IPsec-based Anonymous Networking: A Working Implementation

Csaba Kiraly, Renato Lo Cigno
DISI – Università di Trento, Italy

***Abstract*—Protecting users' privacy is becoming one of the rising issues for the success of future communications. The Internet in particular, with its open architecture, presents several threats to the right of protecting personal and sensitive data.**

One fundamental building block of privacy-respectful communications is protecting the communication parties identities, or, as it is commonly called within the research community *anonymous networks* (ANs). An AN prevents external observers as well as the network to have access to communicating partners identities and addresses. In this paper we propose a novel architecture to realize ANs, as an extension to IPsec. After explaining the rationale and discussing possible alternatives, we present a working prototype implementation and its experimental performance comparison with application level solutions.

I. Background and Privacy Protection

One of the challenges for the "network of the future" is improving the protection of users against frauds and intrusions, including the provisioning of suitable means to protect personal and sensitive information. In some cases preserving privacy implies the right to be anonymous, including in this term pseudonymization techniques, where the identity of a person is disguised, but can be revealed to selected authorities (e.g., lawful investigators authorized by a judge). One example for all is voting, where revealing the identity of voters is always the first step toward election frauds.

Until very recently, it was deemed that secrecy of data communications (i.e., strong enough cryptography) was enough to preserve users' privacy. Anonymity of both addresses and users was not even conceived. Now, instead, privacy protection is recognized as one of the functions that a telecommunication network should provide to its customers and users.

Due to the addressing space of the Internet, source and destination IP addresses can be seen all along the communication path, revealing the location (through IP geolocation) as well as the identity (through reverse DNS; by linking different actions initiating from the same IP address; or by contacting the ISP who assigned the address) of the communicating parties. Once the identity is uncovered, privacy can be compromised. Widespread use of customer databases and activity logs, and the recent evolution of data mining and linkage techniques make it possible to obtain private and sensitive data analyzing traffic flows and communication end-points.

Privacy in the "Internet" was first discussed by David Chaum [1] introducing the concept of anonymous communications, but after this initial discussion the topic remained confined in small communities for a long time. The concept, which was originally thought for anonymous e-mail only, was later extended to low-latency communications by application level overlay systems such as Tor [2], Tarzan [3] or Freedom [4]. Low-latency in this context means that the communications is not based on storing and forwarding *entire application level messages* in intermediate nodes, but rather small information units are forwarded limiting end-to-end delays to seconds.

The contribution of this paper is showing that IP address protection can be provided *within the network layer itself*, namely within the IPsec framework, without the need of building cumbersome and non-scalable application-level overlays as any other AN solution (see [2], [3], [4]) does.

We implemented the proposed architecture using standard IPsec and routing configuration tools available in Linux 2.6. Measurements on an experimental testbed show significant performance advantage of our architecture compared to the Tor architecture, which is the de-facto standard in AN, with thousands of overlay routers in operation and over 200.000 users.

II. Principles of Anonymous Networking

Anonymity at the network level means not revealing the address of one end-point to the other side, and preventing any third party observing the network from understanding who is communicating with whom.

Referring to Fig. 1, the goal of an AN is protecting Alice's identity from Bob, as well as the fact that they are communicating together from Mallory, an external attacker. We call a node of an AN a *mix*. A mix preserves communication privacy, including anonymity, and does this by mixing together traffic belonging to different end-to-end connections.

The threat model we consider is similar in scope to many low-latency ANs [2], it comprises most of the modern traffic analysis techniques. To provide sender-receiver unlinkability at least one of the sender or the receiver addresses must be hidden at any observable point of the network. This way, Mallory, even if being near the source or the receiver, can only see that one of Alice or Bob is communicating with someone, but is not able to understand who the communicating parties are.

A global adversary with unlimited resources to monitor links and routers and correlate information (or with initial clues about communicating pairs) can still uncover who is communicating with whom matching packet size, packet timing, or flow characteristics. Protection against such an attack is possible using traffic flow confidentiality techniques, such

This work was partially supported by the EU IST 6^{th} Framework Program project Discreet, IST No. 027679.

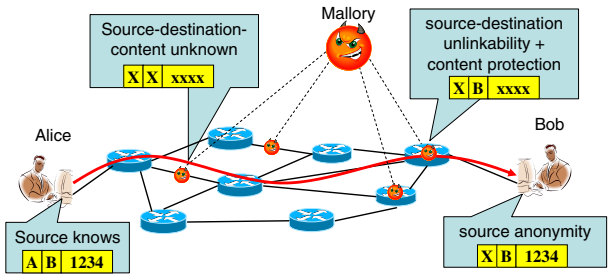

Fig. 1. Protecting anonymity in face of the destination and of an external attacker with a favorable observation point where several links and routers can be controlled.

