A First-Hop Traffic Analysis Attack Against Femtocell*

Saman Feghhi and Douglas J. Leith

School of Computer Science & Statistics, Trinity College Dublin, Ireland.
{feghhis,doug.leith}@tcd.ie

Abstract. We introduce an attack against encrypted web traffic passing over the first hop between a Femtocell device and the mobile core network. The attack makes use only of packet timing information on the uplink and so is impervious to packet padding defences. We demonstrate the effectiveness of the attack at identifying the web sites being visited by a mobile broadband user whose traffic goes through a Femtocell device, achieving mean success rates of 92%. As well as being of interest in its own right, this timing-only attack serves to highlight deficiencies in existing defences and so to areas where it would be beneficial for VPN designers to focus further attention.

1 Introduction

In this paper we consider an attacker of the type illustrated in Figure 1. The attacker can detect the time of packets traversing the first hop encrypted tunnel between the Femtocell device and the mobile core network, but has no other information about the clients activity. A Femtocell is an eNodeB cellular base station with a small physical footprint (similar to a WiFi access point) and limited cell size (typically about 30m radius). It is intended to improve cellular coverage indoors, filling in coverage holes and improving download rates, while also offloading traffic from the macrocell network. Wired backhaul to the cellular operators network is via a user supplied network connection *e.g.* a home DSL line. Since Femtocells are usually user installed, physical access to the backhaul connection is straightforward and it is a simple matter to route backhaul traffic via a sniffer. Mobile operators are, of course, aware of this and backhaul traffic is therefore secured via use of an IPSec encrypted tunnel. The attacker's objective is to guess, with high probability of success, the web sites which the client visits. What is distinctive about the attack considered here is that attacker relies solely on packet timestamp information whereas previously reported attacks against encrypted web traffic have mainly made use of observations of packet size and/or packet count information.

Our interest in timing-only attacks against the first hop is twofold. Firstly, packet padding is a relatively straightforward defence against attacks that rely

* This work was supported by Science Foundation Ireland under Grant No. 11/PI/1177.

primarily on packet size. Secondly, alternative attacks based on packet counting [**?**, **?**] are insensitive to packet padding defences but require partitioning of a packet stream into individual web fetches in order for the number of packets associated with each web fetch to be determined, which may be highly challenging in practice on links where there are no clear pauses between web fetches. In contrast, packet timing-based attacks are not only largely unaffected by packet padding defences but also do not rely upon partitioning of the packet stream. Hence, they are potentially a practically important class of attack against encrypted tunnels e.g. Tor, Femtocell and indeed other VPNs. While some work has been carried out using inter-arrival time information to classify the application (HTTP, IMAP *etc.*) [**?**], to the best of our knowledge there is no previous work reporting the use of timing information alone to construct a successful classification attack against encrypted web traffic.

The main contributions of this paper as as follows: (i) we describe an attack against encrypted web traffic that uses packet timing information alone, (ii) we demonstrate that this attack is effective against clients using the mobile broadband network through a Femtocell, achieving mean success rates of 92%, (iii) we also demonstrate that the attack is effective against traffic streams *i.e.* back to back web page fetches where the packet boundaries between fetches are unknown.

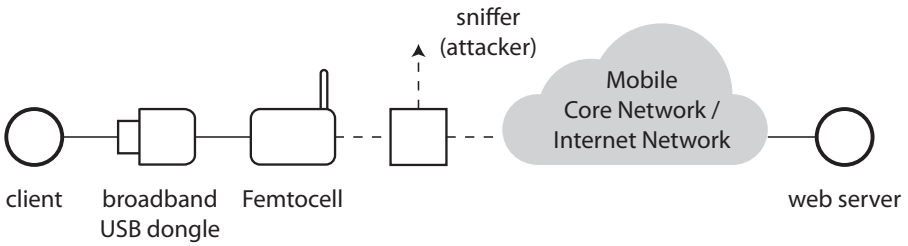


Fig. 1: Schematic illustrating attacker of the type considered. A client machine is connected to the Internet via broadband USB dongle through a Femtocell. The attacker can detect the time when packets traverse the first hop encrypted tunnel, but has no other information.

