Persona: Network Layer Anonymity and Accountability for Next Generation Internet

Yannis Mallios[1], Sudeep Modi[1], Aditya Agarwala[2], and Christina Johns[2]

[1] Carnegie Mellon University, Information Networking Institute, Pittsburgh PA, USA
imallios@andrew.cmu.edu, sdmodi@andrew.cmu.edu
[2] Carnegie Mellon University, Dept. of Electrical and Computer Eng., Pittsburgh PA, USA
adityaag@andrew.cmu.edu, cjohns@andrew.cmu.edu

Abstract. Individual privacy has become a major concern, due to the intrusive nature of the services and websites that collect increasing amounts of private information. One of the notions that can lead towards privacy protection is that of anonymity. Unfortunately, anonymity can also be maliciously exploited by attackers to hide their actions and identity. Thus some sort of accountability is also required. The current Internet has failed to provide both properties, as anonymity techniques are difficult to fully deploy and thus are easily attacked, while the Internet provides limited level of accountability. The Next Generation Internet (NGI) provides us with the opportunity to examine how these conflicting properties could be efficiently applied and thus protect users' privacy while holding malicious users accountable. In this paper we present the design of a scheme, called Persona that can provide anonymity and accountability in the network layer of NGI. More specifically, our design requirements are to combine these two conflicting desires in a stateless manner within routers. Persona allows users to choose different levels of anonymity, while it allows the discovery of malicious nodes.

1 Introduction

Advances in Information and Communication Technologies (ICT) that make data collection and processing fast and efficient, have brought privacy protection to the spotlight. For that reason several anonymity mechanisms have been proposed and implemented [1]. Most of these mechanisms rely on providing anonymity in the higher network layers, like the application or transport layer, while for efficiency and usability reasons they use weaker mechanisms for anonymity protection (e.g. no use of dummy traffic). This however can introduce greater threats to anonymity. For example TOR [2], one of the most popular anonymizing networks, has been proven vulnerable to several attacks that could degrade the level of anonymity provided [3,4].

The disadvantage of making use of the application layer to provide anonymity is that applications are not necessarily bound to using the anonymity service. It is possible to circumvent the anonymizing procedures by directly making use of the functionality of the lower layers. For example, a javascript or flash file embedded in an html page could initiate another connection to a third party server without using the anonymizing application, which could reveal the user's IP address. Thus, it is important to apply

D. Gritzalis and J. Lopez (Eds.): SEC 2009, IFIP AICT 297, pp. 410–420, 2009.

anonymizing procedures at the lowest networking layer possible, so as to avoid application bypasses and lower layer attacks. However, due to the structure of today's Internet, there is no straightforward implementation. For that reason, Next Generation Internet (NGI) can be used as a point of reference and infrastructure, so as to design and explore an efficient and effective anonymity solution. Despite the need for privacy and anonymity, there is also a need for some sort of accountability not only for security purposes but also for purposes such as billing, management, measurement, etc [5].

In this paper we present a solution that incorporates both privacy and accountability in the network layer, in the context of NGI. We introduce Persona, a scheme that describes the design of a network layer which provides routing and addressing services in a manner which ensures that packets are routed and delivered with the highest level of anonymity between the communicating parties. Finally, if required, Persona can be used to reveal anonymity in an appropriate manner, thereby providing the right degree of accountability as required. We must mention that in this paper we decided to focus our design requirements on combining the conflicting desires of anonymity and accountability in the network layer. However, anonymity could be also applied as an overlay in a higher networking layer, as used today. Answering the question whether anonymity techniques should be applied in the network layer, or in an overlay in a higher networking layer, is out of the scope of this paper. Our work is inspired by research like Accountable Internet Protocol [6], and SIFF [7], at least in the context of discussion about NGI. More specifically, we do not refer to the NGI as the means of a new radical design proposal for the internet; rather we try to improve the current network layer, by adding the components necessary to meet the requirements of both anonymity and accountability. It is for these changes that we refer to NGI.

