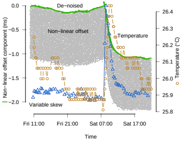
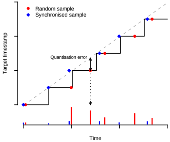
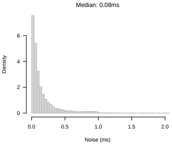
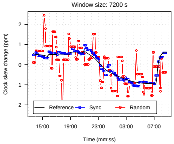
An Improved Clock-skew Measurement Technique for Revealing Hidden Services









Sebastian Zander[1], Steven J. Murdoch[2]

[1]caia.swin.edu.au/cv/szander
[2]www.cl.cam.ac.uk/users/sjm217

SWINBURNE UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

USENIX Security Symposium, 28 July – 1 August 2008, San Jose, CA, US

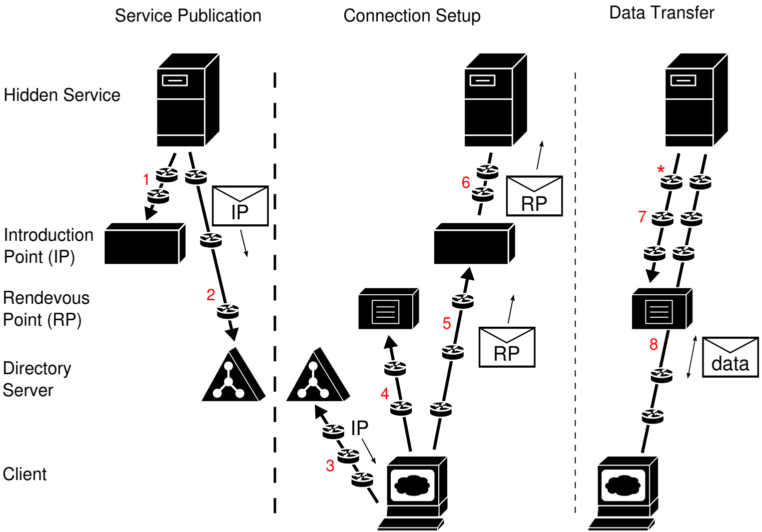
Overview

- What are hidden services
- Revealing hidden services: Clock skew, temperature and network load
- Current clock skew estimation approach and noise sources
- Improved clock skew estimation: Synchronised sampling
- Evaluation of synchronised sampling
- New techniques for revealing hidden services
- Conclusions and future work

Tor is a low-latency, distributed anonymity system

- Real-time TCP anonymisation system (e.g. web browsing)
- Supports anonymous operation of servers (hidden services)
- These protect the user operating the server and the service itself
- Constructs paths through randomly chosen nodes (around 2 500 now)
- Multiple layers of encryption hide correlations between input and output data
- No intentional delay introduced

Hidden services are built on top of the anonymity primitive the Tor network provides



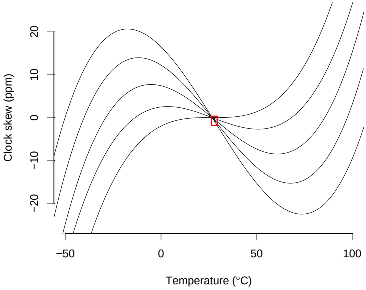
Computers have multiple clocks, some can be queried over the Internet

- A clock consists of an:
  - Oscillator, controlled by a crystal, ticks at a nominal frequency
  - Counter, counts the number of ticks produced by the oscillator
- Some clocks can be queried remotely:

Clock	Frequency	NTP	Firewall	Other
ICMP timestamp request	1 kHz	Affected	Usually blocked	Often disabled in operating systems
TCP sequence numbers	1 MHz	Affected	Cannot be blocked	Linux specific, very difficult to use
TCP timestamp extension	2 Hz – 1 kHz	Unaffected	Hard to block	Cannot be measured over Tor (no end-to-end TCP)
HTTP timestamp header	1 Hz	Affected	Hard to block	Low frequency, can be measured over Tor

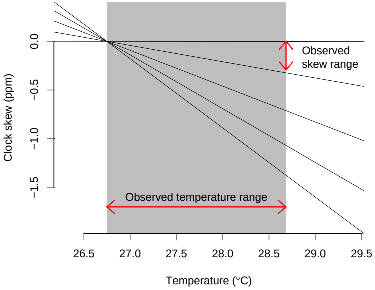
Temperature has a small, but remotely measurable, effect on clock skew