2 Related Work

The general topic of traffic analysis has been the subject of much interest, and a large body of literature exists. Some of the earliest work specifically focussed on attacks and defences for encrypted web traffic appears to be that of Hintz [**?**], which considers the SafeWeb encrypting proxy. In this setup (i) web page fetches occur sequentially with the start and end of each web page fetch known,

and for each packet (ii) the client-side port number, (iii) the direction (incoming/outgoing) and (iv) the size is observed. A web page signature is constructed consisting of the aggregate bytes received on each port (calculated by summing packet sizes), effectively corresponding to the number and size of each object within the web page. In [**?**] it is similarly assumed that the number and size of the objects in a web page can be observed and using this information a classification success rate of 75% is reported.

Subsequently, Bissias *et al* [**?**] considered an encrypted tunnel setup where (i) web page fetches occur sequentially with the start and end of each web page fetch known, and for each packet (ii) the size, (iii) the direction (incoming/outgoing) and (iv) the time (and so also the packet ordering) is observed. The sequence of packet inter-arrival times and packet sizes from a web page fetch is used to create a profile for each web page in a target set and the cross correlation between an observed traffic sequence and the stored profiles is then used as a measure of similarity. A classification accuracy of 23% is observed when using a set of 100 web pages, rising to 40% when restricted to a smaller set of web pages.

Most later work has adopted essentially the same model as [**?**], making use of packet direction and size information and assuming that the packet stream has already been partitioned into individual web page fetches. In [**?**, **?**] Bayes classifiers based on the direction and size of packets are considered while in [**?**] a SVM classifier is proposed. In [**?**] classification based on direction and size of packets is studied using Levenshtein distance as the similarity metric, in [**?**] using a Gaussian Bag-of-Words approach and in [**?**] using KNN classification. Similarly, [**?**] considers Bayes and SVM classifiers and finds that a range of proposed defences are ineffective.

3 Anatomy of a Web Page Fetch

When traffic is carried over an encrypted tunnel the packet source/destination addresses/ports and the payload are hidden. We also assume here that the tunnel pads the packets to be of equal size, so that packet size information is also concealed, and that the start and end of an individual web fetch may also be concealed *e.g.* when the web fetch is embedded in a larger traffic stream. An attacker sniffing traffic on the encrypted tunnel is therefore able only to observe the direction and timing of packets through the tunnel, *i.e.* to observe a sequence of pairs $\{(t_k, d_k)\}$, $k = 1, 2, \cdots$ where t_k is the time at which the k'th packet is observed and $d_k \in \{-1, 1\}$ indicates whether the packet is travelling in the uplink/downlink direction. Since it will be sufficient to mount an effective attack, we will assume a weaker attacker that can only observe the timestamps $\{t_k\}$, $k \in K_{up} := \{\kappa \in \{1, 2, \cdots\} : d_\kappa = -1\}$ of uplink traffic .

Figure 2 plots the timestamps $\{t_k\}$ of the uplink packets sent during the course of fetching five different health-related web pages (see below for details of the measurement setup). The x-axis indicates the packet number k within the stream and the y-axis the corresponding timestamp t_k in seconds. It can be seen that these timestamp traces are distinctly different for each web site,

and it is this observation that motivates interest in whether timing analysis may by itself (without additional information such as packet size, uplink/downlink packet ordering *etc.*) be sufficient to successfully de-anonymise encrypted web traffic.

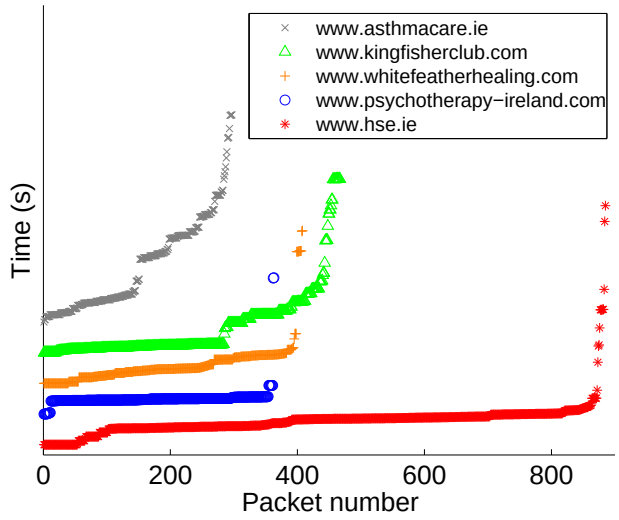


Fig. 2: Time traces of uplink traffic from 5 different Irish healthcare websites are shown. It can be seen that the website time traces exhibit distinct patterns. The traces are shifted vertically to avoid overlap and facilitate comparison.

