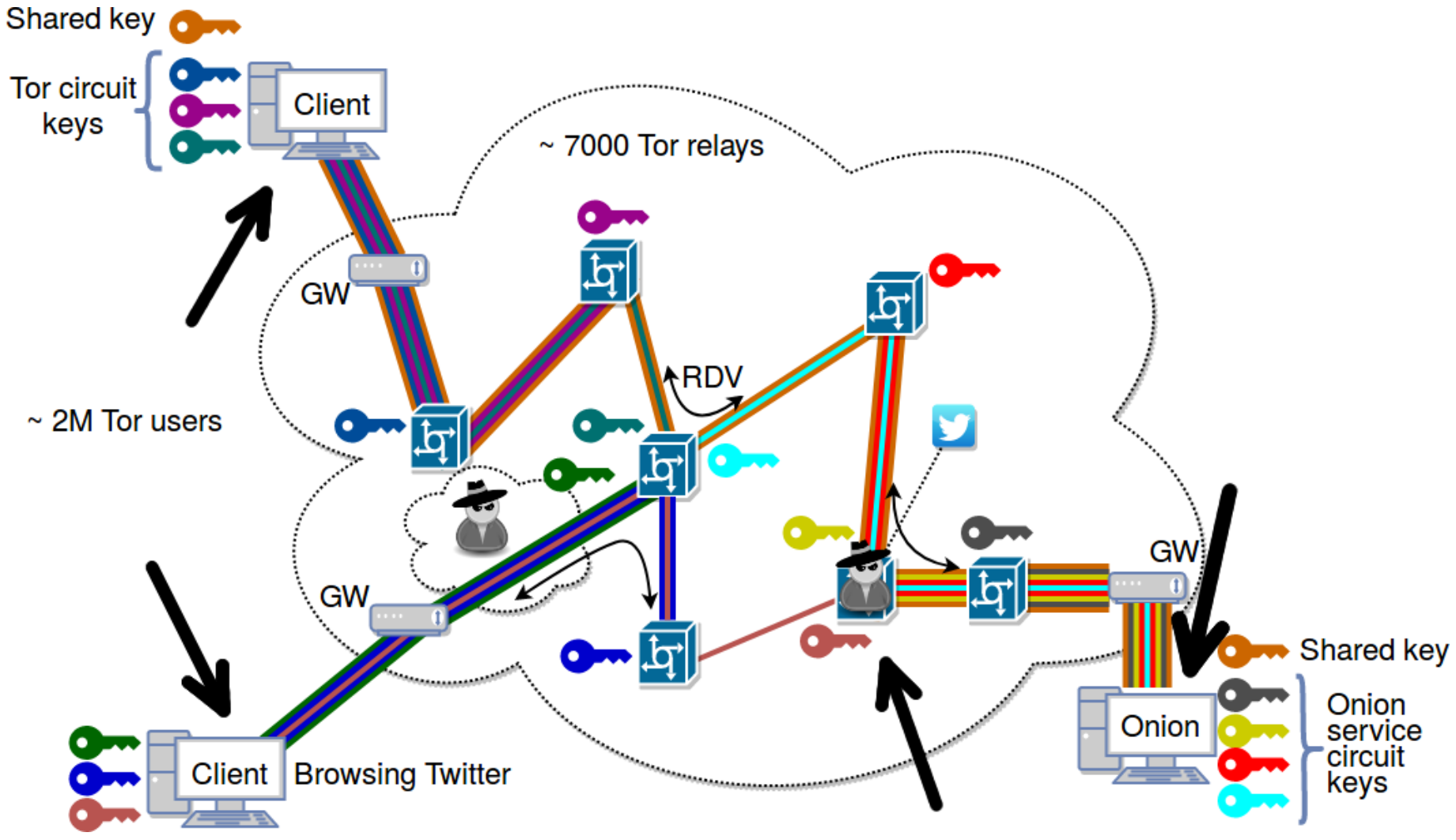
Overview


Shared key
Tor circuit keys
Client
~ 7000 Tor relays
GW
RDV
~ 2M Tor users
GW
GW
Shared key
Onion
Onion service circuit keys
Client
Browsing Twitter

Overview


Shared key
Tor circuit keys
Client
~ 7000 Tor relays
GW
RDV
~ 2M Tor users
Onion Sevice's Guard
GW
GW
Shared key
Onion
Onion service circuit keys
Client
Browsing Twitter

Overview


Shared key
Tor circuit keys
Client
~ 7000 Tor relays
GW
RDV
~ 2M Tor users
GW
GW
Shared key
Onion
Onion service circuit keys
Client
Browsing Twitter

Architectural Principle of the Internet

- Robustness principle (among others):

> *Be conservative in what you do, be liberal in what you accept from others. [RFC761], [RFC1122], [RFC1958]*

- Can lead to strong attacks in deployed anonymity systems

Forward compatibility

- Allows compliance to future version of the protocol.