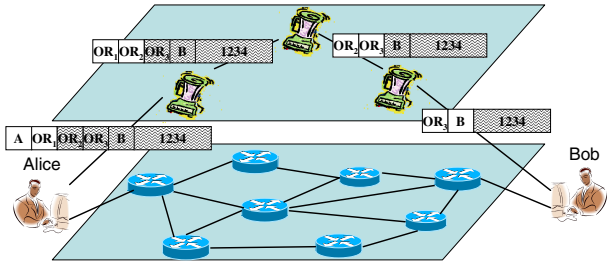

Fig. 2. Anonymous networking realized with Onion Routing

as the IPsec extension introduced in [5] or any other suitable proposal, which are however outside the scope of this paper.

The first system designed to provide ANs was [1], which defined the founding principles of anonymous communications, summarized as follows:

- P.1 Un-traceable delivery: messages are delivered through a chain of mix nodes, which enforce that no correspondence can be found between incoming and outgoing messages;
- P.2 Un-traceable return address: a mechanism that allows the receiver to respond while the sender still remains anonymous;
- P.3 Distributed trust: no universally trusted authority is required to keep the identity secret. Each mix knows only the previous and the next mix on the delivery path: the only trust required is that they will not collaborate to find out the communicating parties.

Technically, these principles are realized using onion encapsulation (see Fig. 2), i.e., embedding one encrypted layer into another recursively. For this reason we will also call mixes *Onion Routers* (ORs). The source selects the path of the message and encrypts it in multiple layers, using the public-keys of the selected mix nodes one after another, including the address of the next mix node in each layer of encryption. Each mix "peels off" one layer, delivering the payload to the next mix in the chain. A chain of two nodes is enough to provide anonymity, but three or more are normally used to improve protection by reducing the possibility that mixes collaborate.

To provide low-latency, high throughput anonymous service, public-key encryption is computationally too heavy. To overcame this limit, faster symmetric key cryptography must be used, dividing communication into two phases. When a node wants to use AN services, first, in a circuit-setup phase, symmetric keys and circuit identifiers are exchanged with public-key cryptography with each OR of the chain. Later, during data transmission on this circuit, faster symmetric keys are used for the onion encapsulation.

This kind of anonymous circuits are normally called 'telescopes' because the source encapsulates all headers related to tunnels toward ORs one into the other, and each OR decapsulate one header, so that the overall structure of nested tunnels is thicker at the source and becomes thinner toward the destination. In what follows we use the terms circuit and telescope interchangeably.

Implicit in the notion of telescope is the use of source routing: it is the source that defines which mixes are to be used. Source routing is not the most efficient routing solution; however, at the state of the art no other viable solution to route packets and protect privacy has been found.

The topological structure of the whole AN can be seen as an overlay of mix nodes on the IP network, serving a large number of telescopes. Mix nodes are interconnected by overlay links, implemented as tunnels crossing multiple standard IP routers. These long-term overlay tunnels are different from nested tunnels of telescopes, and provide some basic functionalities to organize the network. They convey information among mix nodes forming the edges (or links) of the anonymous network. They provide encryption to make packets belonging to different circuits unrecognizable from each other. Finally they can provide traffic flow confidentiality to prevent statistical analysis and confirmation attacks.

Existing ANs [2], [3], [4] make different design choices for the implementation of overlay tunnels, but all of them work at the application level either by using TLS tunnels or proprietary solutions. For implementing the nested tunnels of telescopes, they all use proprietary solutions. All systems are 'all-in-one' solutions, making it difficult to evolve them and to make them highly performant and widespread in use. In [6] we discuss the limitations and performance impairments (from an architectural/theoretical point of view) of using transport (or application) level tunneling.

Summarizing this Section, an AN can be realized with a proper combination of the following techniques.

- A.1 A procedure to open and maintain secure tunnels, both for overlay links and for nested tunnels of telescopes; the telescope tunnels should hide end-points address.
- A.2 A path selection mechanism to choose the mixes forming the telescope.
- A.3 An anonymous packet forwarding mechanism that enforces the specified path, both in the forward and in the reverse direction.
- A.4 A mechanism that allows any host (or generic IP node) to communicate over the anonymous overlay network through anonymity gateways and exit nodes.
- A.5 A telescope setup mechanism to distribute (tunnel's) state information to mixes of the selected path.
- A.6 A management functionality to distribute both to mixes (in order to set up overlay tunnels) and to clients (to

provide information for path selection) the topology information and to distribute public keys and similar maintenance information.

