Hot or Not:

Revealing Hidden Services by their Clock Skew

Author : Stephen Murdoch

Presented at the ACM conference on Computer and Communications Security '06

Presented by : Naman Agarwal

Motivation

- Systems such as Tor offer location hidden services
- Anonymous services help protect the owner allowing for censorship resistant content. Also helps prevent selective DoS attacks.
- Due to the credible threat it is important to evaluate the security not only of deployed systems as well as proposed changes.
- The paper presents a potential attack on anonymity systems based on measuring clock skew.

Assumptions on the attacker

- The main goal of the attacker is to figure out the IP address of the operator
- The attacker is not assumed to be part of the anonymity system but can access the hidden services exposed by it.
- The attacker has a list of a limited number of candidate hosts for a hidden service.
- The attacker cannot observe, inject, delete or modify any network traffic other than that to or fro from his computer

Existing Attacks on Tor

- Tor is an overlay network and therefore machines can be accessed over the anonymous channel as well as directly

- This makes Tor susceptible to attacks based on the analysis of traffic patterns.

- The attacker induces traffic patterns in the network and then probes the latency of possible intermediate nodes looking for correlations. <Add reference here>

- Such attacks can be prevented by establishing a QoS guarantee where every stream passing through a node is essentially isolated from another

- Essentially every Tor node has a given capacity which is divided into several slots. Each circuit is assigned one slot and is given a guaranteed a data rate regardless of others

Clock Skew based attacks

- The key observation behind such attacks is that when circuits carried by a node become idle, the CPU load reduces and the temperature reduces.
- This has a measurable effect on a quantity called clock skew and can be observed remotely.
- Thus an attacker can distinguish between a busy vs an idle CPU.

Background on Clock Skew

- Lets first fix a reference frame for time. For the attack's purpose we will think of the clock with the adversary as our reference for time.
- A clock C is designed to count the time elapsed since some initial time i(C)
- Clock C's *resolution*, r(C), is the smallest unit by which the clock can be incremented, and we refer to each such increment as a *tick.* The inverse of such an increment is called the intended *frequency h(C)*
- A resolution of 10 ms means that the clock is designed to have 10 ms granularity, not that the clock is always incremented *exactly* every 10 ms.
- This induces an *offset o(t)* defined as the difference between the clock's reported time and the actual time (t). The skew *s(t)* is the derivative of *o(t)* at time t.
- We split the skew of the target machine into two components s_c which represents a constant upper bound on the skew and a small time varying component *s(t)* which is assumed wlog to be negative.

Timestamps

- To measure the time of a remote machine the authors make use of TCP timestamps by establishing direct TCP connections with the machine.

- Define $T(t_S)$ to be the timestamp sent at time t_S.

- The timestamp sent is given by

$$T(t_s) = \left\lfloor h \cdot \left(t_s + s_c t_s + \int_0^{t_s} s(t)dt \right) \right\rfloor$$

Timestamps (Cntd.)

- We sample timestamps T_i by continuously choosing a random time between ticks. Therefore the quantization noise due to the floor can be captured by subtracting a random variable *c* uniform in [0,1].

- The time when the remote machine sent the sample according to the remote machine is given by

$$\tilde{t}_i = T_i/h = t_{s_i} + s_c t_{s_i} + \int_0^{t_{s_i}} s(t)dt - c_i/h$$

