

Denial-of-Service Flooding Detection in Anonymity Networks

Jens O. Oberender*, Melanie Volkamer†, and Hermann de Meer†*

*Faculty of Informatics and Mathematics, †Institute of IT-Security and Security Law, University of Passau, Germany

E-Mail: {oberender|volkamer|demeer}@uni-passau.de

Abstract—Denial-of-Service (DoS) flooding attackers benefit from sender anonymity and exit node diversity. Anonymity networks provide this by hiding the communication relationship and therefore hinder attack detection. After the anonymity network purges IP headers, the attributes for clustering of traffic flows remain hidden. Message unlinkability provides network privacy. We design limited message linkability for clustering of traffic flows. Clusters of anonymous traffic are sufficient for flooding attack detection and also enable mitigation. The number of linkable messages is restricted to limit profile size and protect from privacy adversaries. In distributed scenarios, our incentive motivates use of a single entity. Message tags enable detection of flooding attacks. The set of linkable messages is limited, which cuts activity profile. Adversaries cannot influence message linkability of other parties. Senders dynamically govern their message linkability through the message arrival rate. During flooding to a single victim message linkability improves, enabling DoS detection for anonymity networks.

I. INTRODUCTION

Current Internet DoS attacks feature distributed systems, e.g. BotNets. They circumvent IP-based attack mitigation and also provide anonymity to the attack initiator. In order to combat network crimes, network communications are monitored preventively. The collection of IP header data is also enforced by recent EU laws (2006/24/EG). Network forensic investigations rely on such traces.

Network monitoring does not differentiate malicious from normal traffic. Doubts against network monitoring often criticize the loose control of sensitive log data. Privacy-compliant data handling in IP-based networks is difficult. IP header information is necessary for network routing, therefore all routers and network links must conform with privacy requirements. Anonymity networks have been developed to enable network privacy. In case of malicious traffic, the victim host can only determine that it has been attacked through the anonymity network. As a result a policy prohibits further exit traffic onto the target. An adversary can misuse this procedure to permanently block anonymous communications. After an adversary has blocked all data sources, it can easily observe further communications.

A mechanism to detect flooding is proposed, being challenged by anonymous communication. We consider message tags to enable traffic flow classification, which is fundamental to detection and mitigation mechanisms.

This work has been partly sponsored by EPSRC (GR/S69009/01) and EU funding (EuroFGI IST-028022). The authors would like to thank Huei-Ru Tseng and Amine Houyou for fruitful discussions.

This paper is structured as follows. After review of related approaches (Sec. II), we introduce the problem and list security objectives (Sec. III). Sec. IV describes linkability and message tag generation. Defense against DoS attacks and privacy attacks are considered in Sec. V. The paper is completed by future work (Sec. VI) and concluding remarks (Sec. VII).

II. RELATED WORK

The tackled problem is novel in its application. The following mechanisms require adoption to detect anonymous flooding. Responder anonymity protects the receiver against flooding of illegitimate traffic. The secure overlay service (SOS) introduces secret servlets, which communicate through the target firewall [1]. If a secret servlet turns malicious, the target replaces the servlet and reconfigures the target firewall. Tor introduces *location-hidden service* adding an extra level of indirection, namely rendezvous points [2].

Some works focus on the resource consumption during an attack. The *Speakup* protocol increases the traffic rate for all participating senders [3]. Because attackers have no bandwidth reserve, they are unable to increase the traffic rate. In result the legitimate traffic supersedes malicious traffic. The approach cannot differentiate between DoS attackers and participants with limited bandwidth.

A principle problem is unbalanced resource consumption between client and server. The *client puzzle* protocol increases the effort to issue a request. A client must solve a cryptographic challenge [4], i.e. decomposition of a large number into prime factors. Auction protocols assign server resources by the computation effort provided by the client side [5]. Again, this penalizes low capability nodes. During ongoing attacks mobile devices cannot compete on server resources.

A more direct approach to attack mitigation and forensics is *anonymity revocation*. In such systems a trustee guarantees anonymity, unless misbehavior can be proved. Blind signature schemes provide anonymous network access [6]. The trustee creates a ticket which is revocable to the sender identity. Suspect of misuse is determined at the egress node, which can initiate the revocation of the sender anonymity in a mixer cascade [7]. The user privacy is protected by requiring all nodes of the communication path to accept the revocation. However, it is arguable whether exit policies are able to capture criminal activities without tampering network anonymity.

