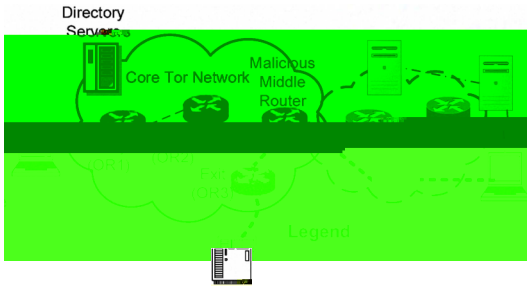
empirical evaluation of the effectiveness of Tor bridges resisting censorship in this paper. We conducted an extensive theoretical analysis on two bridge-discovery approaches and our experimental results demonstrate the effectiveness of large-scale bridge discovery in real-world environments. Although there is one related work on discovering bridges [4], [5], the

of 10MB/s, we discovered 2369



Algorithm 3 Bandwidth Weighted Node Selection Algorithm

Require:

(a) B, the total bandwidth of the nodes in the node list (b) BE, the total bandwidth of the exit nodes in the node list (c) Be, the total bandwidth of the guard nodes in the node list (d) q, the number of the nodes in the list (e) b[i], the bandwidth of the i^{th} node in the list (f WE, the weight of the exit nodes (g) We, the weight of the guard nodes (h) bw, the weighted bandwidth of the nodes (i) totalbw, the totally weighted bandwidth of the nodes G) randbw, the random sampling bandwidth value from the totalbw

Ensure: Find a suitable node from the node list

```
Derive a list of qualified running nodes
Count B, BE and Be
if try to find a exit node then
    WE = 1
else
    WE = 1 - B/(3 × BE)
end if
if try to find a guard node then
    We = 1
else
    We = 1 - B/(3 × Be)
end if
if WE < 0 then
    WE = 0
end if
if We < 0 then
    We = 0
end if
for i = 1 : q do
    if the node is both exit and guard node then
        bw = b[i] × We × WE
    else if the node is entry then
        bw = b[i] × We
    else if the node is exit then
        bw = b[i] × WE
    else
        bw = b[i]
    end if
    totalbw = totalbw + bw
end for
Randomly sample a bandwidth randbw fom totalbw
for j = 1 : q do
    if the node is both exit and guard node then
        temp = temp + b[i] × We × WE
    else if the node is entry then
        temp = temp + b[i] × We
    else if the node is exit then
        temp = temp + b[i] × WE
    else
        temp = temp + b[i]
    end if
    if temp > randbw then
        return the i^th node
    end if
end for
```

Algorithm 4 Selection of an Entry l iddle Node

```
Derive a list of qualified running nodes
if Bandwidth or a guard node is required then
    Use a bandwidth weighted Algorithm 3 to choose one;
    (this is the default branch)
else
    Choose the middle node randomly
end if
```

selecting

$\frac{Bi_E}{+B}$

path. Once the circuit is established, the client can connect to a web server through the circuit.

C. Brid e Clients Usin Tor

In order to access Tor, a bridge client needs to obtain a least one bridge. As we can see fom Figure 1, the bridge client can obtain the information of bridges via email and https. We will further discuss these methods in Section III. The bridge client uses a bridge as a hidden frst-hop relay into the Tor network to avoid censorship. The bridge client then follows the similar procedures discussed earlier, i.e., downloading the information of Tor nodes and choosing the appropriate exit onion router OR3 and middle onion router (e.g., malicious middle router in Figure 1), respectively. Finally, the bridge client creates a circuit and anonymously surf the Internet.

III. THREE ApPROACHES FOR LARGE-SCALE TOR BRIDGE DISCOVERY

In this section, we frst introduce the basic ideas of discovering Tor bridges and then present our experimental strategies.

A. Basic Ideas

In this paper, we investigate the following two categories of approaches to discover bridges:

(i) Email and https enumeration. An attacker can use a Yahoo or gmail account to send an email to the bridge email server (*brid es@torproject.or*) with the line "get bridges" in the body of the mail. The bridge email server promptly replies with three distinct bridges. To avoid malicious enumeration, the bridge email server only replies one email to an email account each day. Alternatively, the user can access the bridge website (https:/lbridges.torproject.org/) to obtain three bridges. To avoid malicious enumeration, the https server distributes three bridges to each 24-bit IP prefx each day as well.