To gain insight into the differences between the packet timestamp sequences in Figure 2 and, importantly, whether they are genuinely related to characteristics of each web page rather than to other factors, it is helpful to consider the process of fetching a web page in more detail. To fetch a web page the client browser starts by opening a TCP connection with the server indicated by the URL and issues an HTTP GET or POST request to which the server then replies. As the client parses the server response it issues additional GET/POST requests to fetch embedded objects (images, css, scripts *etc.*). These additional requests may be to different servers from the original request (*e.g.* when the object to be fetched is an advert or is hosted in a separate content-delivery network), in which case the client opens a TCP connection to each new server in order to issue the requests. Fetching of these objects may in turn trigger the fetching of further objects. We make the following more detailed observations:

1. *Connection to third-party servers.* Fetching an object located on a third-party server requires the opening of a new TCP connection to that server, over which the HTTP request is then sent. The TCP connection handshake introduces a delay (of at least one RTT) and since the pattern of these delays

is related to the web page content it can potentially assist in identifying the web page.
2. *Pipelining of requests.* Multiple objects located on the same server lead to several GET/POST requests being sent to that server, one after another. Due to the dynamics of TCP congestion control, this burst of back-to-back requests can affect the timing of the response packets in a predictable manner that once again can potentially assist in identifying the web page.
3. *Asynchronous requests.* Dynamic content, *e.g.* pre-fetching via AJAX, can lead to update requests to a server with large inter-arrival times that can potentially act as a web page signature.
4. *Connection closing.* When a web page fetch is completed, the associated TCP connections are closed. A FIN/FINACK/ACK exchange closes each connection and this burst of packets can have quite distinctive timing which allows it to be identified. Since the number of connections is related to the number of distinct locations where objects in the web page are stored, it changes between web pages.

Our aim is to use timing features such as these, which vary depending upon the web page fetched, to create a timing signature which allows us to identify which web page is being fetched based on timing data only.

4 Comparing Sequences of Packet Timestamps

Suppose we have two sequences of packet timestamps $\boldsymbol{t} := \{t_i\}$, $i = 1, 2, \cdots, n$ and $\boldsymbol{t}' := \{t'_j\}$, $j = 1, 2, \cdots, m$. Note that the sequence lengths n and m are *not* assumed to be the same. To proceed we need to define an appropriate measure of the distance between such sequences.

4.1 Derivative Dynamic Time Warping

Our interest is in a measure of the distance between packet sequences which is insensitive to the types of distortion introduced by the network, so that the distance between packet streams $\boldsymbol{t}$ and $\boldsymbol{t}'$ associated with fetches of the same web page at different times is measured as being small, and ideally the distance between fetches of different web pages is measured to be large. To this end we use a variant of Dynamic Time Warping (DTW) [**?**]. DTW aims to be insensitive to differences between sequences which are due to stretching/compressing of time and so can be expected to at least partly accommodate the effects of changes in download rate, queueing delay *etc.*and distortions due to using a shared wireless channel.

We define a warping path $\boldsymbol{p}$ to be a sequence of pairs, $\{(p_k^i, p_k^j)\}$, $k = 1, 2, \cdots, l$ with $(p_k^i, p_k^j) \in V := \{1, \cdots, n\} \times \{1, \cdots, m\}$ satisfying boundary conditions $p_1^i = 1 = p_1^j$, $p_l^i = n$, $p_l^j = m$ and step-wise constraints $(p_{k+1}^i, p_{k+1}^j) \in V_{p_k^i, p_k^j} := \{(u, v) : u \in \{p_k^i, p_k^i + 1\} \cap \{1, \ldots, n\}, v \in \{p_k^j, p_k^j + 1\} \cap \{1, \ldots, n\}\}$,

$k = 1, \cdots, l-1$. That is, a warping path maps points from one timestamp sequence to another such that the start and end points of the sequences match (due to the boundary conditions) and the points are monotonically increasing (due to the step-wise constraints). This is illustrated schematically in Figure 3, where the two timestamp sequences to be compared are indicated to the left and above the matrix and the bold line indicates an example warping path.