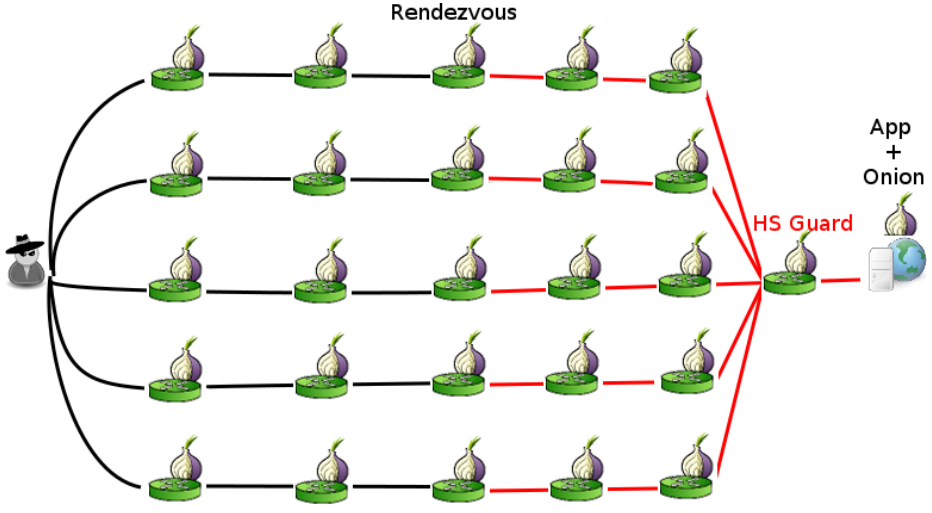
```
static int connection_edge_process_relay_cell(cell_t *cell, ...) {
  ...
  switch(rh.command) {
    case RELAY_COMMAND_DROP: return 0; // do nothing.
    ...
    case RELAY_COMMAND_DATA:
      //process data
      ...
      return 0;
    ...
  }
  log_fn(LOG_PROTOCOL_WARN, LD_PROTOCOL, "Unrecognized command %d", rh.command);
  return 0; /* for forward compatibility, don't kill the circuit */
}
```

Outline

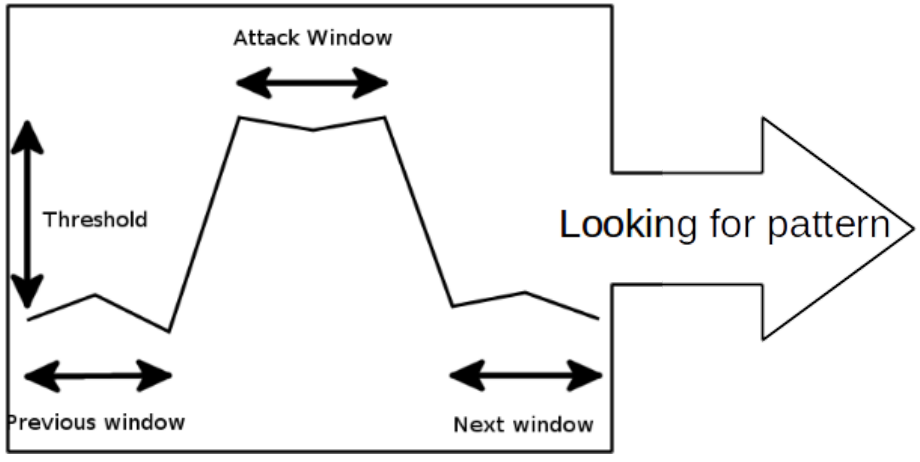
1. Guard Discovery Attack
    - Uses forward compatibility, a path selection trick and a side-channel
2. Dropmark attack
    - An active traffic confirmation attack with interesting properties