(ii) Brid es inference by malicious Tor middle routers. A circuit created by a bridge client traverses both bri Ù it

circuit creation mechanism by using two commands, "SETCONF _ DisablePredictedCircuits=I" and "SETCONF MaxOnionsPending=0". We then use the command "EXTENDCIRCUIT CircuitlD ServerSpec *(, ServerSpec)" to establish our custom circuits. If "CircuitlD" is zero, it is a request that Tor build a new circuit along the specifed path. Otherwise, it is a request that the server should extend an existing circuit with that ID along the specifed path. Note that "ServerSpec" is the nickname of the specifed Tor node. In this way, we can control the Tor network to create a two-hop circuit via distinct exit nodes. Once the circuit is created, the tool *wget* is used to download bridge web pages.

D. Discovering Bridges Via Tor Middle Routers

Figure 1 illustrates the basic idea of discovering Tor bridges via middle Tor routers. We deploy malicious Tor middle routers on PlanetLab to discover bridges connected to these Tor middle routers. Recall a client uses a bridge as an entry node to establish a three-hop circuit for surfng the Internet. Traffc forwarded by the bridges may traverse these middle routers. Then the middle routers can identify the IP addresses of the bridges. The recorded IP addresses will be either fom Tor entry guards or fom bridges. Because the information of entry guards is public, it is trivial to distinguish bridges from entry nodes. We modifed the Tor source code to embed the aforementioned functions, record the incoming connection information, differentiate bridges fom other Tor nodes, and send an email with bridge IPs to us. This approach allows us to automatically retrieve bridges via the controlled Tor middle routers on PlanetLab. Of course, such malicious middle routers can be deployed at any place, including the researchers' home and Amazon EC2 [17]. PlanetLab nodes have very limited bandwidth while home and Amazon EC2 nodes may provide large bandwidth.

Notice that we need to prevent malicious routers fom becoming entry or exit routers automatically because of the rule of Tor. When onion routers advertise an uptime and bandwidth at or above the median among all routers, these routers will be marked as entry guards by directory servers [7]. To prevent malicious routers from becoming entry routers, we need to reduce their bandwidth or control their uptime. By confguring the exit policy, we also prevent those malicious routers from becoming exit routers.

IV. Analysis

In this section, we frst analyze the efectiveness of the bridge discovery via emails and https. We formalize the bridge discovery problem as a weighted coupon collector problem and derive the expected number of samplings for obtaining all bridges. We then analyze the effectiveness of the bridge discovery approach via malicious Tor middle routers.

A. Bridge Discovery via Emails and HTTPS

The approaches that enumerate bridges via emails and https can be formalize as a weighted coupon collector problem. In the classical coupon collector problem [18], all n coupons are obtained with an equal probability of $\frac{1}{n}$. To derive the

$$\frac{B_i}{\sum_{i=1}^{n}}$$

the weighted bandwidth routing algorithm discussed in Section II-B, the bandwidth weight can be derived by,

$$W_E = \begin{cases} 1 - \frac{B+k\cdot b}{3\cdot(B_{exit}+B_{EE})} & : \quad WE > 0, \\ 0 & : \quad WE \quad 0. \end{cases} \tag{4}$$

$$W_G = \begin{cases} 1 - \frac{B+k\cdot b}{3\cdot(B_{entry}+B_{EE})} & : \quad We > 0, \\ 0 & : \quad We \quad 0. \end{cases} \tag{5}$$

The weighted bandwidth $Bexi_t'$, BEE', Ben_try' and $BN\text{-}EE$, can be derived as follows,

$$Bexi_t' = Bexi_t \cdot WE, \tag{6}$$

$$BEE' = BEE \cdot WE \cdot We, \tag{7}$$

$$Ben_try' = Ben_try \cdot We, \tag{8}$$

$$BN\text{-}EE' = BN\text{-}EE + k \cdot b. \tag{9}$$

With the total weighted bandwidth $Bexi_t' + BEE' + Ben_try' + BN\text{-}EE'$ derived above and the total bandwidth of malicious Tor middle routers $k \cdot b$, according to the weighted bandwidth route selection algorithm in Section II-B (the total bandwidth of malicious Tor middle routers divided by the total weighted bandwidth is the probability that malicious middle nodes are chosen for serving circuit), we have the following theorem for calculating the catch probability.