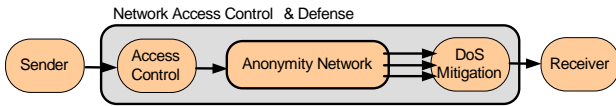


Fig. 1. Anonymous Communication Scenario

III. PROBLEM DESCRIPTION

A. Scenario

Our communication scenario is shown in Fig. 1. The inner block provides network anonymity, i.e. messages arrive anonymously at the DoS detection. The outer block authenticates message senders and authorizes network access. A DoS detection engine protects the receiver at the other end of the anonymity network against anonymous flooding attacks.

David Chaum developed a fundamental scheme for *anonymity* in computer networks [8]. Network anonymity considers correlations between subjects, messages, and actions. A subject remains anonymous if it is not identifiable within a set of subjects, the anonymity set. The technique to hide within the anonymity set is provided by mixer nodes, which provide a special routing protocol on application layer. Mixer nodes receive messages from multiple subjects, whose traffic is forwarded to other mixer nodes and re-encrypted. As soon as many participants contribute messages through the anonymity network, observers become unable to correlate communication relationships, i.e. whether two participants have exchanged messages.

Anonymity networks split into two categories. Low-latency networks span the anonymity set by a large number of contributing subjects. Low-latency networks cannot guarantee perfect secrecy, i.e. adversaries are able to learn about the anonymity set with each sent message [9]. High-latency networks provide perfect secrecy, but disable interactive sessions like HTTP browsing. In our scenario, the access control entities defend the anonymity network itself against adversaries from outside.

The *DoS detection and mitigation* engine protects the receiver from malicious traffic, such as flooding. Incoming messages are clustered into traffic flows, which share the same sender and recipient. From each traffic flow simple characteristics are derived, e.g. the total message count and the average rate of sent messages. A sliding window limits the scope of included data either by time interval or a fixed number of messages. Activity profiling highlights traffic flows with extraordinary sending rates. Sequential change-point detection tracks flow characteristics and is triggered by sudden changes. A sudden increase in a small volume flow can indicate an attack, even if much larger streams coexist [10]. Shaping of traffic flows and re-configuration of network elements are possible responses to suspicious network traffic. The anonymity network replaces IP header information. This disables traffic flow clustering and impedes the application of current mitigation approaches.

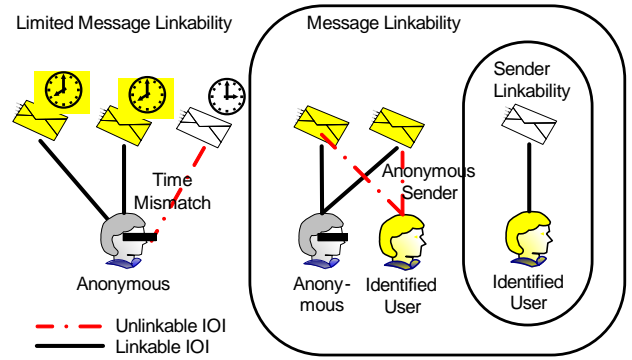


Fig. 2. Linkability concepts

B. Threats

Attackers generate Denial-of-Service flooding against public services. Malicious network participants can *flood* network hosts, limiting the availability to legitimate users. Flooding is an especial attack in anonymity networks, because it requires groundbreaking adaptations for traffic flow clustering. Existing clustering methods rely on IP header information, which is concealed by the anonymity network.

A DoS attack aims at *resource exhaustion* at the victim host, e.g. CPU time and buffer space. A flooding attack can be successful, if the host takes more time to respond to a specific request than the attacker requires for generating these. A size-limited buffer hosts requests from legitimate and malicious users. Therefore a DoS attacker is able to cause dropping of legitimate requests.

Man-in-the-Middle attacks tap ongoing connections. Nested encryption prohibits this attack in the anonymity network. Certificates provide proof of identity for intermediate entities such as access control and detection engine.