- Clock skew: difference in frequency of a clock to the 'true' clock
- Skew of typical clock crystal will change by ± 20 ppm over $150\,^\circ$C operational range
- In typical PC temperatures, only around ± 1 ppm
- By requesting timestamps and measuring skews, an estimate of temperature changes can be derived
- Even in a well-insulated building, changes in temperature over the day become apparent



Temperature has a small, but remotely measurable, effect on clock skew

- Clock skew: difference in frequency of a clock to the ‘true’ clock
- Skew of typical clock crystal will change by ± 20 ppm over $150\,^\circ$C operational range
- In typical PC temperatures, only around ± 1 ppm
- By requesting timestamps and measuring skews, an estimate of temperature changes can be derived
- Even in a well-insulated building, changes in temperature over the day become apparent



Clock skew variations can be extracted with numerical analysis

Measure clock offset of candidate machine(s)

↓

Remove constant skew from offset

↓

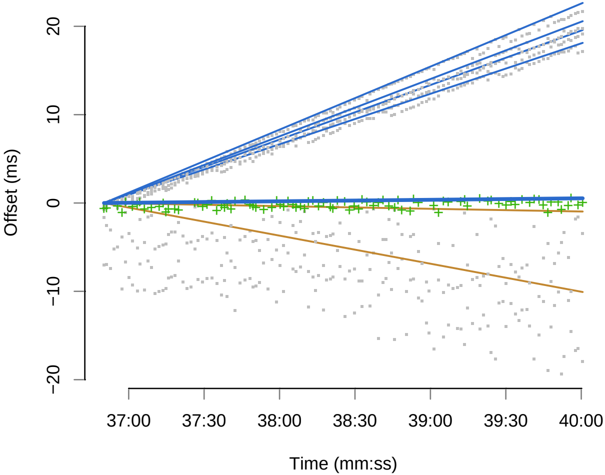
Remove noise

↓

Differentiate and negate

↓

Compare to temperature



Clock skew variations can be extracted with numerical analysis

Measure clock offset of candidate machine(s)

↓

Remove constant skew from offset

↓

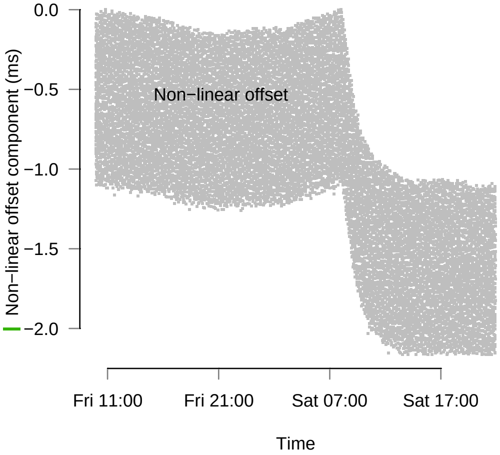
Remove noise

↓

Differentiate and negate

↓

Compare to temperature



Clock skew variations can be extracted with numerical analysis

Measure clock offset of candidate machine(s)

↓

Remove constant skew from offset

↓

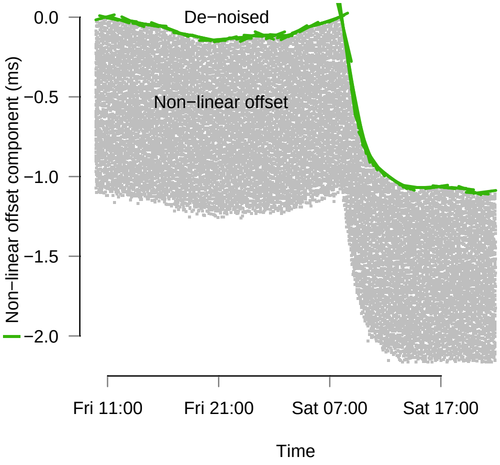
Remove noise

↓

Differentiate and negate

↓

Compare to temperature



Clock skew variations can be extracted with numerical analysis

Measure clock offset of candidate machine(s)