Theorem 2. *The* **catch probability** *can be derived by*

$$P(k,b) = \frac{k \cdot b}{Bexi_t' + BEE' + Ben_try' + BN\text{-}EE,}, \tag{10}$$

where $k = 1,2,3\ldots$ *and* $0 < b < 10MB/s$.

Theorem 2 is intuitive based on the bandwidth weighted path selection algorithm. From Theorem 2, we derive the following corollaries.

Corollary 1. *The catch probability increases with the number of malicious Tor middle routers.*

$$P(r,b) > P(k,b), \text{ where } r > k. \tag{11}$$

Corollary 2. *The catch probability increases with the bandwidth of malicious Tor middle routers, i.e.,* $P(k,b)$ *is a monotonous increasin function in terms of* b.

$$P(k,l) > P(k,b), \text{ where } l > b. \tag{12}$$

The proof of Corollary 1 and Corollary 2 is given in Appendix B of [19]. These two corollaries indicate that the catch probability increases with both the number of malicious Tor middle routers and the bandwidth of malicious Tor middle routers. This is not a surprise.

We would like to know what affects the catch probability, the number of malicious middle routers or the aggregated bandwidth of malicious middle routers. This is important in practice because we may not have so many computers with di f erent IPs. Theorem 3 answers this question.

Theorem 3. *The catch probability is determined by the a re ated bandwidth contributed by malicious Tor middle routers. That is, if* $M = k \cdot b$, $M' = k' \cdot b'$, *and* $M \geq M'$,

$$P(M) \geq P(M'). \tag{13}$$

The equality holds when $M = M'$.

The proof of Theorem 3 is given in Appendix C of [19]. Theorem 3 implies that an attacker may not need to inject many malicious middle routers into the Tor network. A middle router with large bandwidth can achieve the same catch probability as a number of middle routers with small bandwidth. Our experiments in Section V-B validate this observation.

In practice, we also want to know the catch probability vs. the number of created circuits. Theorem 4 answers this question.

Theorem 4. *After* q *circuits are created, the catch probability that at least one brid e connects to one of the malicious* k *routers of bandwidth* b *can be derived by*

$$P(k,b,q) = 1 - (1 - P(k,b))^{Q}, \text{ where } q = 1,2,3,\ldots. \tag{14}$$

Theorem 4 is intuitive. From Theorem 4, we have Corollary 3.

Corollary 3. *The catch probability increases with the number of created circuits.*

$$P(k,b,h) > P(k,b,q), \text{ where } h > q. \tag{15}$$

The proof of Corollary 3 is given in Appendix D of [19].

We also derive the relationship among the catch probability, the total bandwidth of the malicious Tor middle routers and the number of created circuits based on Equation (14).

Corollary 4. *After* q *circuits are created, the probability that at least one brid e connects to one of the malicious routers with the total bandwidth of the Tor nodes* M *can be derived by*

$$P(M,q) = 1 - (1 - P(M))^{Q}. \tag{16}$$

Corollary 4 is intuitive given Theorem 4. From Theorems 3 and 4, we also derive Corollary 5. The proof is given in Appendix D of [19].

Corollary 5. *The catch probability increases with the total bandwidth of the malicious Tor middle routers.*

$$P(M,q) > P(M',q), \text{ where } M > M'. \tag{17}$$

In summary, the catch probability increases with two factors: the total bandwidth of malicious Tor middle routers and the number of created circuits. Our experimental data in Sed