Offset Computation

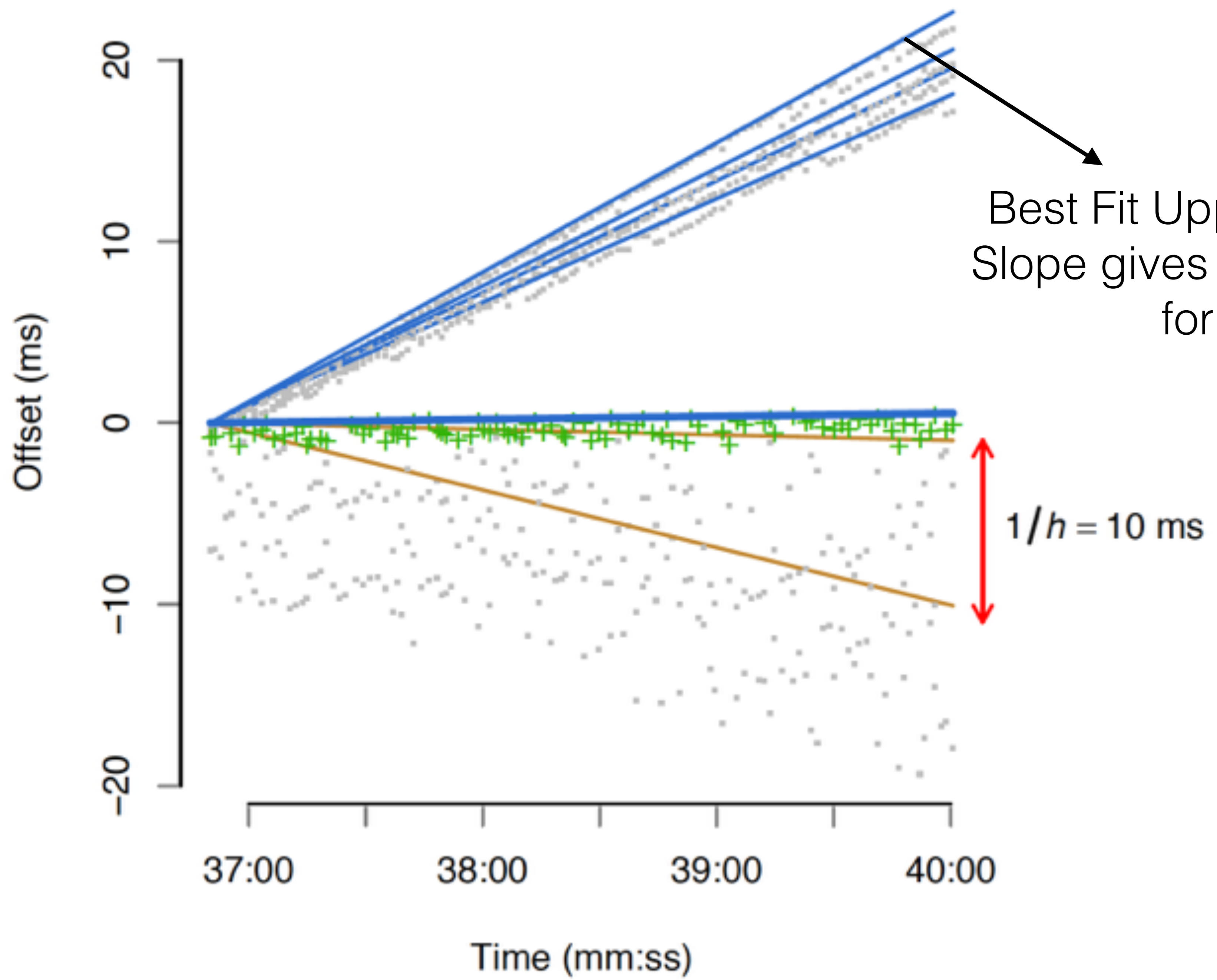
- We cannot compute the sending machine's clock skew but we can compute the offset o(i) between the timestamped time according to the remote machine and our reference. This will be given by

$$\tilde{t_i} - t_{s_i}$$

- However we only have the time when the packet was received. Therefore we need to factor in noise due to latency d_i. The expression for the offset finally looks like