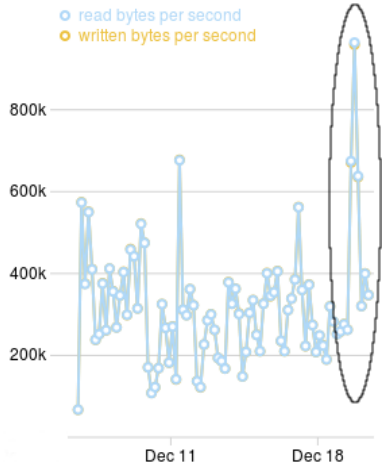
Guard discovery

- Combines a path selection trick, forward compatibility and a side-channel.



Guard discovery





- The public report of bandwidth consumption acts as a side-channel
- What is the probability of success in the wild?

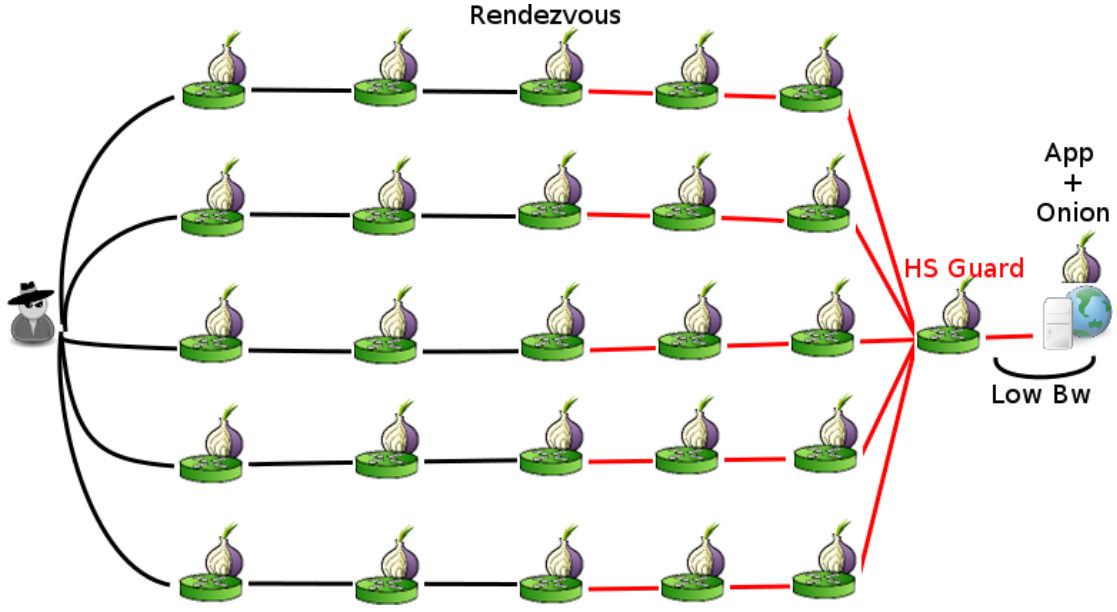
Assumptions

- Consumed bandwidth of the targeted guard is always higher during the attack
  - Silently dropped traffic strengthens this assumption
- Less variance in public measurements during the attack
- Graceful behaviour of the relay
- Graceful behaviour of the relay operator

Given those assumptions, we can use the history of the network to evaluate the attack

Let's investigate different situations

Onion Service's BW << Guard's BW



Triggering the OOM killer algorithm

- One day guard discovery attack with a bug exploit
- We fill the memory of the guard relay
- Can trigger easily the OOM algorithm which generates a counter bug (public information)

