DOI:10.7544/issn1000-1239,2015,20150562
Journal of Computer research and development 52(10):2373-2381,2015

Towards Measuring Unobservability in Anonymous Communication Systems

Tan Qingfeng[1,2,3], Shi Jinqiao[1,2], Fang Binxing[1,2], Guo Li[1,2], Zhang Wentao[1,2], Wang Xuebin[1,2], and Wei Bingjie[4]

[1] (*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)

[2] (*National Engineering Laboratory for Information Security Technologies (Institute of Information Engineering, Chinese Academy of Sciences), Beijing* 100093)

[3] (*University of Chinese Academy of Sciences, Beijing* 100049)

[4] (*National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing* 100029)

Abstract Anonymous communication technique is one of the main privacy-preserving techniques, which has been widely used to protect Internet users' privacy. However, existing anonymous communication systems are particularly vulnerable to traffic analysis, and researchers have been improving unobservability of systems against Internet censorship and surveillance. However, how to quantify the degree of unobservability is a key challenge in anonymous communication systems. We model anonymous communication systems as an alternating turing machine, and analyze adversaries' threat model. Based on this model, this paper proposes a relative entropy approach that allows to quantify the degree of unobservability for anonymous communication systems. The degree of unobservability is based on the probabilities of the observed flow patterns by attackers. We also apply this approach to measure the pluggable transports of TOR, and show how to calculate it for comparing the level of unobservability of these systems. The experimental results show that it is useful to evaluate the level of unobservability of anonymous communication systems. Finally, we present the conclusion and discuss future work on measuring unobservability in anonymous communication systems.

Key words anonymous communications; relative entropy; unobservablity; privacy protection; traffic analysis

Chinese library classification number: TP393.08

Received date: Jun.15, 2015; **Revised date:** Aug.13, 2015

Fund Project: National Fund Project for Science & Technology Pillar Program (2012BAH37B04); Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06030200)

Corresponding author: Shi Jinqiao(shijinqiao@iie.ac.cn)

With the development of Internet technology, the network has been deeply involved in all aspects of people's lives, such as electronic voting, electronic payment, e-mail, and even web browsing, etc. Anonymous communication technology is widely used in the Internet as a major privacy enhancement technology. The existing anonymous communication system uses the Mix network and onion routing technology to hide the identity information of the communication subject and the communication relationship between the two parties in the communication process, thereby providing privacy protection for online users, thereby providing privacy protection for online users. However, with the development of network traffic analysis technology, attackers can not only detect the traffic fingerprint characteristics of the anonymous communication systems, but also further monitor the anonymous network, thereby cracking its anonymity and even blocking the connection between the Internet user and the anonymous communication system. For example, the "Prism" plan shows that the US government agencies (located at the NSA Intelligence Center in Utah) have conducted large-scale information collection on anonymous networks such as TOR (the onion router)[1]. Therefore, although the anonymous communication technology can guarantee the untraceability of the communication subject and the privacy of the communication object, it cannot cover up the fact that the communication subject is using the technology, that is, for the adversary, it can be detected that the user is using the anonymous communication systems to communicate.

In this context, researchers at home and abroad began to study the unobservability of communication protocols based on anonymous communication systems. That is to eliminate the communication behavior and traffic characteristics of anonymous communication systems by means of protocol masquerading, traffic fuzzification, access point hiding and differentiated release, etc., to defend against deep flow analysis, scanning attacks, witch attacks, etc., thereby enhancing the anonymity and undetectability of the system.

A lot of literatures [2-5] have proposed anonymous support for concealed communication methods. The literature [2-4] proposes the method of protocol masquerading and traffic fuzzification. The main idea of protocol masquerading is to imitate or camouflage the popular bunker protocol to evade network censorship, similar to “lean on a moneybag” at the protocol level. The literature[5] claims to run the real target protocol and carry the target protocol onto popular bunker protocols (such as VOIP, P2P, UGC, etc.) to realize the unobservability of communication behavior. According to the literature research published on the Internet, it is still a key challenge to quantify the degree of unobservability in anonymous communication systems. Therefore, how to quantify the degree of unobservability of these protocols is a very important for instructing the design of future systems and evaluating the degree of unobservability for communication protocols.

We model anonymous communication systems as an altering Turing machine, and analyze adversaries’ threat model. Based on this model, this paper propose a relative entropy approach that allows to quantify the degree of unobservability for communication protocols, for example, how much the degree of observability is to be detected by an attacker.