↓

Remove constant skew from offset

↓

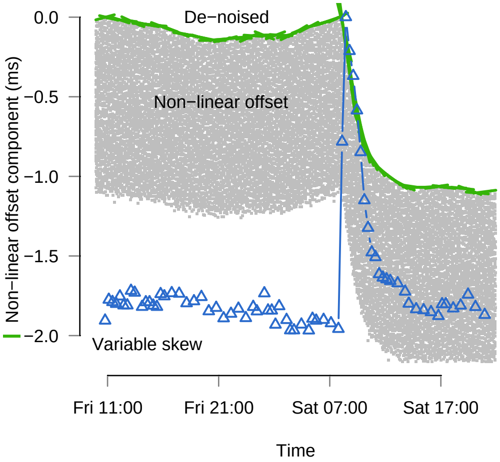
Remove noise

↓

Differentiate and negate

↓

Compare to temperature



Clock skew variations can be extracted with numerical analysis

Measure clock offset of candidate machine(s)

↓

Remove constant skew from offset

↓

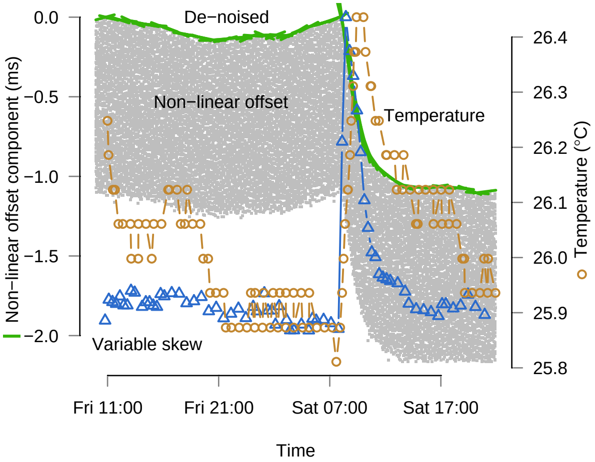
Remove noise

↓

Differentiate and negate

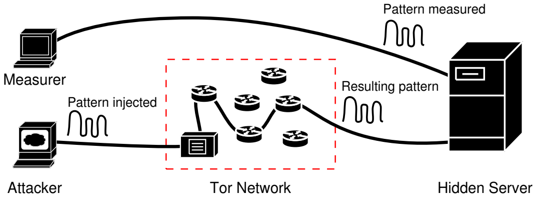
↓

Compare to temperature



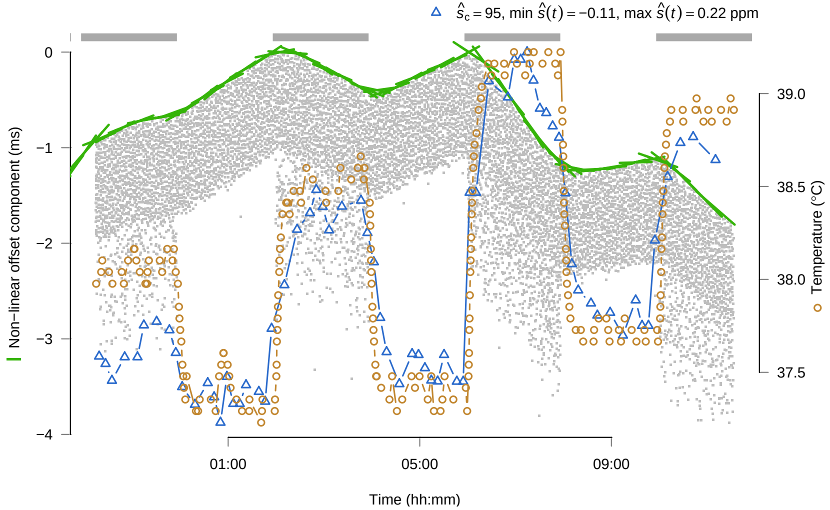
Network load of hidden service can be estimated by measuring temperature induced clock skew

- Attacker induces load pattern by making requests to hidden server via Tor
- At the same time the attacker directly measures clock-skew patterns of candidates (set of IP addresses)
- If the patterns match, the hidden service is revealed

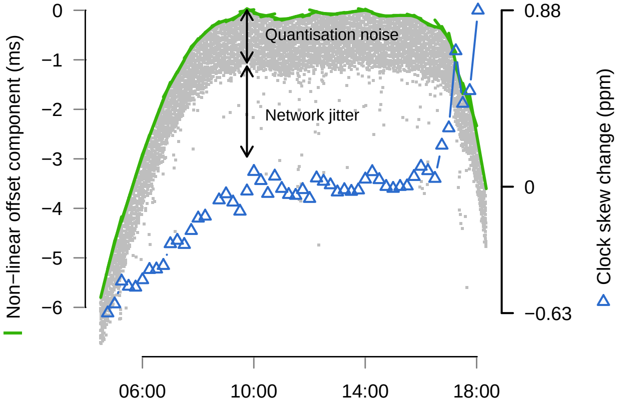


Network load of hidden service can be estimated by measuring temperature induced clock skew