Adversaries head to reveal the identity of anonymous subjects, i.e. break the confidentiality of the communication relationship. One approach learns about the paths through the mixer nodes. Therefore the adversary tags messages within the anonymity network and observes the (unencrypted) tags at dishonest nodes [11]. This way the adversary can learn about the applied routing and possibly reveal communication partners. Although there are several security issues to be considered [12], this paper focuses threats that appear within the proposed DoS detection mechanism.

Profiling is the act of acquiring information from observed messages. Several messages are explicitly linked together either by packet header attributes or by application-level data, like an HTTP session cookie. Data that could be misused for profiling can be expunged by client-based filters, like Privoxy. Such filters impede adversaries to gain message linkability from within the message contents. The anonymity network removes profiling attributes from the packets. However, query keywords and timing correlation between messages can enable probabilistic messages correlation.

The act of communication refers to the message being sent and two involved subjects as sender and receiver. For an

adversary several items of interests are desirable to be revealed. Items of interest are subjects (an observed user/server), messages (criminal content or externally tagged), and actions (tunnel establishment). Anonymity networks protect against several grades of *linkability* (cf. Fig. 2). Message linkability enables an adversary to correlate two messages that originate from the same, but anonymous, sender. Sender linkability is even stronger as it reveals the identity of the message sender. By transitivity sender linkability enables the correlation of any communication from a specific sender. Both share the threat that an adversary is able to profile all messages ever sent. A remedy to profiling is limited linkability, e.g. by chronological limits, shown in Fig. 2 left. Although the third message originates from the same sender, it is unlinkable because of the different time. User activities split by time into several profiles, which cannot be linked together.

C. Security Objectives

Flooding protection in anonymous communications is based on the following three properties.

- In order to cluster traffic flows, a **message tag** is necessary at the DoS detection engine. The tags must map bursts of message onto the same traffic flow, because they indicate a flooding attack. This measure enables flooding detection and prevents the victim of malicious resource exhaustion.
- Message linkability must be **limited** to protect subjects against adversary profiling. Flood attacks focus attacker resources onto a single victim. Collaboration between detection engines is not necessary. Comparability of message tags between detection engines of different receivers is not necessary and weakens privacy.
- **Control** of message linkability must be independent from participating subjects. Resulting profiles should result deterministically from the message sending behavior. Malicious sender must be unable to camouflage their attack traffic. Also no actions of an adversary should weaken the anonymity of other subjects.

Together these properties protect receivers from flooding and legitimate users from profiling and sender linkability.

IV. MESSAGE TAGGING WITH LIMITED LINKABILITY

The distinction of traffic flows is a prerequisite for attack detection. Tag generation must ensure that flooding messages become mapped onto one tag. This enables DoS detection mechanisms to correlate anonymous messages.

A. Message Linkability Continuum

The grade of linkability is driven by the message tag generation. Properties within the linkability continuum are illustrated. Finally we motivate limited linkability.

The amount of messages with an equal tag defines the *message linkability continuum* (cf. Fig. 3). In lifelong linkability the cluster attributes remains static, a prerequisite for traffic flow recognition and profiling. Other applications restrict message linkability completely. Broadcast (anonymity)

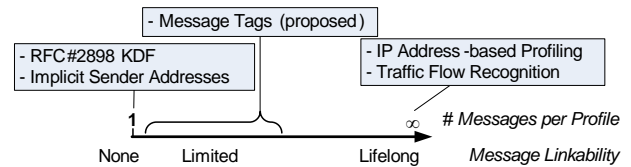


Fig. 3. Message Linkability Continuum

networks keep the receiver address confidential. Even if an adversary taps all links message broadcasts protect the identity of the intended receiver. Non-deterministic *implicit addressing* encrypts the address together with random data. The random bits prevent an adversary from correlating messages [13]. The grade of linkability depends on the value renewal. An extension of hashing, the key derivation function (KDF), protects a passphrase from dictionary attacks. Before being hashed, the passphrase is extended by salt (random data). This protects against bruteforce password attacks, because the hash values depends on the salt value [14]. In order to discourage list-based password attacks, KDFs disrupt linkability of hash values between authentication sessions. Encryption together with salt protects against linkability, e.g. during transmission through an anonymity network. For the DoS detection engine, fresh salt per message is too strong. Flooding detection demands for message linkability within a short time interval. Therefore salt should be derived from the message header. A criteria must define limited linkability, i.e. how many messages become linkable.