III. IPsec Based Anonymous Networking

IPsec natively supports configurations where several layers of tunnels are nested in each other [7]. Here we show how this capability, together with additional properties of IPsec and related signaling protocols can be manipulated and used to fulfill the points A.1, ..., A.5 that emerged at the end of Sect. II as critical requirements to implement AN. The realization of A.6 is not discussed in this paper.

Let $\mathrm{OR}_1, \cdots \mathrm{OR}_n$ be the IPsec routers that implement the overlay. Long-term tunnels between these nodes are the edges of the overlay. Let $T^O_{i,j}$ be the overlay tunnel established between OR_i and OR_j. Overlay tunnels are bidirectional, so $T^O_{i,j} \equiv T^O_{j,i}$.

A. Nested secure and anonymous tunnels

Let $\mathrm{OR}_{c_1} \cdots \mathrm{OR}_{c_k}$ be the nodes of the anonymous circuit c between clients A and B. We implement the circuit through k nested IPsec tunnels (T_{A,c_1} .... T_{A,c_k}), each one having A as one end-point and $\mathrm{OR}_{c_i}; i = 1, \ldots, k$ as the other. Also these nested tunnels are bidirectional.

Setting up the telescope described above does not provide anonymity. Packets passing from A to B are routed on the selected circuit and the IP addresses of A and B are protected from external eavesdroppers by the overlay tunnel encryption, but the address of A, as the initiator of each T_{A,c_i} tunnel, is seen by every OR_{c_i}.

To enforce anonymity we need to masquerade the IP address of A in these tunnels. During signaling phase, the client negotiates a private IP address with each node of the circuit (A_{c_i} with OR_{c_i}, etc.), and this address is used as source IP address in T_{A,c_i}.

IPsec allows the creation of all the aforementioned tunnel structures setting Security Policies (SP) and creating related Security Associations (SA), thus fulfilling the requirement A.1. For details about Security Policies and Security Associations see [7].

B. Path selection

As already mentioned all ANs use source routing as path selection mechanism. The source A chooses the k nodes $\mathrm{OR}_{c_1} \cdots \mathrm{OR}_{c_k}$ of the circuit randomly with uniform distribution among the available $\mathrm{OR}_1 \cdots \mathrm{OR}_n$. Previous works showed that $k = 3$ guarantees strong anonymity protection while keeping the overhead low [2]. Once the path has been chosen, it is fixed in A's SP database. The standard IPsec packet *selector* associated with this SP allows fine-grained configuration of what traffic should be anonymized and thus routed over c. The following nested SP is created in A to satisfy criterion A.2:

$$\textit{selector} \rightarrow \mathrm{PKT} \supset T_{A,c_k} \supset \cdots \supset T_{A,c_1} \supset T^O_{A,c_1}$$

where the operator $\supset$ means tunnel encapsulation, a per-packet (PKT) operation, and *selector* stands for a standard IPsec SP selector. By creating similar SPs with different selectors, several anonymous circuits can be used at the same time.

Random path selection is just the simplest possible solution. Other path selection strategies are possible ([8]), but details about path selections are irrelevant to our work.

C. Anonymous forwarding

The system implements policy based routing in the ORs. For a given circuit c each participating OR_{c_i} installs SPs to receive and forward packets in the right direction with the necessary encapsulation/decapsulation operations, both in the forward and in the reverse path.

In the forward direction, OR_{c_i} terminates both the overlay tunnel $T^O_{i-1,i}$ and the nested tunnel T_A, c_i. The packet is routed toward $\mathrm{OR}_{c_{i+1}}$ based on the destination address of the internal IP header, and therefore it is automatically encapsulated in $T^O_{i,i+1}$. The following SPs are created for each circuit:

$$(A_{c_i}{+}1, \mathrm{OR}_{c_{i+1}}) \rightarrow (\mathrm{PKT} \ \subset T^O_{c_{i-1},c_i} \subset T_{A,c_i}) \supset T^O_{c_i,c_{i+1}} \tag{1}$$

where (S, D) is a selector matching IP packets with S as source and D as destination, and the $\subset$ operator means decapsulation.