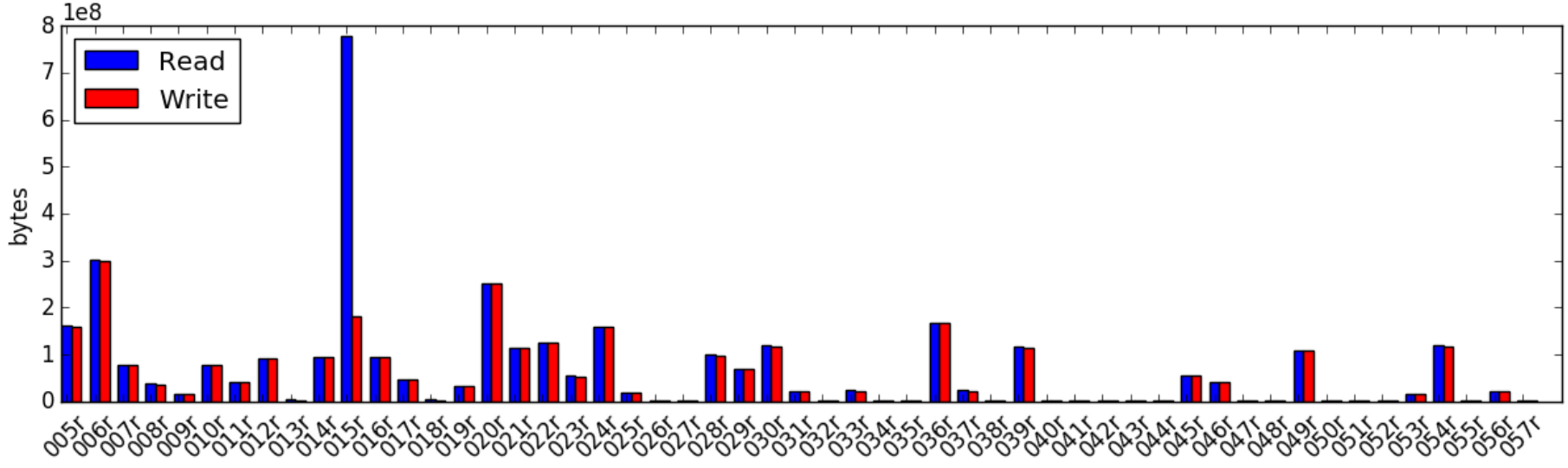
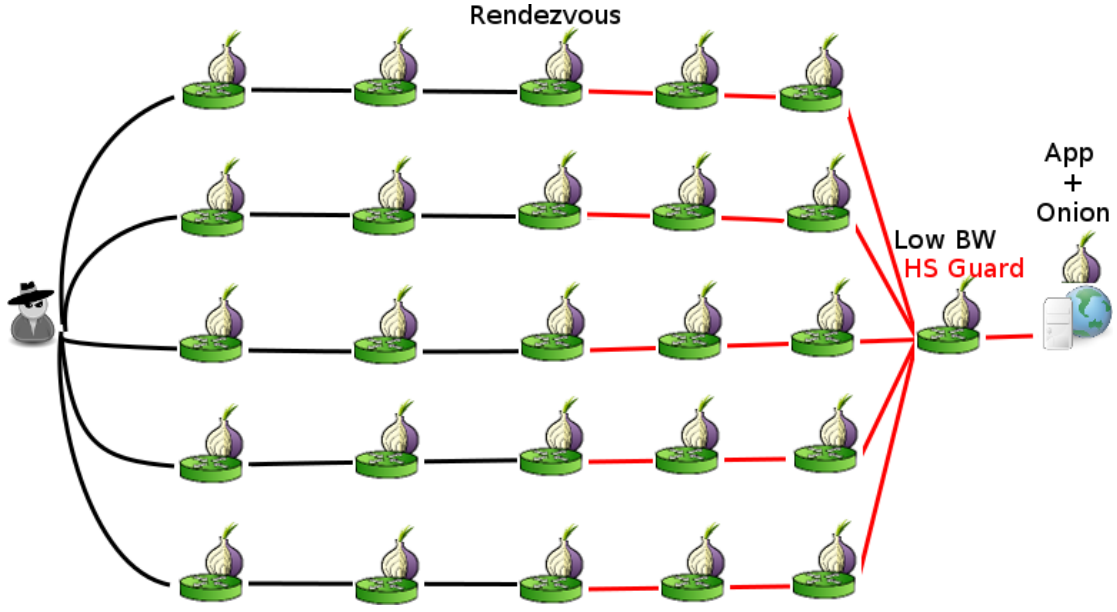