Deterministic generation of message tag can be based on message header attributes, like receiver and sending time. To maintain communication anonymity, preimage resistance is important. I.e. an adversary cannot derive message attributes from a tag. Tag creation based on cryptographic hash functions enables traffic flow classification. The impact of collisions is considered in Sec. V-A. If message tags match, the (anonymized) attributes have been equal. This integrates traffic flow clustering into anonymous communication.

Traffic flow clustering enables further specification of limited linkability. Generally speaking, linkability should be proportional to the probability of flooding attack. Only high arrival rates, which typically occur in flooding attacks, should be linkable. Clustering techniques apply a sliding window in order to limit involved data, either by a fixed number or time interval. Linkability with messages outside the window is not necessary for clustering. This establishes the idea of limited linkability. Because lack of meta information, the receiver's message arrival curve is unknown during tag generation. But tag generation can include numbered time intervals and the message volume into message tagging. Time intervals are well suited for limited linkability. The parameter $\lfloor \text{time}/1 \text{ min} \rfloor$ applies the current minute into tag generation. However, only traffic flows within the same minute are linkable, messages in different intervals result in different tags. The detection engine identifies the change of message tags continuously as part of the normal operation. Each communication participant steers

message linkability through the arrival rate. Flooding with high message rate gain high linkability.

Another solution is message counting, which is clock-independent. A trusted party stores a separate message counter c_{src} assigned to each sender identity. The counter is used to derive a tag parameter, e.g. $\lfloor c_{src}/100 \rfloor$ keeps the tag parameter for one hundred messages constant. Adversaries benefit from this method, because a static profile size simplifies analysis. Instead, message linkability should scale proportionally with the message volume, because a high message volume possibly indicates to a flooding attack. We therefore suggest a logarithm scale for the tag parameter. In result, the set of linkable messages $\{x\}$ increases exponentially:

$$\forall t > 0 : \|\{x \mid \lfloor \log_2 x \rfloor = t\}\| = 2^t$$

In the long run, an increasing number of messages become linkable together, therefore enabling flooding detection. Because the message counters of all participating senders vary, collisions of different length interfere with profiling. A regular user establishes communication with multiple receivers and message frequency remains low. For $t = 10$ thousand messages are distributed over time and multiple receivers. In case of a flooding attack a high message frequency towards one victim is sent, therefore the DoS detection engine is able to mitigate the complete set of thousand messages.

B. Time-based Generation

Next, we propose an approach for message tagging, which limits message linkability to fixed time-intervals. Then we investigate an alternative approach based on message volume.

The objective of message tag generation is the detection of DoS flooding attacks at the detection engine. This assumes flooding is distinguishable from regular traffic based on the flow arrival curves. The activity profiling detection mechanism rates each traffic flow separately. The arrival curve of incoming traffic flows become ordered by intensity. As a simple measure for messages intensity with arrival times (t_0, t_1, \dots, t_n) we suggest the sum of inverted exponential arrival times $r = \sum 1/e^{t_i-t_0}$. A message tag with a high arrival rate outperforms tags with average traffic. Later in this section example graphs will be discussed.

The message tags are assigned by the access control entity (cf. Fig. 1a). The entity authenticates the sender identity and then applies the key derivation function to compute the tag.

$$g(m) = \text{KDF}(\text{src}) \quad (1)$$

The entity forwards then tag plus message through the anonymity network towards the recipient. The message tag is included in the application content and encrypted during passage; it cannot be used for path observation. The DoS detection engine extracts the message tag and eventually forwards the message to the receiver. Eq. (1) violates the security objective of limited linkability, as its tags depend on the sender identity only. Limited message linkability must also tear down cross-server profiling. This can be realized by including the destination into the message tag. Additionally,