In the reverse direction, after decapsulation from the overlay tunnel, the packet is identified as belonging to c based on the unique $(\mathrm{OR}_{c_{i+1}}, A_{c_i}{+}1)$ address tuple. Thus, a layer of telescope encryption is added, and PKT is forwarded to $\mathrm{OR}_{c_{i-1}}$ through $T^O_{i,i-1}$. The following SPs enforce these operations:

$$(\mathrm{OR}_{c_{i+1}}, A_{c_i}{+}1) \rightarrow (\mathrm{PKT} \supset T^O_{c_{i+1},c_i}) \subset T_{A,c_k} \subset T^O_{c_{i-1},c_i} \tag{2}$$

The enforcement of SPs (1) (2) in $\mathrm{OR}_{c_1} \ldots \mathrm{OR}_{c_k}$ guarantees the anonymous forwarding in both directions satisfying A.3.

D. Anonymity gateways and Exit nodes

To make AN ubiquitous (satisfying A.4), there are two possibilities: i) extend AN services to any IP host; and ii) setup trusted anonymity gateways allowing any node to make anonymous connections. The first choice is clearly impractical, while the second one is feasible defining *anonymity gateways* (AG) and *exit nodes* (EN).

An AG is a node that carries on anonymization services for client nodes that cannot handle (or does not want to handle) anonymization by themselves. It is implemented by extending the scope of A's SP *selector* to include forwarded traffic besides traffic originating from A itself. Such an AG can be placed in the default route of the client (like e.g., a home router), or can be contacted on-demand through an IPsec tunnel or using SOCKS.

An EN employs standard stateful source NAT to allow anonymous connections through the overlay toward any IP host. When OR_{c_k} is an exit node, after the decapsulation of T_{A,c_k}, it receives a packet with source A_{c_k} destined for an ordinary node B. It uses source NAT to change the source address to OR_{c_k} and sends out the packet. B responds to OR_{c_k} which changes the destination address based on NAT state to A_{c_k}, and thus policy based routing works as described before.

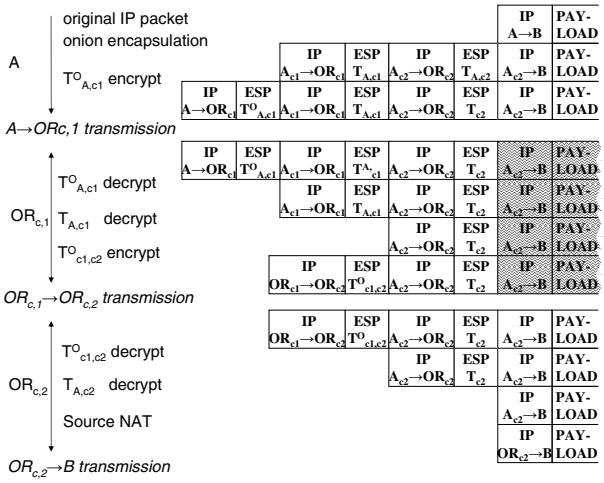

Fig. 3. Structure of onion encrypted packet leaving the source and subsequent packet transformations (ESP trailer omitted for simplicity); the shaded part of the packets is encrypted and thus not 'visible' to OR_{c_1}

Fig. 3 summarizes packet transformations along c in the forward path, for the simplified case where A is part of the overlay and $k = 2$. Since return path operation is the exact opposite of forward path operation, reverse path transformations can be seen reading Fig. 3 bottom-up.

E. Telescope setup mechanism

For setting up a telescope, shared symmetric keys should be negotiated between A and all the ORs of the path. During the negotiation, A should remain anonymous. [9] generalizes signaling that can be used for telescope setup, and the symmetric keys negotiated by the introduced signaling can be used to configure SPs and SAs through local management protocols such as the PF_KEY API [10] or the IPsec SPD Configuration MIB [11]. In our proof-of-concept implementation pre-shared keys are used for telescope encryption for the sake of simplicity.

IV. Performance Measures and Analysis

The IPsec-based AN described in Sect. III is a complex architecture with many facets and details. One of the main reasons to propose network-layer AN is the conjecture that its performance will be much better than application-level overlays. To prove this conjecture we implemented a proof-of-concept AN with the key features over Linux (2.6 kernel).

Our proof-of-concept includes all encapsulation and decapsulation procedures, management of SPs and SAs, and EN, as well as all information transfer procedures; it does not include all signaling mechanisms, since the performance during information transfer is not influenced by signaling. The implementation is fully functional and implemented in kernel space, and it is available from the authors upon request.

