TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGIES
*Trans. Emerging Tel. Tech.* (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/ett.2697

RESEARCH ARTICLE

# A first look at traffic patterns of Siri

L. Caviglione*

Institute of Intelligent Systems for Automation (ISSIA), National Research Council of Italy (CNR)

## ABSTRACT

The evolution of handheld devices and wireless communication ignited a new wave of speech-driven services using an "intelligent" back-end to perform complex tasks, for example, real-time dictation and data retrieval. Hence, the network is of crucial importance, and its usage must be properly analysed. This paper discusses some basic behaviours of Siri in terms of traffic patterns, possible models and a short privacy/security assessment. Copyright © 2013 John Wiley & Sons, Ltd.

***Correspondence**
L. Caviglione, CNR – ISSIA, Via de Marini 6, I-16149, Genova, Italy.
E-mail: luca.caviglione@ge.issia.cnr.it

Received 20 March 2013; Revised 9 May 2013; Accepted 12 July 2013

## 1. INTRODUCTION

Advancements of mobile devices and wireless communications for accessing the Internet are one of the most important drivers for the creation of innovative services. While Web 2.0 fosters real-time interactions among people, *de-facto* standard features, such as the Global Positioning System or the built-in camera, enable sharing geo-tagged multimedia contents. Despite the power of mobile appliances, almost all functionalities are often located within a remote back end. Accordingly, devices are more similar to terminals in *à la* mainframe organisation rather than *smart* entities. This design is at the basis of popular services like Google Goggles, Shazam and Siri (http://www.apple.com/ios/siri/). In more details, Google Goggles enables data search through "visual queries", whereas Shazam can guess song titles from short audio clips recorded via the device's microphone. Instead, Siri is a more comprehensive tool for speech interacting with the phone, for example, for dictating text and commands or retrieving answers to questions formulated in a natural language form. To implement such utilities, it is composed of a "thin" operating system (OS)-wide client interface in charge of (i) sampling and compressing voice with the Speex audio codec; (ii) sending data chunks to a remote computing facility and (iii) routing responses to the OS. Consequently, the majority of information and algorithms are located at the server-side, which is in charge of recognising and processing the vocal stimuli and sending back directives to the requestor. Possible replies are textual recognition of each word and information gathered from the Internet or a well-defined *service cloud*. Hence, to assure a proper degree of responsiveness, the device continuously communicates with the back-office, leading to a traffic intensive behaviour. This is even more critical when using Universal Mobile Telecommunication System (UMTS) or long term evolution loops, which can have time varying bandwidth and delays and battery or data billing issues. Nevertheless, when in presence of a vast volume of users, the produced load could be highly fragmented, thus forcing the network operator to deploy proper quality of service mechanisms [1]. Therefore, investigating the traffic produced by modern speech-driven mobile applications is of crucial importance. Even if more general studies are available in the literature, at the author's best knowledge, this is the first work focused on the basic network behaviours of voice-assisted services. Specifically, reference [2] analyses the traffic produced by smartphones, also highlighting some interactions with the power management module of the radio subsystem. Reference [3] characterises the network usage of mobile appliances from several viewpoints, for example, in terms of users' session and diurnal/nocturnal patterns.

The contributions of this work are a basic traffic analysis of Siri, including a concise review of models [4], and a short assessment of privacy/security issues. The remainder of the paper is structured as follows: Section 2 briefly introduces the Siri protocol. Section 3 showcases its characterization, whereas Section 4 reviews limits of state-of-the-art models. Section 5 presents some privacy/security considerations, and Section 6 concludes in this paper.