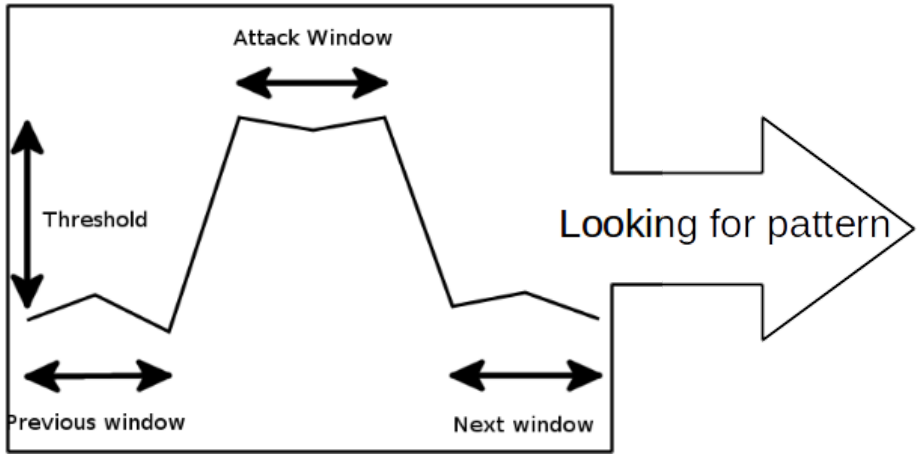
Figure: Chutney experiment triggering the OOM killer algorithm of the onion service's guard