In this paper we make two major contributions. First, we introduce a novel approach to provide anonymity in the network layer. This paper is the first one, to our knowledge, that provides anonymity per packet, rather than per session, and in stateless manner in the routers. Second, we discuss how this approach is applicable to the NGI and we show how our novel approach provides accountability in case of misbehaving nodes. The paper is organized as follows. In section 2 we provide definitions of the relevant terms, and we elaborate on our assumptions, the attacker model and the attacks that our scheme defends against. In Section 3 we present the design and functionality of Persona, while in section 4 we discuss the evaluation of our scheme in terms of anonymity, efficiency and applicability. In section 5 we discuss some related work and finally, in section 6 we present some ideas for further research.

2 Definitions, Assumptions and Attacker Model

Anonymity is a concept that has received wide research attention, due its ability to protect privacy. For that reason a precise set of formal definitions has been proposed for the concepts of anonymity and its relevant terms [8]. In this context, anonymity of a subject is defined as the property by which the subject is not identifiable within a set of subjects, the anonymity set [8]. Since most communications are a bi-directional, anonymity is often distinguished to sender and receiver anonymity. Sender anonymity is achieved when it is not possible to identify the sender within a set of possible subjects

that sent the message. Similarly receiver anonymity is achieved when it is not possible to identify the receiver of the message within a set of possible receivers [8]. Another important term relevant to anonymity is pseudonymity which is defined as the use of pseudonyms as IDs. Pseudonyms refer to identifiers of a subject other than one of the subject's real names [8]. An advantage of pseudonymity over the previous terms is that accountability for misbehavior can be enforced. Accountability can be defined as the state of a subject being held responsible for a certain action taken by that subject. It is easy to see that if a subject is held accountable for a particular action, that subject is no longer anonymous.

Our proposed architecture is described within the context of some assumptions. The first set of assumptions is with regards to the network infrastructure. Specifically, we assume that ISPs will not have any legacy systems or routers. All links between the sender and receiver are assumed to go through new hardware that supports our solution. As far as the hardware infrastructure is concerned we make the following assumptions. First, we assume routers that have the computational power to perform encryption and decryption with symmetric keys in hardware. Additionally we assume that routers will come equipped with a Trusted Platform Module (TPM). A TPM is a microcontroller that stores keys, passwords and digital certificates [9]. This is a safe assumption considering most new desktops and laptops already come equipped with these. Finally, we assume that TPM units are actually as secure and tamper resistant as they are claimed to be. We do not try and define a secure TPM protocol but assume the ones defined are secure and work as described [9]. It must be mentioned that the assumption for hardware capabilities of encryption and decryption, is a weak assumption; the architecture would still be effective without this assumption, and its efficiency would be decreased by only a small factor as we analyze later.

Finally, to define a valid solution, assumptions need to be made on the capabilities of the attacker. In our scheme we assume a rather strong attacker model as defined in [10]. Following this attacker model, several attacks on anonymity protocols have been proposed, with traffic analysis attacks being considered the most potent ones. In this context, the attacks, against which we provide anonymity guarantees, are Brute Force Attacks, Communication Pattern Attacks, Timing Attacks and Packet Counting Attacks [11].

3 Persona

As mentioned in section 2 anonymity and accountability are the two conflicting notions. However, pseudonymity enables users to hide their true identity, until some event is triggered by which a third party can reveal it. In this paper we focus on providing sender anonymity and accountability by exploiting the notion of pseudonymity. Persona is structured around the following concept. While a packet is being routed through the network and towards the destination, we obfuscate the source address (in each hop) to provide anonymity (through pseudonymity). Additionally, we want the ability to trace back the origin of the packet, for accountability and routing replies. These two properties can be achieved through symmetric cryptography. More specifically, in the "forward path" encryption helps obfuscate the source address, while in the "trace-back path"

decryption helps reveal the original path. It can be seen that the approach used for packet replies is also used for accountability. The only difference is the context in which the traceback functionality is provided. For this reason, we focus on describing the technical details of tracing back packets, as accountability has several additional policy related issues that are out of the scope of this paper. However, we do provide a description on how accountability can be achieved, using our scheme.

3.1 First Hop Communication

Today, when a user registers with an ISP, it is common for the ISP to provide the user with a router in order to connect to the ISP and the Internet. As mentioned in the previous section, we assume that routers come installed with a TPM. Embedded within the TPM are symmetric keys that the router shares with the ISP. The ISP also has routers that use TPMs, to ensure trusted and secure software execution, attestation, and key storage.