## 2. THE SIRI PROTOCOL IN BRIEF

Siri is released under a closed-source policy, thus its communication layer has been documented via a collective reverse engineering effort. The application uses the client-server model, where data is transmitted through an Hypertext Transfer Protocol (HTTP) channel. To guarantee sufficient degrees of security and confidentiality, the resulting stream is sent via a Secure Socket Layer encrypted connection. All the used network protocols are standard, except for the HTTP, which is tweaked in a byzantine manner. In fact, data are delivered by an indefinitely held single connection, rather than by utilizsing a *back channel*, as recommended for streaming-like applications [5]. Another peculiarity is the use of the Adaptive Communication Environment custom framework, resulting in the Content-Length header permanently set to $\sim$ 2 Gbyte and the absence of the Content-Encoding one. Also, the server-side manages connections by using a Close header. To offer a suitable level of responsiveness, when the user activates Siri, it continuously exchanges information between the phone and the server farm. This generates a non-negligible consumption of transmission resources, which is mitigated by compressing data with the *zlib* algorithm. Again, a nonstandard usage can be observed because the proper control header is placed only within the first chunk rather than in every packet composing the flow. Also, the compressed stream is produced with the sync-flush option, resulting into units sent without the checksum and the size of the original uncompressed information. Differently from similar applications, Siri does not perform any additional processing to voice-related packets. Messages noncontaining voice fragments are generated in a binary form and then encapsulated within a property list (.plist), that is, a dictionary-like structure ruled by (*key, value*) associations. For completeness, we mention the main types: keepalives, blocks containing the encoded voice and server replies. As an example, when using Siri to dictate text, for each word, the related audio block is sent, and the ordinary reply consists in the word in textual form, a score about the accuracy of the recognition, and a timestamp.

## 3. TRAFFIC ANALYSIS

This section deals with the traffic analysis of Siri in order to characterise some key features of modern speech-driven applications. To this aim, we performed trials in a controlled setup composed by a Siri device (iPhone 4S) connecting o the Internet via a 100 Mbit/s link bridged by a dedicated IEEE 802.11n access point (AP). To capture data, we mirrored traffic to a target Linux machine running Wireshark. To produce realistic patterns, we selected five paradigmatic operations: a medium-size Facebook status update (50 words), a long-size note dictation (150 words), a 160 character long SMS (including the needed address book lookup via vocal commands), a weather forecast interrogation and the creation of an entry in the calendar. Such use-cases are assorted enough to capture and compose a variety of tasks, for example, posting data to other online platforms or querying different remote services. For this round of tests, we issued commands by using the English language. The usage template has been iterated until averages had a confidence interval of 95%. The highly homogenous nature of Siri traffic makes 25 trials sufficient. In more details, data is principally composed of voice fragments using a "rigid" encoding scheme, and responses adhere to the .plist format, dramatically