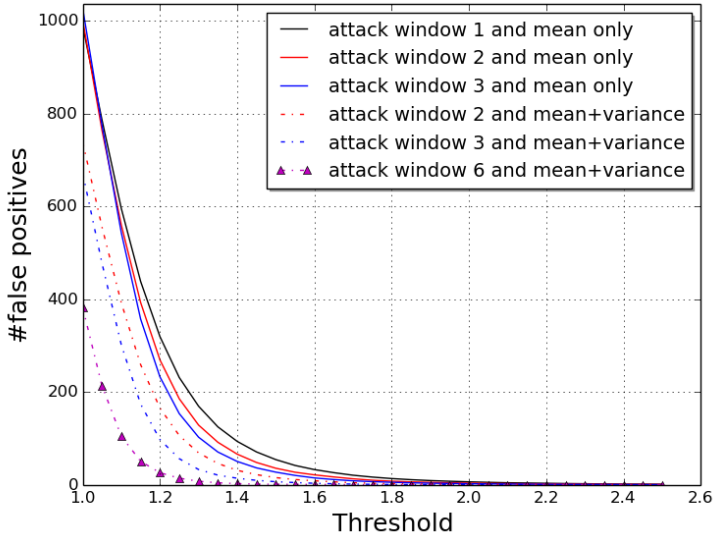
Let's investigate different situations

Onion Service's BW >> Guard's BW

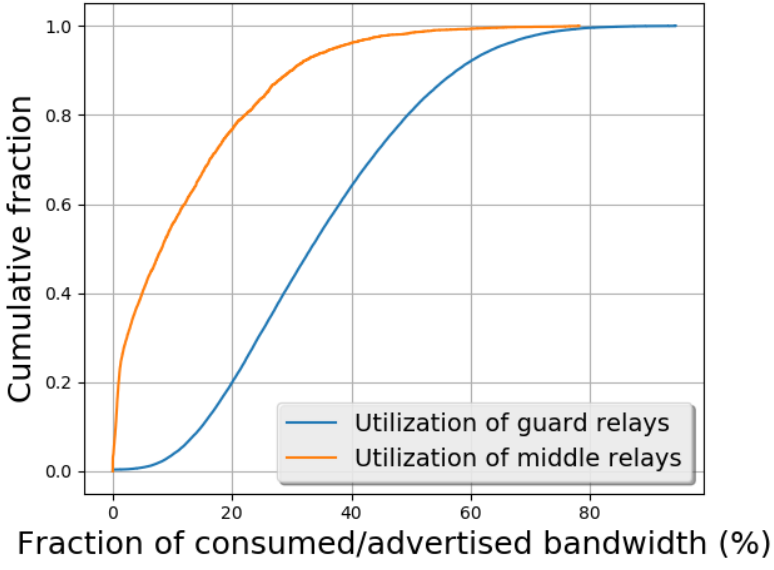


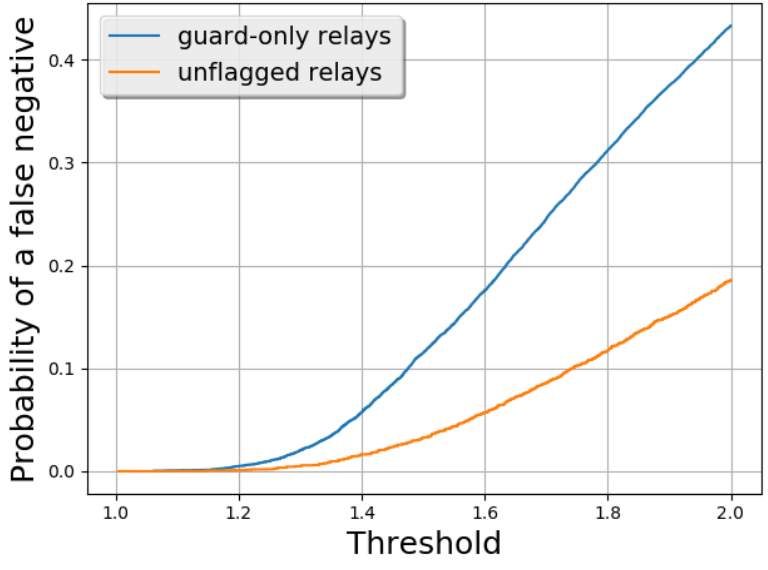
Evaluation


Attack Window
Threshold
Looking for pattern
Previous window
Next window


attack window 1 and mean only
attack window 2 and mean only
attack window 3 and mean only
attack window 2 and mean+variance
attack window 3 and mean+variance
attack window 6 and mean+variance
#false positives
Threshold
0
200
400
600
800
1000
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
2.6

Evaluation

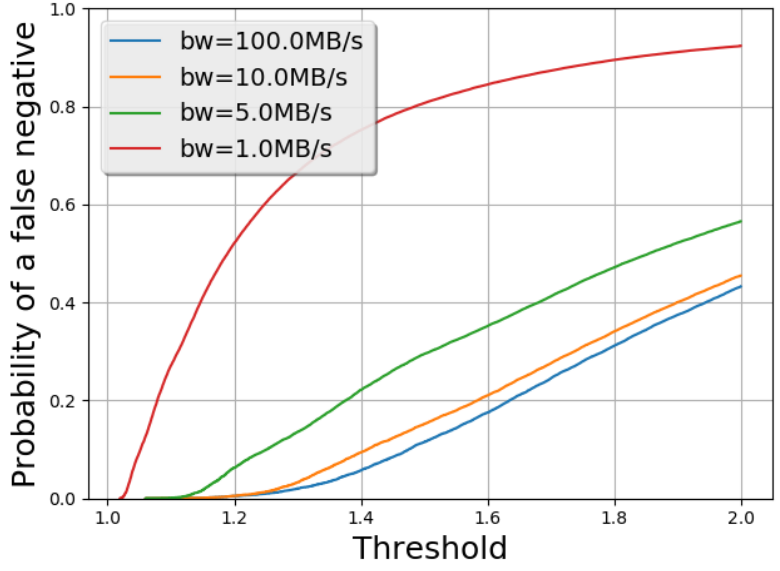




- Evaluating spare resources of guards
- The attack would be inneficient if the Onion service guard is already overloaded

Evaluation

- With the assumption that the counter exploit is fixed



- $\approx 96\%$ success rate to retrieve only one (correct) guard in a few days, against a few MB/s onion service
- This attack cost less than a sandwich

Countermeasures

- Multiple suggestions to counter the guard discovery
  - The Tor Project chose to perform a **volume analysis**, and to **increase the bandwidth reporting interval**
  - Onion service operators: decreasing the available bandwidth reduces the risk (once the counter exploit is solved)
  - See @mikeperry-tor/vanguards on GitHub for mitigations