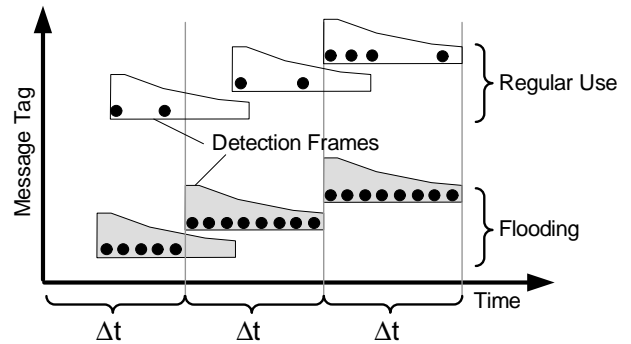


Fig. 4. Time-based Message Tagging

the scope of profiling should be limited to a short message intervals. Such a constraint can be achieved with time-based or volume-based approaches.

$$g(m) = \text{KDF}(\text{src}, \text{dest}_m, t_{i_m}) \quad (2)$$

Equation (2) additionally involves the current time interval $t_{i_m} = \lfloor t/1 \text{ min} \rfloor$, in which message m is sent, plus the destination address dest_m of the specific message. Local time is divided into time intervals numbered in minute steps beginning at zero (Jan, 1st 1980).

The viewpoint of a DoS detection engine is shown in Fig. 4. For convenience the message tags are grouped by sender identity. Cryptographic hash functions hide the similarity of input values. The upper three message tags belong to one sender with an average message arrival rate. The traffic flow is split into three detection frames (displayed as polygon), because sending in different time intervals lead to three different message tags. The arrival curve rating is updated on each incoming request. The last three message tags in Fig. 4 belong to a second sender. The engine detects the excessive arrival rate and marks the traffic flows suspicious (grey shaded polygons).

Time-based message tags enable traffic flow classification. At the same time they limit linkability by splitting profiles into time intervals. Cross-server profiling is inhibited because different tags are issued per message destination. Messaging tagging is robust against influences from malicious parties. Each sender is able to avoid message linkability if it omits further transmissions for two minutes. The scalability of this approach is limited, because it assumes that all tag generators are bound to the same time base. If multiple entities provide network access, their clocks must be synchronized. Otherwise an attacker is able to gain different message tags splitting traffic between separate entities. The attacker is able to hide within other traffic flows. Time synchronization within a large set of authorities is costly and in distributed scenarios often not an accepted solution. Time-based message tagging is therefore not applicable with multiple access control entities.

C. Distributed Generation of Message Tags

A scalable architecture relies on parallel instances. Such requires multiple instances of access control entities. Senders

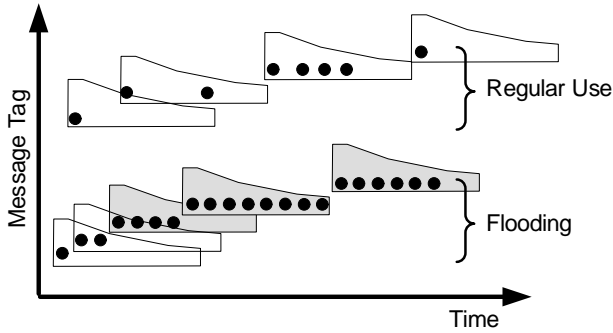


Fig. 5. Volume-based Message Tagging

can freely decide whether they deliver all traffic over a single entity or split it up using several access control entities. To enable scalability we propose message tagging applicable to distributed scenarios.

An efficient defense mechanism combines attack countermeasures with disincentives for malicious activities. Supported by game theory fundamentals we study strategic players, which maximize their own benefit by adopting their own strategy. As cost function we choose unlinkability. Concretely, strategic players minimize their message linkability by shrinking the size of their profiles, i.e. the number of messages that carry the same message tag. For simplification, we assume that senders communicate with a single receiver only. The set of strategies varies the number of access control entities over which the traffic is split. We construct an incentive mechanism out of the strategy set and this goal. By increasing linkability in the user of parallel access control entities, we enforce strategic players to send their traffic over a single instance.