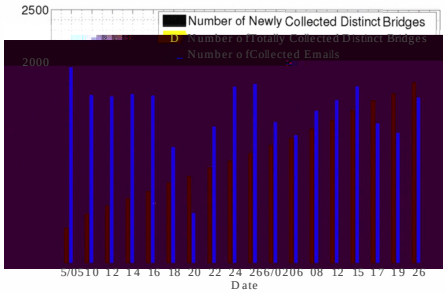


Fig. 2. Discovered Bridges via Emails

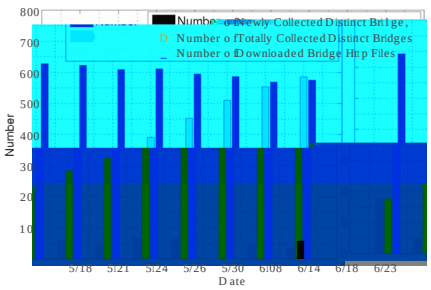


Fig. 3. Discovered Bridges via HTTPS

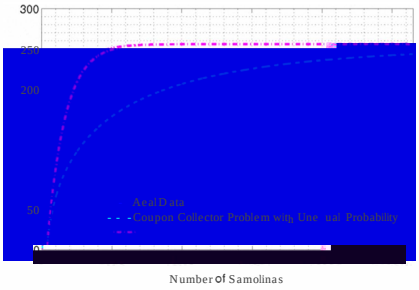


Fig. 4. Number of Samplings v.s. Number of Distinct Bridges

A. Bridge Enumeration via Email and HTTPS

To evaluate bridge-discovery approaches via emails and https, we conducted large-scale experiments from PlanetLab from May to June, 2010. Figure 2 gives the number of newly collected distinct bridges, number of totally collected distinct bridges, and number of collected emails over the time. Because the Yahoo SMTP server may not successfully deliver emails sent from PlanetLab, we could not receive all replies all the time. From Figure 2, we can see that more emails produce more distinct bridges. On May 5, 2010, we received 2000 emails and collected more bridges than other dates.

Figure 2 also shows that the number of totally collected bridges increases over time. Actually, we are told that Tor has more than 10,000 bridges! This is the reason why the number of bridges keeps increasing steadily. However, this set of experiments show that the discovery approach works effectively because it discovered the new bridges. To enumerate all bridges, we only need to continue experiments. Figure 3 gives the data obtained via https. The number of discovered bridges via https has a similar trend to that in Figure 2.

We now verify Theorem 1 in Section IV using real-world data. Recall that Tor distributes different pools of bridges (there is crossover among pools) via email and https servers over time. However, experiments on a certain day can be formulated as a weighted coupon collector problem because the pool has not been shifed. One retrieved bridge can be treated as one sampling. Figure 4 shows the relationship between the number of samplings and the number of distinct bridges. It can be observed that the curve of the not-weighted classical coupon collector problem is much steeper than the curve for the real data at the beginning. This implies that the bridges are not distributed uniformly.

In order to verify that the bridge distribution is a weighted coupon collector problem, we assume that the bridge bandwidth distribution is similar to the public Tor router bandwidth distribution. We randomly pick up a set of public Tor routers and use their bandwidth to simulate bridge bandwidth (note that we do not know bridge bandwidth). We then obtain the curve of the weighted coupon collector problem based on Equation (3). Figure 4 shows that the theoretical curve is only slightly lower than the real data generated curve. Hence, it is highly possible that bridges are distributed with their bandwidth as the weight. Such a weighted distribution is also consistent with Tor's weighted routing algorithm for performance enhancement. Actually, Tor developers later confrmed this fact to us.

B. Bridge Discovery via Tor Middle Routers

Figure 5 shows the public Tor router bandwidth cumulative distribution function on July 10, 2010. There were 1604 active Tor routers, including 326 pure entry onion routers, 525 pure exit onion routers, and 132 EE routers. Using the real-world data, Figure 6 shows the relationship between the theoretical catch probability and the number of controlled Tor middle routers based on Theorem 2. As we can see, the catch probability is 14.7% when 512 Tor middle routers with bandwidth 50KB/s are used, i.e., $P(512,50) = 14.7\%$. Based on Theorem 4, Figure 7 illustrates the catch probability when the bridge clients create q circuits, that is $P(512,50,q)$. From Figure 7, we can see that in theory, the catch probability approaches 99%, after the bridge clients created 30 circuits, i.e., $P(512,50,30) \approx 99\%$. In addition, from Equation (17), we know that by only confguring three nodes as malicious Tor middle routers, we can obtain the catch probability $P(3*10000,30) > P(512*50,30) \approx 99\%$.

To demonstrate the correctness of our theory, we used 512 PlanetLab nodes as malicious Tor middle routers and set their bandwidth as 50KB/s (because of the limited bandwidth of PlanetLab nodes). To avoid afecting the Tor network, we only conducted a short experiment for 2 days. We set up a Tor client to create 430 circuits via our own Tor bridge in an apartment. We found that the 21^{th} circuit passed through our Tor middle routers in PlanetLab. The experimental results match the theoretical results above well.