1 Relevant work

In the past 30 years, since Chaum[6] first proposed the concept of Mix (message mixing) technology and anonymous communication in 1981, the research on anonymous communication technologies on the Internet has attracted more and more attention both from academia and the general public. At the same time, research institutions have been working on the design, evaluation, analysis and improvement of anonymous communication systems. At present, the anonymous communication system used mainly includes TOR[7] and JAP[8], etc., and is deployed and run in the actual network environment. However, anonymous communication systems have not been well designed on the demand for anti-censorship at the very beginning. Therefore, the anonymous communication protocol itself is easy to detect, allowing attackers to monitor Internet users using anonymous communication tools and even block users' connections to anonymous networks.

Pfitzmann et al. [9] proposed unobservability in the literature on the definition of anonymity and privacy-related terms, and gave the definitions. In the various existing literatures, the unobservable communication system on the Internet is usually interpreted as satisfying anonymous and undetectable communication systems. Since then, researchers at home and abroad have proposed various unobservable communication systems [2-5]. Houmansadr et al. gave detailed evaluation and analysis to protocol masquerading and imitation in the literatures[10], pointing out that it is very difficult to imitate and disguise, and it is necessary to correctly implement the specific specifications of the protocol and the dependencies within and between protocols. Geddes et al. [11] pointed out that various problems can appear (such as frame mismatch, channel mismatch, transmission content mismatch, etc.) even if covert channels (such as Freeware) were embedded based on popular protocols. However, these papers do not give comparable quantitative assessments for the degree of unobservability. At present, the measuring the unobservability in anonymous communication systems has not attracted enough attention, and there is almost no relevant research work. However, there are quite a lot of research work on measuring the anonymity in anonymous communication systems, some of which will be explained below.

Reiter et al. [12] proposed the use of 1-*p* for anonymity, where *p* is the probability that an attacker can identify a particular user from an anonymous set, which can be used to measure the degree of identifiability between different objects in the same anonymous set. Berthold et al. [13] proposed a measurement for the degree of anonymity based on the size of an anonymous set, namely defining the degree of anonymity as d=1bN, where N is the number of objects in an anonymous set. However, under this definition, the measurement for the degree of anonymity only considers the number of users in the system, and does not take into account the number of users that an attacker can distinguish after a period of attack. Therefore, this measurement cannot describe the change of anonymity after an attack on the anonymous communication system. Diaz et al. [14] proposed a Shannon entropy approach that allows to quantify the degree of anonymity. The advantage of this approach is that it not only considers the size of the anonymous set, but also takes in to account the probability distribution of the identifiable state among different members in the anonymous set. Therefore it can better measure the degree of anonymity for different members in the anonymous communication system after the attacker obtains the relevant information.

Hamel et al. [15] believe that there are many defects for the entropy-based anonymity measurement, they then focuses on how adversaries' attacking affects the anonymity. They suppose adversaries' attacking ability is equivalent to the amount of bandwidth resources that can control the anonymous network, and then assign a certain bandwidth budget and explore the impact on anonymity attacks under certain bandwidth resources to evaluate the degree of anonymity in anonymous communication systems.

2 Problem statement and definition

2.1 Problem Statement

More and more users use anonymous communication technology to evade network censorship. Anonymous communication technology can protect the communication content security of the communication parties and the identity information of the communication subject, but it cannot hide the fact that the user is using the technology, that is, an attacker can easily identify that a user is using anonymous communication tools. In addition, modern anonymous communication technology can not eliminate the explicit behavior characteristics of network packets, such as the packet size distribution, network latency, and packet interval, etc. These seemingly useless protocol fingerprint features, actually can effectively help to identify the identity information of network users for the adversary with flow analysis attack capability.

For the flow analysis attacks, the most common countermeasure is protocol masquerading, which is to disguise and imitate the specific implementation of a popular protocol, to confuse the explicit behavior characteristics of the protocol, for example, SkypeMorph[2] to disguise TOR traffic as Skype video traffic, StegoTorus [3] to disguise TOR traffic as Skype and HTTP traffic, and CensorSpoofer [4] to imitate SIP-based VOIP protocol, etc. In fact, in order to fight agaisnt flow analysis, TOR has deployed transport layer plug-ins (such as obfs[16], fte[17] and Meek[18] , etc.) to support protocol obfuscation and protocol masquerading. Therefore, measuring the degree of unobservability for these protocols is of great significance for evaluating the concealment of the protocol and instructing the design and implementation of future protocols.

The availability problem in low-latency anonymous communication systems is essentially dependent on the degree of unobservability for its communication protocol. This paper focuses on the measurement and evaluation of the unobservability in anonymous communication systems.