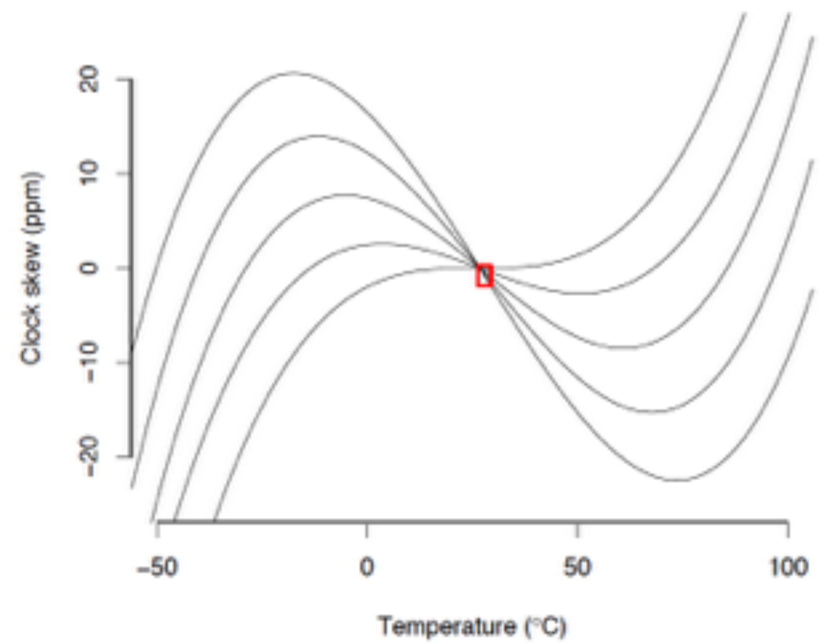
$$o_i = \tilde{t_i} - t_{r_i} = s_c t_{r_i} + \int_0^{t_{r_i}} s(t)dt - c_i/h - d_i$$