When the user first connects to the network through his router, the keys in the TPMs are used to encrypt the information exchanged between the ISP and the user. This includes 1) the addresses of the routers that the user can contact as "first hop" from an ISP perspective (i.e. the routers that will eventually provide the anonymizing service), and 2) the shared keys between these routers and the user. After the "handshake", each user connected to the ISP follows a traffic sending pattern in order to exchange information in a secure and anonymous way. First the TPM of the user's router pseudorandomly chooses one of the newly received trusted routers to forward the packet. Thus, each packet will be forwarded to the Internet through a different router. In the case of a node compromise the attacker only has a probabilistic opportunity of identifying the sender. The second pattern that each user follows is that all packets sent are of same length. This length can be ISP specific, and can be established through the initial TPM handshake. This way traffic analysis attacks are prohibited and the attacker cannot correlate between messages' sizes. Next, the users of the ISP continuously send packets, by using dummy traffic whenever the user has no data to send. Thus, timing attacks and traffic pattern analysis are also difficult to achieve. This means that the link capacity between the users and the ISP will be always completely utilized. It has been shown that dummy traffic is the only way to protect against timing attacks [4]. Our model makes weaker and more efficient assumptions than previous proposed ones (like mix nets [12]), since the only part of the network that will utilize its maximum capacity at all times will be the connection between the user and the ISP. In conclusion, by requiring the usage of dummy traffic only in the first hop of the communication, we strike a balance between strong anonymity guarantees and efficiency (i.e. realistic use of dummy traffic). Finally, the receiver's address is encrypted using the shared key between the TPM of the user's router and the TPM of the ISP's router.

For the rest of the paper we assume that the receiver's address is sent in clear text after the first hop (i.e. the ISP of the sender). If a receiver's key is known (either public or shared), the router (i.e. the network layer) can also encrypt the payload, so as to provide confidentiality to the upper layers, especially if the higher level applications do not use encryption.

3.2 Persona Network Operation

As mentioned in the introduction, our approach introduces the concept of per packet anonymity. In order to achieve this, the packets need to be uniquely identifiable, at least for a given time interval, in order to avoid collisions. For that reason, the user (sender) has a Sequence Number (SN) which is incremented for each packet sent. We identify each packet by two unique fields. The first is the sender's IP, which will be unique in the Internet, and the second is the SN. If the SN is 128 bits long each user can send 2^{128} distinct packets before there is a duplicate packet in the network.

After the user has created a packet following the principles described in the previous section, she forwards the packet to the ISP. The main goal of the ISP is to hide the source identity of the packet. Each router holds a secret key that it can use for encryption. Using that key and a symmetric encryption function, the router maps the source IP address of the sender, to another randomly selected IP address from the range that its ISP holds. After the IP address has been changed the packet is forwarded to the Internet, according to the routing tables that the router has. This way the ISP shuffles the address of each packet and the attacker cannot determine the user that actually sent each packet. Finally the ISP routers batch the packets to be sent before actually forwarding them to the network. The communication and messages exchanged between the user and the router are as follows[1]:

User	**Router**
U: E_{KUR}(destination) => Q	R: E_{KR} (IP_1 || SN_1) => (IP_x || SN_x)
U=>R: (IP_1 || SN_1) || $Address_R$ || Q || payload	R: D_{KUR} (Q) => destination
	R=>Internet: (IP_x || SN_x) || destination || payload

Fig. 1. Message exchanged between User and Router

where KUR is the key shared by the user and the specific router, $Address_R$ is the IP address of the router, and destination is the IP address of the recipient. This operation will be done by each router until the packet reaches the destination as shown in Figure 2 (IPv4 addresses are used due to familiarity reasons with addressing).

It is easily understood that if sender anonymity is to be achieved, the receiver will not know how to reply to the packet. The reason is that there is no tunnel established, and thus there is a need to keep some state in order to return the reply to the sender. However by using the above scheme, packets can be routed to the original sender even if no state is kept. To better illustrate this, we will use an example. Let us assume that

[1] The semantics of the equations of the network operations are the following: The left column is used to denote the parties that take place in a given operation. If there is a single party, for example "a" then the right column is the action performed locally by that party. If there is a statement of the format "$a => b$", then this means that a sends to b, the message that exists in the right column. The right column denotes either actions or message contents. $E_k(m) => q$, denotes encryption of m under key k and q as the result of the encryption. $D_k(q) => m$, means decrypt q using key k and get m as the result of the decryption. If the right column is a message, then the symbol || is used to denote concatenation of the information that are included in the packet.