- Here, a periodic 2 hour on, 2 hour off pattern was used

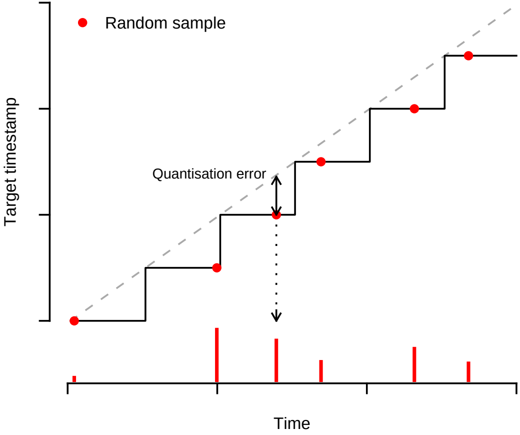


Measurement errors have two sources: quantization noise and network jitter



Many samples, over a long time, are needed to eliminate this noise

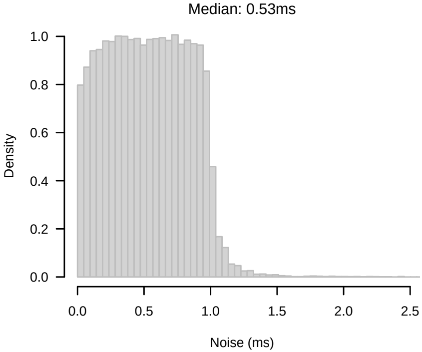
Quantization noise of a sample depends on how close it was to a clock-edge



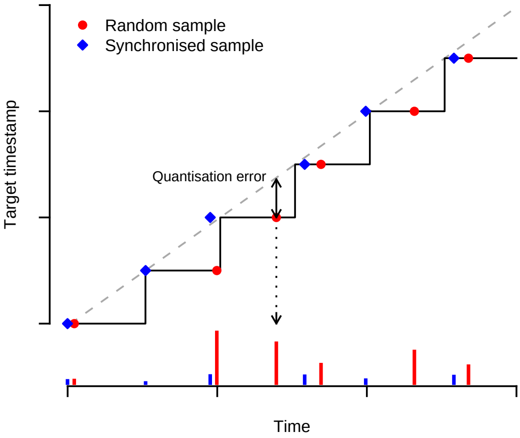
Only the samples made near clock edges contribute to the accuracy of the skew measurement

Current attack is limited by quantisation noise

- For 1 kHz clock shown here, max. quantization error is 1 ms
- Clock-skew cannot be accurately measured via Tor because available 1 Hz HTTP timestamps have a 1 s period
- Temperature change must be induced sending larger amounts of traffic across Tor
    - May not be possible (Tor has low capacity and server may limit requests)
    - Even if possible it would likely raise suspicion



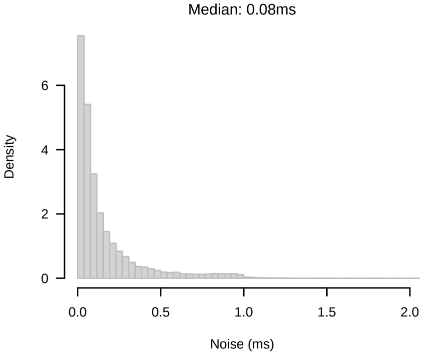
Quantization noise can be effectively eliminated by sampling just before or after clock ticks



Now the noise level is independent of clock frequency

Synchronised sampling algorithm

- Algorithm first locks onto target's clock tick, and predicts position (before or after tick)
- Then it alternately samples before and after clock ticks (determined by bounds)
- If actual position equals expected position, bounds are tightened, otherwise they are opened
- It also adjusts the sampling interval based on relative skew between attacker and target
- Resulting noise is far lower than random sampling