The access control entities track the message frequency of all senders. This is a candidate as parameter for message tagging. The message tags must be based on locally available information. The architecture does not scale well, if sender-specific information must be distributed between the access control entities. A message counter tracks the amount of sent messages for each sender identity. Referring to the design considerations in Sec. IV-A, each access control entity counts the sent messages per sender. Eq. (3) adds the floor of the message counter logarithm as tagging parameter.

$$g(m) = \text{KDF}(\text{src}, \text{dest}_m, \lfloor \log_2 c_{\text{src}} \rfloor) \quad (3)$$

It derives message tags from the source identity, the message destination, and the message counter for the sender identity c_{src} . Note, that this modifies the message tagging only. It does not change the flooding detection method. The result of the traffic flow clustering is visualized in Fig. 5. The arrival curves of senders 1 and 2 remain unchanged. Both send messages to one receiver. Because sender 1 starts with an empty message counter, 1, 2, 4, and 8 messages receive the same message tag. The engine builds a new detection frame with the first message of unknown tags. The arrival curve of messages within a frame determines, whether the flow is considered suspicious. Although the third frame contains four messages, their arrival

curve does not reach the flooding threshold. Sender 2 floods the receiver with high message frequency (bottom of Fig. 5). The tagged messages allow linkability, and the arrival curve strengthens detection, because traffic flows detected at the engine feature extraordinary arrival curves (indicated by grey shades). The detection engine mitigates the flood. Summarizing, flooding detection improves over time, because the scope of the limited linkability extends with the attack duration.

V. ANALYSIS

Two security vulnerabilities are addressed by distributed message tagging. Flooding must be detectable using the message tags. Profiling must be prevented in order to maintain sender anonymity.

A. Flooding Attacks

Message tags enable application of traffic flow analysis, based on the message linkability described in the previous section. The DoS detection engine senses flooding traffic based on the intensity of the arrival curve.

Collisions may occur under the KDF during tag generation, but only very few messages will be delivered to the same receiver. In this case, a match of the message tag does not truly indicate that it originates from the same sender identity. If messages of a regular application become shaped, a repeated request will be sent after a timeout. Message tagging parameters develop further, so that the collision at the receiver does not persist for long time.

Attackers could bypass countermeasures in order to perform flooding attacks. Game theory models an attacker as strategic player, who adopt its behavior to reduce message linkability at the detection engine. The free choice of access control entities allows an attacker to distribute flooding messages over several access control entities. The act of tag generation has three parameters. Sender and receiver identity are fixed for all messages in a flooding. The message counters of the involved entities are independent, but a distributed flooding results in high values at all entities.

In the first flooding scenario the attacker simultaneously starts using two access control entities. Because of the identical tag sequence the number of linkable messages doubles. A mixed-rated flooding scenario sets different message rates r and $2r$ per entity (cf. Fig. 6 up). The engine will detect the follow up the slow flooding anyway because the initial flood increased sensitivity for these message tags. Again, the resulting message linkability increases. An attacker could start floodings over two entities at different times (cf. Fig. 6 bottom). Because of the logarithm property, messages become linkable as soon as the tag parameter synchronizes. An attacker could lower message linkability if it increase the number of access control entities. The limited number of entities having independent message counters, restricts the duration with low linkability floods. Alternatively, an attacker could increase the counter by sending cover messages to other parties. An unlinkable sequence towards one victim could be the messages $(2^0, 2^1, 2^2, \dots, 2^{n-1}, 2^n, \dots)$. To maintain a constant message

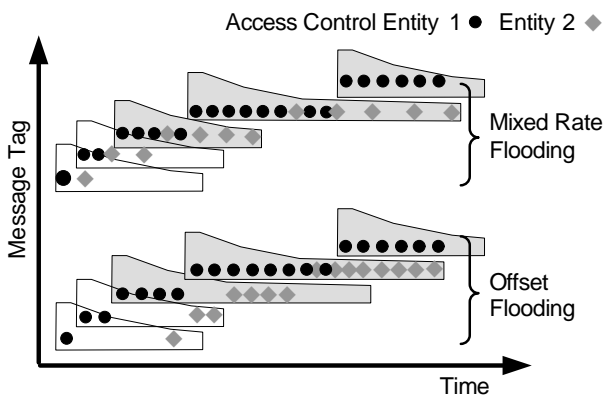


Fig. 6. Attacks on Volume-based Message Tagging

rate, the attacker requires an exponential increase of cover traffic. Otherwise the flooding intensity mitigates fast.

B. Anonymity Adversaries

The communication relationship is protected by the anonymity network. Certification of the access control entities by a trusted third party impedes man-in-the-middle attacks. Adversaries can find other ways to threaten network anonymity of the participating users.