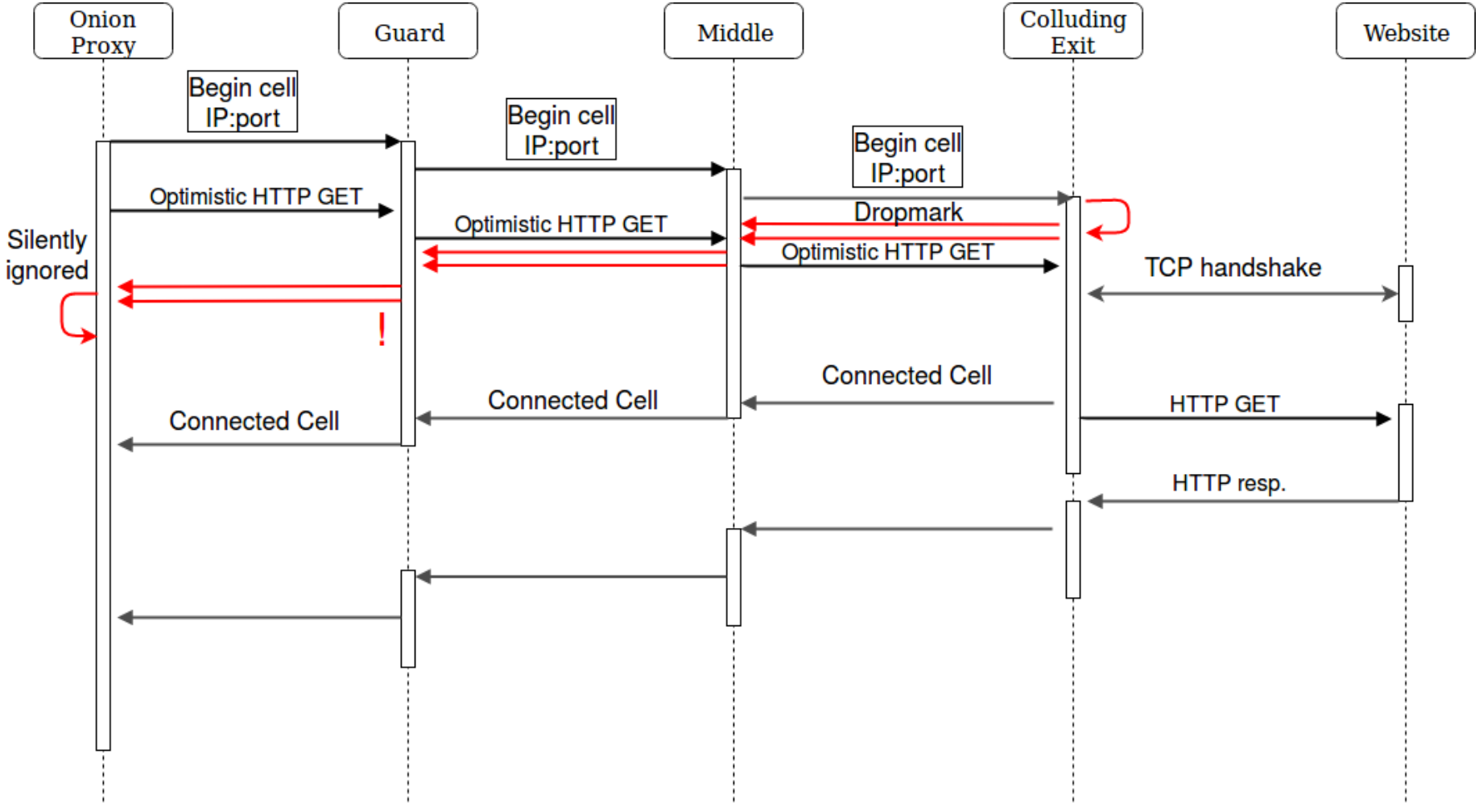
Outline

1. Guard Discovery Attack
    - Uses forward compatibility, a path selection trick and a side-channel
2. **Dropmark attack**
    - **An active traffic confirmation attack with interesting properties**

Dropmark attack

- Active end-to-end correlation attack with interesting properties
  - Does not need the victim to transfer any packet to succeed
  - The application level traffic does not influence the success rate
- Uses forward compatibility and a side-channel
- Assumes colluding exit and guard (or network observer on client-guard)
- Implemented and tested in Shadow with $\approx 99.86\%$ TPR and $\approx 0.03\%$ FPR
- Can be applied in many different scenarios

Dropmark attack



Contributions/Conclusion

- Contributions:
  - Identification of potential weaknesses resulting from Tor's forward compatibility
  - New guard discovery attack
  - New traffic confirmation attack with intriguing properties
  - Many more attacks out there to hunt ...
  - Implementations and tutorial to reproduce our results available on GitHub
- Discussion
  - Removing forward compatibility?
    - Complicates the integration on novel ideas
    - May reduce the Tor network diversity or slow down deployment of new versions
    - Increases code complexity
    - May not solve the problem ...