Let $P^l_{mn} \subset V^l$ denote the set of all warping paths of length l associated with two timestamp sequences of length n and m respectively, and let $C_{\boldsymbol{t},\boldsymbol{t}'}(\cdot) : P^l_{mn} \to \mathbb{R}$ be a cost function so that $C_{\boldsymbol{t},\boldsymbol{t}'}(\boldsymbol{p})$ is the cost of warping path $\boldsymbol{p} \in P^l_{mn}$. Our interest is in the minimum cost warping path, $\boldsymbol{p}^*(\boldsymbol{t},\boldsymbol{t}') \in \arg\min_{\boldsymbol{p}\in P^l_{mn}} C_{\boldsymbol{t},\boldsymbol{t}'}(\boldsymbol{p})$. In DTW the cost function has the separable form $C_{\boldsymbol{t},\boldsymbol{t}'}(\boldsymbol{p}) = \sum_{k=1}^{l} c_{\boldsymbol{t},\boldsymbol{t}'}(p^i_k, p^j_k)$ where $c_{\boldsymbol{t},\boldsymbol{t}'} : V \to \mathbb{R}$, in which case optimal path $\boldsymbol{p}^*(\boldsymbol{t},\boldsymbol{t}')$ be efficiently found using the backward recursion,

$$(p^i_k, p^j_k) \in \arg \min_{(p^i,p^j)\in V_k} C_{k+1} + c_{\boldsymbol{t},\boldsymbol{t}'}(p^i, p^j) \tag{1}$$

$$C_k = C_{k+1} + c_{\boldsymbol{t},\boldsymbol{t}'}(p^i_k, p^j_k) \tag{2}$$

where $V_k = (p^i, p^j) \in \{(u,v) : (p^i_{k+1}, p^j_{k+1}) \in V_{u,v}\}$, $k = l-1, l-2, \cdots$ and initial condition $C_l = c_{\boldsymbol{t},\boldsymbol{t}'}(n,m)$. When there is more than one optimal solution at step (1), we select (p^i_k, p^j_k) uniformly at random from amongst them.

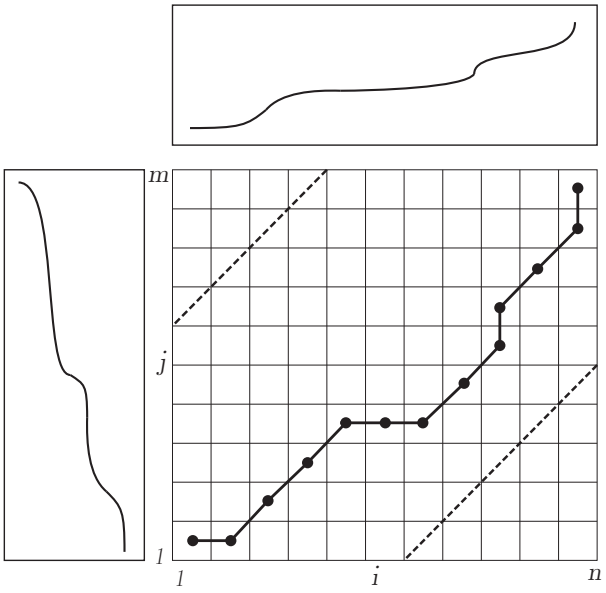


Fig. 3: Illustrating a warping path. The dashed lines indicate the warping window.

A common choice of element-wise cost is the Euclidean norm $c_{\boldsymbol{t},\boldsymbol{t}'}(p^i, p^j) = (t_{p^i} - t'_{p^j})^2$. However, to improve robustness to noise on the timestamp values (in addition to misalignment of their indices), following [**?**] we instead use the following element-wise cost

$$c_{\boldsymbol{t},\boldsymbol{t}'}(p^i, p^j) = (D_{\boldsymbol{t}}(p^i) - D_{\boldsymbol{t}'}(p^j))^2 \tag{3}$$

where $D_{\boldsymbol{t}}(i) = \frac{(t_i - t_{i^-}) + (t_{i^+} - t_{i^-})}{2}$, $i^- = \max\{i-1, 1\}$ and $i^+ = \min\{i+1, |\boldsymbol{t}|\}$. Observe that $D_{\boldsymbol{t}}(i)$ is akin to the derivative of sequence $\boldsymbol{t}$ at index i. Further, we constrain the warping path to remain within windowing distance w of the diagonal (*i.e.* within the dashed lines indicated on Figure 3) by setting $C(\boldsymbol{p}) = +\infty$ for paths $\boldsymbol{p} \in P^l_{mn}$ for which $|p^i_k - p^j_k| > \max\{w \min\{n, m\}, |m - n|\}$ for any $k \in \{1, \cdots, l\}$.