Multi-receiver profiling is frustrated by the message tag generation, which diverges for different receivers. Therefore adversaries can only profile activities at a single receiver. When sending large amounts of messages towards a single receiver, the sender either intends a flooding or it has a trust relationship with the receiver and accepts the risk of profiling.

Pseudonyms are the traditional approach to provide message linkability while covering sender identities. Adversaries benefit from unlimited linkability provided through the pseudonym. The profiles derived detail the network behavior of the corresponding person. If a person acquires a new profile, but behaves similarly the large sized profiles can be linked together. This motivates user-independent control of limited linkability. The proposed scheme enforces frequent message tag changes, which results in limited linkability.

VI. FUTURE WORK

A clear limitation is the requirement of trusted access control entities. Cryptographic blinding will enable distributed message tagging, however it increases the effort on the client side. Participants can then join the anonymity network. Current peer-to-peer overlays do not provide responder anonymity yet. Also, anonymity network nodes supply resources which maintain anonymity of third party communication. Together with the limited capabilities of mobile devices, trust into an access control entities is rational.

In future work we will examine revocation of anonymity in order to permanently eliminate malicious traffic. During ongoing attacks the victim collects proofs of malicious traffic. After the attack proofs have been verified, the access control entities deny further communications with the victim.

Although Distributed Denial of Service attacks are not a primary problem in anonymous communication, identity theft introduces a new class of attacks. We examine whether anonymous communication can fulfill the requirements of IP-based DDoS detection mechanisms.

VII. CONCLUSION

Exit policies are not sufficient to mitigate attacks through anonymity networks. Network privacy can only be maintained applying adequate attack response. The design of a traffic flow cluster attribute must preserve limited message linkability and respect sender anonymity.

The time-based message tagging approach enables limited message linkability. The receiver can decompose traffic flows and detect DoS flooding. Volume-based message tagging enables system scalability. An incentive mechanism refrains communication participants from using multiple access control entities. The analysis shows that attackers cannot derogate linkability of their attack traffic. The approach is also resistant to network profiling from adversaries. The proposed mechanism is scalable, robust against flooding attacks, and also protects sender anonymity. In conclusion, flooding attacks over anonymity networks are detected.

REFERENCES

- [1] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: secure overlay services," in *SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 61–72.
- [2] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *13th USENIX Security Symposium (SSYM'04)*, 2004, p. 21.
- [3] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker, "DDoS defense by offense," in *SIGCOMM '06*. ACM, 2006, pp. 303–314.
- [4] A. Juels and J. G. Brainard, "Client puzzles: A cryptographic countermeasure against connection depletion attacks," in *Network and Distributed System Security Symposium (NDSS)*, 1999.
- [5] X. Wang and M. K. Reiter, "Defending against denial-of-service attacks with puzzle auctions," in *Symposium on Security and Privacy (SP '03)*. IEEE, 2003, p. 78.
- [6] J. Claessens, C. Diaz, C. Goemans, B. Preneel, J. Vandewalle, and J. Dumortier, "Revocable anonymous access to the internet," *Journal of Internet Research* 13(4), vol. 13(4), pp. 242–258, 2003.
- [7] S. Köpsell, R. Wendolsky, and H. Federrath, "Revocable anonymity," in *Emerging Trends in Information and Communication Security (ETRICS)*, ser. LNCS, vol. 3995. Springer, 2006, pp. 206–220.
- [8] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 4, no. 2, 1981.
- [9] A. Pfitzmann and M. Hansen, "Anonymity, unobservability, and pseudonymity: A consolidated proposal for terminology," Draft, July 2007, v0.29, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml.
- [10] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.
- [11] N. Mathewson and R. Dingledine, "Mixminion: Strong anonymity for financial cryptography," in *8th International Conference on Financial Cryptography*, vol. 3110. Springer, 2004, pp. 227–232.
- [12] D. Kesdogan and C. Palmer, "Technical challenges of network anonymity," *Computer Communications*, vol. 29, no. 3, pp. 306–324, Feb. 2006.
- [13] A. Pfitzmann and M. Waidner, "Networks without user observability—design options," in *EUROCRYPT '85*. Springer, 1986, pp. 245–253.
- [14] B. Kaliski, "PKCS #5: Password-based cryptography specification version 2.0," RFC 2898, 2000.