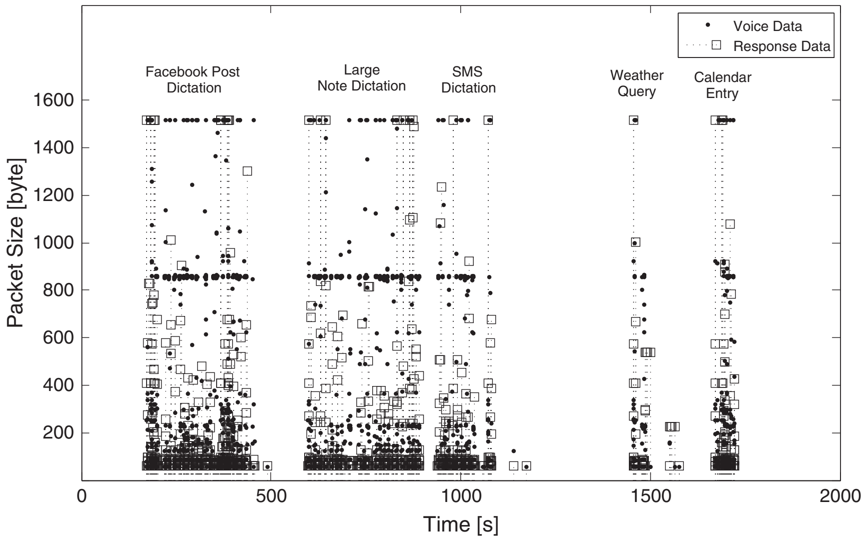

**Figure 1.** Packet trace of a reference Siri session.

DOI: 10.1002/ett

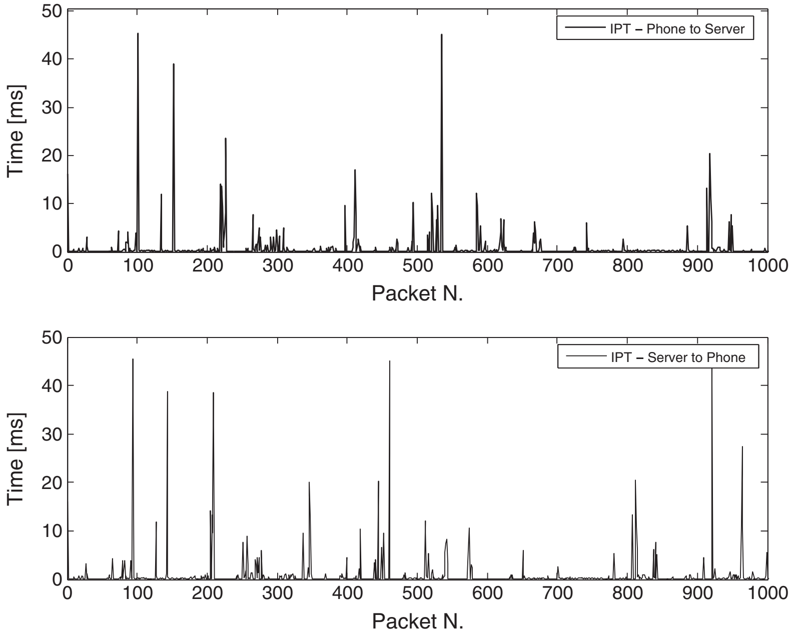

**Figure 2.** Inter packet time of a Siri-generated stream.

reducing the variety of exchanged information especially if compared with web browsing [2].

Figure 1 depicts a reference trace for a single repetition of the defined use-cases. Because the device produces traffic mostly to send voice samples to the back office, patterns are very similar even if with various durations.

Homogeneities in the different generation processes support the intuition that the "amount of voice" (i.e. the word count) rules the volume rather than its significance. To avoid latencies, as soon as the required processing is completed, the server sends data back to the device. As a result, its packet rate and the duration of bursts are tightly coupled with the received stimuli. Another viewpoint to characterise the offered load is given by the inter packet time (IPT), that is, the time gap between two adjacent packets. Figure 2 depicts a typical reference outcome (ACK segments are ∼40%). For the sake of clarity, the Siri traffic has been subdivided according to the direction. Because bottlenecks were absent, delays among consecutive packets are very bounded and constant, thereby the application is not affected by *jitter*. On the average, the IPT is of 8.12 and 4.73 ms from phone to server and vice versa, respectively. We observe that larger values are usually due to hesitations or pauses performed by the talker. Thus, when in presence of enough bandwidth, both the hardware of the phone and the remote computational facilities are sufficient to assure an excellent degree of responsiveness. Because network topology could play a role, we performed additional trials by using a standard 10 Mbit/s fiber to the home Internet access. Also in this case, the IPT values were very similar. For such reasons, saturation appears as the main cause for a bottleneck, rather than specificities in the network architecture.

An important tool to develop realistic models is the cumulative density function (CDF) of the size of protocol data units (PDUs) generated by Siri. Figure 3 showcases the CDF computed on the entire dataset. Data produced by the phone are mostly composed by voice, resulting in PDUs mainly in the range of 800 − −1500 byte. This is consistent with the "richness" of audio information, even if encoded and compressed. Nevertheless, the resulting payload is large enough to avoid packetisation delays without generating too small packets. On the contrary, PDUs carrying server responses are mainly in the 100 − −600 byte span, reflecting their textual nature. Further aggregation has to be avoided, because collides with the constraint of pushing back information as fast as possible, even at the cost of producing tiny packets. This prevents latencies or a poor quality of experience (QoE) but increases the bandwidth usage (mainly due to additional HTTP/TCP/IP headers). Figure 4 shows a reference throughput generated by Siri produced by implementing the defined operations, but imitating different speaker's behaviours, as to emphasise the correlation among human characteristics and the related traffic. In more details (i) peaks are due to sudden and fast "ramblings" and (ii) drops are consequences of hesitations or pauses. Instead, the throughput produced by the server-side is mainly composed of textual entries in response to remote stimuli. Consequently, it is less resource consuming, and its bursts match the rates of information produced by the talker.