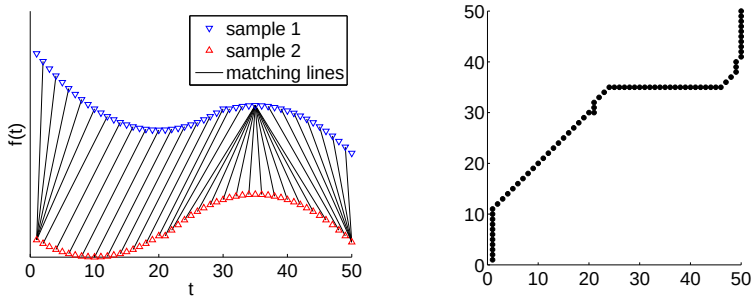
Figure 4b illustrates the alignment of points between two sequences obtained using this approach and for comparison Figure 4a shows the corresponding result when using Euclidean cost. The figure shows the warping paths on the right-hand side and an alternative visualisation of the mapping between points in the sequences on the left-hand side. Observe that when Euclidean cost is used the warping path tends to assign many points on one curve to a single point on the other curve. In comparison, use of the derivative distance tends to mitigate this effect and select a warping path with fewer horizontal and vertical sections.

4.2 *F*-Distance Measure

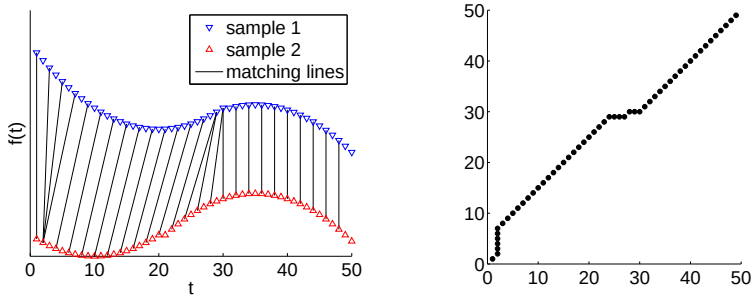
Given two timestamp sequences, the warping path is a mapping between them. With reference to Figure 3, sections of the warping path which lie parallel to the diagonal correspond to intervals over which the two sequences are well matched. Sections of the warping path that are parallel to the x- or y-axes correspond to intervals over which the two sequences are poorly matched. This suggests using the fraction of the overall warping path which is parallel to the x- or y-axes as a distance measure, which we refer to as the F-distance.

In more detail, let $\boldsymbol{p} = \{(p^i_k, p^j_k)\}$, $k = 1, \cdots, l$ be a derivative DTW warping path relating timestamp sequences $\boldsymbol{t}$ and $\boldsymbol{t}'$, obtained as described in the previous section. We partition the warping path into a sequence of subpaths within each of which either p^i_k or p^j_k remain constant and we count the subpaths which are longer than one. For example, for the setup shown in Figure 5 there are five subpaths: $(1,1)$; $(2,2),(2,3)$; $(3,4),(4,4),(5,4)$; $(6,5)$; $(7,6)$. Two of these subpaths consist of more than one pair of points, namely $(2,2),(2,3)$ and $(3,4),(4,4),(5,4)$, and these correspond, respectively, to the vertical section and the horizontal section on the corresponding warping path shown in Figure 5b.

Formally, define $\kappa_1 := 0 < \kappa_2 < \cdots < \kappa_{r-1} < \kappa_r := l$ such that for each $s = 1, \cdots, r-1$ (i) either $p^i_{k_1} = p^i_{k_2}$ $\forall k_1, k_2 \in \{\kappa_s + 1, \cdots, \kappa_{s+1}\}$ or $p^j_{k_1} = p^j_{k_2}$ $\forall k_1, k_2 \in \{\kappa_s + 1, \cdots, \kappa_{s+1}\}$ and (ii) either $\kappa_{s+1} = l$ or condition (i) is violated for some $k_1, k_2 \in \{\kappa_s, \cdots, \kappa_{s+1} + 1\}$ *i.e.* each subsequence is maximal. Note that $p^i_k \neq p^j_k$ for all $k = 1, \cdots, l$ (due to warping path step-wise constraints) and so in condition (i) it is not possible for both p^i_k and p^j_k to be constant. We are now