From the perspective of user's communication behavior, the unobservability is a computational puzzle, that is, the attacker is not statistically able to confirm or distinguish whether a user is using a particular protocol. Therefore, how to quantify the degree of unobservability for the anonymous communication protocol is quite challenging. This paper will formalize the model of anonymous communication systems, and study whether the anonymous communication protocol and the disguised protocol have statistically distinguishable communication behavior under the model. That is, calculate the relative entropy of the explicit behavioral characteristics of the data packet during the communication to quantify its observability.

2.2 Relevant conception definitions

In this section, we will give an accurate definition of unobservability in anonymous communication systems. This paper adopts the definition given by Pfitzmann et al. in literature[9].

Definition 1. Anonymity. Anonymity refers to the state in which a communication subject is unrecognizable in a anonymous set. If an attacker can associate with a sender in the anonymous collection from the obtained information, such as the IP address of the sender, the user is considered identifiable.

Definition 2. Undetectability. From an attacker's perspective, the attacker cannot distinguish whether the item of interest exists.

Therefore, anonymity is to study the communication relationship of the communication subjects, and protect their identity information. The research object of undetectability is the item of interest, and protect the communication behavior and communication object.

Definition 3. Perfect undetectability. When the item of interest is completely indistinguishable, the communication object is considered to be perfectly undetectable.

Definition 4. Unobservability. Unobservability refers to the state in which the item of interest is indistinguishable from any other communication sets of the same type. The unobservability includes two meanings: the anonymity of the communication subject and undetectability of the communication object.

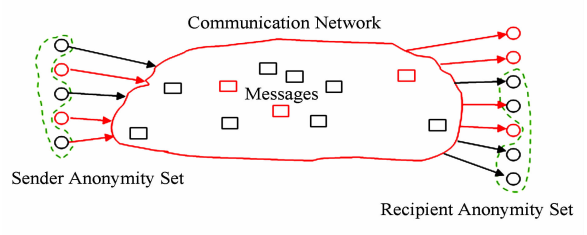

Fig.1 Network graph for anonymous communications.

This paper focuses on the measurement and evaluation of unobservability in anonymous communication systems. Here, we only consider the sender's unobservability, that is, for a particular message in the sender's anonymous set, the attacker wants to find out whether this message appears and from which sender during the communication process. The sender here is defined as all possible users.

The degree of unobservability defined in this paper is based on the probability of occurrence for a particular message. That is, after the attacker monitors the anonymous communication system for a period of time, the attacker assigns a probability value to the explicit behavior characteristics of the message that appears during the communication process for each communication subject, which is used to characterize the probability of occurrence for this message.

3 System Modeling

Anonymous communication system modeling: Defines the anonymous communication system as a directed graph $G=<V,E>$, where V is the node of the anonymous communication system, such as the client, server, relay node, while E is the routing path.

Defines E as the set of routing paths for the anonymous communication system, denoted $E=Path(G)$.

V is the set of vertices on G, and defines V as a probabilistic polynomial program, modeled as a five-tuple $V=(S, I, O, f, g)$, where:

S is a state set of clients, servers or relay nodes. Any client, server or routing node is in one state of $n=S$, denoted as $s \in S$;

I is a possible input set of the client, server or routing node. The next state of the client, server or routing node is determined jointly by the input and state transition function f;

O is the possible output set for each state. We can call this output a sequence of packets, denoted as $O=\{\{o_{n,i}\}_{i=1}^{M}\}_{n=1}^{N}$, where N is the number of packets, the number of explicit behavioral features that can be observed on each packet, and $O_{n,i}$ is the i^{th} matrix eigenvalue on the n^{th} packet;

f for the state transition function: $f: S \times I \rightarrow S$;

g for the output function: $g: S \times I \rightarrow O$.

3.1 Threat Model

The degree of observability in the system depends on the attack ability of the adversary, the resources the adversary possesses, and the observability of the explicit behavioral characteristics of the communication protocol itself. In this paper, the adversary is modeled as a passive and local adversary. That is, the adversary can monitor the traffic of the user at the network boundary, thereby observing the explicit behavior characteristics of the communication protocol (such as the packet size and the packet interval, etc.), but cannot falsify, inject or discard the packet. In addition, we also assume that the adversary can only deploy traffic analysis devices at the network boundary, and cannot control the user's computer. The adversary has limited computing and storage resources, and cannot perform complex calculations in a large-scale backbone network environment. In this section, we will define the attack capabilities of the adversary from three aspects: attack ability, observability and computing resources owned by the adversary:

1) Attack ability

①Passive attack. The adversary can deploy a flow analysis systems at the network boundary to monitor and analyze the network flow and the communication behavior of the communication subject.