Table I summarises average values (computed on ∼300 000 PDUs using both the English and the Italian datasets as will be discussed in Section 4) of the most relevant characteristics of the Siri traffic, as well as best fitting

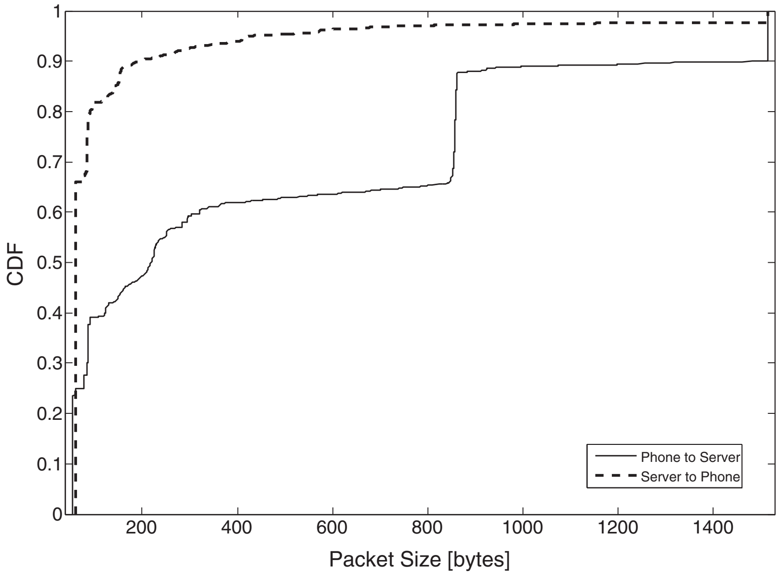

**Figure 3.** Cumulative density functions of protocol data unit sizes produced by the Siri framework.

CDFs for the IPTs. Distributions have been computed with the MatLab Statistic Toolbox, and only those with a confidence greater than the 95% have been considered. As a final remark, we did not encountered any transport/network issues (e.g. TCP retransmissions) during our trials.

### 3.1. The impact of mobility

When IEEE 802.11 is not available, Internet connectivity is usually provided via cellular networks or satellite links if in rural or emerging areas. Accordingly, it is important to understand their impacts over Siri-like applications. To this aim, we routed data from the access point through a Linux machine running netem to emulate different conditions, as resumed in Table II. Tests have been made with the same cases and repetitions presented in Section 3. A deep analysis of Siri over cellular networks is out of the scope of this paper. Rather, we want to summarise the most relevant issues that should be addressed when in presence of different wireless channels.

The major findings are (i) when using highly asymmetrical links, data can be received at a reduced rate, thus degrading the perceived interactivity or causing service

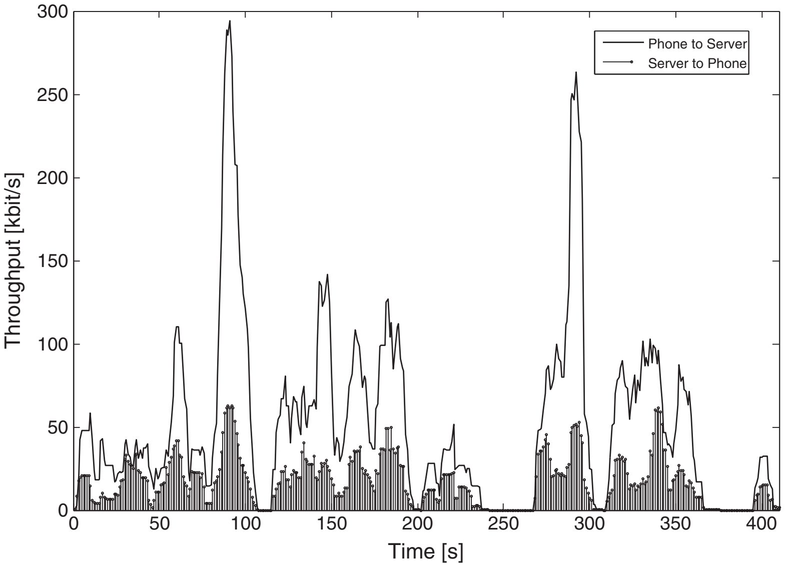

**Figure 4.** Breakdown of the throughput produced by Siri.

**Table I.** Traffic summary computed on the entire dataset - 25 trials per language.

| — | Phone → Server | Phone ← Server |
|---|---|---|
| Avg. IPT [ms] | 8.12 | 4.73 |
| Avg. tput [kbit/s] | 68.2 | 35.8 |
| Avg. session volume (overall dataset) [Kbyte] | 1,998 | 923 |
| Avg. session volume (English) [Kbyte] | 1,791 | 900 |
| Avg. session volume (Italian) [Kbyte] | 2,022 | 956 |
| Best fitting CDF for IPT | Normal | Weibull |
| | $\mu = 0.7166$ | $a = 0.135$ |
| | $\sigma = 6.957$ | $b = 0.414$ |