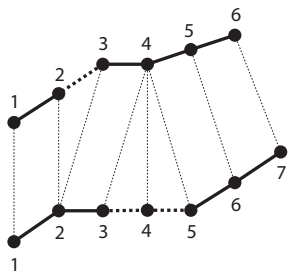


(a) Euclidean cost

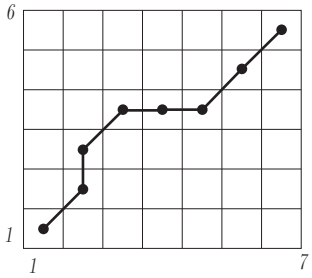


(b) Derivative cost

Fig. 4: Example DTW alignment and warping paths between two sequences vs cost function $c_{t,t'}$ used, window $w = 0.1$. In this example the length l of the warping path is 73 when a Euclidean cost is used and 54 with the derivative cost.



(a) Sequence alignment.



(b) Warping path

Fig. 5: Illustrating method for calculating the F-distance between two timestamp sequences.

is a position to define the F-distance measure between timestamp sequences $\boldsymbol{t}$ and $\boldsymbol{t}'$, namely:

$$\phi(\boldsymbol{t}, \boldsymbol{t}') := \frac{\sum_{\substack{s \in [1, r-1] \\ \kappa_{s+1} - \kappa_s > 1}} \kappa_{s+1} - \kappa_s}{n + m} \tag{4}$$

where κ_s, $s = 1, \cdots, r$ are the constant subsequences in minimal warping path $\boldsymbol{p}^*(\boldsymbol{t}, \boldsymbol{t}')$. It can be seen that $\phi(\boldsymbol{p})$ takes values in interval $[0, 1]$, and is 0 when sequences $\boldsymbol{t}$ and $\boldsymbol{t}'$ are identical (in which case the warping path $\boldsymbol{p}$ lies on the diagonal in Figure 3). For the example in Figure 5 the F-distance $\phi(\boldsymbol{p})$ is $(2 + 3)/13 = 0.385$.

5 De-anonymising Web Fetches Over Femtocell

In this section we present measurements of web page queries carried out over the encrypted Femtocell channel and evaluate the accuracy with which the web page being fetched can be inferred using only packet timing data. The dataset consists of 100 fetches of the home pages of each of the top 100 Irish health, finance and legal websites as ranked by www.alexa.com under its Regional/Europe/Ireland category in September 2014, yielding a total of 10000 individual web page fetches. The web pages were fetched during February 2015 where all samples of each website are fetched consecutively over an hour. To study the effect of time on dynamic pages, a second data set is collected on May 2015 where this time, for each website a sample is taken every hour for a duration of 6 days. A `watir-webdriver` script on Firefox 36.0 was used to perform the web page fetches and `tcpdump` to record the timestamps and direction (uplink/downlink) of all packets traversing the Femtocell connecting the client to the network although only packet timestamps on the uplink were actually used. Further experiments are also done showing similar results for downlink, and different versions of websites (caching *etc.*) which demonstrates their impact on the results.

5.1 Hardware/Software Setup

The client machine is a Sony VGN-Z11MN laptop running on Intel core 2 duo 2.26GHz CPU and 4GB of memory. It is running Ubuntu Linux 14.04 LTS Precise. It uses a Huawei K3770 HSPA USB Broadband Dongle to connect wirelessly to the internet via the Femotcell. The Femtocell is a commercial Alcatel-Lucent 9361 Home Cell V2-V device. The Femtocell wired backhaul is connected to the campus network via a NetGear EN 108 TP Ethernet hub. A monitor computer which is running on a AMD Athlone 64 X2 Dual Core Proc 5000+ CPU and 4GB memory is also connected to this hub and logs all packets.

In the setup considered here, the Femtocell backhaul is over a university gigabit ethernet connection and we used `tcpdump` to log packets passing over this link.

5.2 Classifying Measured Timestamp Sequences

We used the F-distance measure $\phi(\cdot,\cdot)$ described in Section 4 to compare measured uplink timestamp sequences, with windowing parameter $w = 0.2$ unless otherwise stated. A K-Nearest Neighbours approach was used for classification (use of a naive Bayes classifier was also investigated, but the K-NN classifier was found to consistently offer better performance). In the K-NN classifier, for each web page i we sort the measured timestamp sequences $\boldsymbol{t}' \in T_i$ used for training in ascending order of sum-distance $\sum_{\boldsymbol{t}\in T_i} \phi(\boldsymbol{t},\boldsymbol{t}')$ and select the top 5 to use as exemplars to represent this web page. When presented with a new timestamp sequence, its distance to the exemplars for all of the training web pages is calculated and these distances are sorted in ascending order. Classification is then carried out by majority vote amongst the top K matches.