②Active attack. The adversary can manipulate the network flow (such as latency, packet loss, packet content modification and injection attack, etc.).

③Attack. The adversary can simulate the client of the communication system to send packets to actively detect the network status, node behavior characteristics and so on.

2) Visibility

①Locally visible. The adversary can only monitor the network boundary (in and out of a network) or part of the network node.

②Globally visible. The adversary can monitor the entire communication network node and network traffic.

3) Resources

①Limited. The adversary has limited computing and storage resources.

②Sufficient. The adversary has sufficient computing and storage resources to collect and capture large-scale network traffic for storage and processing from different network locations.

This paper assumes that the adversaries in real world are local and passive, and have limited computing and storage resources.

4 Measurement Approach

4.1 Relative Entropy

In information theory, information entropy is defined as the probability of occurrence of stochastic discrete events. Information entropy is usually used to measure the degree of uncertainty of information. Assuming the discrete random variable is X, which has N possible values, namely $X \in \{x_1, x_2, \ldots, x_N\}$, and the probability of occurrence of each value is $p_i=P_r(X=x_i)$, where x_i is the value of the discrete random variable X, then the entropy of the discrete random variable X is defined to be:

$$H(X) = -\sum_{i=1}^{N} p_i \mathrm{lb}(p_i). \quad (1)$$

Relative entropy, also known as Kullback–Leibler divergence (KLD), is a measure of how one probability distribution P is different from a second one Q. Typically, P represents the true distribution of the data and Q represents the theoretical distribution, the model distribution, or the approximate distribution of P. Assuming that $p(x)$ and $q(x)$ are two probability density functions of the random variable X, the relative entropy between the two is defined to be

$$D[p(x), q(x)] = \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)}, \quad (2)$$

,if and only if $p(x)=q(x)$, $D[p(x), q(x)]=0$.

4.2 How to measure the degree of unobservability

According to the definitions in Section 3, if the anonymous communication protocol π is a specific implementation of the target protocol τ, for the given set O= $\{\{o_{n,i}\}_{i=1}^{M}\}_{n=1}^{N}$, each communication protocol can be modeled as a probability distribution of the data packet on the explicit behavioral features. Assuming that P_π is the probability distribution of the anonymous communication system on the i^{th} matrix eigenvector O_i, similarly, p_τ is the probability distribution of the target protocol on O_i, where O is the explicit behavior feature set of the observed packet after the attacker attacks, the relative entropy between the anonymous communication protocol π and the target protocol τ under state s is defined to be

$$D_s[p_\pi \cdot q_\tau] = \sum_{o_{n,i} \in O_i} p_\pi(o_{n,i} \mid s) \ln \frac{p_\pi(o_{n,i} \mid s)}{q_\tau(o_{n,i} \mid s)}. \quad (3)$$

Of these, is the state of the anonymous communication node, such as handshake state and idle state, etc.

Therefore, based on the relative entropy, we can statistically quantify the degree of unobservability for the communication behavior in the anonymous communication system and the masquerading protocol. When the attacker sees the probability distribution of the explicit behavior characteristics of the anonymous communication protocol on O are completely consistent with that of the target protocol, the relative entropy between them is the smallest, that is, the degree of unobservability for the anonymous communication protocol is the smallest.

Definition 5. Assuming S is the state set of the anonymous communication system node. $\hat{D}$ is defined to be the relative entropy of the anonymous communication system, denoted as

$$\hat{D} = \frac{1}{|S|} \sum_{i=1}^{|S|} (D_i[p_\pi \cdot q_\tau] - D_i[p_\tau \cdot q_\tau]). \quad (4)$$

Definition 6. The maximum relative entropy of the anonymous communication system is defined to be

$$D_M = \max_{s \in S} (D_s[p_\pi \cdot q_\tau]). \quad (5)$$

Definition 7. The degree of unobservability in the anonymous communication system is defined to be

$$d = 1 - \frac{D_M - \hat{D}}{D_M}. \quad (6)$$

Therefore, the degree of unobservability in the anonymous communication system can quantify the degree of distinguishability between the anonymous communication system and other protocols in terms of communication behavior. The value range of d is *[0, 1]*. The smaller the d is, the better the unobservability of the protocol is. When *d=0*, the protocol is completely indistinguishable.