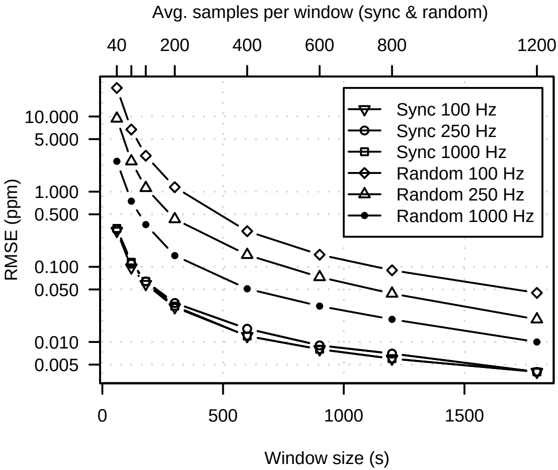


Evaluation compares synchronised sampling with random sampling in different scenarios

- True clock skew cannot be measured, so what baseline to compare against?
  - Use 1 MHz reference clock realised by exchanging µs resolution timestamps over UDP
  - Reference clock does not provide true skew, but has minimal quantisation error
- Compare clock skew estimates based on TCP or HTTP timestamps with reference using root mean square error $RMSE = \sqrt{\frac{1}{N} \sum_i (\hat{x}_i - x_i)^2}$
- Use same average sample rates for random and synchronised sampling
- One clock-skew estimate is computed for w samples (window)
- Use over-sampling to get more frequent clock-skew estimates

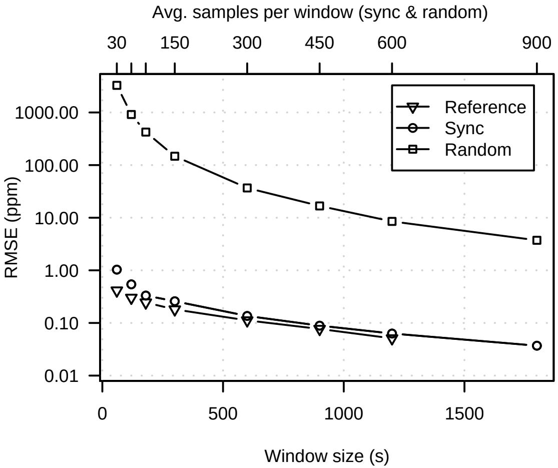
With synchronised sampling the accuracy is independent of quantisation noise

- Compare synchronised and random sampling in LAN
- Obtained clock frequencies by rounding target's timestamps



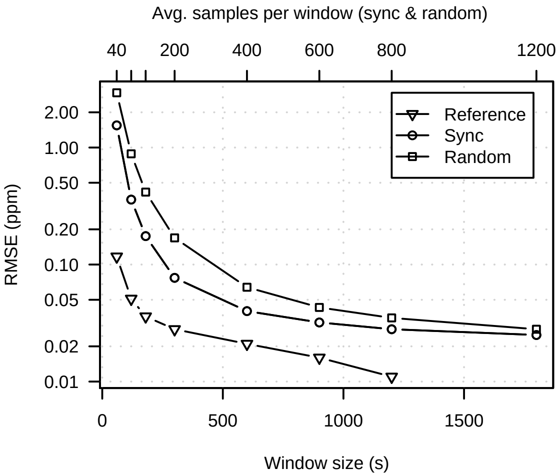
Low-resolution HTTP timestamps become usable for clock-skew estimation

- Compare synchronised and random sampling in LAN
- Target was running Apache 2.2.4 (no extra load)



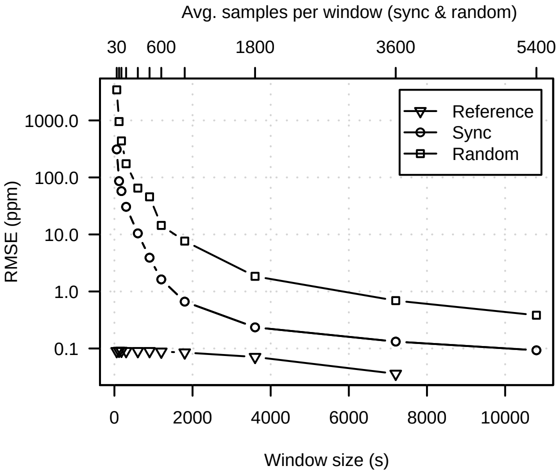
Even on long-distance path the noise reduction is significant as network jitter is often small

- 22 hops (average RTT of 325 ms, but RTT/2 jitter was $\leq$0.5 ms)
- Used 1 kHz TCP timestamps