We now show data supporting the fact that high bandwidth routers have a higher chance to be selected as middle routers. Figure 8 gives the empirical cumulative distribution function (ECDF) of the bandwidth of onion routers selected as middle routers for these 430 circuits. Recall that we are able to record routers selected for a circuit at the client. We can see that 60% of those routers have a bandwidth more than IMB/s. However, as shown in Figure 5, only 10% of Tor routers have a bandwidth of larger than IMB/s. This implies that the higher bandwidth the routers have, the higher chance these routers are selected as middle routers for serving circuits.

Figure 9 illustrates the theoretical catch probability that a circuit passes malicious Tor middle routers in terms of router bandwidth advertisement and the number of malicious middle routers, based on Theorem 2. We can see that the theoretical probability is monotonously increasing with both the number of controlled middle routers and their bandwidth advertisement. These observations match our analytical results in Theorems 1 and 2 well. As expected, the catch probability

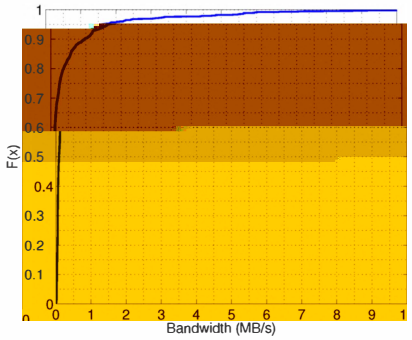


Fig. 5. Empirical CDF of Bandwidth of All Routers in the Tor Network

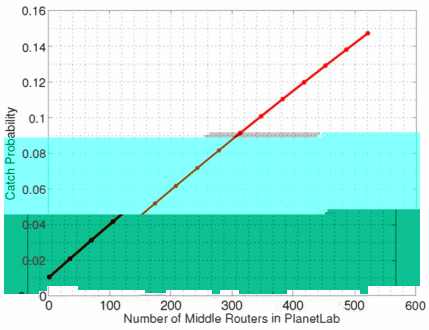


Fig. 6. Probability that a Circuit Chooses the Middle Routers in PlanetLab vs. Number of Tor Middle Routers in PlanetLab

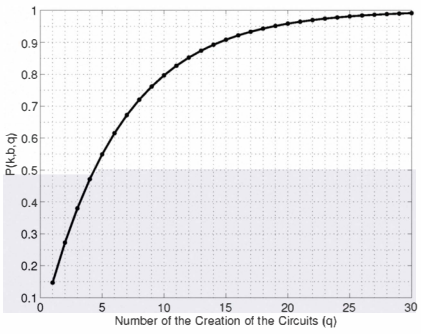


Fig. 7. Probability That at Lea t a Circuit Traverses Through the Controlled Middle Routers After Bridge Clients Create q Circuits

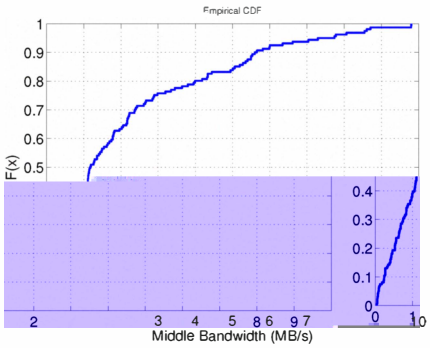


Fig. 8. Empirical CDF of Bandwidth of Selected Middle Routers

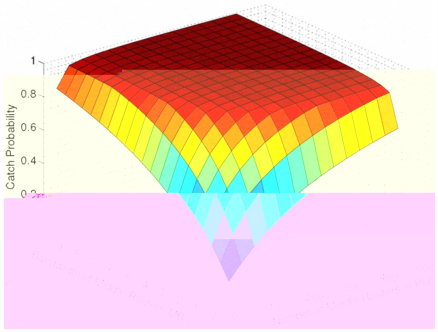


Fig. 9. Probability that a Circuit Chooses the Tor Middle Routers in PlanetLab vs. Number of Tor Middle Routers & Bandwidth Advertisement of Tor Middle Routers in PlanetLab

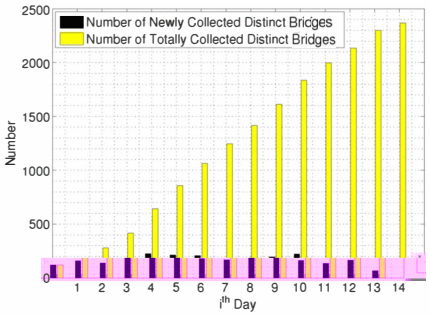


Fig. 10. Discovery Bridges via a Tor Middle Router

approaches 90% when there are 20 malicious middle routers with 10MBIs bandwidth, i.e., $P(20, 10000) \approx 90\%$.