5 Measuring the degree of unobservability for TOR transport layer plug-ins

TOR is the most widely used low-latency anonymous communication system in the world. Currently, TOR anonymous communication system has more than 6,000 routing nodes and 4,000 non-public Bridge nodes, with more than 2 million [19] online users at the same time every day. However, the TOR protocol itself is not strong enough in terms of unobservability, so it is difficult to defend against flow analysis attacks. He Gaofeng et al. from Southeast University proposed two identification approaches to effectively identify the flow of TOR, that is, one is based on TLS fingerprint identification and the other is TOR anonymous communication system flow identification based on packet length distribution. Researchers at home and abroad have proposed various protocol masquerading methods. The TOR project team has also developed various transport layer plug-ins to enhance the unobservability of the protocol. This paper takes TOR's bridging approach and the Meek plug-ins released recently as examples to give the degree of unobservability.

5.1 Bridge mechanism introduction

In order to defend against network censorship and flow analysis attacks, TOR proposed a non-public Bridge mechanism. The Bridge node can only be obtained by application to the TOR mail server through Gmail mailbox or from the TOR official website (https://Bridges.torproject.org). In order to defend against enumeration attacks, TOR restricts each email address to only obtain a certain number of Bridge nodes within a certain time interval. For webpage mode, TOR also restricts each IP address to only obtain a certain number of Bridge nodes within a certain time interval. In addition, in order to defend against flow analysis attacks, TOR's Bridge communication protocol imitates the Firefox browser's TLS handshake process to confuse its traffic characteristics.

5.2 Meek mechanism introduction

Meek is a transport layer plug-in recently released by the TOR project team. Its goal is to disguise TOR traffic as the traffic to access to the cloud platform. When TOR's data traffic reaches the cloud computing platform, it will be forwarded to the TOR anonymous network via TOR's Meek-Server, and finally gets to the real target address. Different from other transport layer plug-ins, it does not imitate a protocol, but runs the target protocol directly. That is, the TOR traffic is encapsulated into the HTTPS payload of Firefox, and the HTTPS protocol header is the protocol header for Firefox to communicate with the cloud platform. After reaching the server of the cloud platform, the HTTPS content is resolved and forwarded to TOR anonymous network via Meek-Server. Therefore, the Meek transport layer plug-in has the following two advantages:

1) Highly concealed. The Meek transport layer plug-in runs the target protocol directly rather than masquerades, so its protocol characteristics are very similar to those of the target protocol.
2) There is no need to directly publish the access point address like the Bridge mechanism. Instead, the front-end domain name mechanism of the cloud platform to is used to allow users to request the domain name of the cloud platform, and then resolve to obtain the IP address. The process is exactly the same as the user's access to Google's search service ,Amazon and Microsoft cloud platform.

5.3 Experimental methods

In order to measure the degree of unobservability for the transport layer plug-in in OST anonymous communication system, we downloaded the latest version of the TOR client software (torbrower-install-4.5.1_zh-CN) from the TOR official website for Windows and Linux platforms, and the built-in Firefox version is Firefox 31.7.0.

Since both Meek and Bridge approaches are masquerading protocols, the method of this experiment is to respectively capture the traffic to visit a website in two TOR modes: Meek-Azure mode and Bridge mode. And then directly access the same URL by Firefox browser. In each mode, the traffic is repeatedly collected 3 times under Windows 7 and Ubuntu respectively. The traffic collected in each mode includes the phases of HTTPS protocol: the handshake phase, the idle phase (waiting for approximately 3 minutes), and the user input data phase (all accessing the same URL). In addition, in order to calculate the degree of unobservability for the Meek and Bridge protocol as well as the disguised Firefox (with an version of Firefox ESR31.7.0), this experiment also collected the URL of the Meek-Azure platform directly accessed by the user using the Firefox browser. The collected traffic also includes the protocol handshake phase, the idle phase (waiting for approximately 3 minutes), and the user input data phase (all accessing the same URL).

In order to measure the c for the TOR transport layer plug-in, we chose the size distribution and packet interval reaching time as the explicit behavior characteristics of the network communication behavior observability. In order to visualize whether the characteristics selected in this paper are distinguishable, we firstly calculated their cumulative distribution function (CDF) for client-to-server (C2S) and server-to-client (S2C) on different platforms (Windows and Ubuntu), as well as the CDF values for direct accessing to HTTPS sites(Azure platform), as shown in Figure 2 and Figure 3. Then, we calculated the relative entropy in different platforms and states according to the approaches proposed in Section 4.2 of this paper, as shown in Table 1 and Table 2. Finally, we gave the measure values for the degree of unobservability in the two TOR modes, respectively Meek mode and the Bridge mode, according to the calculation formula for the degree of z in the anonymous communication system proposed in Section 4.2. In order to show the effect more intuitively, we preprocessed the data and filtered all TCP three-way handshake packets. We made the packet interval accurate to 10ms. In addition, for the case of $P(x) = 0$, we technically assigned it a small value in this paper, with a probability value of one ten-billionth.