Fig. 4 describes the measurement setup. We compare the performance of Tor and our IPsec-based AN solution. To evaluate the "cost" of privacy we choose two measures: *web page download time* and *traffic overhead*, the first one being

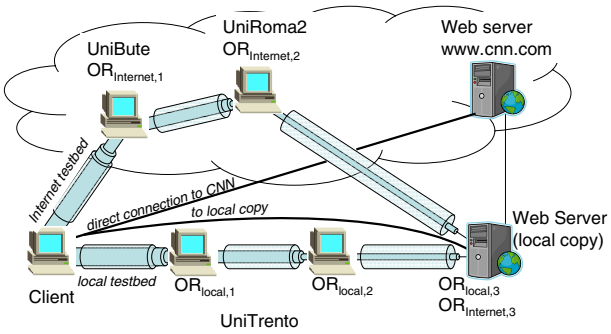

Fig. 4. Testbed configuration and experiment setup

the price a user has to pay for anonymity, while the second measuring the cost for the network to support it. As an example we have selected the download of the main page of CNN with Firefox, but other web-pages, the use of other protocols (e.g., ftp or POP/smtp for e-mail), or the use of different browsers do not change the scenario, yielding similar results.

Our goal is to verify if the IPsec based solution can perform better than Tor, a service which has already about 200.000 users despite its performance bottlenecks, which are normally described as a price which must be paid to have anonymity. We have configured IPsec tunnels with the same ciphers as used in Tor (AES-CBC 128-bit encryption and SHA-1 data authentication) in order to guarantee a fair comparison between the systems and be sure that differences are not due to the different efficiency of the encryption algorithm.

To better understand the impact of the overlay networking and the anonymization 'price' we have used two sets of overlay nodes for our tests: a circuit of three nodes in our lab (*local testbed*, lower part of Fig. 4); and a circuit of three nodes in different European universities (*Internet testbed*, upper left part of Fig. 4). In both circuits, the exit node was the same router in our University, therefore the last unencrypted part of the path to CNN was always the same. In both circuits we had Tor as well as IPsec-based anonymous routing installed[1], thus having four different cases. We use direct, non anonymous download as a reference. These five cases are shown on the right side of performance figures.

Additionally, to evaluate the performance also in an "ideal" scenario, where the Internet does not interfere with the download speed, and there is no need for an EN, we have prepared a local copy of the CNN main page on our own server, having five more cases marked with *local copy* (we show these on the left side of performance figures). Notice that the on-line CNN page is dynamic and changes continuously, while the local copy was always the same (indeed, it turned out to be a page slightly larger than the average), so that on-line and local results are not directly comparable.

The *direct connection* results refer to non anonymous download, thus representing the baseline performance of the service.

[1]To achieve comparable results, our private Tor network was not connected to the global Tor network thus guaranteeing that no other traffic is using the same node modifying performances

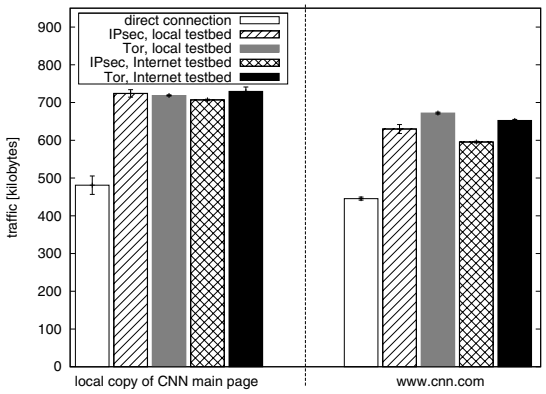

Fig. 5. Traffic received to complete the web page download

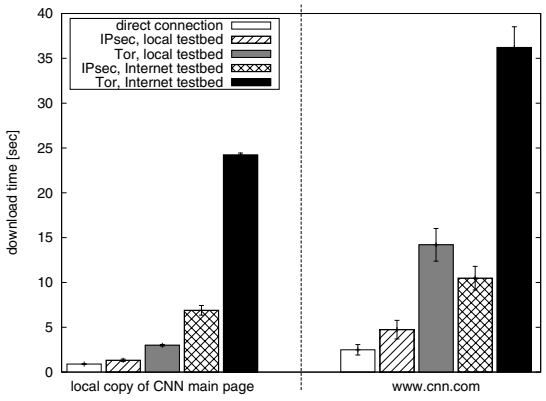

Fig. 6. CNN web page download time

All downloads were repeated 10 times averaging the results; error bars refer to $\pm$ the standard deviation.