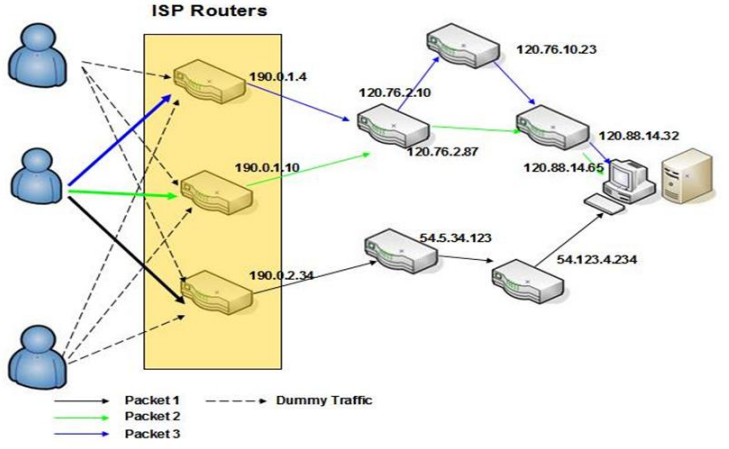

Fig. 2. Example of Persona Routing

after the first packet was forwarded to the Internet as explained above. Table 1 in Figure 3 depicts the operations that will take place in the last 2 hops (i.e. the router immediately before the receiver, and the receiver itself; the sequence of hops is $R_{n-1} => R_n => Receiver$). Now, the receiver has to create a packet that will have as destination the duple $(IP_d||SN_d)$ and forward it to Rn in order to send a reply, following the steps depicted in Table 2 in Figure 3.

R_{n-1}=>R_n:	$(IP_i \parallel SN_i) \parallel$ destination $\parallel$ payload
R_n:	$E\ (IP_i \parallel SN_i)_{KRn} => (IP_d \parallel SN_d)$
R_n=>Receiver:	$(IP_d \parallel SN_d) \parallel$ destination $\parallel$ payload

Table 1

Receiver => R_n:	$IP_{Receiver} \parallel (IP_d \parallel SN_d) \parallel$ payload
R_n:	$D\ (IP_d \parallel SN_d)_{KRn} => (IP_i \parallel SN_i)$
R_n=>R_{n-1}:	$IP_{Receiver} \parallel (IP_i \parallel SN_i) \parallel$ payload
…	
R_1=>R:	$IP_{Receiver} \parallel (IP_x \parallel SN_x) \parallel$ payload
R:	$D\ (IP_x \parallel SN_x)_{KR} => (IP_1 \parallel SN_1)$
R:	$E\ (IP_{Receiver})_{KUR} => W$
R=>U:	$W \parallel (IP_1 \parallel SN_1) \parallel$ payload
U:	$D\ (W)_{KUR} => IP_{Receiver}$

Table 2

Fig. 3. Process of Routing Back Replies to Sender

Using this scheme, packets can be routed backwards, without the need for the routers to keep any state. The only distinction that needs to be made by the routers is whether the packet is being sent "forward" or "backwards", so as to know whether to encrypt or decrypt. This can be done by an identifier, for example a single bit, which would be set by the receiver before sending his reply.

Given the size of the Internet in terms of hosts, routers and packets being sent, we use the SN in each packet to minimize the possibility of collision in an intermediate router (no packet is the same in the network since there are 10^{128} unique packets per

address). If no SN was used, a router could end up assigning the same output IP address to two different incoming packets due to its limited IP address range. For example, if a router has an input of 2^{10} distinct IPs, an output range of 2^{10} distinct IPs and a SN of 128 bits then there must be 2^{138} packets (128 bits of SN and 10 bits of IP range) sent before there is a possibility of collision of two different packets, with different sources. Thus by using SN it can be ensured that there will be no collision until the SN space is exhausted[2]. One assumption that we made was that each router has the same size of input and output IP ranges. However this might not always be the case. Let us assume that a given router R has some IP range that it can use in order to map outgoing IP addresses (denoted as "/x" output, where x is essentially the number of bits the router can manipulate), while it receives input from other routers that also have some IP range (denoted as "/y" input). It is expected that a lot of routers in the Internet will have a smaller output space than input space, and thus our router R will need y-x additional bits in order to perform one-to-one mapping. In order to include these additional bits to the packet, we use piggybacking.