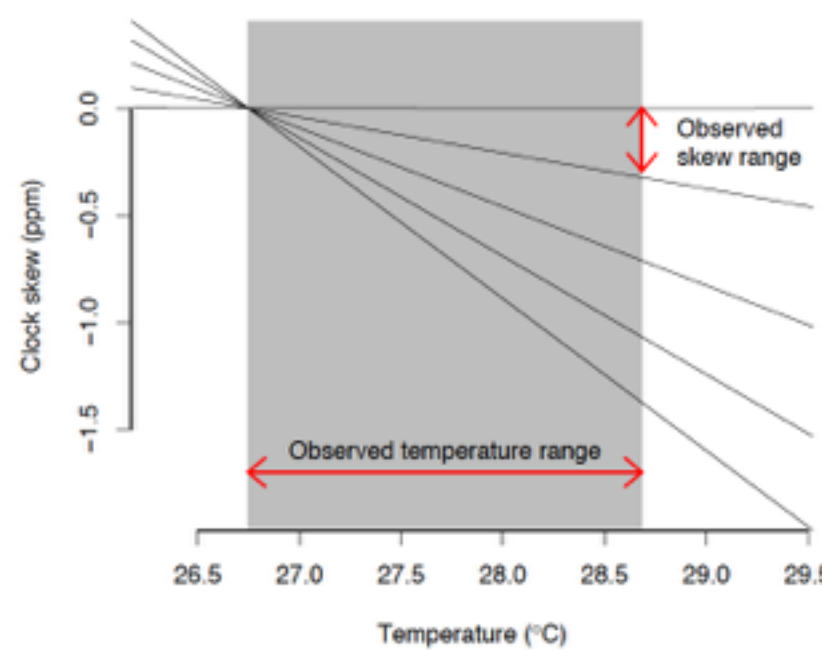
Computing the constant skew



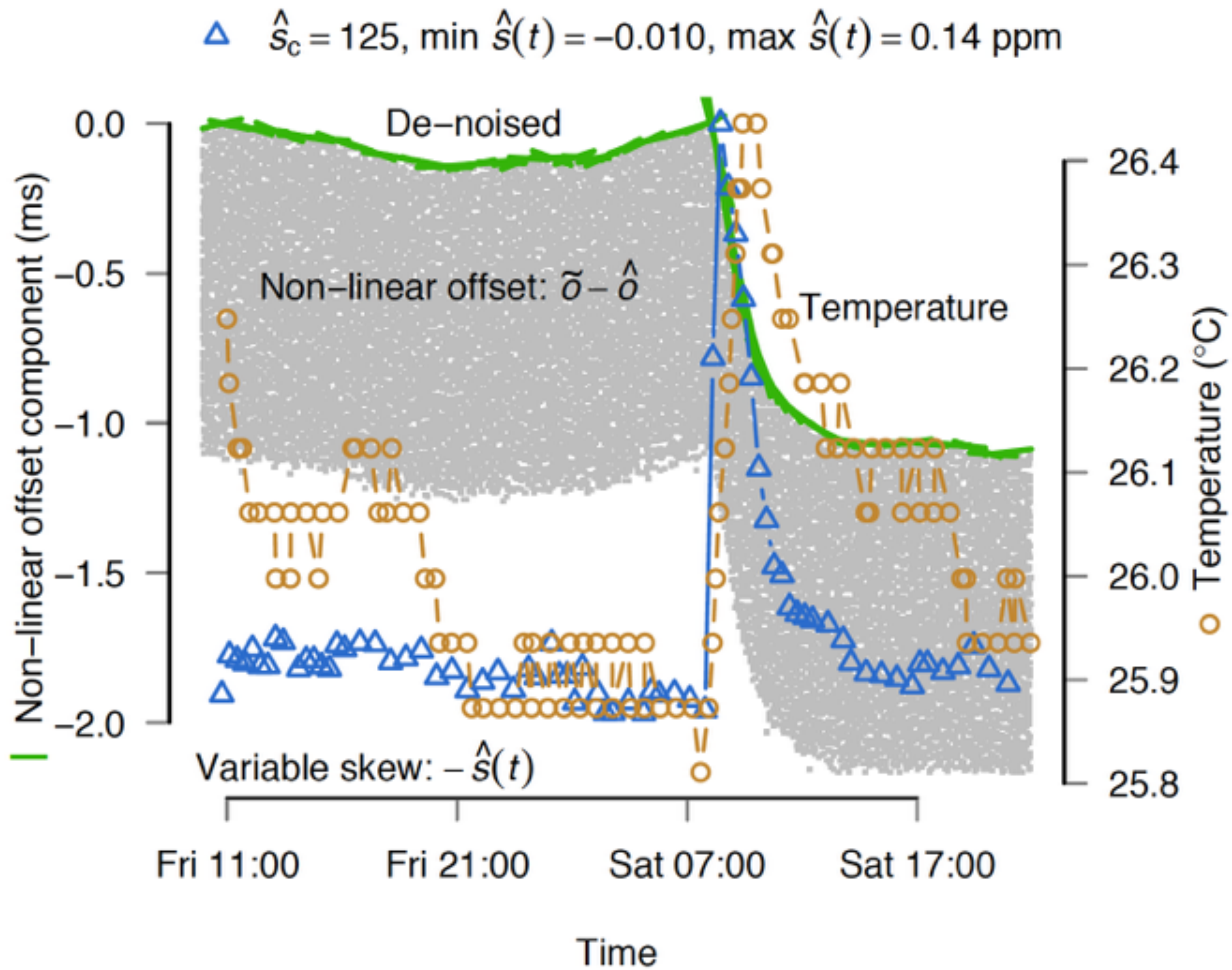
Best Fit Upper Bound:
Slope gives an estimate for s_C

Impact of temperature





Effect of temperature on the s(t)