5.4 Experimental results and analysis

In this section we will discuss the experimental results and analyze them accordingly. Figure 2 shows the packet interval distribution of Meek, Bridge and HTTPS (HTTPS of Firefox) with direct-access. From Figure 2, we see that there is little difference for 70% to 80% of the packet interval distribution among Meek, Bridge and HTTPS. However, for Meek's packet, there is a significant jitter at 0.5~2s; while for HTTPS, there is about 20% of the packet interval at 10s.

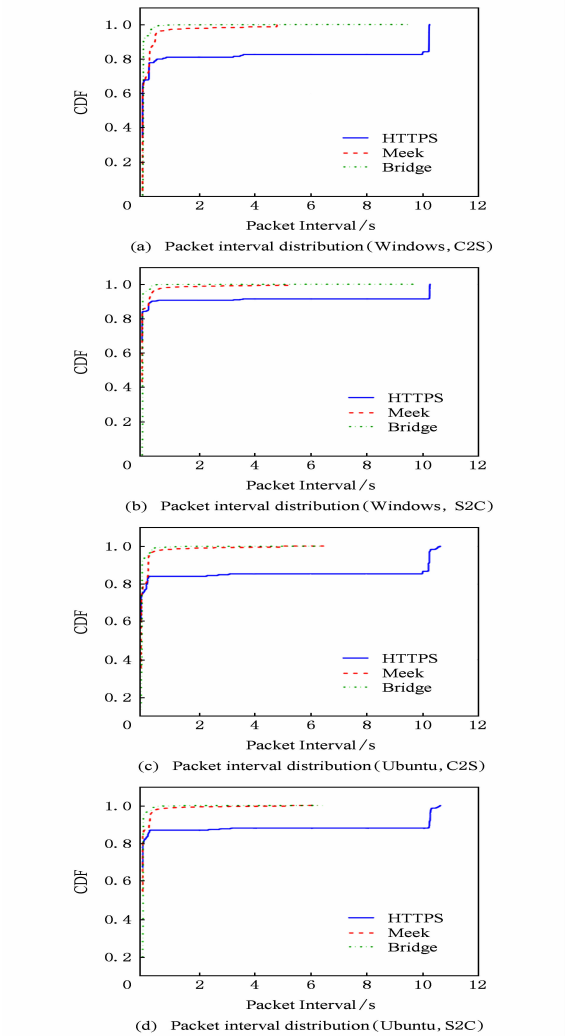

Fig.2 Packet interval distribution

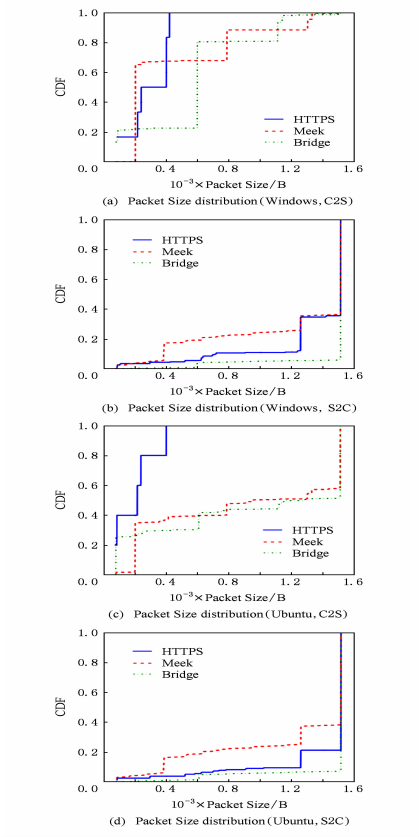

Fig.3 Packet size distribution.

After manually analyzing the source code of TOR, it is found that Meek's client will automatically send a heartbeat packet when the connection is idle in order to maintain a long connection to the cloud platform. The heartbeat packet has a random interval between 0.1 and 5s. When directly access the Azure cloud platform with Firefox, if there is no data, a heartbeat packet will also be sent, and the heartbeat packet interval is 10s.

Figure 3 shows the packet size distribution for Meek, Bridge, and HTTPS (HTTPS of Firefox). From Figure 3, we can see the following two obvious phenomena:

1) Phenomenon 1. In the Bridge mode, the C2S packet size is concentrated at 600B. After analysis, it is found that the TOR packet (cell) is a fixed size, which is 597B under the Windows platform, and 609B under the Ubuntu platform.
2) Phenomenon 2. In the Meek mode, there is a high proportion for C2S packet size of about 200B and S2C packet size of about 400B. After analyzing the source code of TOR, it is found that these are the size of heartbeat packets in TOR's Meek mode. The size of the heartbeat packet is slightly different under different platforms, and there is no difference under the same platform.