**Table II.** Access network parameters to emulate mobility.

| Parameter | Min | Max |
|---|---|---|
| Round trip time [ms] | 20 | 250 (520 for GEO) |
| Bit error rates (see e.g. [6]) | $10^{-6}$ | $10^{-5}$ |
| Bandwidth [kbit/s] | down = 177.6; up = 118.4 | down = 384; up = 128 |
| | (EDGE class 10, MCS-9) | (UMTS) |

Round trip time is assumed as two-way and uniformly partitioned between the uplink and downlink.

unavailability due to timeouts. Similar issues happen when in presence of high *delay·bandwidth* channels (e.g. GEO satellite). This can also reduce the QoE of Siri when used in rural areas; (ii) because the decision logic is remotely placed, intermittent connectivity (due to physical obstacles impeding proper carrier coverages) or huge delays (due to handovers) prevent the framework to correctly behave; (iii) Siri produces a one-way traffic volume comparable to a voice over IP (VoIP) application, thus its ample usage can quickly saturate the allotted data plan; (iv) voice is not sent to a remote human, which can handle unintelligible words via past experience, or by analysing the context. Therefore, high bit error rates have a worse impact than in standard VoIP communications and (v) responses are carried by tiny PDUs with low IPT bursts (see Figure 3 and Figure 2, respectively). Hence, large user populations can produce a too aggressive queuing behavior, leading to additional latencies and intermittencies [7]. For what concerns the protocol behaviours, excessive latencies and errors reflect into connection timeouts or retransmissions triggered by congestion control algorithms of the TCP, thus worsening the QoE needed to operate voice-based services.

## 4. ON THE DEVELOPMENT OF A BEHAVIOURAL MODEL

Traffic of VoIP applications has been extensively studied, for example, reference [8] revises network characterization of most popular voice codecs. However, Siri imposes the talker to act differently from when in presence of a remote human. Unlike from other voice services, speech-driven mobile applications have the following additional characteristics: (i) dictation is an open-loop process, (ii) the number of words to reach a goal is language-dependent and (iii) server responses vary according to the specific task. To capture the essence of this class of services, new models should be developed or at least tweaks to classic methodologies (e.g. semi-Markov processes or continuous-time Markov chains [9]) have to be made. Besides, this is mandatory for quantifying transmission/computing resources, bottlenecks and energy consumptions via synthetic generators. Although the definition of a general behavioural model is outside the scope of this paper, we focus on specific traits of Siri. Thus, we investigate two datasets (having the same rules discussed in Section 3) differing from the used language, that is, English and Italian. Additional features of Siri-generated patterns can be summarised as follows: (i) because human interaction is absent, the talkspurt $W$ could differ from the one of VoIP colloquies. A rough on/off inspection reveals an average of $W \sim 14$ s when dictating large notes; (ii) Italian is less concise than English: the same operations (also including the proper translation of text to be dictated) increase the data volume of $\sim$13% and (iii) let us define $T_v$ and $T_a$ as the volumes of data produced by the phone and server, respectively, and $k \in \boldsymbol{K}$ as the finite set of operations, which can be performed by Siri. A possible formal template is $T_a = f(T_v, k)$, where $f(\cdot)$ has to be found in an class of model $\boldsymbol{\Gamma}$. Defining such a functional dependence could be complex, yet it can be trivial for some cases. For instance, if $k$ is a dictation operation, $f(\cdot) = \alpha_k$, that is, it reduces as a traffic scale constant. Then, $T_a \simeq \alpha_k T_v$, with $\alpha_k = 0.41$ in our trials. The intuitive interpretation is that voice is sent to the server, and responses are a 1-to-1 match in textual form, hence smaller in term of data volumes. Conversely, when $k$ is a query, a more elaborate $f(\cdot)$ is needed because results can be mixed, for example, composed of texts, pictures, HTML or small multimedia objects.