Having explained the operation of the routers, we now discuss in detail the structure of the packets that our scheme uses. As mentioned previously, our protocol ensures collision free operation until the SN space is exhausted, after which each router changes its key. For this solution to be effective, the router needs to add some information in each packet, about the key it used for changing the IP address of the particular packet. The information that needs to be stored in each packet in order to make our protocol more efficient includes: 1) Sequence Number, 2) index of Router's secret key, 3) size of input space of addresses, 4) size of output space of addresses[3], and 5) the level of anonymity required. The level of anonymity is an optional variable that could be used to route the message through paths that provide better anonymity, but have more latency. Essentially all this information could be piggybacked in the packet. Thus all that needs to be added in the IP header will be the indexing of this information.

3.3 Persona Accountability

In the previous sections we discussed how our scheme operates in terms of routing and anonymity preservation. In this section we are going to describe how our scheme ensures accountability. As mentioned in the beginning of section 3, the operational principles of accountability are structured around backwards routing. Thus we will describe

[2] It must be mentioned that although there are 64 bits for addressing, not all of them are available to a particular router. If that was the case, and every router was assigning output (pseudo) addresses, based on all 64 bits, then it would be really difficult to keep track of the routing of packets (i.e. routing tables). Thus in order not to modify the routing tables, and yet allow Persona to fully operate, we assume that each router will only output addresses that he has been assigned; this means that each router will be able to manipulate only a number of bits (that correspond to its IP range), and not all the 64 available ones. This is depicted in the example where we assume a 10 bit manipulation.

[3] The input and output space, are the /x and /y that the router used during the encryption of the message. In a dynamic environment like the Internet, relationships between ASes and IPSs might change, and thus x and y are not expected to remain static, and thus the router needs to know what were the variables used for encryption at a particular time.

the context in which backwards routing can be used for accountability. We will classify accountability into two categories; short term and long term. By short term accountability we refer to the accountability about attacks that are "currently" taking place like DoS attacks, DDoS attacks, network scanning, etc, which essentially require the identification of the attacker as soon as possible. For better illustrating Persona's accountability operations, we assume that there exist other mechanisms that deal with IP spoofing. In short term accountability, ISPs can cooperate in order to identify and stop malicious attackers. For example, let as assume that a DoS attack takes place against a specific IP address (e.g. webserver). The router that forwards the packets to that IP address, can backtrace the routers from which it received the packets by decrypting the IPs reported by the webserver. Then it can contact these routers, reporting that a DoS attack takes place. If this procedure is applied recursively backwards, the originator of the attack can be found, and if the routers cooperate (i.e. routers can query other routers for packet throttling and the recipients of the requests indeed apply that throttling), the attack can be mitigated.

Long term accountability refers to examples like fraud detection, were the attacker is found after days or moths of investigation or forensics. In that case, the routers could be queried for past key usage, and since they keep a table of all keys used in the past in the corresponding table, the attacker could be easily tracked down (if memory constraints are placed on the table, past keys can always be stored on external backup media).

4 Persona Evaluation

In this section we are going to describe how our scheme defends against the attacks mentioned in section 2 and provide an efficiency analysis of Persona. Persona resists anonymity attacks as follows. 1) Brute Force Attack: In this attack, the attacker follows the life of every single packet that enters the network. Persona, by encrypting the packet during the first hop (possibly adding some nonce to each packet), and the subsequent source IP change, renders this attack useless. The user is at least provided with k anonymity where k is the number of active senders of ISP. 2) Communication Pattern Attacks: In this scenario the attacker monitors the two ends of a communication channel and tries to correlate entering and exiting packets. This attack is thwarted fairly well with the dummy traffic introduced in our solution. The use of dummy traffic and change of source IP address prevents the attacker from knowing which entering packet corresponds to the exiting packet. The best the strongest attacker can do is to deduce the originating ISP. The attacker still cannot determine which of the ISP's customers sent the packet. 3) Timing Attacks: Here, the attacker can deduce the origination of a packet based on the amount of time it took to reach the destination. In our scheme the ISP batches the messages and thus it is difficult for the attacker to identify which client of the ISP actually sent the message. Additionally, if more routers implement batching, the level of anonymity achieved is greater. Finally, even if the packets follow the same route, the attacker will not be able to apply timing attacks. This is because the source address will be changing for each packet and the attacker will not be able to tell if there are multiple senders sending to a single receiver. Thus she will not be able to deduce the identity of the senders or even the number of senders. 4) Packet Counting

Attacks: The attacker can connect unusual bursts of outgoing traffic with unusual bursts of incoming traffic. Since our model uses dummy traffic and same packet size there are no bursts of traffic to identify.