Table 1 Relative Entropy for Packet Interval Distribution

Pattern	OS	Handshake		Idle		Data Transmission	
		C2S	S2C	C2S	S2C	C2S	S2C
Meek	Windows	3.56	1.20	18.27	18.14	10.35	5.19
	Ubuntu	2.78	1.63	18.94	18.02	4.29	3.17
Bridge	Windows	18.15	18.47	21.82	21.82	15.64	19.20
	Ubuntu	13.07	13.14	21.85	21.82	14.59	13.76
HTTPS	Windows	1.67	2.05	2.56	4.64	1.96	0.77
	Ubuntu	3.57	2.05	17.03	16.86	1.51	1.54

Table 2 Relative Entropy for Packet Size Distribution

Pattern	OS	Handshake		Idle		Data Transmission	
		C2S	S2C	C2S	S2C	C2S	S2C
Meek	Windows	17.57	4.76	20.91	20.20	19.89	10.86
	Ubuntu	18.14	5.48	10.25	15.01	18.48	5.72
Bridge	Windows	18.00	0.77	21.92	21.92	20.19	4.88
	Ubuntu	18.69	0.73	21.95	21.92	18.44	4.30
HTTPS	Windows	0	0	0	0	0	0.77
	Ubuntu	0	0	0	0	0	1.45

It can be seen directly from Figure 2 and Figure 3 that the TOR's transport layer plug-ins (Bridge and Meek) still statistically have a significant difference from the masquerading protocol (HTTPS). Now, we will calculate the relative entropy for the packet interval distribution and the packet size distribution respectively in TOR's Meek mode, TOR's Bridge mode and HTTPS protocol in different states, as well as the relative entropy of HTTPS compared with itself (the packet captured under the same platform at different times), for example, the relative entropy for C2S packet size distribution and packet interval distribution.

The results in table 1 and table 2 show that relative entropy can well describe the statistical similarity between the target protocol and the masquerading protocol. Compared to the packet interval distribution, the packet size distribution can better characterize the degree of distinguishability of explicit behavioral features for TOR transport layer plug-ins. In this paper, we calculated the relative entropy for packet size distribution and packet interval distribution under the same network environment, the same operation system and the same state, so the different calculation results are comparable.

Therefore, according to formula (6), the degree of unobservability for the Meek and Bridge transport layer plug-ins can be calculated. For the Meek plug-in, the degree of unobservability under Windows platform is $d=0.54$; while $d=0.36$ under the Ubuntu platform. For the Bridge plug-in, the degree of unobservability under the Windows platform is $d=0.72$; while $d=0.53$ under the Ubuntu platform. As can be seen from the above values, the degree of unobservability for the Meek is better that for the Bridge mode no matter under the Windows platform or the Ubuntu platform. That is to say, it is statistically more difficult to distinguish the traffic in Meek mode. However, it still has stronger traffic characteristics compared with HTTPS traffic, that is, it is easy to distinguish the traffic of TOR in both Bridge and Meek modes from HTTPS traffic.

6 Conclusions and future work

This paper formalizes the communication model of anonymous communication system, and proposes an relative entropy to measure the degree of the unobservability in anonymous communication systems based on this model. This method can effectively measure the the degree of the unobservability in anonymous communication systems, and also can be applied to measure the degree of unobservability for TOR transport layer plug-ins. It is also applicable to measure he degree of unobservability in other anonymous communication systems. Although scholars at home and abroad and the TOR project team have been working to improve the security for TOR communication behavior, the theoretical analysis and experimental results in this paper show that the unobservability claimed by the TOR official has not been realized for TOR transport layer plug-ins. At present, the measure analysis on the unobservability in anonymous communication systems proposed in this paper only takes into account the unobservability of the anonymous communication system by means of the information entropy at a static time point. The information entropy provides a good analytical method for the security assessment of the anonymous network itself, but for any given user, it is usually necessary to accurately assess how long it takes and how much the probability is to effectively identify the protocol of anonymous communication systems, thereby further cracking its anonymity. These are of great significance to measure and evaluate the anonymous communication system.