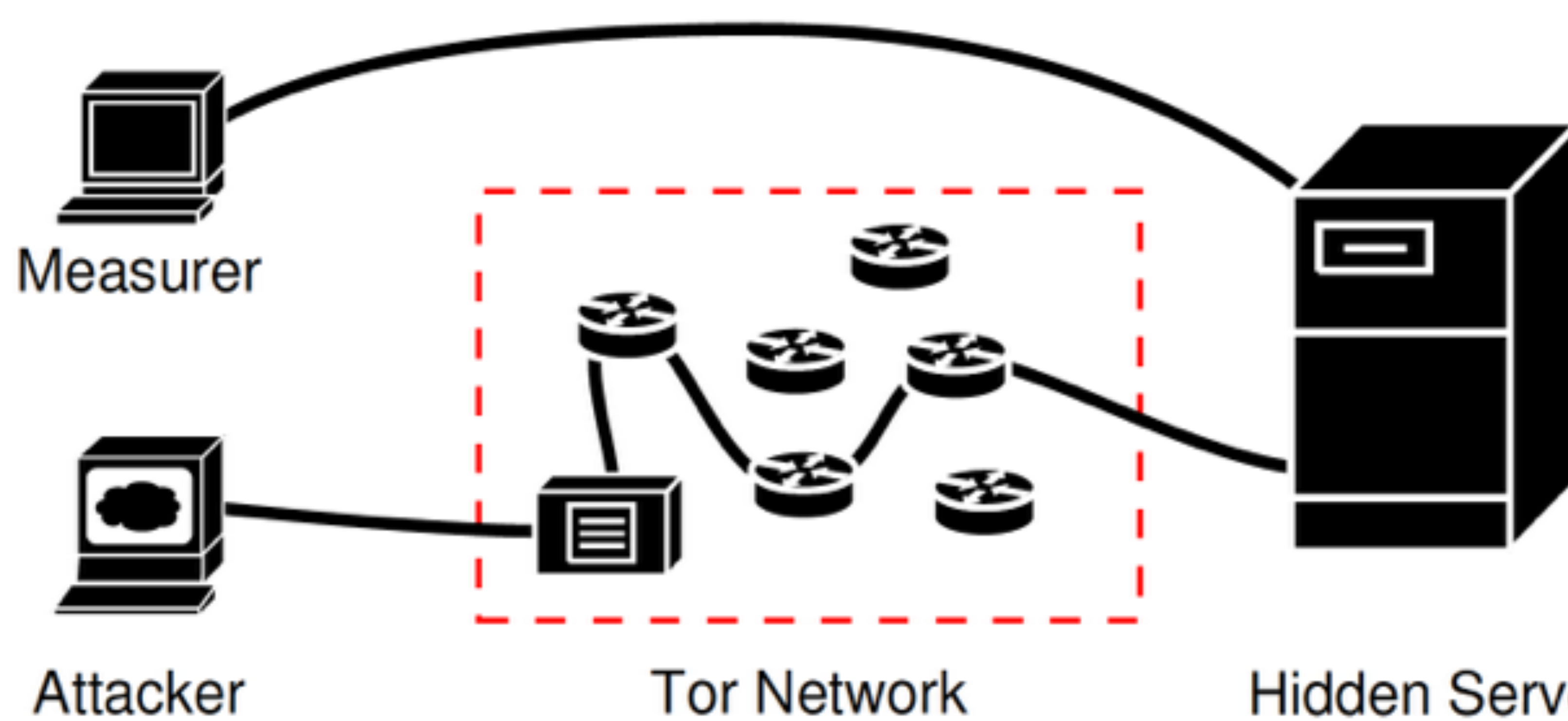


Attacking Tor - Setup

- The authors now simulate a Tor scenario to show that observing the clock skew and hence the temperature can be used to corelate the CPU usage of a target machine.