Our solution incorporates most of the suggested countermeasures proposed in mix-nets, namely encryption, same packet size, batching and dummy traffic [11]. These countermeasures are applied at minimum to the first hop, between the user and the ISP. Thus at minimum, each user will have a level of k-anonymity, where k is the number of active users connected to the ISP at a particular moment. This makes our scheme resilient to additional attacks that are defeated with these countermeasures, whilst the level of anonymity our scheme provides, increases with the number of routers that actually implement the abovementioned properties.

In order to provide the abovementioned anonymity guarantees and functionality, Persona requires a lot of cryptographic operations to take place in each router and for each packet. We know that for symmetric encryption there can be approximately 10^7 operations per second in 1 GHz processor if done in hardware. Let us assume that users are using a line speed of 100Mbps, and that an average IPv6 packet size is used (i.e. 20000 bytes without Ethernet limitations). Each user will be sending, and each router will be receiving, 10^3 packets per user, and thus, given that it can calculate 10^7 operations per second, 10^4 users per router per second can be accommodated. For the first router that belongs to the ISP, multiple routers can be used, both for increase anonymity and better efficiency. The only bottleneck would be top level routers. These routers already have to forward packets at line speed, so there is a tradeoff between the anonymity and the efficiency that NGI will offer. In that case the optional bits for level of anonymity discussed in the previous section could be used.

Finally it has to be mentioned that we approached the problem of anonymity and accountability from a network layer's perspective. That is we did not take into account the above protocols, and we only defined the services that are going to be provided to them. However, one question that might be of importance is how our protocol can assist and support connection oriented services (e.g. TCP). In that case we see two possible solutions. The first one is the redesign of the transport layer so as not to use IP addresses as point of reference for keeping sessions' state. The second solution would be to have a handshake between the sender and the server, so as the sender is granted with a session identifier that he can use for session identification. This identifier can be encrypted alongside with the payload, so as to avoid traffic analysis based on its identification in the various packets.

5 Related Work

Currently, most of the techniques used for anonymity make use of the concept of mix networks introduced by Chaum [12]. Mix networks, are networks composed of mix servers which batch a set of messages encrypted by their corresponding public keys. They then decrypt these messages and send them out in a rearranged order such that an external observer cannot tell which outgoing messages correspond to incoming messages. Babel [13], Mixmaster [14], Mixminion [15] and Onion routing [16] are some of the most important systems based on mix networks. A common aspect of all the

solutions mentioned above is that none of these systems have accountability as one of their main goals, as Persona does. In addition, in Persona the sender does not need to know the path that the packet will follow, as done in Onion Routing for example. Moreover Persona does not encrypt the packet in an "onion style" with the public keys of the routers which has additional efficiency advantages.

Two systems that make use of pseudonymity in the network to achieve anonymity are Freedom [17] and Tarzan [18]. Both these techniques are similar, in the sense that the user connects to a node anonymously. This node then sends the packet to the destination but changes the source address of the packet by assigning it a pseudonym. Once a packet is sent the receiver sends a reply addressed to the pseudonym. The node that had performed the network address translation while sending maps the pseudonym to the original address and sends the reply to the sender using the anonymous channel. The two approaches differ in the way a user connects to the node doing the network address translation. However, in both cases the sender requires information about the path between itself and the exit node performing the NAT. Our approach differs from these approaches in three ways. Firstly, we do not need to keep any state in the nodes to translate the IP addresses back to the original address. This saves a lot of memory and the time to look up the translations in memory. Secondly, we do the translation per packet and not per session. This increases the Unlinkability between the sender and receiver as each packet is routed independently giving the attacker little information. Thirdly, in our scheme the sender does not need to set up the path every time before sending a packet. This path is preconfigured when the router is installed. Not having to set up the path avoids wasting time.