Fig. 5 shows the total amount of traffic generated in each scenario. Privacy protection with onion routing induces significant overhead in bytes: around 50% for both solutions. The price of privacy protection is relatively high, but not excessive, and it can be applied selectively on sensitive traffic. From the network perspective, further overhead is induced by routing packets over multiple overlay links instead of using shortest path routing.

Fig. 6 shows average download times. With no bottlenecks (local testbed - local copy), layer 3 operation (2^{nd} column) is clearly much faster than Tor (3^{rd} column). There is almost no delay that can be ascribed to IPsec-based OR operation, in contrast to Tor where the average download time increases threefold (from 1 s to 3 s). Tor's performance degradation is rooted in architectural choices and the use of TLS tunneling in each overlay hop, and it is not due to bad implementation. The Internet testbed (shown in the 4^{th} and 5^{th} column) obviously increases RTT and thus download time, since packets are routed around Europe to distribute trust to different countries. Here Tor's disadvantage grows significantly (24 s vs. 6 s with IPsec, again threefold), having a page download time clearly noticeable by the user and thus annoying.

Looking at downloads directly from CNN through the local testbed (right side of Fig. 6), our IPsec based solution increases the download time compared to direct download by roughly 50%. Surprisingly, Tor over the local testbed is slower (14 s) than IPsec over the Internet testbed (10 s), indicating that the performance bottlenecks are to be sought for in Tor itself and not in the Internet. Finally, the performance of Tor with the Internet testbed makes it almost unusable with interactive services.

V. Conclusions and Future Work

This paper introduced the idea of supporting anonymous networking (AN) as an extension of IPsec. AN has not received much attention by the networking community, being considered, until recently, a problem to be tackled at the application level. By supporting privacy-aware, anonymous communications at the network layer with a proof-of-concept implementation of an anonymous overlay embedded within IPsec, this paper demonstrates that it is possible to enforce anonymity within IPsec.

A performance comparison with Tor showed that on the one hand the overhead (in terms of additional transmitted bytes) of IPsec and application level solutions are almost identical, while on the other hand the performance in terms of download delay is 3–4 times faster with the IPsec solution. This performance result, measured using real services over the Internet, is extremely promising and calls for further research to implement signaling solutions to make deployment easy and 'plug&play'.

Acknowledgment

The authors thank Giuseppe Bianchi and Simone Teofoli for the useful discussions and support.

References

[1] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[2] Dingledine R. Mathewson N. and Syverson P. Tor: The Second-Generation Onion Router. In *Proc. of the 13th USENIX Security Symposium*, Aug. 2004.

[3] Freedman M.J. and Morris R. Tarzan: A Peer-to-Peer Anonymizing Network Layer. In *Proc. of CCS 2002*, Washington, DC, Nov. 2002.

[4] Boucher P. Shostack A. and Goldberg I. Freedom Systems 2.0 Architecture. White paper, Zero Knowledge Systems, Inc., Dec. 2000.

[5] C. Kiraly, S. Teofili, G. Bianchi, R. Lo Cigno, M. Nardelli, and E. Delzeri. Traffic Flow Confidentiality in IPsec: Protocol and Implementation. In S. Fischer Hübner et Al, editor, *The Future of Identity in the Information Society*, volume 262 of *IFIP*. Springer, July 2008.

[6] C. Kiraly, G. Bianchi, and R. Lo Cigno. Solving Performance Issues in Anonymization Overlays with a L3 approach. Technical Report DISI-08-041, Univ. of Trento, August 2008. http://www.disi.unitn.it/locigno/preprints/TR-DISI-08-041.pdf.

[7] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), December 2005.

[8] R. Snader and N. Borisov. A Tune-up for Tor: Improving Security and Performance in the Tor Network. In *Proc. of NDSS'08*, Feb. 2008.

[9] L. Øverlier and P. Syverson. Improving Efficiency and Simplicity of Tor Circuit Establishment and Hidden Services. In *Proc. of PET 2007)*, Ottawa, Canada, June 2007.

[10] D. McDonald, C. Metz, and B. Phan. PF_KEY Key Management API, Version 2. RFC 2367 (Informational), July 1998.

[11] M. Baer, R. Charlet, W. Hardaker, R. Story, and C. Wang. IPsec Security Policy Database Configuration MIB. RFC 4807 (Proposed Standard), March 2007.