References

[1] Bamford J. The NSA is building the country's biggest spy center (watch what you say) [OL]. [2015-05-23]. http://www.wired.com/2012/03/ff_nsadatacenter

[2] Moghaddam H, Li B, Derakhshani M, et al. SkypeMorph: Protocol obfuscation for tor bridges [C] //Proc of the 19th ACM Conf on Computer and Communications Security. New York: ACM, 2012: 97-108

[3] Weinberg Z, Wang J, Yegneswaran V, et al. StegoTorus: A camouflage proxy for the Tor anonymity system [C] //Proc of the 19th ACM Conf on Computer and Communications Security. New York: ACM, 2012: 109-120

[4] Wang Q, Gong X, Nguyen G T K, et al. CensorSpoofer: Asymmetric communication using IP spoofing for censorship-resistant Web browsing [C] //Proc of the 19th ACM Conf on Computer and Communications Security. New York: ACM, 2012: 121-132

[5] Houmansadr A, Riedl T J, Borisov N, et al. I want my voice to be heard: IP over Voice-over-IP for unobservable censorship circumvention [C] //Proc of the 20th Annual Network and Distributed System Security Symp. Reston, VA: The Internet Society, 2013: 1-17

[6] Chaum D L. Untraceable electronic mail, return addresses, and digital pseudonyms [J]. Communications of the ACM, 1981, 24(2): 84-90

[7] Dingledine R, Mathewson N, Syverson P. Tor: The second-generation onion router [C] // Proc of the 13th Conf on USENIX Security Symp. Berkeley, CA: USENIX Association, 2004: 303-320

[8] Department of Business Informatics, University of Regensburg. JAP: The JAP anonymity & privacy homepage [OL]. [2015-05-23]. http://anon.inf.tu-dresden.de

[9] Pfitzmann A, Hansen M. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management [OL]. [2015-05-23]. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf

[10] Houmansadr A, Brubaker C, Shmatikov V. The parrot is dead: Observing unobservable network communications [C] //Proc of the 34th IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2013: 65-79

[11] Geddes J, Schuchard M, Hopper N. Cover your ACKs: Pitfalls of covert channel censorship circumvention [C]//Proc of the 20th ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 361-372

[12] Reiter M K, Rubin A D. Crowds: Anonymity for Web transactions [J]. ACM Trans on Information and System Security (TISSEC), 1998, 1(1): 66-92

[13] Berthold O, Pfitzmann A, Standtke R. The disadvantages of free MIX routes and how to overcome them [G] //Designing Privacy Enhancing Technologies. Berlin: Springer, 2001: 30-45

[14] Diaz C, Seys S, Claessens J, et al. Towards measuring anonymity [C] //Proc of the 2nd Int Conf on Privacy Enhancing Technologies. Berlin: Springer, 2003: 54-68

[15] Hamel A, Gregoire J, Goldberg I. The misentropists: New approaches to measures in TOR. CACR 2015-18 [R]. Waterloo, Ontario, Canada: University of Waterloo, 2011

[16] Dingledine R. Obfsproxy: The next step in the censorship arms race [OL]. [2015-05-23]. https://blog.torproject.org/blog/obfsproxy-next-step-censorship-arms-race.

[17] Tor Project. Tor Meek [OL]. [2015-05-23]. https://trac.torproject.org/projects/tor/wiki/doc/meek

[18] Dyer K P, Coull S E, Ristenpart T, et al. Protocol misidentification made easy with format-transforming encryption [C] //Proc of the 20th ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 61-72

[19] Tor Project. Tor Metrics [OL]. [2015-05-23]. https://metrics.torproject.org

[20] He Gaofeng, Yang Ming, Luo Junzhou, et al. Online identification of Tor anonymous communication traffic [J]. Jornal of Software, 2013, 24(3): 540-556 (in Chinese)
(何高峰, 杨明, 罗军舟, 等. Tor匿名通信流量在线识别方法[J]. 软件学报, 2013, 24(3): 540-556)

Tan Qingfeng, born in 1981, PhD candidate, assistant professor. Member of China Computer Federation. His main research interests include anonymous communication and covert communications.

Shi Jinqiao, born in 1978, PhD, professor, PhD supervisor. Member of China Computer Federation. His main research interests include network and information security, especially privacy enhancing techniques, and data leakage detection.

Fang Binxing, born in 1960, Professor, and PhD supervisor. Academician of Chinese Academy of Engineering. His current research interests include computer architecture, computer network and information security.

Guo Li, born in 1969, Professor, and PhD supervisor. Member of China Computer Federation. Her research interests include network information security.

Zhang Wentao, born in 1989. Master. Member of China Computer Federation. His main research interests include anonymous communications and information retrieving.

Wang Xuebin, born in 1986. Master. Member of China Computer Federation. His current research interests include visual analytic, computer network and information security.

Wei Bingjie, born in 1987. Recevied her PhD degree in compute software and theory from Institute of Computing Technology, Chinese Academy of Sciences. Her main research interests include microblog retrieval.