6 Conclusions and Further Research

This paper has presented Persona, a scheme that incorporates anonymity and accountability in the network layer of the Next Generation Internet. We have proposed a novel approach by which each packet is provided with anonymity, thus achieving stronger properties from previous solutions. Additionally we adopted proven solutions from previous research on anonymity techniques, so as to ensure a maximum level of anonymity, at minimum within the ISP of a client. Finally, we described how our scheme can be used for achieving accountability, in cases of malicious and misbehaving users. Continuing our effort, we examine how we can optimize our solution and provide additional types of anonymity (e.g. recipient anonymity). Additionally we examine how our scheme can be applied to the current Internet. Finally we are planning to do simulations so as to identify the tradeoffs of our approach in terms of current IP deployment and existing anonymity techniques.

Acknowledgements. The authors would wish to thank Professor Adrian Perrig for his support, guidance and constructive suggestions, and the anonymous reviewers for their helpful comments. We greatfully acknowledge support for this research by CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, or the U.S. Government or any of its agencies.

References

1. Danezis, G., Diaz, C.: A Survey of Anonymous Communication Channels. Microsoft Research technical report MSR-TR-2008-35 (January 2008)
2. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The Second-Generation Onion Router. In: The Proceedings of the 13th USENIX Security Symposium (August 2004)
3. Murdoch, S.J., Danezis, G.: Low-cost traffic analysis of Tor. In: Proceedings of the 2005 IEEE Symposium on Security and Privacy (May 2005)
4. Murdoch, S.J.: Covert channel vulnerabilities in anonymity systems. Technical report, University of Cambridge (August 2007)
5. Bellovin, S.M., Clark, D.D., Perrig, A., Song, D.: A Clean-Slate Design for the Next-Generation Secure Internet. In: NSF Workshop on a clean-slate design for the next-generation secure Internet (2005)
6. Andersen, D.G., Balakrishnan, H., Feamster, N., Koponen, T., Moon, D., Shenker, S.: Accountable Internet Protocol (AIP). SIGCOMM Comput. Commun. Rev. Journal 38(4), 339–350 (2008)
7. Yaar, A., Perrig, A., Song, D.: SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. In: Proceedings of the IEEE Symposium on Security and Privacy (May 2004)
8. Pfitzmann, A., Hansen, M.: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management A Consolidated Proposal for Terminology, Version v0.31 (2008)
9. McCune, J.M., Parno, B., Perrig, A., Reiter, M.K., Seshadri, A.: Minimal TCB code execution (Extended abstract). In: Proceedings of the 2007 IEEE Symposium on Security and Privacy (May 2007)
10. Diaz, C.: Anonymity Metrics Revisited. In: Dolev, S., Ostrovsky, R., Pfitzmann, A. (eds.) Anonymous Communication and its Applications (2006)
11. Raymond, J.F.: Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In: Federrath, H. (ed.) Designing Privacy Enhancing Technologies. LNCS, vol. 2009, pp. 10–29. Springer, Heidelberg (2001)
12. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the Association for Computing Machinery 24(2), 84–88 (1981)
13. Gulcu, C., Tsudik, G.: Mixing E-mail with Babel. In: Network and Distributed Security Symposium - NDSS 1996. IEEE, Los Alamitos (1996)
14. Mller, U., Cottrell, L.: Mixmaster Protocol - Version 2, Unfinished draft (January 2000)
15. Danezis, G., Dingledine, R., Mathewson, N.: Mixminion: Design of a Type III Anonymous Remailer Protocol. In: The Proceedings of the 2003 IEEE Symposium on Security and Privacy, pp. 2–15 (May 2003)
16. Syverson, P.F., Goldschlag, D.M., Reed, M.G.: Anonymous connections and onion routing. In: IEEE Symposium on Security and Privacy, Oakland, California, pp. 44–54 (1997)
17. Boucher, P., Shostack, A., Goldberg, I.: Freedom Systems 2.0 Architecture, Zero Knowledge Systems, Inc. White Paper (December 2000)
18. Freedman, M.J., Morris, R.: Tarzan: A Peer-to-Peer Anonymizing Network Layer. In: The Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), Washington, DC (November 2002)