Connects directly; continuously requests TCP timestamps

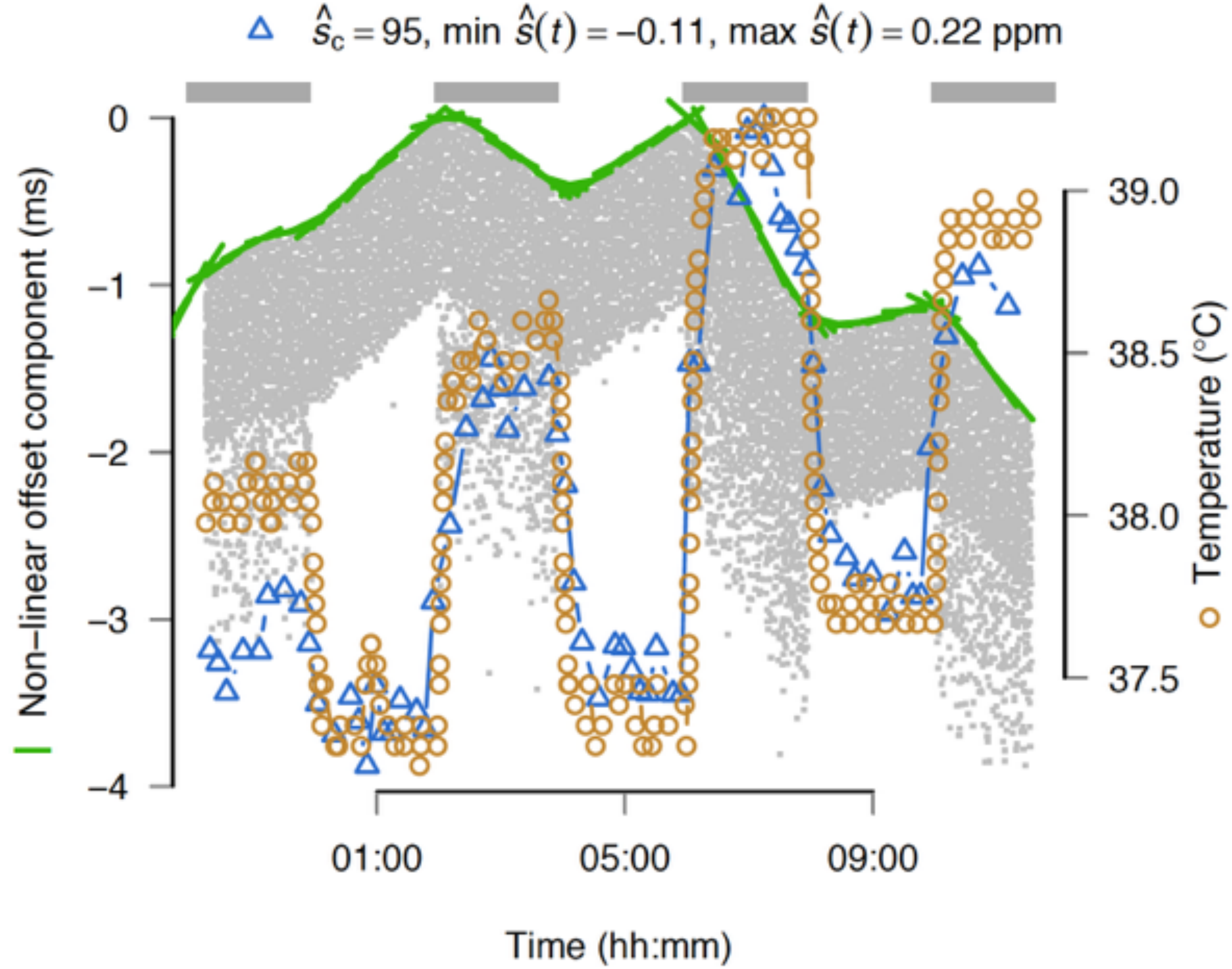
Tor Client : Repeatedly requesting the file from the server