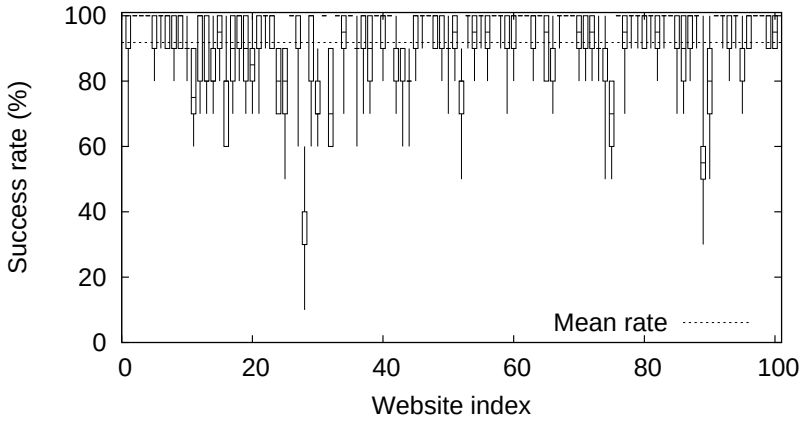
5.3 Classification Performance

For each of the 100 web pages studied, Figure 6 details the measured classification accuracy using the K-NN approach. We use 10-fold cross validation, where the 100 samples of each website are divided into 10 random subsets and for each subset we use the remaining 90 samples as the training data to find the exemplars and use the 10 samples in the subset as the validation data. The rates for these 10 subsets for each website are summarized and displayed in the figure. Each of the boxes indicate the 25%, 50% and 75% quartiles and the lines indicate the maximum and minimum values.

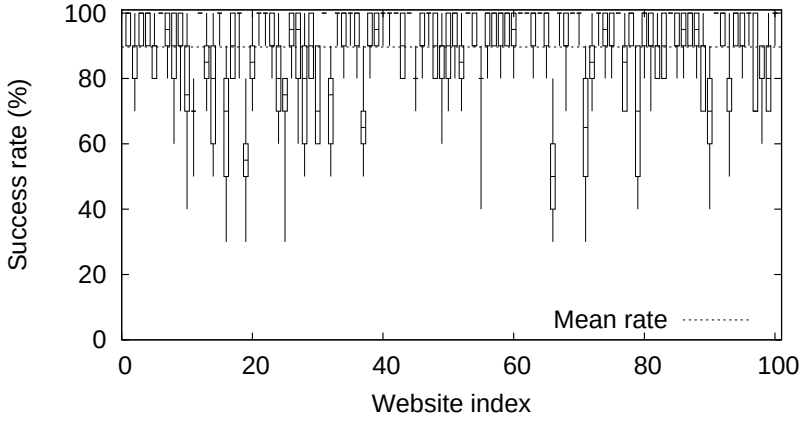
The mean success rate is 91.8%. This compares with a baseline success rate of 1% for a uniform random classifier over 100 web sites and so is likely to represent a significant compromise in privacy. For comparison, when 50 rather than 100 web sites are used the success rate is 95.7%, indicating that our results are relatively insensitive to the number of web pages. A mean success rate of 89.7% is obtained when the samples are fetched every hour over a duration of 6 days. The insignificant change in classification accuracy shows the effectiveness of the attack even on dynamic websites whose content evolves over time.

The mean success rate of 91.8% is also compared with 90.5% for when the attack is conducted against downlink traffic, which shows little difference in success rates. Moreover to show the effect of caching, a small but similar experiment is conducted (using the remainder of our data allowance) where the caching is enabled on the browser and the success rate is 86.7%. The results are summarized in Table 1.

For comparison, we also created a second dataset by repeating the 10000 web page fetches on a standard ethernet channel where the traffic is routed from the client to the internet. Figure 7 details the results obtained. Overall, a mean success rate of 95.0% was obtained. It can be seen that using a mobile broadband with a Femtocell device makes little difference to the classification accuracy compared to normal network use. The measured performance for other parameter settings is also summarised in Table 1.



(a) Consecutive fetching over one hour



(b) One fetch every hour for 6 days

Fig. 6: K-Nearest Neighbours classification performance when using mibile broadband over Femtocell, no browser caching, $K = 5$.