To verify Theorem 3 that the bridge discovery is determined by the aggregated bandwidth of the malicious Tor nodes, we confgured a high bandwidth middle router of 10MB/s. We recorded the bridges that pass through this middle router from July 10 to 23, 2010. Figure 10 gives the number of newly discovered distinct bridges and the number of totally collected distinct bridges. The number of totally collected bridges increases over time. Eventually, we obtained 2369 bridges, indicating that discovering bridges via middle routers can be very effective and effcient and the catch probability is mainly determined by total bandwidth contributed by malicious middle routers. Notice that to prevent the middle router from becoming an entry router in 7 days, we restarted the router on the 6th day.

VI. DISCUSSION OF COUNTERMEASURES

We have demonstrated the impact of our bridge-discovery approaches. We now discuss possible countermeasures against these malicious discovery. We note that those countermeasures can reduce the effectiveness of the malicious bridge discovery, but none of them will be an ultimate solution.

(i) Explore the human interactive proofs (HIPs) method to resist automatically lar e-scale brid e discovery: CAPTCHA as a known HIP-based approach could be used to increase di f culty to large-scale automatic bridge enumeration through emails and https. A CAPTCHA consists of a visual challenge in the form of alphanumeric characters, which are distorted in such a way that available computer vision algorithms have di f culty segmenting and recognizing the text. The Tor bridge https and email server may adopt CAPTCHA to this end. However, there have been various e f orts in automatically deciphering CAPTCHA [21].

(ii) Desi n new router selection strate ies: As we discussed, the weighted bandwidth routing strategy speeds up the bridge discovery through controlled Tor middle onion routers. One naive approach to reduce the impact of malicious Tor middle routers is e% m R̄v" " ⨛ therT! routing strategy a

volunteer to act as bridges. Nevertheless, routing schemes based on DHT have security issues as well [22], [23], [24].

VII. RELATED WORK

Because of space limit, we only review the most related work. McLachlan *et al.* [5] investigated the weakness of current bridge architecture, leading to a few advanced attacks on the anonymity of bridge operators. Their results indicate that the existing attacks may expose clients to additional privacy risks and Tor exit routers should be considered as sharing a single IP prefx that is mentioned in the bridge design [20]. Vasserman *et al.* [4] presented the attacks against Tor bridges and discussed countermeasures using DHT based overlay networks. Bauer *et al.* [25] showed that an adversary who controls only 6 malicious Tor routers can compromise over 46% of all clients' circuits in an experimental Tor network with 66 total routers. Edman *et al.* [26] identifed the risk associated with a single autonomous system (AS), which observes both ends of an anonymous Tor connection is greater than previously thought. Their results showed that the growth of the Tor network had only a small impact on the network's robustness against an AS-level adversary.

VIII. CONCLUSION

In this paper, we conducted extensive analysis and large-scale empirical evaluation on Tor bridge discovery via email, https and malicious Tor middle routers. To discover bridges automatically, we developed a command-and-control architecture on PlanetLab to send emails via Yahoo SMTP to the bridge email server and download bridge webpages fom the bridge web server, respectively. We formalized the email and https bridge discovery process as a weighted coupon collector problem and analyzed the expected number of retrieved bridges with a number of samplings. We also exploited the Tor weighted bandwidth routing algorithm and studied the bridge discovery via malicious Tor middle onion routers deployed on PlanetLab and in an apartment. We formally analyzed the catch probability of discovering bridges via middle onion routers. Our real-world implementation and large-scale experiments validated the effectiveness and feasibility of the three bridge discovery approaches. **We have discovered 2365 Tor bridges via email and https and 2369 bridges by only one controlled Tor middle router in 14 days.** Our study shows that the bridge discovery approach based on malicious middle routers is simple, effcient and effective to discover bridges with little overhead. We also discussed potential mechanisms to counter bridge discovery.