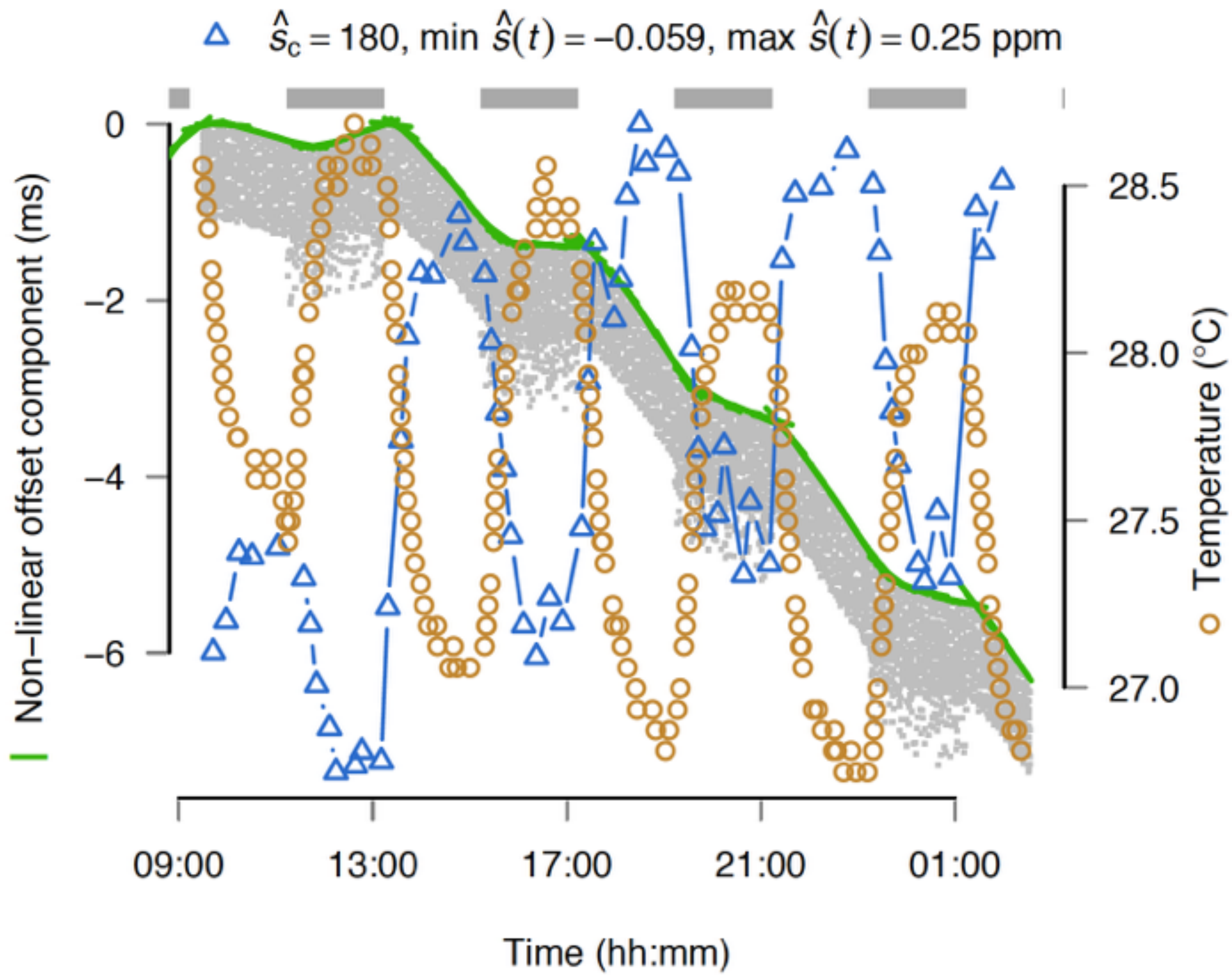
2 Tor Directory Servers
5 Normal Servers

Hosting a 10 MB File

Results - 1


$\hat{s}_c = 95$, min $\hat{s}(t) = -0.11$, max $\hat{s}(t) = 0.22$ ppm
Non−linear offset component (ms)
0
−1
−2
−3
−4
01:00
05:00
09:00
Time (hh:mm)
39.0
38.5
38.0
37.5
Temperature (°C)

Results - 2


$\hat{s}_c = 180$, min $\hat{s}(t) = -0.059$, max $\hat{s}(t) = 0.25$ ppm
Non−linear offset component (ms)
0
−2
−4
−6
28.5
28.0
27.5
27.0
Temperature (°C)
09:00
13:00
17:00
21:00
01:00
Time (hh:mm)

Possible defenses

- Use of more expensive over controlled crystal oscillators which behave better with temperature
- Always run at Maximum CPU load
- External access to timing information can be restricted or jittered

Summary

- QoS guarantees help towards preventing traffic analysis based attacks on anonymity systems.
- Even in presence of such guarantees the idle/busy period on a CPU gets reflected on its temperature and in turn on its clock skew.
- The clock skew can be measured remotely using TCP timestamps

Strengths

- Although the technique of measuring clock skew to finger-print devices had been established, the paper is the first to apply temperature modulation in conjunction with skew measurements to reveal hidden information

- The attack circumvents the QoS based defenses proposed to counter traffic analysis based attacks.

- The attack establishes a more general paradigm of attack where “high” confidentiality level information is leaked to a agent with access to “low” confidentiality level information through the hardware.

Weaknesses

- More of a proof-of-concept than an actual attack.
- A very small scale experiment done on a private network with conditions designed to be favorable.
- The issue of latency noise having an effect of measurements has not been strongly considered.

Extensions

- Geolocation
- Noise Mitigation
- Classical Covert Channels