5.4 Finding a web page within a sequence of web requests

In the experiments presented so far we have assumed that within the observed packet timestamp stream the boundaries between different web fetches are known. This is probably a reasonable assumption on lightly loaded links where the link is frequently idle between web fetches. However, not only might this assumption be less appropriate on more heavily loaded links but it also allows for a relatively straightforward means of defence, namely insertion of dummy packets to obscure the boundaries between web fetches. In this section we therefore extend consideration to links where web fetches are carried out in a back

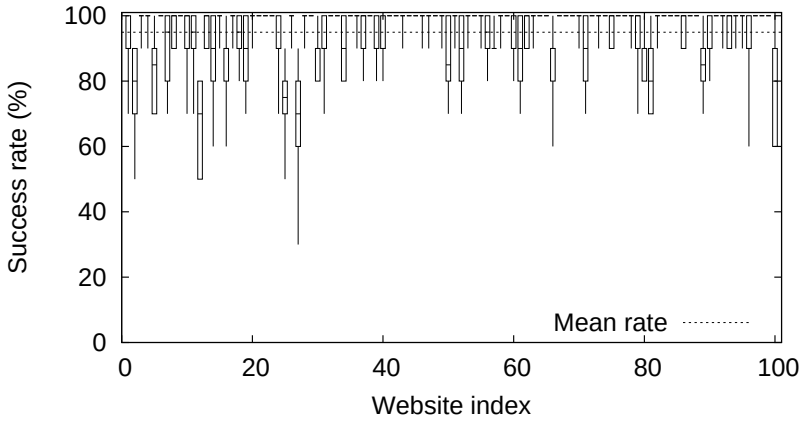


Fig. 7: K-Nearest Neighbours classification performance when using standard ethernet channel, no browser caching, $K = 5$.

	Exemplars	Web Pages	K		
			1	3	5
Femtocell (Uplink)	3	100	92.6%	91.8%	91.83%
	3	50	95.72%	95.20%	94.86%
	3	100*	90.2%	90.08%	89.65%
Downlink	3	100	91.80%	90.95%	90.50%
Cached	3	25	91.48%	89.92%	86.68%
Ethernet (Uplink)	3	100	95.01%	94.97%	94.98%
	3	50	97.16%	97.18%	97.04%

Table 1: Summary of the measured success rate of the timing-only attack. (*) is for the case were samples are collected over a 6 day period.

to back fashion such that the boundaries between web fetches cannot be easily identified.

The basic idea is to sweep through a measured stream of packet timestamps trying to match sections against the timing signature of a web page of interest. This exploits the fact that our timing-only attack does not fundamentally depend on knowledge of the start/end times of the web fetch (unlike approaches which use packet counts to classify web pages).

In more detail, to locate a target web page within a stream of packet timestamps we first select three measured packet timestamp sequences for that web page to act as exemplars (as previously). Then, we sweep through the stream of timestamps in steps of 10 packets, extract a section of the stream of the same length as each exemplar (plus 10 to cover the step size) and calculate the distance between the section and the exemplar. After sweeping through the full stream we select the location within the stream with least distance from the exemplars

to be the most likely location of the target web page within the stream. This process assumes that the target web page is present within the packet stream.

We constructed a test data set as follows. For each sample a website is chosen as the target website. It is then fetched among 1, 2, 3 and 4 other websites selected randomly from the list. The fetches are permuted and fetched consecutively with a pause time chosen at random from $[5, 30]$ seconds. This is done 100 times and once for each website from the list.

Using the classification approach described above we attempted to identify the location of target websites within each packet stream. Corresponding to consecutive fetching of 2, 3, 4 and 5 websites, we achieved success rates of 85%, 82%, 72% and 70% for locating the target web page within each packet stream with a position error of $w.l_s$ packets, where w is the window size at which DTW operates (0.2 in our setting) and l_s is the average length of the 3 exemplars which is calculated for each website s separately. Given the limited information being used, this is a remarkably high success rate and indicates the power of the timing-only attack.

6 Summary and Conclusions

We introduce an attack against encrypted web traffic passing through a Femtocell device and to the cellular network. The attack makes use only of packet timing information on the uplink and is therefore impervious to packet padding defences. We demonstrate the effectiveness of the attack at identifying the web sites being visited by a user, achieving mean success rates of 92%. In addition to being of interest in its own right, this timing-only attack serves to highlight deficiencies in existing defences and so to areas where it would be beneficial for VPN designers to focus further attention.