## 5. PRIVACY/SECURITY CONSIDERATIONS

The intrinsic “personal” nature of handheld devices, jointly with the increasing *social* vocation of the Internet, makes privacy and security more critical than ever. As a consequence of its design, Siri sends voice through the network even in presence of data that has to be considered confidential and locked down in the device (e.g. a personal note). For such reason, all the traffic is encrypted by exploiting Secure Socket Layer then inheriting its flaws like certificate theft. Furthermore, voice data has well-known features making possible statistical-based attacks (see e.g. [10], and references therein, for a detailed discussion on the topic). As an example, the size of PDUs and the IPT makes feasible the disclosure of a speaker nationality, even using a cyphered channel. Then, such a leaked information can be used to conduct *social engineering* attacks. A possible countermeasure is using *traffic morphing* [11], which can alter the produced stream by masking voice characteristics and the used codebook, or performing pseudo random data padding. Alas, it needs additional hardware, software and bandwidth resources to be effective. Although such requirements could be satisfied by mobile nodes (but paying an increased price in terms of battery drain), the cumulative overhead experienced at the server side (which can be used to serve hundreds of million devices) to de-morph traffic could be prohibitive.

Lastly, the usage of the static DNS record guzzoni.apple.com to locate the server farm makes the retrieval of connection endpoints a minor effort. In this perspective, iOS nodes can be identified without using protocol resource consuming procedures, such as stack fingerprinting or deep packet inspection.

## 6. CONCLUSIONS

In this paper, we used Siri as an archetype to investigate different traffic features of modern speech-driven mobile applications. Moreover, we briefly discussed the development of a behavioural model, and we underlined potential security and privacy threats. Future work aims at enriching the analysis, as well as developing synthetic models to support performance evaluations through simulations, for example, to help carrier operators to exploit traffic engineering in the access networks.

## ACKNOWLEDGEMENTS

Acknowledgement to Prof. Mauro Coccoli for the precious suggestions and the hardware equipment.

## REFERENCES

1. Partridge C, Carvey PP, Burgess E, *et al.* A fifty-Gb/s IP router. *IEEE/ACM Transactions on Networking* 1998; **6**(3): 237–245.
2. Falaki H, Lymberopoulos D, Mahajan R, Kandula S, Estrin D. A first look at traffic on smartphones, In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC10)*, New York, NY, USA, 2010; 281–287.
3. Falaki H, Mahajan R, Kandula S, Lymberopoulos D, Govindan R, Estrin D. Diversity in smartphone usage, In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys10)*, New York, NY, USA, 2010; 179–194.
4. Caviglione L. Extending HTTP models to web 2.0 applications: the case of social networks, In *Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing (UCC11)*, Melbourne, Australia, December 5–7, 2011; 361–365.
5. Popa L, Ghodsi A, Stoica I. HTTP as the narrow waist of the future internet, In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets-IX)*, Monterey, CA, 2010; 1–6.
6. Elaarag H. Improving TCP performance over mobile networks. *ACM Computing Surveys* 2002; **34**(3): 357–374.
7. Caviglione L. Traffic analysis of an internet online game accessed via a wireless LAN. *IEEE Communications Letters* 2006; **10**(10): 698–700.
8. Menth M, Binzenhofer A, Muhleck S. Source models for speech traffic revisited. *IEEE/ACM Transactions on Networking* 2009; **17**(4): 1042–1051.
9. Daigle JN, Langford JD. Models for analysis of packet voice communications systems. *IEEE Journal of Selected Areas in Communications* 1986; **4**(6): 847–855.
10. Wright CV, Ballard L, Coull S, *et al.* Spot me if you can: uncovering spoken phrases in encrypted VoIP conversations, In *Proceedings of the IEEE Symposium on Security and Privacy*, The Claremont Resort Oakland, California, USA, 2008; 35–49.
11. Wright CV, Coull S, Monrose F. Traffic morphing: an efficient defense against statistical traffic analysis, In *Proceedings of the 16th IEEE Network and Distributed Security Symposium*, San Diego, CA, 2009; 237–250.