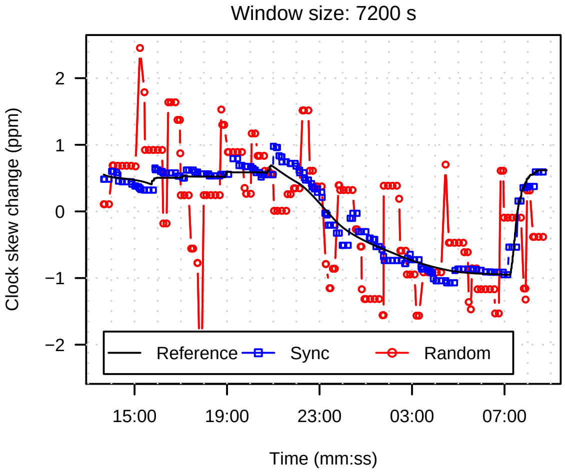
Clock skew can be estimated across Tor

- Currently performance/reliability of Tor hidden services is poor
- Used private 19-node Tor testbed running on Planetlab nodes
- Average RTT was 885 ms and RTT/2 jitter up to 50 ms



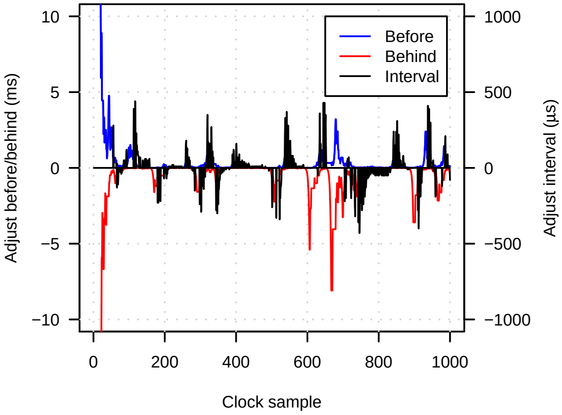
Daily temperature-change patterns are visible

- Synchronised sampling shows temperature decreasing during night and increasing during day
- Random sampling does not show pattern (same window size)



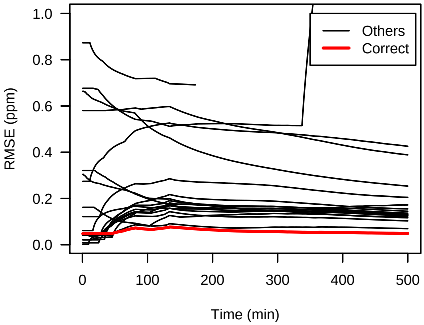
Initial synchronisation is quick even across Tor

- Takes about 2.5 minutes for algorithm to synchronise
- But high network jitter forces regular opening of bounds and sample interval adjustments



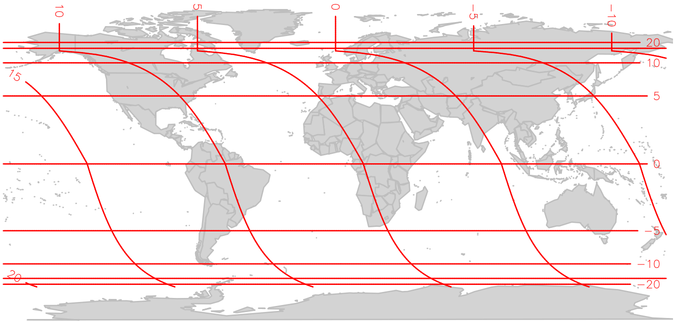
More efficient variants of the original attack (1)

- Attacker measures clock skew of the hidden server via Tor and of candidates directly
- Compare fixed skew or variable skew over time (shown here) to identify hidden server
- Generates only fraction of traffic needed of original attack (here one probe every 2 seconds)
- Requires only fraction of time of original attack, especially if fixed skew can be used (here 139 minutes)



More efficient variants of the original attack (2)

- Attacker measures clock skew of hidden service and estimates geographic location
- Generates only a fraction of traffic and does not require direct access to the target
- Does not provide an unambiguous identification if candidate locations are geographically close



Conclusions and future work

- Synchronised sampling significantly improves accuracy of clock-skew estimation
- Synchronised sampling enables accurate clock-skew estimation from low-frequency clocks
- Improves previous attack and enables new more efficient attacks
- Improves other clock-skew-based techniques, such as remote fingerprinting

- Extend evaluation (analyse duration and traffic volume of new attacks, use real Tor network)
- Improve timing accuracy (use real-time kernel or kernel implementation)
- Algorithm parameter tuning