ACKNOWLEDGEMENT

This work was supported in part by National Key Basic Research Program of China under Grants 201OCB328104, National Natural Science Foundation of China under Grants 60903162, 60903161, 61070158, 61070161, 61003257, China Specialized Research Fund for the Doctoral Program of Higher Education under Grants 200802860031, Jiangsu Provincial Natural Science Foundation of China under Grants BK2008030, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants 93K-9, and USA NSF grants 0942113, 0958477, 0943479 and CNS-1117175 and the Army Research Laboratory under grant W911NF-II-I-0193. Any opinions, fndings, conclusions, and recommendations in this paper are those of the authors and do not necessarily refect the views of the funding agencies.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in Prceedings of the 13th USENI Security Symposium, August 2004.

[2] "Tor and censorship: lessons learned," https://blog.torproject.orglblogltor-and-censorship-lessons-learned, 2010.

[3] J. B. Kowalski and K. Gabert, "Tor network status," http:/torstatus.blutmagie.del, 2010.

[4] E. Vasserman, R. Jansen, J. Tyra, N. Hopper, and Y. Kim, "Membership-concealing overlay networks," in Prceedings of the 16th ACM conference on Computer and communications security (CCS), November 2009.

[5] J. McLachlan and N. Hopper, "On the risks of serving whenever you surf: Vulnerabilities in Tor's blocking resistance design," in Prceedings of the Workshop on Privacy in the Electronic Society (WPES), November 2009.

[6] The Trustees of Princeton University, "Planetlab - an open platform for developing, deploying, and accessing planetary-scale services," http:/www.planet-Iab.orgl, 2010.

[7] R. Dingledine and N. Mathewson, "Tor directory protocol, version 3," http:/tor.ef.orglsvn/trunk/doc/spec/dir-spec.txt, 20I0.

[8] ——, "Tor path specifcation," http:/tor.ef.orglsvn/trunk/doc/spec/path-spec.txt, 2008.

[9] "Imacros," http:/www.iopus.comlimacros/. 2010.

[10] "Relays in the tor network," http:/metrics.torproject.orglconsensus-graphs.html, 2010.

[11] B. N. Chun, "Pssh," http:/www.theether.orglpssh/, 2010.

[12] M. A. Muquit, "Mailsend - send mail via smtp protocol," http:/www.muquit.comlmuquitsofwarelmailsend/mailsend.html, 2008.

[13] M. Lambers, "Mpop: A pop3 client," http:/mpop.sourceforge.net, 2010.

[14] "Freepops," http:/www.freepops.orglen/, 2010.

[15] R. Dingledine and N. Mathewson, "Tc: A tor control protocol (version I)," https:/svn.torproject.orglsvn/tor/trunk/doc/speclcontrol-spec.txt, 2010.

[16] "Vidalia," http:/www.torproject.orglvidalia, 2010.

[17] "Amazon.com: Amazon elastic compute cloud (Amazon EC2)," http:/aws.amazon.comlec2/, 2010.

[18] P. Berenbrink and T. Sauerwald, "The weighted coupon collector's problem and applications," in Prceedings of the 15th Annual Interational Conference on Computing and Combinatorics, 2009.

[19] Z. Ling, X. Fu, W. Yu, J. Luo, and M. Yang, "Extensive analysis and large-scale empirical evaluation of tor bridge discovery," http:/www.cs.uml.edu/ xinwenfu/paperlBridge.pdf, University of Massachusetts Lowell, Tech. Rep., 2011.

[20] R. Dingledine and N. Mathewson, "Design of a blocking-resistant anonymity system," https:/svn.torproject.orglsvn/projects/design-paper/blocking.html, 2008.

[21] "Captcha king," http:/www.captchaking.coml. 2010.

[22] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, "Denial of service or denial of security? how attacks on reliability can compromise anonymity," in Proceedings of the 14th ACM Conerence on Computer and Communications Security(CCS), October 2007.

[23] P. Mittal and N. Borisov, "Information leaks in structured peer-to-peer anonymous communication systems," in Prceedings of the 15th ACM Conference on Computer and Communications Security(CCS), October 2008.

[24] A. Tran, N. Hopper, and Y. Kim, "Hashing it out in public: Common failure modes of dht-based anonymity schemes," in Proceedings of the Workshop on Privac in the Electrnic Society (WPES), November 2009.

[25] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-resource routing attacks against tor," in Proceedings of the Workshop on Privacy in the Electrnic Society (WPES), Washington, DC, USA, October 2007.

[26] M. Edman and P. F. Syverson, "As-awareness in tor path selection," in Prceedings of the *2009* ACM Conference on Computer and Communications Security (CCS), November 2009.